INTERNATIONAL JOURNAL OF RESEARCH IN COMPUTING

Volume 04 Issue 01

IJRC

International Journal of Research in Computing (IJRC)

Volume 04 Issue 01

January 2025

ISSN 2820-2139

Published by

Faculty of Computing, General Sir John Kotelawala Defence University,

Ratmalana, Sri Lanka

Tel: +94-11-2635268

E-Mail: editor@ijrcom.org & editorijrc@kdu.ac.lk

Website: http://ijrcom.org/

EDITORIAL COMMITTEE

MEMBERS OF THE EDITORIAL BOARD

CONTENTS

A Comprehensive Review: Enhance Logistics Performance by Optimizing Supply Chain Routes with Dynamic Factors using Genetic Algorithm

GSM Jayasooriya[1#] and ADAI Gunasekara[1]
[1]Department of Computer Science, Faculty of Computing, General Sir John Kotelawala Defence University, Ratmalana, 10390, Sri Lanka.
[#]39-bcs-0015@kdu.ac.lk

ABSTRACT As supply chain networks grow increasingly complex, achieving optimal logistics has become essential for industries to remain competitive and adapt to dynamic demands. Traditional route optimization methods often fail to accommodate real-time factors such as traffic congestion, unpredictable weather conditions, and shifting customer requirements, leading to inefficiencies in logistics performance. This study aims to address these challenges by exploring the potential of Genetic Algorithm (GA) as a robust solution for multi-objective route optimization. A thematic literature review was conducted to evaluate existing algorithms and models, revealing significant gaps in their ability to manage dynamic, multi-factor logistics environments effectively. The review identified that Genetic Algorithm excel in integrating real-time data, enabling the optimization of delivery routes with greater efficiency and adaptability. Real-world applications of GA in diverse industries demonstrated reductions in delivery times, improved resource utilization, and enhanced customer satisfaction. These findings establish GA as an intelligent and scalable approach to modern logistics challenges, offering significant implications for advancing supply chain management practices.

INDEX TERMS Dynamic factors, Genetic algorithm, Real-time data integration, Route optimization, Supply chain logistics management, Multi objective optimization

I. INTRODUCTION

In today's globalized economy, efficient supply chain logistics is crucial for maintaining competitiveness and meeting customer expectations. Efficient, timely, and reliable delivery is the backbone of modern supply chain logistics, where the competition for customer satisfaction and cost-efficiency is at more intense. As companies seek to improve their operational performance, route optimization has become a vital strategy for minimizing delivery costs and reducing transportation times while enhancing overall service quality.

In recent years, technological developments and logistics management improvements have highlighted the significance of dynamic, multi-factor route optimization within supply chain systems. Conventional methods of route planning often rely on simple distance calculations or static/fixed models and they struggle to adapt to the dynamic nature of real-world conditions. [1] This has led to the rise of more sophisticated computational techniques and the use of more advanced optimization techniques as a practical approach for complex optimization challenges in logistics.

In the real world, the task of route optimization in supply chain logistics is complex and intricate. It encompasses not only the shortest or least expensive route, but also the various dynamic and unpredictable variables. These variables, which include weather conditions, traffic congestion, road status, and customer urgency or demand, each play a vital part in the decision-making process. [2] Traditional optimization strategies often struggle to address all these interrelated aspects, resulting in inefficiencies in planning routes and distribution.

Genetic Algorithm (GA), a type of evolutionary nature-inspired algorithm, has emerged as a powerful and robust Algorithm for solving complex optimization problems. Their capability to handle large search spaces and adapt to changing environments makes them an appropriate choice for route Optimization in logistics, were multiple, often competing, factors must be taken into account. GA uses principles such as selection, crossover, and mutation to iteratively enhance a population of possible solutions, making them well-suited for the dynamic and multi-objective challenges of route optimization.[2] The application of GA in supply chain logistics has shown promise in reducing costs, minimizing delivery times, and improving overall efficiency.

Current research on route optimization in supply chain logistics primarily centered on single-factor optimization, with using the factors such as distance or cost, without considering the interconnected nature of multiple influencing factors like weather conditions, traffic patterns, and customer urgency. Although Genetic Algorithm has been applied successfully in single-objective optimization, their potential for handling multi-factor multi-objective scenarios remains underexplored. These gaps in existing research present a significant opportunity for advancing the field, as multi-factor optimization can provide a more comprehensive and accurate solution for route planning in dynamic supply chain environments.
The aim of this study is to provide a comprehensive analysis of application of Genetic Algorithm (GA) in multi-factor route optimization for supply chain logistics. It seeks to investigate the effectiveness of GA compared to conventional optimization techniques, identify existing challenges, and highlight real-world applications, thereby offering insights into the potential of GA for route optimization in logistics and enhancing efficiency and adaptability in dynamic environments.

Objectives:

1) To evaluate current supply chain logistics optimization strategies, highlighting their advantages and disadvantages.

2) To evaluate and compare GA's performance against existing and conventional techniques.

3) To investigate how Genetic Algorithm (GA) can provide flexibility and adaptability in addressing the difficulties and challenges in multi-factor route optimization.

4) To demonstrate the usefulness and practical advantages of GA in logistics through real-world applications.

II. LITERATURE REVIEW

This literature review uses a thematic approach to explore the application of Genetic Algorithm (GA) in multi-factor route optimization in supply chain logistics by organizing the findings into four major themes. The thematic structure helps in systematically analyzing the existing research and identifying key gaps for future exploration.

A. Alternative Route Optimization Techniques in Supply Chain Logistics

In the field of supply chain logistics, a number of optimization strategies have been developed to address the challenges of optimal transportation and distribution of goods between the supplier and the customers. Apart from Genetic Algorithm (GA), some advanced models and techniques, including Ant Colony Optimization (ACO), Multi-Agent Systems (MAS), and other mathematical modeling, are frequently used.[2] For example, ACO, which takes inspiration from how the ants discover the best or optimal routes, has been successful in addressing route optimization problems by continuously adapting to changing environments. Research shows that, ACO is capable of adjusting to various scenarios, making it fitting for situations where the environment is predictable and relatively stable. This approach is crucial in overseeing transportation networks with the goals of minimizing travel distances and minimizing logistical expenses.[3]

For many years, conventional methods like the Dijkstra Algorithm and the Traveling Salesman Problem (TSP) have been used for route planning. The Dijkstra Algorithm effectively identifies the shortest path between two nodes (destinations) within a graph, making it suitable for network-related tasks such as logistics routing. Conversely, the TSP seeks to find the most effective route that visits each location exactly once before returning to the starting point, a challenge often faced in planning delivery routes.[4] However, these algorithms can struggle with scalability as the complexity of the problem increases, necessitating the need for more sophisticated optimization strategies.

Metaheuristic techniques like Simulated Annealing (SA) and Particle Swarm Optimization (PSO) have been introduced to route optimization in logistics to address these challenges.[5] SA, which inspired by the annealing process in metal treatment, helps prevent the algorithm from getting stuck in local optima by facilitating exploration across a broader range of the solution space. This technique has been shown to be beneficial and working well, when a near-optimal solution is sufficient within a reasonable timeframe.[6] In contrast, PSO is inspired by the social behaviors of birds in flocks or schools of fish. It is a population-driven optimization method that updates possible solutions based on their current positions and velocities within the solution space, proving effective in tackling complex logistics challenges that include various factors and interconnected variables.[7]

However, Genetic Algorithm (GA) has the ability to escape the local minima and investigate a variety of solutions through crossover and mutation process. That makes GA unique among other techniques. GA are more flexible in their changing solutions than PSOs, which makes them especially suitable for dynamic logistics scenarios where several elements need to be balanced at once[1].

B. Challenges in Multi-Factor Route Optimization

Multi-factor route optimization in supply chain logistics Becomes very complex due to the involvement of interlinked factors and variables such as cost, time, Distance, and environmental concerns.[4] A critical Challenge is the variability and unpredictability of external factors, such as accidents leading to congestion of traffic and Weather conditions. For instance, studies show that not being able to adapt dynamically to changing real-world conditions Has a significant impact on the efficiency of logistics operations and overall customer satisfaction.[8]

Getting accurate and updated information for these variables in order to train the models is another big hurdle. This route optimization can be done most effectively based on real-time Data taken from traffic patterns, weather conditions, and road closures through integrations with live traffic monitoring systems and weather forecasting APIs.[9] IoT devices and Cloud-based services can be used to make sure that the logistics system draws live reports to perform route planning with the most updated information. Such a model will also be helpful in dynamic routing and, hence, more economically efficient with enhanced delivery of services.[10] On the other hand, computational complexity, real-time data integration, and the need for adaptive algorithms make it more challenging.

Another factor would be the demand from customers and the urgency, which is from time to time from customer to customer, hence unpredictable. This variability affects how goods are scheduled for delivery, from vehicle load capacities to time windows of delivery. This variability makes it hard to effectively plan routes since those routes are optimized based on one set of demand conditions, which may turn out to be suboptimal if demand patterns change in an instance.[1] Therefore, models need to be optimized based on data and predict changes with the use of predictive analytics coupled with historical data in order to manage supply chain logistics better. However, the challenging criterion is that these techniques also require more computational power to train the algorithms and find and evaluate the models using appropriate weights.

While there exist a number of traditional and existing models, Most of them are narrow in their focus, typically based on very few key factors or variables like the minimum distance or minimum Cost[10]. Conventional methods, such as the

Traveling Salesman Problem or Dijkstra's Algorithm, take into consideration only the shortest path or shortest distance, not including real-time data input or any fluctuation in customer needs. [11] Especially in logistics, customers are increasingly expecting quicker and more reliable services. Thus, the demand for more advanced models is increasing to generate increasingly adaptive and efficient solutions. [8]

Key Challenges:

1) Route optimization in logistics involves balancing multiple variables such as cost, time, distance, and environmental conditions, making the process complex and challenging.

2) Unpredictable external conditions like traffic congestion, weather changes, and fluctuating fuel costs can disrupt pre-planned routes, affecting overall efficiency.

3) Real-time data is crucial for effective route optimization, and IoT devices and cloud-based services help integrate this information despite challenges.

4) Customer demand variability can impact delivery schedules, necessitating swift adaptation of models and often necessitating predictive analytics for efficient routes.

5) The integration of real-time data and model training requires more computational power, making it more challenging to evaluate models with appropriate weights based on changing facts.

C. Why Genetic Algorithm (GA) for Route Optimization

Genetic Algorithms are one of the emerging and nature-inspired optimization techniques, especially in solving complex and multi-variable problems, which often cannot be solved easily by other methods. GA got the inspiration from principles of natural selection and evolution to explore the solution space effectively through processes of selection, crossover, and mutation to gradually arrive at an optimum or near-optimum solution.[12] Because of the nonlinear, multi objective, and NP-hard ability of GA, they have already been used in extensive applications on vehicle routing, network design, and scheduling in supply chain logistics.[13]

A major benefit of GA is that they can survey large solution spaces effectively without getting trapped in local optima , a problem often faced by deterministic algorithms. In fact, it has been shown that GA can be hybridized with other advanced heuristic methods like Simulated Annealing to obtain higher-quality solutions and better computation times for routing problems.[8] This hybrid methodology is able to advance the algorithm's capability to adapt in dynamic situations, such as those involving immediate fluctuations in customer needs or conditions around traffic, which are very crucial for logistics distribution planning.[14]

Moreover, GA provide the capability for encoding a number of objectives into their fitness function, allowing them to be applied for multiobjective optimization. For example, a GA-based model could optimize cost, delivery time, and customer satisfaction simultaneously, offering a set of Pareto-optimal solutions where the decision-maker can choose based on priorities.[1] Recent research illustrated the efficiency of GA in managing these trade-offs and, hence, their potential to support strategic decisions in supply chain management by balancing cost efficiency and service quality.[12]

To solve complex optimization problems, GA have to be computationally efficient and scalable, especially in large-Scale applications. Recent developments have enhanced These attributes are through parallelization and distributed computing frameworks. The significant methodologies are scalable and parallel Genetic Algorithms, which increase the performance in large integer linear programming models by handling up to 2258 decision variables. [12] Distributed computing frameworks, such as Apache Spark, have enabled scalable GA to manage large datasets and complex problems, achieving scalability across up to 3000 dimensions and one billion generations while preserving population diversity. [15]

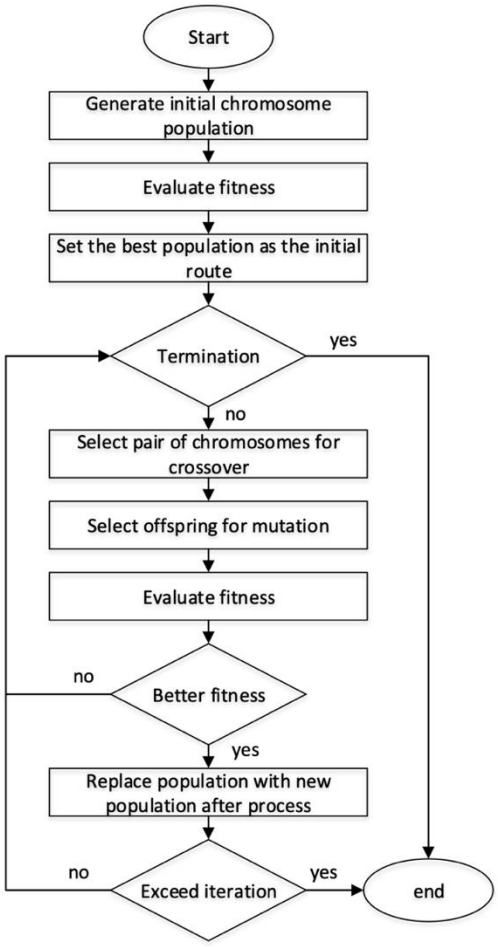

Figure 1. How Genetic Algorithm Works
Source: Authors

Additionally, improvements in algorithmic design, such as population initialization and mutation techniques, have tailored GA for large-scale issues, achieving significant speedup in runtime. These advancements underscore the robustness and scalability of GA, yet challenges remain in balancing computational resources and maintaining solution quality, particularly in dynamic environments.

D. Real-World Applications of Genetic Algorithm in Route Optimization on Logistics

GA has been an effective algorithm in many real-world Logistics applications, especially those problems that are complex and multi-constrained, as well as multi objective. Their applications are found within many optimization problems in a wide range. It has been applied to solve the vehicle routing problem and many variants, such as multi-depot VRP, VRP with time windows, and dynamic VRP.[16] For instance, research has shown that GA can optimize delivery schedules by minimizing travel distance and ensuring timely deliveries, enhancing overall efficiency, and reducing operational costs for logistics companies. [4]

Logistics applications of GA, such as routing available vehicles from several depots to customer locations, are similar to the multi-depot VRP.[16] Conventional routing methods often do not consider dynamic customer demands and may even fail to take up real-time changes. However, GA succeeded in satisfying these dynamic needs and leading to more adaptive and robust solutions. Hybrid approaches combining GA with other heuristics, such as simulated annealing, have also been implemented with the purpose of enhancing local search capability and preventing algorithms from getting stuck in suboptimal or local minimum/maximum solutions.[17]

Furthermore, GA has been utilized in the field of supply chain network optimization, where it supports the determination of the most appropriate configuration regarding suppliers, production facilities, and distribution centers. In these applications, GA efficiently balances various goals on cost minimization, service level maximization, and resource utilization. This multi-objective approach is critical in dealing with large-scale supply chains where conflictive goals often appear that need to be pursued simultaneously. Studies have shown that GA creates Pareto-optimal solutions with higher strengths compared to traditional optimization techniques, thus helping companies increase customer satisfaction and smooth operations.[8]

GA, in the context of urban logistics, has been integrated into "last-mile" delivery optimization systems, whereby the challenge of efficiently navigating and working through complex cityscapes and managing to change customer demands. Companies have used models involving the use of GA for dynamic routing of delivery vehicles to minimize delays and fuel consumption and also ensure on-time delivery in urban congestion areas.[1] Similarly, GA has been applied in warehouse management, allowing for optimized layout and storage allocation within extensive facilities. Through estimation of the most efficient place of goods, GA helps reduce time spent on retrieval, hence improving general warehouse productivity. They are accommodating in the e-commerce operation and deal with massive inventories.[10]
Another perfect application is logistics in air cargo, where GA is used in the scheduling of cargo flights and efficient space allocation, keeping weight limits and flight schedules so as to be able to carry maximum loads. This will reduce operation costs, hence improving the profitability of the services offered in air freight.[9]

Depot management also plays a significant role in the military field in terms of how military supplies, equipment, and munitions are stored, protected, and managed safely during wartime. GA has also been employed here to solve complex problems related to depot location, stock management, and vehicle routing. A study tried to use GA on route optimization for the material resupply to soldiers in operation areas and utilized technologies like Google Maps and RoboFlow API to check locations and conditions. It was able to showcase the efficiency of GA in both critical and secure environments.[18]

Overall, GA has an excellent capability for optimizing complex logistical problems. Their flexibility and efficiency make them exceptionally strong in handling multi-objective problems Related to vehicle routing and network design concerning supply chains. While there are many open challenges, GA has proved its value in enhancing operational efficiencies for various logistic scenarios.

III. METHODOLOGY

The development of this review paper followed a structured approach involving the exploration of literature, identification of critical gaps, and systematic synthesis of findings. The methodology can be summarized as follows:

1) The initial step was to identify a research topic within the area of interest - machine learning. After exploring various domains, the focus was directed towards the application of Evolutionary Algorithms in multi-factor route optimization for supply chain logistics.

2) The study conducted a thorough search across various academic databases, including IEEE Xplore, ScienceDirect, and Google Scholar, using keywords like "Optimization Algorithms," "route optimization," "vehicle routing," and "multi-objective optimization" to identify research papers that directly contribute to the chosen research area.

3) A close review and analysis revealed existing models had certain limitations, especially in their ability to handle real-life complexities. Most of the conventional optimization methods did not consider various vital parameters such as environmental variables, real-time traffic conditions, road quality, and variations in customer demand and urgency. The lack of these crucial parameters was thus helpful in underlining the need for robust and flexible optimization models. Likewise, I identified some critical gaps in this research.

4) Then, the data gathered from relevant literature was systematically reviewed to identify key themes and trends. Papers were categorized into four main themes:

i. Other Optimization Techniques in Supply Chain Logistics
ii. Challenges in Multifactor Route Optimization
iii. Genetic Algorithm (GA) in Optimization
iv. Real-world Applications of GA in Logistics

This categorization, therefore, empowered the review to be conducted in a structured manner, hence allowing the adoption of knowledge regarding conventional methods, the emerging challenges, and the applications of GA. It also helped compare the capabilities of GA against existing methods, in particular, for addressing the identified critical gaps.

After that, a further step in this study is the critical analysis of information that has been extracted from selected studies. More importantly, this entailed efforts to identify emergent vital themes such as how traditional optimization was carried out, what unique challenges multi-factor route optimization faced, and what new approaches GA has introduced. Particular attention was paid to how well each addressed the complex conditions found in real-world situations, changes in road and weather conditions, and fluctuating demands by customers.

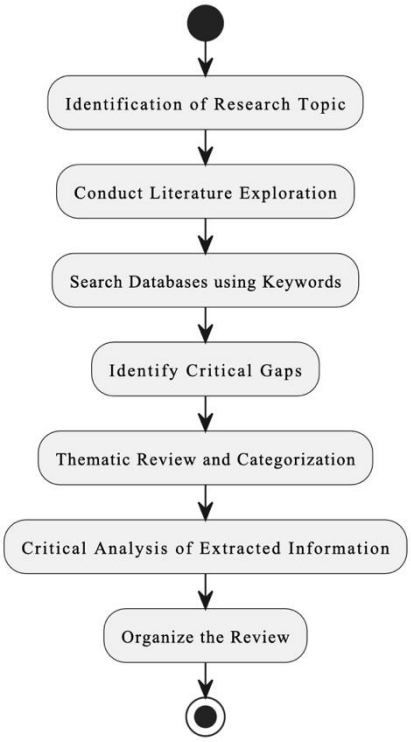

Figure 2. Workflow of the study
Source: Authors

Figure 02 provides a visual representation of the methodology followed in this review paper; the flowchart above outlines each step taken from the initial selection of the research topic to the final synthesis of the findings.

This methodology guaranteed a comprehensive analysis of the subject, providing coherent overview that connects theoretical understanding with real-world applications and lays the groundwork for further studies in Genetic Algorithm-based multi-factor route optimization in logistics.

IV. RESULTS AND DISCUSSION

Below is a breakdown of the main findings, reflecting how GA is being utilized effectively across various applications, as well as the challenges they present

A. Effectiveness of Genetic Algorithm in Logistics Route Optimization

GA is proven to be effective in multi-objective, complex tasks, balancing cost, time, and environmental impact. They are used in logistical issues such as vehicle routing and supply chain network design. GA provides better results than traditional methods, especially in problems like Vehicle Routing Problems, supply chain design, and depot management. [1] They also have the ability to adapt to real-world complexities, such as traffic patterns, road conditions, and changing customer demand. This allows for more flexible and reliable logistics plans, a significant advantage over rigid traditional methods that struggle to adapt to dynamic environments. [16]

B. Comparative Analysis: Genetic Algorithm vs Traditional Route Optimization Techniques

GA offers advantages over traditional methods like linear programming, Dijkstra's algorithm, or simulated annealing in finding near-optimal solutions in non-linear, multi-variable scenarios. Conventional methods are effective for simple, single-objective problems, while GA works best in non- linear, multi-variable scenarios where multiple objectives must be balanced. [11] For example, linear programming can handle simple cost-minimization tasks, but it cannot handle complex situations involving cost, speed, and fuel efficiency. GA provides Pareto-optimum solutions, allowing decision-makers to achieve balanced trade-offs at any given time.[16]

C. Real-World Applications and Case Studies

GA has practical applications in various real-life scenarios, such as urban delivery route refinement, supply chain design, And military logistics. Companies have saved millions of dollars by optimizing travel paths based on real-time traffic and customer delivery schedules. GA also helps determine optimal warehouse and distribution center locations, minimizing shipping costs while maintaining high service levels.[1] Even air cargo logistics and Military logistics use GA to solve supply routing problems during operations, demonstrating versatility in handling complex and unpredictable environments with security, terrain, and real-time conditions.[18]

D. Scalability of GA for the real-world applications in Logistics

In modern logistics operations with dynamic factors, systems often need to handle vast amounts of data from various sources-such as sensors, tracking devices, and databases in real time. Genetic Algorithms (GAs) are well-suited for these environments because they can be scaled up to process and optimize this diverse and large-scale data. By leveraging modern distributed computing environments like cloud platforms, GAs can run parallel processes that tackle different parts of the problem simultaneously. This means that even as the network grows and complexity, GAs can efficiently manage and optimize logistics operations, ensuring that decisions are made quickly and effectively across the entire system.[19]

Here are some key features of GA, that contribute to the real-world logistic networks.

i. Handling Diverse and High-Volume Datasets: Modern logistics generate vast, varied data—from real-time tracking to fluctuating demand. GAs, supported by robust data preprocessing and encoding techniques, efficiently manage both structured and unstructured data without performance slowdowns.

ii. Optimized with Distributed and Cloud Computing: Utilizing cloud platforms allows GAs to run parallel computations across multiple populations. This distributed approach ensures that even as data volume and complexity grow, optimization remains fast and scalable.

iii. Adaptive Parameter Tuning: Instead of manual adjustments, adaptive techniques enable GAs to automatically fine-tune parameters such as mutation and crossover rates. This dynamic tuning helps maintain optimal performance under varying logistics conditions.

iv. Integration with Other Optimization Methods: Combining GAs with other optimization techniques improves solution quality and speeds up convergence. These hybrid models effectively tackle the multifaceted challenges in large-scale logistics.[20]

v. Handling Real-Time Constraints and Dynamic Environments: Logistics networks are constantly changing due to factors like traffic or sudden demand shifts. Scalable GA implementations manage these real-time constraints, allowing continuous adaptation without complete re-optimization.

Key findings from a few significant research papers are compiled in the table below. The optimization factors considered, the main findings, and the specific application areas are highlighted in the table below. This summary illustrates GA's versatility in addressing various logistics-related optimization problems.

Table 1. Summary of key research finidngs

Paper	Factors Considered	Key Findings (Application Area)
[9]	Distance travelled and the Cost, Demand of Customer.	Faster delivery and a significant savings in costs achieved. (Product Distribution)
[4]	Time efficiency, Cost, Delivery area division.	Improved route efficiency, reduced computational time. (Logistics Distribution -TSP)
[1]	Cost, Service level, Resource utilization.	produced efficient Pareto-optimal solutions that balanced service and cost. (Supply Chain Network Design)
[8]	Time windows, Dynamic routing, and Usage of fuel	Reduced operating expenses and improved service were the results of effective routing. (Vehicle Routing)
[18]	Location, stock management, security, and real time updates.	Effective routing for military supplies under complex scenarios. (Military Logistics)

Source: Authors

E. Challenges in Implementing GA for Multi-Factor Optimization

While GA offers more advantages, there are disadvantages as well. The computational burden of GA is high, especially when working with complex datasets and parameters. [9] This may cause problems for real-time data integration situations. However, this problem has diminished with parallel processing and cloud computing. Incorporating real-world data, such as environmental conditions or customer requirements, into GA models could be challenging. Reliable and consistent data input is essential for GA to function effectively and consistently.[10]

Many logistics networks rely on legacy software, and making GA implementation would be challenging. Microservices architecture, hybrid GA approaches, and cloud-based API deployments can ease integration without disrupting existing workflows.

Another challenge that has been identified is Parameter Tuning Complexity. Manually adjusting GA parameters like mutation rates and crossover probabilities can be inefficient. Techniques such as self-adaptive GA, Bayesian Optimization, and reinforcement learning-based tuning allow automated parameter adjustments for better performance.[14]

F. Practical Recommendations for Implementing a System

Implementing GA-based solutions in real-world logistics networks can be highly effective and it helps to overcome the modern logistics problems. Through this review, several key insights have emerged regarding the practical implementation of Genetic Algorithm based solutions in logistics networks.

i. Data Acquisition and Preprocessing: Efficiency of GA-based solutions relies heavily on high-quality, real-time data.[21] Studies indicate that logistics operations integrate diverse data sources, including traffic congestion tracking, warehouse inventory, and customer demanding. Preprocessing techniques such as normalization and feature encoding are essential to ensure GA can handle structured and unstructured datasets without compromising performance.

ii. Tools and Computational Platforms for GA Implementation: The review identifies widely used platforms such as- Python libraries (DEAP) and MATLAB for developing and simulating GA models [4], R-based evolutionary algorithms for statistical and optimization analysis, and Cloud-based platforms for scalable, distributed GA implementations.

iii. Hybrid Optimization Approaches: The review identifies a growing trend in hybrid optimization models, where GA is combined with other heuristic and machine learning techniques for improved efficiency. Simulated Annealing (SA) helps GA avoid premature convergence by broadening the solution search space. Particle Swarm Optimization (PSO) enhances routing decisions by leveraging real-time and historical data patterns.[20]

V. CONCLUSION

This review outlines the strengths of Genetic Algorithm in resolving complex multi-factor optimization problems in logistics. Unlike the conventional methods, GA are more adaptive. Hence, they can handle dynamic variables such as cost, time, and environmental factors simultaneously.

The thematic review highlighted key gaps in conventional optimization techniques and established GA as a viable alternative, particularly in scenarios requiring real-time data integration and multi-objective optimization. Key findings indicate that Genetic Algorithmic approaches significantly improve logistics performance by reducing delivery times and costs, optimizing resource utilization, and enhancing overall service quality.

Comparative analyses show that GA outperforms conventional methods like Dijkstra's Algorithm, the Traveling Salesman Problem (TSP), and Simulated Annealing in handling dynamic and large-scale logistics networks. Real-world applications, ranging from urban delivery routing to military logistics, further validate the effectiveness of GA in practical scenarios.

While there are some challenges related to computational expenses and integration of data, ongoing advancements in GA, like hybrid approaches and the integration of new emerging technologies, will undoubtedly further increase their efficiency and applicability.

In summary, this paper uniquely contributes by exploring how Genetic Algorithms (GA) optimize multiple factors in logistics, such as traffic, weather, and customer demand, unlike traditional methods that focus on a single factor like distance or cost. It compares GA with older techniques, highlighting their limitations and demonstrating GA's ability to make smarter, more adaptable decisions. The paper also identifies key challenges, including computational complexity and real-time data integration, and suggests solutions like hybrid models and distributed computing. Additionally, it provides practical recommendations for industry adoption, such as integrating GA with real-time data, using AI-driven predictive models, and combining GA with other optimization techniques to improve efficiency and scalability.

VI. FUTURE WORKS

Future work on this will involve the implementation of a real-world GA-based route optimization system in supply chain logistics that can be used across various industries. The system will focus on dynamically planning and adjusting delivery routes, considering real-time factors such as traffic conditions, road quality, weather updates, and customer demand/urgency, with an ultimate goal of cost optimization and increasing the efficiency of deliveries. The final implementation will be a software platform that embeds the GA model with real-time data integration by fetching from APIs for updates.

The system will include a demand forecasting mechanism to predict future customer orders, in order to plan routes more effectively in advance. With that, logistics managers will be able to change the weights of the parameters according to the current situation, which includes cost, speed, and environmental impact. It will provide them with finer control over operations and the potential for the system to respond in cases of sudden changes, such as traffic congestion and sudden spikes in demand, by applying robust and intelligent methods to address current problems in logistics management.

Additionally, future work will explore hybrid models that integrate Genetic Algorithms with other optimization techniques or Machine Learning-based heuristics, to enhance performance and convergence speed. By improving both the adaptability and efficiency of GA-based systems, this work aims to develop a more scalable and resilient solution for modern logistics operations.

REFERENCES

[1] F. Altiparmak, M. Gen, L. Lin, and T. Paksoy, "A genetic algorithm approach for multi-objective optimization of supply chain networks," *Comput. Ind. Eng.*, vol. 51, no. 1, pp. 196–215, Sep. 2006, doi: 10.1016/j.cie.2006.07.011.

[2] L. Xin, P. Xu, and G. Manyi, "Logistics Distribution Route Optimization Based on Genetic Algorithm," *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–9, Jul. 2022, doi: 10.1155/2022/8468438.

[3] G. D. Sensi, F. Longo, G. Mirabelli, and E. Papoff, "ANTS COLONY SYSTEM FOR SUPPLY CHAIN ROUTES OPTIMIZATION".

[4] N. Mouttaki, J. Benhra, and G. Rguiga, "GENETIC ALGORITHM FOR OPTIMIZING DISTRIBUTION WITH ROUTE RESTRICTION CONSTRAINT DUE TO TRAFFIC JAMS," *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. XLIV-4/W3-2020, pp. 295–301, Nov. 2020, doi: 10.5194/isprs-archives-XLIV-4-W3-2020-295-2020.

[5] O. Samuel Sowole, "A Comparative Analysis of Search Algorithms for Solving the Vehicle Routing Problem," in *Optimization Algorithms - Classics and Recent Advances*, M. Andriychuk and A. Sadollah, Eds., IntechOpen, 2024. doi: 10.5772/intechopen.112067.

[6] "Research on multi-path optimization problem based on particle swarm optimization algorithm," *Theor. Nat. Sci.*, vol. 43, no. 1, pp. 156–161, Jul. 2024, doi: 10.54254/2753-8818/43/20240857.

[7] M. Zolfpour-Arokhlo, A. Selamat, and S. Z. M. Hashim, "Route planning model of multi-agent system for a supply chain management," *Expert Syst. Appl.*, vol. 40, no. 5, pp. 1505–1518, Apr. 2013, doi: 10.1016/j.eswa.2012.08.040.

[8] T. Rajora, A. Gaur, T. Kapoor, A. Kushwaha, Y. Prashar, and J. Parashar, "Implementation of Genetic Algorithm on Vehicle Routing System," *Eng. Technol.*, vol. 11.

[9] L. Judijanto, T. R. Fauzan, and B. Fisher, "Enhancing Logistic Efficiency in Product Distribution through Genetic Algorithms (GAs) for Route Optimization," *Int. J. Softw. Eng. Comput. Sci. IJSECS*, vol. 3, no. 3, pp. 504–510, Dec. 2023, doi: 10.35870/ijsecs.v3i3.1872.

[10] S. K. Jauhar and M. Pant, "Genetic algorithms in supply chain management: A critical analysis of the literature," *Sādhanā*, vol. 41, no. 9, pp. 993–1017, Sep. 2016, doi: 10.1007/s12046-016-0538-z.

[11] S. Nagy-Bota, L. Moldovan, M.-C. Nagy-Bota, and I. E. Varga, "Mathematical Models Used in the Optimizations of Supply Chains," *Acta Marisiensis Ser. Technol.*, vol. 20, no. 1, pp. 27–31, Jun. 2023, doi: 10.2478/amset-2023-0005.

[12] N. Topuria and O. Kikvidze, “Application of Genetic Algorithm in Common Optimization Problems,” *Int. Ann. Sci.*, vol. 8, no. 1, pp. 17–21, Jul. 2019, doi: 10.21467/ias.8.1.17-21.

[13] J. XueJing and Y. Xu, “Application of Genetic Algorithm in Logistics Path Optimization,” vol. 2, no. 1.

[14] H. Hu, “System Parameter Optimization based on Genetic Algorithm,” *Int. J. Mech. Electr. Eng.*, vol. 2, no. 3, pp. 11–16, May 2024, doi: 10.62051/ijmee.v2n3.02.

[15] M. F. Ibrahim, M. M. Putri, D. Farista, and D. M. Utama, “An Improved Genetic Algorithm for Vehicle Routing Problem Pick-up and Delivery with Time Windows,” *J. Tek. Ind.*, vol. 22, no. 1, pp. 1–17, Feb. 2021, doi: 10.22219/JTIUMM.Vol22.No1.1-17.

[16] W. Ho, G. T. S. Ho, P. Ji, and H. C. W. Lau, “A hybrid genetic algorithm for the multi-depot vehicle routing problem,” *Eng. Appl. Artif. Intell.*, vol. 21, no. 4, pp. 548–557, Jun. 2008, doi: 10.1016/j.engappai.2007.06.001.

[17] L. Ran, S. Ran, and C. Meng, “Green city logistics path planning and design based on genetic algorithm,” *PeerJ Comput. Sci.*, vol. 9, p. e1347, May 2023, doi: 10.7717/peerj-cs.1347.

[18] S. Kesik and C. Altıntas, “Development of a Genetic Algorithm for Vehicle Routing Problem in Military Logistics Distribution,” in *2023 4th International Informatics and Software Engineering Conference (IISEC)*, Ankara, Turkiye: IEEE, Dec. 2023, pp. 1–7. doi: 10.1109/IISEC59749.2023.10390997.

[19] Faizatulhaida Md Isa, Wan Nor Munirah Ariffin, Muhammad Shahar Jusoh, and Erni Puspanantasari Putri, “A Review of Genetic Algorithm: Operations and Applications,” *J. Adv. Res. Appl. Sci. Eng. Technol.*, vol. 40, no. 1, pp. 1–34, Feb. 2024, doi: 10.37934/araset.40.1.134.

[20] Z. Liu, J. Liu, F. Zhou, R. W. Liu, and N. Xiong, “A Robust GA/PSO-Hybrid Algorithm in Intelligent Shipping Route Planning Systems for Maritime Traffic Networks”.

[21] A. Maroof, B. Ayvaz, and K. Naeem, “Logistics Optimization Using Hybrid Genetic Algorithm (HGA): A Solution to the Vehicle Routing Problem With Time Windows (VRPTW),” *IEEE Access*, vol. 12, pp. 36974–36989, 2024, doi: 10.1109/ACCESS.2024.3373699.

ABBREVIATIONS AND SPECIFIC SYMBOLS

GA – Genetic Algorithm

ACKNOWLEDGMENT

The author would like to express sincere gratitude to the supervisors for their invaluable guidance and constructive feedback, which significantly contributed to the completion of this review paper. Additionally, the author extends appreciation for General Sir John Kotelawala Defence University for providing necessary resources and a supportive academic environment.

AUTHOR BIOGRAPHIES

GSM Jayasooriya, the first author of this review work, is a final-year undergraduate at General Sir John Kotelawala Defence University, Sri Lanka, pursuing a Bachelor of Science Honor’s Degree in Computer Science. Focused on advancing expertise in Artificial Intelligence and Machine Learning while preparing to transition into the professional world to make meaningful contributions to the field.

ADAI Gunasekara, the second author of this review work is a Senior Lecturer and former Dean of the Faculty of Computing at General Sir John Kotelawala Defence University, Sri Lanka. Currently he is contributing to the development of the field through lecturing, research, and industry collaborations.

Deep Learning Approaches for Classifying Informal and Formal English Texts Using Linguistic Features

KMGS Karunarathna[1#], RAHM Rupasingha[2] and B.T.G.S Kumara[3]
[1]Faculty of Graduate Studies, Sabaragamuwa University of Sri Lanka
[2]Faculty of Social Sciences and Languages, Sabaragamuwa University of Sri Lanka
[3]Faculty of Computing, Sabaragamuwa University of Sri Lanka
[#]gayathrisarangika599@gmail.com

ABSTRACT Effective techniques for automatically classifying texts are becoming increasingly necessary due to the exponential expansion of digital material. Differentiating between formal and informal documents can help students identify appropriate resources for their assignments and improve the effectiveness of information retrieval systems. Although machine learning is extensively utilized in classification of text, there is a lack of research focused to the effective differentiation of formal and informal writings through linguistic features. This gap highlights the necessity for advanced methodologies that improve classification accuracy and enhance the value of digital content in academic and retrieval systems. Our research addresses the problem by utilizing deep learning methodologies and a wide range of 13 linguistic attributes to get enhanced efficacy in text classification. Artificial Neural Networks (ANN), Convolutional Neural Networks (CNN), and Long Short-Term Memory Networks (LSTM) were considered. A dataset , including both formal (news articles, formal documents) and informal (personal letters, personal blogs) texts, were gathered from several web sources. We considered linguistic markers such as colloquialisms, contractions, modal verbs, slang, acronyms, pronouns, phrasal verbs, grammar complexity, vocabulary complexity, voice, and language type to generate the feature vector. The feature vectors were utilized to train and assess the classification models using several cross-validation techniques, particularly 3, 5, 7, and 10 folds. The efficacy of the models was evaluated using performance indicators, f-measure, accuracy, precision, and recall. With the highest accuracy of 99.8% and resilience in differentiating between formal and informal texts, the LSTM model outperformed than the others. Future research will examine big datasets, more linguistic characteristics, sophisticated deep learning models, and real-time and multilingual classification systems.

INDEX TERMS ANN; CNN; Document Classification; Formal Documents; Informal Documents; LSTM

I. INTRODUCTION

In the current highly competitive global environment, students and researchers have the internet as their main source of educational information, thus navigating a web of resources. It has also changed the way instructions are given and provided access to a vast amount of instructional information. As important reference sources, educational websites offer a variety of materials including academic papers, journal articles, educational manuals, and current event updates. The Ken showed the how important of using formal writing in academic materials [1]. The authors who write for these environments can use either the formal or the informal writing style.

In the field of education, researchers and students employ various papers to enhance their knowledge, gain understanding, and obtain data. These resources include a variety of text types and the learners have to choose the most appropriate one at any given time. Thus, the ability to distinguish between formal and informal writing is crucial to the process of learning. The importance of the distinction between different writing styles is becoming more important and is highlighted in the example described. The contrast between these examples highlights the importance of knowing the audience and how to use specific language for the audience. The ability to recognize and understand the differences between formal and informal language and understand how to use them appropriately has become more important as digital technologies and the internet develop. If it could be automated whether the document is formal or informal, it would make the students' task easier and permit them to take out probably more concise the relevant information.

Currently, information retrieval is a laborious and time-consuming operation due to the unorganized nature of web resources. Agnes represented from his research how formal and informal language act through the internet [2]. As of today, information retrieval is a laborious and time-consuming process due to the unorganized nature of web resources. A possible solution is to apply document classification, which is the practice of labelling previously unlabelled documents. This method increases extracting the useful information from various large number of digital resources. This automatic classification system can be used to dramatically improve the access and use of educational resources by students. Ultimately, this will improve students' success rates in academics.

The objective of our study is to automatically classify documents as either formal or informal style using language features distinctive of the respective styles. We aim to construct a model that helps learners make distinctions between formal and informal documents, enabling the learner to more effectively choose the appropriate resources for their individual learning needs.

For this study computational linguistic gives the significant support. Computational linguistics involves the utilization of computer technology for the study and understanding of both formal and informal language. Computational Linguistics is an academic domain that integrates linguistics, computer science, and artificial

intelligence to analyse language from a computational perspective [3].

Recent studies highlight that incorporating linguistic features such as syntax, sentiment, and morphology improves text classification accuracy. Research on psychological state classification demonstrates the effectiveness of linguistic feature integration [4]. A 2024 survey on computational morphology underscores its role in enhancing classification models [5]. Additionally, a comparative study found that combining neural networks with linguistic features leads to more precise results [6] , reinforcing the importance of linguistic analysis in style-based classification.

In the field of Computational Linguistics, the classification of texts by style requires linguistic component analysis. Specific decisions on syntax and word and phrase choice affect style, according to Karlgren. The use of vocabulary becomes an indicator of style; for instance, long, complex sentences are typically interpreted as being formal, but sentences that incorporate informal idiomatic expressions are considered to be informal most often. In addition, numerous additional factors influence whether a piece of writing is formal or informal, and formal and informal writing has a wide range of sub-genres. It is essential to first determine the primary traits that set formal writing apart from informal writing to create an efficient classification model. These features include variations in vocabulary utilization overall as well as word choice, phrases, and expressions. First of all, we need to define what are the main characteristics that distinguish formal writing from informal writing to obtain an effective classification model. These characteristics are reflected in overall vocabulary usage as well as specific word choices, phrases, and expressions. By capturing these we can build a model that can consistently classify documents and hence provide students with a tool for navigating the vast amount of material offered. This characteristic will enable students to more easily and efficiently access the most relevant and suitable information needed for their learning tasks saving also time.

A. Formal and informal writing style

From the formal and informal language that uses the most precise, objective, and specific terminology one can refer to a formal style. This type of language is used in academic books and articles, technical reports, research papers as well as legal documents. Formal style finds in scholarly writing and professional documentation the kind of expression required by accuracy, clarity, and attention to detail.

An informal style of writing, or speaking, is a friendly, warm approach that is used to conduct writing or in everyday conversation. It is employed when writing for or conveying to close friends, relatives, etc, and in general conversation. This sort of style may be used in personal emails and face-to-face conversations.

The following table 1 show the key features of formal and informal writing styles. Here we identified important 13 linguistic features with referencing [7] and [8].

Table 1. Key features and examples for formal and informal writing styles

Features	Formal	Informal
Colloquialisms [7]	Do not use colloquialisms Ex: Want to	Use colloquialisms Ex: Wanna
Contractions [7]	Do not use contractions Ex: Can't	Use contractions Ex: Can not
Pronouns [7]	Third person pronouns Ex: This could be good research	First-person pronouns Ex: I think this good research
Phrasal verbs [7]	Do not use phrasal verbs Ex: Spend time	Use phrasal verbs Ex: Hang out
Grammar [8]	Complex grammar Ex: Global warming, which results in several major environmental concerns, is one of the most prevalent environmental challenges.	Simple grammar Ex: Global warming is an environmental challenge.
Modal Verbs [8]	Used modal verbs Ex: This methodology can be applied to the research	Do not use modal verbs Ex: This methodology used for the research
Vocabulary [8]	Ex: Purchase	Ex- Buy
Abbreviations [8]	Do not use abbreviations Ex: Examination	Use abbreviations Ex: Exam
Acronyms [7]	Do not use acronyms Ex: As soon as possible	Use acronyms Ex: ASAP
Slangs [7]	Do not use slang Ex: Friend	Use slangs Ex: Mate
Initialism [7]	Do not use initialisms Ex: United Kingdom	Use initialism Ex; UK
Language [8]	Formulaic language Ex: The experiment's result showed the validity of the accuracy.	Direct language Ex: My parents are very well.
Voice [9]	Passive Voice Ex: dissertations are written by scholars.	Active Voice Ex: Scholars write dissertations

B. Research questions and objectives

Figure 1 shows the mapping between research questions and objectives.

Distinguishing between formal and informal writing styles is crucial for learning and understanding. This study aims to automatically classify documents as formal or informal using above mentioned 13 language features distinctive of each style. After collecting formal and informal documents, we applied different pre-processing techniques and then extracted the features. Then, research evaluates and compares different deep learning models such as

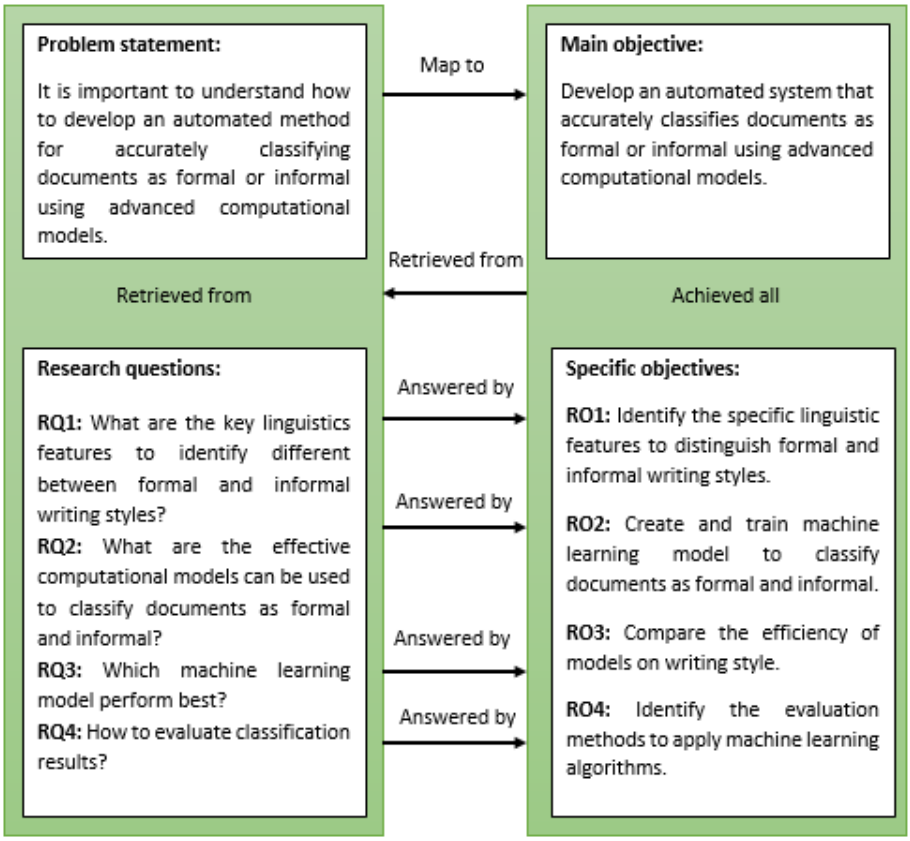

Fig. 1.Mapping between research questions and objectives

ANN, CNN, and LSTM to identify the best classification method. The evaluation is performed based on accuracy, precision, recall, and f-measure to better understand how these models are in classifying texts based on their writing style. This will ensure that the performance of models will be systematically analysed to choose the best model for document classification tasks between formal and informal texts.

The rest of this study is organized as follows: Related works on writing style-based document classification are investigated in Section 2. In Section 3 we present a novel automated mechanism for labelling documents in formal and informal style. The studies conducted on the effectiveness of our method, as well as the metrics and measurement methods applied, are detailed in Section 4.

Section 5 concludes the paper, summarizes findings, and proposes future research directions.

II. LITERATURE REVIEW

A. Related work

In the literature review, we look at previous studies and approaches to the problem of classifying documents according to writing style, with a particular emphasis on attempts to automate the differentiation between formal and informal styles by linguistic and computational methods.

Classes of Hindi poetry were identified by [10] as part of their suggested categorization scheme: Shringar, Karuna, and Veera. They generated their feature vectors using the Bag of Words model. Five machine learning classification techniques were classified: Support Vector Machines (SVM), Random Forest, Decision Tree, K-nearest Neighbors (KNN), and Naïve Bayes. Tests revealed that in this instance, the accuracies of Random Forest and Naïve Bayes outperformed those of the other algorithms. A university website was utilized as a case study in [11] and the Waikato Environment for Knowledge Analysis (WEKA) machine learning workbench served as the framework for the machine learning effort. The classification model was created using the Naïve Bayes method. The study's findings demonstrated that the Naïve Bayes algorithm is capable of accurately classifying massive online content. On the other hand, the researchers employed a single classification method. Various feature vector types were covered by [12] to represent and subsequently classify documents. Regarding their impact on vector and document classification, the study contrasted the Binary, Count, and Term Frequency–Inverse Document Frequency (TF–IDF) feature vectorsThey employed the Naïve Bayes classification for every feature vector representation. Once more, just one classification method was utilized in this study. In [13]examined many categorizations of methods, including Decision Tree, SVM, and Naïve Bayes, in their survey study. Additionally, they examined assessment metrics that may be used for different types of study, including accuracy, F-measure, and G-measure. They only examined three classification methods, though. Using the Naïve Bayes methodology, [14]) developed a classification system for Turkish papers. F-measure, recall, and accuracy were used to gauge performance. Once more, the model in this study was constructed using a single classification method. A research paper classification method that can group research articles into relevant groupings based on related issues was suggested [15]. The TF-IDF values of each

paper were utilized to classify whole papers using the K-mean clustering approach. Keywords in the abstracts were represented using the Latent Dirichlet Allocation technique. Here, the drawback was that the research publications were classified using just one classification system.

The WEKA text mining technology was utilized by [16] to assist in the classification of online frauds. The classification models were constructed using the J48 Decision Tree, Sequential Minimal Optimization, and Naïve Bayes techniques. They discovered that J48 produced the best results among these techniques in terms of accuracy and error rate. The authors of [17] presented a neural network design that demonstrated how effectively both compression and iteration layers could encode character input. They evaluated the suggested model on eight extensive document classification tasks and contrasted it with a character-level convolution-only model.

In eight languages, a new subset of the Reuters Corpus with balanced class priors was proposed by [18]. In terms of phrases and morphology, they studied four extremely diverse languages: Italian, Russian, Japanese, and Chinese. Multilingual word and phrase embedding in all language translation directions, provide a solid basis. The Bidirectional Encoder Representations from Transformers (BERT) model was initially used to document categorization [19]. They demonstrated how BERT may be used to create a basic classification model for four widely used datasets.

In [20] presented a novel neural network-based model that measures uncertainty using a newly developed dropout entropy technique. A state of uncertainty concerning the appropriate course of action is referred to as uncertainty. Additionally, they presented a metric learning system based on dropout approaches that enhances feature representation for uncertainty. In their accurate prediction studies, this exhibited negligible predictive variability. Specifically, their model's accuracy increased from 0.78% to 0.92% when 30% of the most ambiguous results were applied to the "20 Newsgroups" dataset used by anthropologists. A two-stage text-document classification system that combined automated and conventional feature engineering was presented by [21]. The suggested process consisted of deep component modification using filters in conjunction with a neural network selection algorithm. The "20 Newsgroups" dataset and the BBC News dataset, two of the most popular public datasets, were utilized to assess this technique. The findings of the evaluation indicate that the suggested methodology surpasses the current state-of-the-art approaches that rely on deep learning and machine learning. For "20 Newsgroups," the margins were 7.7%, while for BBC News, they were 6.6%. Note that neural networks were the main focus of this study.

In [11] used the LitCovid database, an expanding collection of 23,000 research publications on the new 2019 Coronavirus, to conduct a study of alternative multiple-label document classification methods. By a narrow margin, they discovered that the pre-trained language models outperformed all other baseline models on this dataset, with BioBERT and micro F1 achieving accuracy scores of around 86% and 75% of the test set, respectively. In further research, [22] examined the processing of documents from different time intervals with document classification systems that were trained on documents from certain intervals, taking into account both seasonal and non-seasonal intervals. In [23] introduced a notably intricate document representation format that employs in-depth learning to automatically classify financial documents into predetermined categories. The two primary components of their model architecture were document classification and representation. Their results demonstrated that the best document classification performance for the INFUSE dataset was obtained by a hidden three-layer nutrient network consisting of 1024 neurons. In [24] concentrated on classifying information using KNN and Naïve Bayes, two of the six primary techniques. The TREC Code of Conduct, which can be downloaded and contains over 3,000 text documents with over 20 categories, served as the corpus for this study. After processing the data with RapidMiner, k=13 was found to be the ideal value for k in the KNN. This number for k resulted in an average accuracy of 55.17%, which was higher than Naïve Bayes' 39.01%. We do see, though, that they only employed two classification techniques.

In [25] used a majority vote to classify assignments using five well-known classification techniques for the Urdu language corpus. 21,769 news articles across seven categories could be found in the corpus. The papers were pre-processed data using tokenization, stop word removal, and a rule-based stemmer because the algorithms were unable to operate directly on the data. Following pre-processing, the data has 93,400 characteristics taken out. With a majority vote, the outcomes had a 94% recall and accuracy rate. Lemmatization and lower-case conversion were not used as pre-processing methods in this study. The goal of the study [26] was to use a Naïve Bayes classification algorithm to estimate a song's singer based only on the lyrics. The 207 songs that Metallica and Nirvana played were included in the dataset that was developed. With an F-measure of 0.94, a recall of 0.95, and an accuracy of 0.93, the model assessment procedures produced extremely positive results. As a result, words classified with Naïve Bayes were deemed successful. This study, however, only employed one classification method.

Machine learning and automated document classification approaches are discussed in [27] article, which lessens the workload for domain experts by managing a huge number of news items or web page information. The authors spoke about a variety of design choices and factors that often come up while creating automatic classifiers with the help of XML document structure. Using two datasets, experimental research on automated classification has been conducted in this work. New texts were classified using the learned Naïve Bayes classifier. The outcomes were contrasted using a hierarchical classifier and a KNN. Real-time web apps may employ this kind of capability because of the incredibly optimistic categorization performance demonstrated by these trials.

The paper [28] addressed an important issue in computational linguistics: distinguishing between formal

and informal styles of text in document classification and text generation. They proposed two main techniques to solve this task namely building a model to classify any text or sentences and a second technique based on natural language generation (NLG). They started with summarizing characteristics of formal and informal writing styles and manually collected parallel lists of formal and informal words. Then they built the model using a Decision Tree, Naïve Bayes, and a Support Vector Machine (SVM). The evaluation results showed that that model can predict a class and formal and informal for any text or sentence with high accuracy. Then they built a system that can generate formal and informal sentences by using NLG techniques. However, they did not focus on every characteristic of formal and informal writing styles as well as they used only three machine-learning models for the experiments. The study done by, [29] aimed to work with machine learning techniques to deliver a model that can distinguish formal and informal texts. They explored the Decision Tree, Random Forest, and Logistic Regression as machine learning algorithms. As a result, they found that features are crucial in text formality

classification. As well as their study included how to transform informal text into formal text and vice versa. In this study, they experimented with only five key features to identify the difference between formal and informal writing styles and also used only machine learning algorithms.The research article [30] proposed a document classification method based on formal and informal styles. That research used 200 text documents and used the tf-idf feature. In the classification process Decision Tree (J48), Random Forest, Multilayer Perception (MLP), and SVM as machine learning models. The result was investigated using 5-fold cross-validation and the Random Forest algorithm showed the highest accuracy. This experiment used a few data sets, and because of this, there can be problems with accuracy. The paper [31] presented a method to identify automatically the style of a particular

document. For the feature extraction used tf-idf and five classification algorithms with an ensemble learning approach. From them, Random Forest showed the highest accuracy but after being compared with the Ensemble learning algorithm it represented the highest accuracy. As the results they investigated a higher accuracy can be obtained by adding several algorithms. In this study not used any deep learning algorithms to combine with other algorithms.

The research article [1] proposed a method to identify the academic documents and as the methodology they used Tree tagger. According to their finding it was essential to identify academic documents for the educational purposes. The paper [2] presented usage of formal and informal language for the communication. They used qualitative method for investigations. As the finding they showed that for mobile communication most probably used the informal language.

Recent studies have advanced the classification of formal and informal English texts by integrating deep learning techniques with linguistic features. [32] conducted a systematic study comparing statistical, neural-based, and Transformer-based methods for formality detection, highlighting the effectiveness of neural networks in capturing linguistic nuances. Additionally, [33] study proposed a concept for using linguistic knowledge to support binary text classification, emphasizing the importance of linguistic-driven feature selection in enhancing model performance. Furthermore, [34] study introduced deep learning models, such as Formality-LSTM and Formality-BERT, for formality prediction

Table 2. Summary of literature

Paper	Paper purpose	Methodology	Limitations
[1]	How used formal writing in academic writing	Tree Tager	Not used the deep learning algorithms
[2]	Usage of formal and informal language for communication	Qualitative method	Not used qualitative method to get the accuracy. Not used deep learning algorithms
[10]	Classification approach for Hindi poetry	Random forest SVM Decision Tree KNN Naïve Bayes	Not using any deep learning algorithms.
[28]	Distinguish between formal and informal styles of text in document classification and text generation.	Decision Tree Naïve Bayes SVM	Focused only on eight characteristic Used only three machine learning models for the experiments.
[29]	Work with machine learning techniques to deliver a model that can distinguish formal and informal texts.	Decision Tree Random Forest Logistic Regression	Experimented with only five key features. Used only three machine learning models for the experiments.
[30]	Document classification method based on formal and informal styles.	Decision Tree Random Forest MLP, SVM	Used a few data sets. Not specify what are the characteristics used.
[31]	Identify automatically the style of a particular document.	Decision Tree Random Forest MLP, SVM, Naïve Bayes Ensemble learning approach	Not using any deep learning algorithms. Not discussed features separately.

without the need for feature engineering, demonstrating the potential of deep learning approaches in this domain.

These studies collectively underscore the pivotal role of combining deep learning models with linguistic features to effectively classify texts based on formality.

B. Research gap

Despite significant progress in the field of computational linguistics, automatic document style classification, that is, the identification of formal and informal writing styles, is still a difficult and emerging field of study. However, the studied literature provides certain helpful insights and outlines the general approaches and methods that need further research due to the existing gaps and limitations.

As for prior studies, they have successfully investigated language features that are essential for text formality determination; however, the scope of their findings has often been limited to several features. It is essential to address a gap in the literature by compiling 13 different linguistic features into a single feature vector. Some of these characteristics are subjectivity and objectivity markers, use of slang, number of words in a sentence, number of formal and informal words used, use of contraction, number of passive voices used, etc. By systematically dealing with these processes, classification models improve their stability and accuracy and thus provide a more accurate classification of text formality. The most frequently used algorithms include Decision Tree, Naïve Bayes, Random Forest, SVM, and other similar methods. The use of complex deep learning techniques such as CNN, LSTM, and ANN networks has not been explored in detail, although there is potential in these approaches. The lack of research on deep learning-based solutions due to obstacles including the requirement for substantial processing resources and the availability of large, high-quality datasets, which are frequently limited. The intricate nature of implementation and the restricted interpretability of deep learning models could discourage academics from investigating these methodologies. Conventional methods, due to their greater accessibility and interpretability, are frequently favored across several domains. But it is possible that deep learning methods may be more efficient in detecting complex patterns in textual data. It is also possible to increase the classification accuracy by using deep learning algorithms.

Our research bridges important knowledge gaps through the implementation of 14 linguistic elements to construct an extensive feature selection method for document classification. Detailed analysis through this approach produces better distinctions between formal texts and informal texts in classification tasks. The comparison of advanced deep learning models ANN, CNN and LSTM for specific performance stands as our unique contribution which provides essential insights into their effectiveness. The combined elements in our study represent a major breakthrough for computational linguistics and text classification research direction.

Some of the studies involved a small number of documents in the dataset, which may have affected the results' reliability. Expanding the size of the dataset to include more documents could improve the performance of the classification algorithms.

Our work attempts to address these research gaps by developing a better classification model that involves several language features, advanced deep learning algorithms, and comprehensive evaluation metrics. It might significantly enhance the effectiveness of resources and information in educational and other digital content systems. Table 2 showed the limitations of the previous research as well.

III. METHODOLOGY

The following Figure 2 represents the proposed approach for the study.

A. Data Collection

In the present research, we selected several 5000 text files from several reliable web sources. These data are considered secondary because these data were originally collected for other purposes on the Internet. A suitable number of formal and informal text documents are incorporated into the collection.

We gathered new articles and some formal documents from the Kaggle website [35] and the Washington Post news website [36] to represent formal documents. These articles are excellent examples of formal writing because of their organized framework, formal tone, and respect for journalistic norms. We gathered personal letters and personal blogs for informal documents from the following websites: Writing Help Central [37], AnswerShark [38], Letters Free [39] and extra documents from Kaggle [35]. These personal letters are great examples of informal writing since they are usually written in a more relaxed and conversational tone.

The dataset only included text files in English to maintain linguistic consistency. Using the criterion of writing style and origin, every document was pre selected as formal or informal. This process of selection and classification helped us generate a large number of documents that are representative and diversified enough to be used for testing and training our classification algorithms. Table 3 showed some examples for formal and informal writing styles.

Table 3. Examples of formal and informal writing styles

Formal Writing Style	**Informal Writing Style**
The project's budget has been updated in light of current financial modifications.	We've changed the project's budget because of some recent financial adjustments
After the week, we will offer suggestions after thoroughly reviewing the plan.	Right now, we're looking over the proposal and will get back to you by the end of the week

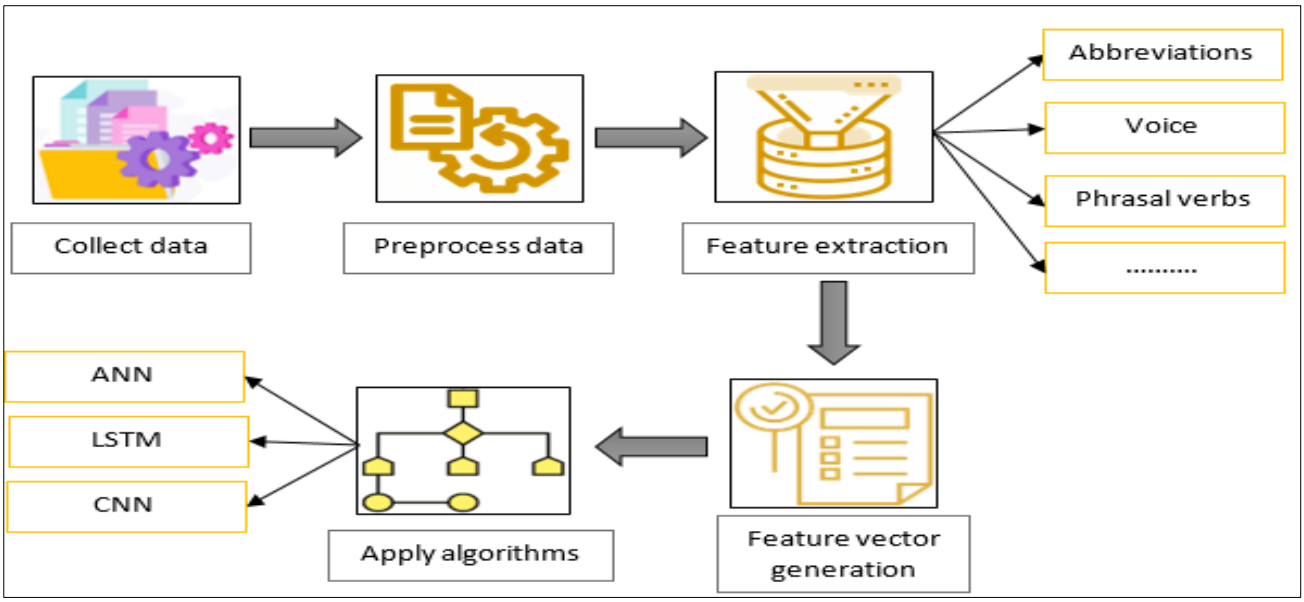

Fig. 2. Proposed Approach

B. Pre-process Data

Text preprocessing mainly involves the process of splitting the text file into words and removing unwanted elements. Four essential preprocessing procedures were involved: This includes lemmatization, lower-casing, stop word removal, and tokenization. These are important processes that help to clean and normalize the text data in a format that is suitable for further analysis and classification.

Tokenization: Tokenization is a process of segmentation of a text sequence into words, phrases, symbols, or other meaningful parts called tokens. This is a very important phase in text processing because it divides the text into segments that can be analysed separately, this is important for further stages.

Lower casing: Before the classification, all capital characters should be converted to lowercase. This phase ensures that all the text data is in the same feature space by projecting every word in the text file. For instance, after lower casing the strings "Education" and "education" as well as "EDUCATION" would be considered similar since it eliminates the differences in case sensitivity.

Stop word removal: To focus on the more informative and content-related words that are more relevant in distinguishing between the formal and informal texts, the stop words are removed before classification. They are words such as 'and', 'the', 'is', etc. which are normally insignificant when used in a categorization context. They have the capability of skewing the data and therefore compromise the efficiency of the classification process.

Lemmatization: Lemmatization is a technique of natural language processing that removes the affixes and prefixes from the words to bring them back to their stem. Lemmatization considers the context and returns the actual base form of the word. One might lower the terms "running," "ran," and “runs” to their stem, “run”.We were certain that the unprocessed text data was well-pre-processed by following the above-mentioned preprocessing steps. It will help enhance the current classification models to be able to differentiate between the formal and informal writing styles better.

C. Feature Extraction

One of the most important steps in converting pre-processed text input into measurable metrics that machine learning algorithms may use is feature extraction. We retrieved features that capture 14 different linguistic aspects indicative of formal and informal writing styles from our dataset of 5,000 text files. The following is a description of the characteristics and how they were extracted. Using algorithms developed in the Python programming language, the 14 distinct linguistic elements representative of formal and informal writing styles were extracted. In particular, we used libraries for natural language processing tasks like spaCy and NLTK. We converted our pre-processed text data into measurable metrics that machine learning algorithms could evaluate and use by applying these algorithms to them. The 13 distinct linguistic elements are as follows:

Colloquialism: The phrases and expressions used in colloquialism. These are the kinds of terminology used in informal writing. For every document, we numbered and recognized. First, we gathered all colloquialisms through the internet and applied the algorithm to count the colloquialisms in our data set.

Abbreviation: In informal communications, abbreviations are more frequently utilized. Each document's usage of abbreviations was counted. Gathered a list of abbreviations through the internet and according to that identified the abbreviations in every document using an algorithm.

Contraction: Writing that uses contractions is considered informal. We tallied how many contractions each text

included (e.g., "don't," "can't"). We created seven rules to identify the contractions of the dataset. Rules are:

- "Rule 1": "(?:\w+)'m"
- "Rule 2": "(?:\w+)'s"
- "Rule 3": "(?:\w+)'re"
- "Rule 4": "(?:\w+)'ve"
- "Rule 5": "(?:\w+)'d"
- "Rule 6": "(?:\w+)'ll"
- "Rule 7": "(?:\w+)'t"

Modal Verbs: Probably, formal writings employ modal verbs. For every document, the frequency of modal verbs was determined. The word list is "shall", "should be", "can", "could", "will", "would", "may", "must", and "might". Those words are identified in the data set.

Slang: Slang is an effective means to determine whether writing is informal. A list of slang phrases was compiled with the sample slang gathered from the internet.

Acronyms: Both formal and informal situations employ acronyms, however they may not be used as frequently. Each document's total number of acronyms was tallied reference acronyms gathered from the internet.

Initialism: Every document's initialization frequency (e.g., "FBI," "ASAP") was tallied. Though they are also used informally, initializations are frequently used in formal settings.

Pronouns: First, identify the first person and third person pronouns and then according to that separately take it as two characteristics.

First-Person Pronouns: How many first-person pronouns (such as "I," or "we") were used? In informal writing, first-person pronouns are more frequently used.

Third-person pronouns: (such as "he," "she," and "they") were also counted. In formal writing, third-person pronouns are frequently employed.

Phrasal Verbs: Two rules were created for identifying the phrasal verbs in the dataset as follows and separately take it as two characteristics.

Verb + Adverb: Phrasal verbs, which are verbs followed by an adverb (such as "give up"), were recognized and tallied.

Verb + Preposition: Phrasal verbs, which are verbs followed by a preposition (such as "look into"), were also counted. Phrasal verbs are usually more prevalent in writing that is informal.

Grammar: Define a set of dependency labels that indicate complex structures of the sentences in a dataset. The following two features separately take as two characteristics.

Complex Grammar: A count was conducted to determine the frequency of complex grammatical structures, such as compound-complex phrases. Formal compositions tend to use more complex syntax.

Simple Grammar: The usage of basic grammatical constructions, such as simple phrases, was also counted. Writing that is informal frequently uses simple grammar.

Vocabulary: The percentage of complex phrases in each text was calculated to determine the vocabulary's complexity. Considering words with more than six letter as complex. More complicated language in writing is a sign of formal writing.

Voice: By creating the rule identify the active and passive voice sentences in every document. We will look for the 'VB' (verb base form) or 'VBD' (past tense verb) followed by 'VBN' (past participle verb) in the tagged words. If we find this pattern and 'BY' is not present in the sentence, we consider it passive and both are considered separately as two characteristics

Active Voice: We tallied how many times each document used an active voice structure. Informal writing tends to use the active voice more frequently.

Passive Voice: The instances of passive voice compositions were also tallied. In formal writing, the passive voice is generally more prevalent.

Language: First, count the frequency of each token. Then calculate the total number of tokens. According to that calculate the frequency of common phrases in the sentence as well as calculate the ratio of common phrases to total tokens. After that classify the language based on the ratio. If it is formulaic language return 0, direct language return 1, and for the mixed language return 2.

This allowed us to transform the textual data into quantitative representations that reflect the nature of

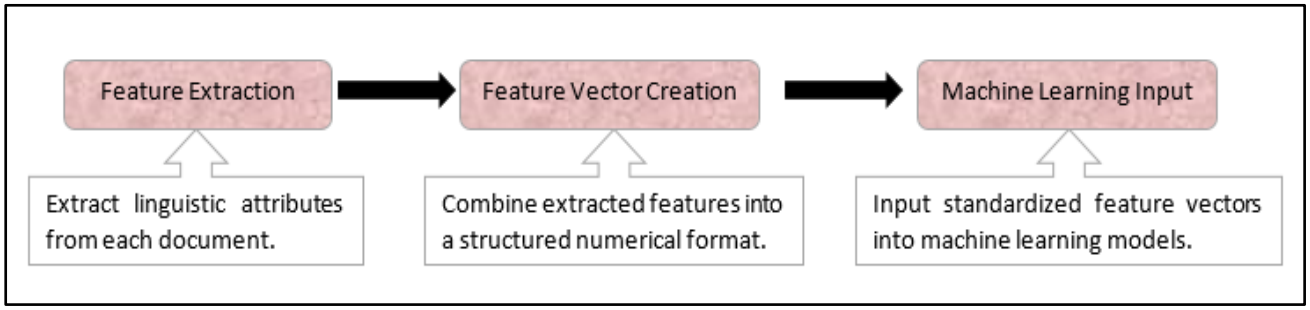

Fig.3. Process of feature vector generation

Colloquialism	Abbreviation	Contraction	F/S Pronouns	Third Pronouns	Phrasal verbs (verb+adverb)	Phrasal verbs (verb+preposition)	Simple Grammar	Complex Grammar	Modal Verbs	Vocabulary (Complexity %)	Slangs	Active Voice	Passive Voice	Acronyms	Initialism	Language Formulaic Language =0 Direct Language =1 Mixed Language =2	Target Variable
0	63	0	0	0	2	1	4	5	2	33	4	8	1	5	1	1	0
0	66	0	0	0	0	2	4	6	1	30	0	9	1	3	1	1	0
1	70	0	0	0	1	0	3	12	0	39	0	12	3	1	1	1	0
0	55	2	0	0	2	0	5	6	3	43	0	10	1	0	3	1	0
1	73	1	0	0	0	1	9	9	1	33	3	15	3	2	0	1	1
1	72	0	0	0	0	0	3	12	1	40	1	12	3	2	2	1	1

Fig. 4 Sample for feature vector generation

formal and informal writing with the help of these characteristics extracted. These attributes are fed into our machine learning models which in turn enables them to learn and differentiate between different writing styles.

D. Feature vector generation

As shown in Figure 3, feature extraction is the first step of transformation which is used to transform the text data into feature vectors that are used by machine learning models.

In this step, linguistic features are detected and counted in each document. Linguistic features can be, for example, contractions, abbreviations, or colloquialisms. In the next step, i.e., feature vector creation all detected linguistic features are put in predefined structured numerical representation. The same representation is used for all documents to ensure the standardized form final numeric representation that is given as an input to machine learning models which analyze generated patterns and characteristics to classify writing styles and increase the predictive capabilities of machine learning models.

Figure 4 shows the sample for feature vector generation. Here explain the different features (characteristics) in different columns as mentioned above and the last column is for two target variables as formal (1) or informal document (0). Structuring the data in this manner ensures that our methodology properly captures the distinctive aspects of formal and informal documents using all 13 linguistic features. This generated feature vector provides to the classification models and it enhances the overall efficacy of our methodology.

E. Apply algorithms

The study utilized ANN, CNN, and LSTM algorithms to analyse the dataset. ANN handles complex pattern recognition, CNN excels in feature extraction, particularly for grid-like data, and LSTM is effective for sequential data due to its ability to capture long-term dependencies.

ANN: Each layer of an ANN is composed of several neurons or perceptron. An ANN only processes inputs in a forward direction and therefore is also called a feed-forward neural network [40]. Figure 5 explains the architecture of ANN algorithm

CNN: Today CNNs are one of the most popular models for deep learning. This neural network computational model includes one or more layers of convolution that can be either pooled or fully connected. It is based on a type of multilayer perceptron [40]. Figure 6 showed the architecture of CNN algorithm.

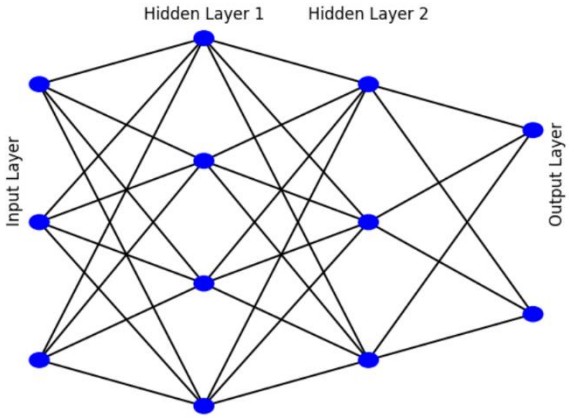

Figure 5. Ann Architecture

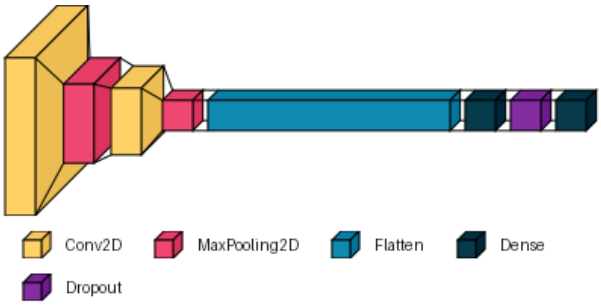

Figure 6. CNN Architecture

LSTM: In the LSTM network, the memory cell is regulated by the input, forget, and output gates. These gates define the input, output, and the addition of data in this memory cell. Therefore, LSTM networks can learn long-term dependencies and the information that flows through the network is either retained or rejected [41]. LSTM model showed by following Figure 7.

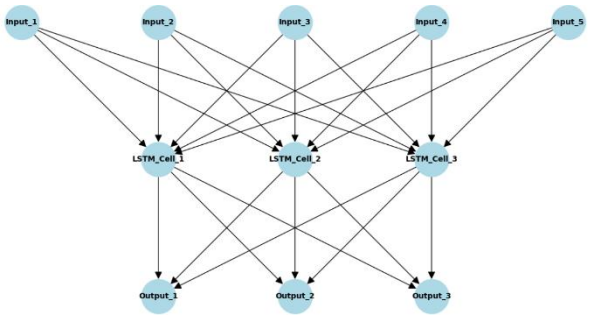

Figure 7. LSTM Architecture

Table 4. Confusion matrix

	5 Fold				10 Fold				3 Fold				7 Fold			
	TP	**TN**	**FP**	**FN**	**TP**	**TN**	**FP**	**FN**	**TP**	**TN**	**FP**	**FN**	**TP**	**TN**	**FP**	**FN**
ANN	411	437	88	64	198	234	40	28	693	724	142	107	294	324	55	41
LSTM	588	392	8	12	291	194	6	9	960	639	26	41	425	282	3	5
CNN	449	427	73	51	234	201	36	29	766	680	123	98	337	292	58	28

IV. RESULTS AND DISCUSSION

The experimental platform used Microsoft Windows 10 on a PC with Processor Intel® Core (TM) i5-8250U CPU @ 1.60GHz, RAM 4.0GB.

For this study used Python programming language and Jypyter Notebook as the platform with the cross-validation namely 5 fold, 10 fold, 7 fold, and 3 fold. These metrics have been calculated for test dataset. The test dataset comprised 20% of the overall dataset, facilitating a comprehensive assessment of the models' efficacy in distinguishing between formal and informal texts.

A. Confusion matrix

The confusion matrices for the various cross-validation folds (5, 10, 3, and7) of the classification algorithms (ANN, CNN, and LSTM) offer a comprehensive. understanding of each model's performance in terms of

true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). This study was conducted using the data that was supplied as shown in table 4.

C. Accuracy of classification algorithm

In this study, we used several cross-validation folds (3-fold, 5-fold, 7-fold, and 10-fold) to assess the performance of three machine learning algorithms: ANN, CNN, and LSTM.

Figure 8 below provides an overview of each algorithm's

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

All folds consistently demonstrated that the LSTM model outperformed the other, achieving the highest accuracy of 99.8% in the 7-fold cross-validation. Even in the 3-fold cross-validation scenario, the LSTM model's lowest accuracy of 96.4% surpasses the maximum accuracies attained by the ANN and CNN models. This highlights the superiority of the LSTM model in our classification task due to its adeptness.

D. Results of precision, recall and f-measure

We used precision, recall, and f-measure across various cross-validation folds (3-fold, 5-fold, 7-fold, and 10-fold) assess the performance of the classification algorithms, which are ANN, CNN, and LSTM, in addition to accuracy. All three methods had their precision, recall, and F-measure computed; the computations for these metrics are detailed in the following (2), (3), and (4).

$$Precision = \frac{\text{TP}}{TP + FP} \quad (2)$$

$$Recall = \frac{\text{TP}}{TP + FN} \quad (3)$$

$$F - Measure = \frac{2 * Precision * Recall}{Precision + Recall} \quad (4)$$

Table 5 below provides a summary of the comprehensive result.

Table 5. Results of the performance matrix

		ANN	**LSTM**	**CNN**
3 Fold	**Precision**	0.85	0.97	0.87
	Recall	0.85	0.97	0.83
	F-measure	0.85	0.97	0.85
5 Fold	**Precision**	0.86	0.98	0.88
	Recall	0.86	0.98	0.84
	F-measure	0.86	0.98	0.86
7 Fold	**Precision**	0.86	1.00	0.88
	Recall	0.86	1.00	0.85
	F-measure	0.86	0.99	0.86
10 Fold	**Precision**	0.86	1.00	0.88
	Recall	0.86	0.97	0.84
	F-measure	0.86	0.98	0.86

All folds consistently demonstrated precise precision, recall, and f-measure values, highlighting the outstanding performance metrics of the LSTM model. The 7-fold cross-validation obtained the highest values, achieving an F-measure of 0.99, with recall and precision both reaching 1.00. The LSTM exhibited proficiency in distinguishing between formal and informal texts, showcasing remarkable metrics across 3-fold and 10-fold cross-validations.

D. Results of MAE and RMSE

Then, using the formulas in (5) and (6), we determined the Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) for each of the six methods. In this case,

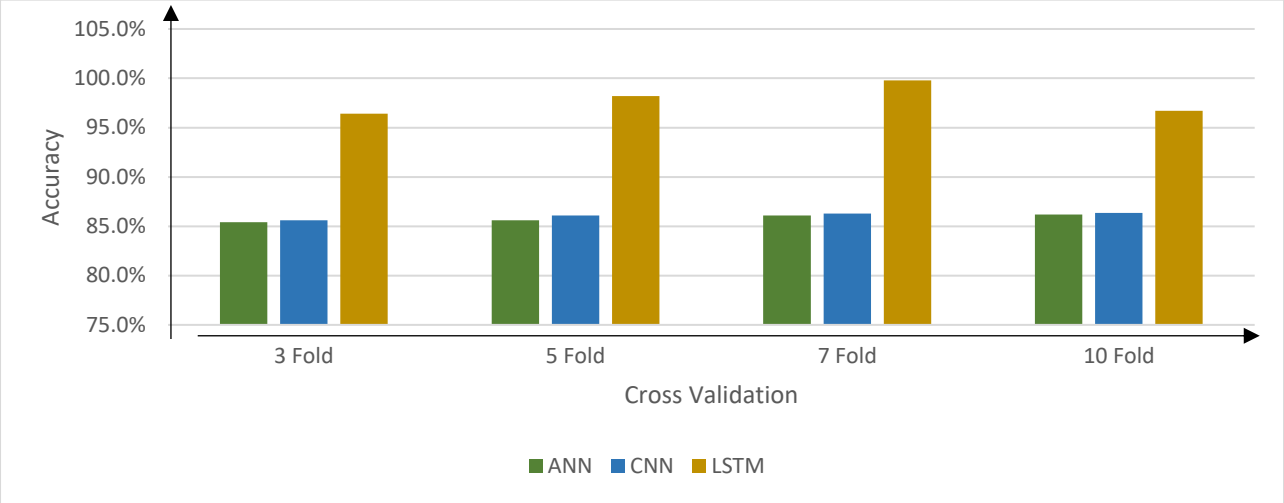

Fig 8. Accuracy of classification algorithms

Mvx is the expected result, Pvx is the current labeled value depending on the outcome, and T is the number of projected values.

$$MAE = \frac{1}{T}\sum_{x=1}^{T} | \, p_{vx} - M_{vx} \, | \qquad (5)$$

$$RMSE = \sqrt{\frac{1}{T}\sum_{x=1}^{T}\left(p_{vx} - M_{vx,}\right)^2} \qquad (6)$$

To assess the models' performance, the MAE and RMSE was also computed, as shown in Figure 10 and 11.

The MAE values of both the ANN and LSTM models remained consistent across all folds, with the ANN slightly surpassing the LSTM in terms of error rate. Likewise, similar to the ANN model, the LSTM model showed steady RMSE values (ranging between 0.37 and 0.38) across each. The low and stable error rate of the LSTM model indicates reliable performance.

Furthermore, each classifier's hyperparameter and optimum values were identified as indicated in table 6. According to the results obtained from the ANN, LSTM and CNN showed better accuracies in epochs 200, 200, and 10 respectively. Difference between the LSTM model and the ANN and CNN models is manifested in the classification results of documents into formal and informal categories. The superior performance of LSTM in correctly identifying the true positive and true negative is evident concerning the false positive and false negative across all folds. However, the ANN, and CNN, were inferior to the LSTM in classification.

Table 6. Configuration parameter values

Configuration parameters for classification algorithms	
Algorithm	**Hyperparameter and Optimum value**
LSTM	Epochs = 200, batch size = 32, activation = “relu”, optimizer = ”adam”, verbose = 0
ANN	Epochs = 200, batch size = 32, activation = “relu”, solver = “adam”
CNN	Epochs = 10, batch size = 32, verbose = 0, activation = “relu”, optimizer = “adam”

E. Comparison with existing approach

Figure 9 compares the results of our study with previous ones. It's important to note that each study used different datasets, with all studies using datasets smaller than the 5000 data set we utilized.

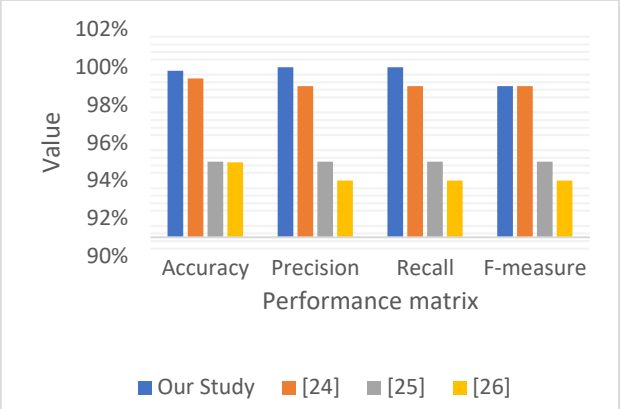

Figure 9. Comparison results with previous studies

In the study [24], SVM achieved the highest accuracy with a dataset of 1000. In the research [25], Random Forest yielded the highest values with a dataset of 1000, while the remaining study [26] used a dataset of 200, where SVM showed the highest performance. However the dataset and methodology are different, our proposed approach showed the better accuracy, precision, recall and f-measure than the existing approaches.

V. CONCLUSION

This is crucial in the recent digital age, where there are huge amounts of different text generated on numerous platforms, to identify and classify a given text in Formal or Informal ways. In the end, we aimed to establish effective methods for differentiating between these writing styles by exploiting and evaluating cutting-edge machine learning models and natural language processing techniques. We wanted to enhance the accuracy & efficiency of textclassification by focusing on an extensive feature extraction process and converting these features into some proper indexed numerical representation. This study provides a valuable methodology for computational linguistics. The study collected 5000 data sets, which were

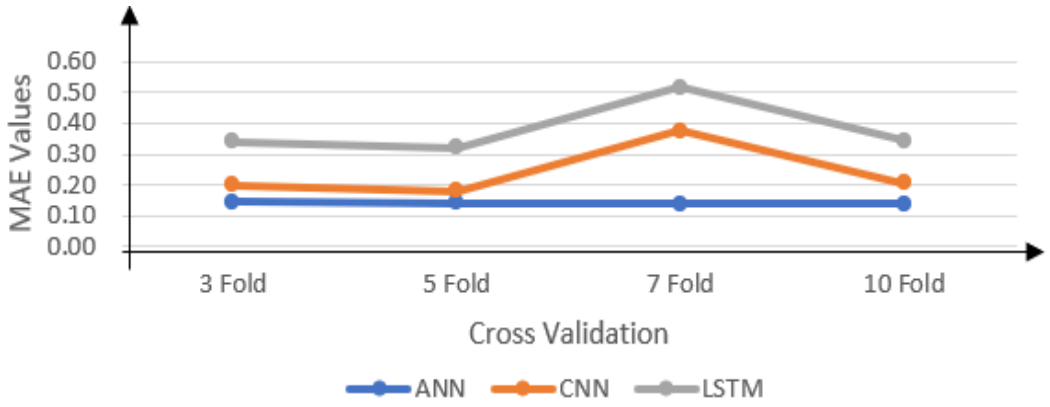

Fig 10. Results of MAE

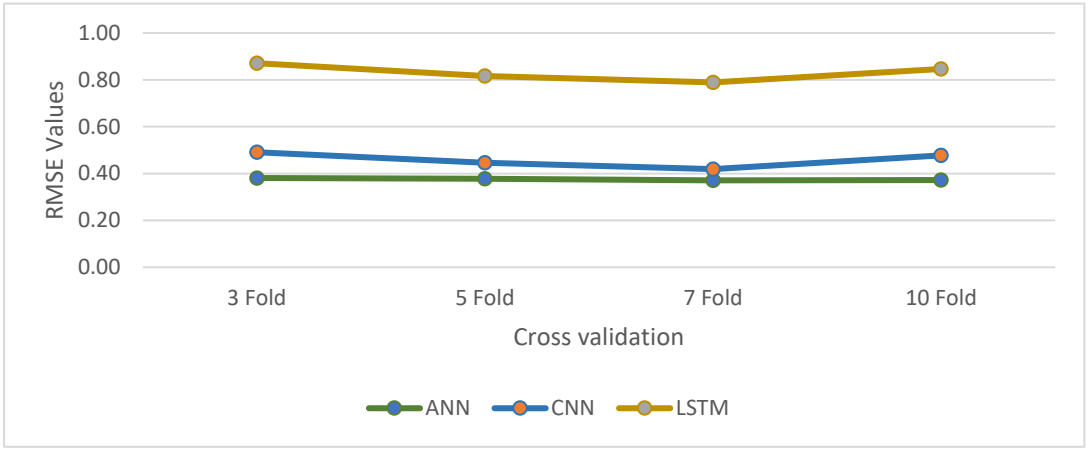

Fig 11. Results of RMSE

then pre-processed. Herein, the characteristic extraction process encompassed counts for colloquialisms, abbreviations, contractions, modal verbs, slangs, acronyms, initialisms, pronouns, phrasal verbs, grammar complexity, vocabulary complexity, voice and language type that gave a complete set of features which was very necessary for the classification task. These characteristics were transformed into machine learning models usable format during the feature vector creation process. We analyzed three deep-learning models to classify formal and informal documents. considered LSTM, CNN, and ANN. The models were used for their classification performance using metrics such as accuracy, precision, recall, f-measure, MAE, and RMSE over multiple cross-validation folds. Results showed that the LSTM model had a better performance than ANN or CNN in terms of accuracy, precision, recall, and f-measure. In particular, the LSTM performed better at reducing both false positives and negatives as well as achieving the highest accuracy (99.8%). To conclude, the outcomes of this study contribute to computational linguistics by giving valuable ways that distinguish between formal and informal writing styles through machine learning and deep learning techniques.

These findings are practical in improving recommendation algorithms for professional and educational writing tools.

A. Future Work

The future work involves collecting larger datasets from different domains and incorporating the additional linguistic features as well. As well as investigating advanced deep learning architecture improvements in accuracy. Last, but not least combining machine learning and deep learning models into a hybrid approach could leverage the strength of both techniques for better performance.

This study's findings, hold considerable potential for practical applications. For instance, the classification framework could be implemented into automated educational programs to help students identify between formal and informal writing styles, boosting their academic writing skills. Likewise, these models could improve writing helpers by delivering prompt feedback on text style, assuring suitability for diverse settings such as professional emails, academic papers, or personal correspondence. Such applications show the broader societal influence and practical importance of our study.

REFERENCES

[1] R. Johnson, K. Hyland, and K. Jiang, “Clear Language and Avoiding Ambiguity in Academic Writing,” pp. 1–27, 2012, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0889490616301016

[2] A. Kukulska-Hulme, “Language as a bridge connecting formal and informal language learning through mobile devices,”

Seamless Learn. Age Mob. Connect., pp. 281–294, 2015, doi: 10.1007/978-981-287-113-8_14.

[3] K. Yasar, "What is computational linguistics?" [Online]. Available: https://www.techtarget.com/searchenterpriseai/definition/computational-linguistics-CL

[4] J. Hu, Y., Wang, L., & Zhou, "Enhancing text classification with linguistic feature integration," *J. Comput. Linguist.*, vol. 48, no. 3, pp. 215-230., 2022.

[5] K. Smith, A., & Lee, "Advances in computational morphology," *A Surv. Nat. Lang. Process. J.*, vol. 52, no. 1, pp. 45–67, 2024.

[6] Y. Brown, T., Kumar, P., & Chen, "Neural networks vs. linguistic features: A comparative study in text classification," *A Comp. study text Classif. Mach. Learn. NLP Rev.*, vol. 29, no. 2, pp. 134–150, 2022.

[7] R. Hopkins, "Formal and Informal Language," *Educ. Action*, pp. 120–136, 2022, doi: 10.1163/9789004523876_009.

[8] B. Counsil, "10 differences between formal and informal language." Accessed: Jan. 01, 2024. [Online]. Available: https://www.londonschool.com/blog/10-differences-between-formal-and-informal-language/

[9] Murray, "Active and Passive Voice (Handout)," *Gramm. Mech. Act. Passiv. Voice*, pp. 19–20, 2018.

[10] K. Pal and B. V. Patel, "Automatic multiclass document classification of hindi poems using machine learning techniques," *2020 Int. Conf. Emerg. Technol. INCET 2020*, pp. 11–15, 2020, doi: 10.1109/INCET49848.2020.9154001.

[11] A. B. Adetunji, J. P. Oguntoye, O. D. Fenwa, and N. O. Akande, "Web Document Classification Using Naïve Bayes," *J. Adv. Math. Comput. Sci.*, vol. 29, no. 6, pp. 1–11, Dec. 2018, doi: 10.9734/jamcs/2018/34128.

[12] B. Agarwal and N. Mittal, "Text classification using machine learning methods-a survey," in *Advances in Intelligent Systems and Computing*, Springer Verlag, 2014, pp. 701–709. doi: 10.1007/978-81-322-1602-5_75.

[13] B. Kaur and G. Bathla, "Document Classification using Various Classification Algorithms: A Survey," *Int. J. Futur. Revolut. Comput. Sci. Commun. Eng. IJFRCSCE*, 2018, [Online]. Available: http://www.ijfrcsce.org

[14] M. Baygin, "Classification of Text Documents based on Naive Bayes using N-Gram Features." [Online]. Available: https://drive.google.com/open?id=1Idp5VK1Q91vyqb940WjeoM

[15] C. S. Lim, K. J. Lee, and G. C. Kim, "Multiple sets of features for automatic genre classification of web documents," *Inf. Process. Manag.*, vol. 41, no. 5, pp. 1263–1276, Sep. 2005, doi: 10.1016/j.ipm.2004.06.004.

[16] E. B. B. Palad, M. S. Tangkeko, L. A. K. Magpantay, and G. L. Sipin, "Document Classification of Filipino Online Scam Incident Text using Data Mining Techniques," *Proc. - 2019 19th Int. Symp. Commun. Inf. Technol. Isc. 2019*, pp. 232–237, 2019, doi: 10.1109/ISCIT.2019.8905242.

[17] P. H. Seo, Z. Lin, S. Cohen, X. Shen, and B. Han, "Hierarchical Attention Networks," *ArXiv*, pp. 1480–1489, 2016, [Online]. Available: http://arxiv.org/abs/1606.02393

[18] H. Schwenk and X. Li, "A corpus for multilingual document classification in eight languages," *Lr. 2018 - 11th Int. Conf. Lang. Resour. Eval.*, pp. 3548–3551, 2019.

[19] A. Adhikari, A. Ram, R. Tang, and J. Lin, "DocBERT: BERT for Document Classification," 2019, [Online]. Available: http://arxiv.org/abs/1904.08398

[20] M. Da Silva Conrado, V. A. Laguna Gutiérrez, and S. O. Rezende, "Evaluation of normalization techniques in text classification for Portuguese," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2012. doi: 10.1007/978-3-642-31137-6_47.

[21] M. N. Asim, M. U. G. Khan, M. I. Malik, A. Dengel, and S. Ahmed, "A robust hybrid approach for textual document classification," *Proc. Int. Conf. Doc. Anal. Recognition, ICDAR*, pp. 1390–1396, 2019, doi: 10.1109/ICDAR.2019.00224.

[22] X. Huang and M. J. Paul, "Examining temporality in document classification," *ACL 2018 - 56th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf. (Long Pap.*, vol. 2, pp. 694–699, 2018, doi: 10.18653/v1/p18-2110.

[23] Z. Kastrati, A. S. Imran, and S. Y. Yayilgan, "The impact of deep learning on document classification using semantically rich representations," *Inf. Process. Manag.*, vol. 56, no. 5, pp. 1618–1632, 2019, doi: 10.1016/j.ipm.2019.05.003.

[24] Z. E. Rasjid and R. Setiawan, "Performance Comparison and Optimization of Text Document Classification using k-NN and Naïve Bayes Classification Techniques," *Procedia Comput. Sci.*, vol. 116, pp. 107–112, 2017, doi: 10.1016/j.procs.2017.10.017.

[25] M. Usman, Z. Shafique, S. Ayub, and K. Malik, "Urdu Text Classification using Majority Voting," *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 8, pp. 265–273, 2016, doi: 10.14569/ijacsa.2016.070836.

[26] D. Buzic and J. Dobsa, "Lyrics classification using Naive Bayes," *2018 41st Int. Conv. Inf. Commun. Technol. Electron. Microelectron. MIPRO 2018 - Proc.*, pp. 1011–1015, 2018, doi: 10.23919/MIPRO.2018.8400185.

[27] R. A. Calvo, J. M. Lee, and X. Li, "Managing content with automatic document classification," *J. Digit. Inf.*, vol. 5, no. 2, pp. 1–15, 2004.

[28] F. A. Sheikha and D. Inkpen, "Linguistic Issues in Language Technology-LiLT Submitted," 2012.

[29] S. Jin, A. P. de Vries, A. Szuba, and D. Hiemstra, "Classification and Interchange of Informal and Formal English Text," 2022.

[30] K. M. G. S. Karunarathna, R. A. H. M. Rupasingha, and B. T. G. S. Kumara, "Classifying Documents based on Formal and Informal Writing Styles using Machine Learning Algorithms," *ICARC 2022 - 2nd Int. Conf. Adv. Res. Comput. Towar. a Digit. Empower. Soc.*, pp. 373–378, 2022, doi: 10.1109/ICARC54489.2022.9753774.

[31] K. M. G. . Karunarathna, R. A. H. . Rupasingha, and B. T. G. . Kumara, "An Ensemble Learning Approach to Classifying Documents Based on Formal and Informal Writing Styles," p. 2022, 2022.

[32] A. Dementieva, D., Babakov, N., & Panchenko, "Detecting Text Formality: A Study of Text Classification Approaches," *Proc. Int. Conf. Recent Adv. Nat. Lang. Process. (RANLP 2023)*, pp. 239–247, 2023.

[33] X. Liu, Y., & Zhang, "Linguistic Driven Feature Selection for Text Classification as Stop Words.," *J. Adv. Inf. Technol.*, vol. 14, no. 4, pp. 796–803, 2023.

[34] M. Chen, J., & Li, "Sentence Formality Prediction with Deep Learning," in *Proceedings of the IEEE 23rd International Conference on Information Reuse and Integration for Data Science (IRI)*, 2022, pp. 1–8.

[35] Kaggle, "No Title." Accessed: May 27, 2022. [Online]. Available: https://www.kaggle.com/

[36] The washington post, "The Washintong Post." Accessed: Jan. 10, 2024. [Online]. Available: https://www.washingtonpost.com

[37] writinghelp-central, "writinghelp-central." Accessed: Jan. 10, 2022. [Online]. Available: http://www.writinghelp-central.com/

[38] answershark, "answershark." Accessed: Jan. 15, 2022. [Online]. Available: https://answershark.com

[39] lettersfree, "lettersfree." Accessed: Jan. 15, 2022. [Online]. Available: https://www.lettersfree.com

[40] geeksforgeeks, "What is LSTM, ANN and CNN." Accessed: Jun. 27, 2024. [Online]. Available: https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/?ref=header_search

[41] geeksforgeeks, "What is LSTM – Long Short Term Memory?" Accessed: Nov. 14, 2024. [Online]. Available: https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/?ref=header_outind

AUTHOR BIOGRAPHY/IES

K.M.G.S Karunarathna is a postgraduate student at the Sabaragamuwa University of Sri Lanka. She has been involved with MPhil in Computing and Information System and now she is second year student. Her research interest include educational practice and methods, document classification and machine learning.

R.A.H.M. Rupasingha received her BSc in 2013 from Sabaragamuwa University in Sri Lanka. She obtained her MSc and PhD in 2016 and 2019, respectively, from the School of Computer Science and Engineering, the University of Aizu, Japan. Currently, she is a senior lecturer in Sabaragamuwa University in Sri Lanka. Her research interests include machine learning, ontology learning, data mining and recommendation

B. T. G. S. Kumara received the bachelor's degree in 2006 from Sabaragamuwa University of Sri Lanka. He received the master's degree in 2010 from University of Peradeniya, Sri Lanka and he received the PhD from School of Computer Science and Engineering, University of Aizu, Japan in 2015. Currently, he is a professor in Sabaragamuwa University in Sri Lanka. His research interests include semantic web, data mining, machine learning, web service discovery and composition.

Conversational AI for Cinnamon and Coffee Exports: Insights on Price and Yield

KGPH Samanthi[1#] , TGI Fernando[1] and MKA Ariyaratne[1]
[1]Department of Computer Science, Faculty of Applied Sciences, University of Sri Jayewardenepura, Sri Lanka
[#]hansikasamanthi914@gmail.com

ABSTRACT

This research covers the development of an AI-powered chatbot that will help develop the agricultural industry in Sri Lanka by answering queries regarding coffee and cinnamon, besides giving weekly producer's price predictions for them. It uses an SVM classifier that selects suitable responses from a given query in Sinhala, translates into English, generates the response, and then translates back to Sinhala for presentation. It implements an LSTM model to forecast prices of export crops from 2016 to 2022. It was observed that there is a great correlation between crop prices and the start date of the week they are valid, with a Pearson coefficient of over 0.70 for both coffee and cinnamon, while others are below 0.60. The chatbot returned to an accuracy rate of 70% in the classification of queries, while poor performance was obtained for harvest prediction due to a lack of sufficient data. The successful integration of predictive models and the chatbot proves the potential of AI in improving agricultural decision-making, productivity, and efficiency. This research consists of a Sinhala language-based chatbot, providing customized advisory services and weekly price predictions, contributing to localized technological advancements in Sri Lankan agriculture.

INDEX TERMS LSTM, SVM, Export Crops, Artificial Intelligence.

I. INTRODUCTION

Conversational AI seeks to create natural, human-like interactions using text or voice, leveraging data, natural language processing, and machine learning. It is increasingly used in chatbots across various industries, enhancing customer support, healthcare, finance, and more. The technology's ability to mimic human conversations and provide valuable insights makes it a crucial tool in modern applications.

Conversational AI can be utilized in agriculture to offer farmers advisory services and information, enhancing productivity in this vital sector. Agriculture contributes about 7% to Sri Lanka's GDP and employs over 25% of the population, with the country's diverse climate supporting a wide range of crops. Key export crops like tea, coconut, rubber, and spices generate significant foreign exchange and have global demand. Farmers in export crop cultivation prioritize quality to maximize income, but face challenges like environmental conditions and diseases that can impact harvests. Expert advice on environmental needs, pest control, fertilization, and weed management is crucial for successful cultivation.

Farmers in Sri Lanka traditionally seek advice from regional agriculture advisors who visit fields or are available at regional centers. The Department of Agriculture's website provides detailed information on crops, pest management, and climate, along with advisory contacts, in Sinhala, Tamil, and English [1]. The National Agriculture Information and Communication Center (NAICC) launched the "Govi Sahana Sarana Sevaya" program in 2006, offering daily advice via a 1920 call center [2]. The NAICC also introduced the "Krushi Advisor" mobile app, which provides information on regional suitability, climate conditions, fertilizer and pesticide needs, and harvesting procedures. Additionally, the Krushi SMS alert system sends synchronized information on 10 selected crops to registered users, aligned with their farming activities.

Hayles Agriculture offers an advisory service called "Agvize" to educate people on agricultural practices, disease management, and soil management [3]. In 2015, Dialog launched the "Govi Mithuru" service, providing timely, customized advice on land preparation, cultivation, and harvest, with voice messages sent to registered farmers in Sinhala [4]. Farmers can also access messages by calling 616, but the service is limited to Dialog and Hutch subscribers. Sri Lanka's agricultural advisory system includes physical meetings with advisors, call centers, mobile apps, message services, pre-recorded voice messages, and websites. These diverse methods ensure farmers receive the necessary guidance to manage their crops effectively.

II. Related Works

The literature reveals a range of chatbot and prediction models developed for various purposes, with many focusing on specific domains or languages. For instance, Hettige & Karunananda [5] developed the first Sinhala chatbot using Java and SWI-Prolog, which operates in both Linux and Windows environments. Their system, designed on a client-server model, utilizes multi-agent technology but is not domain-specific. Similarly, a Sinhala chatbot was created for user inquiries about degree programs at the University of Ruhuna [6], targeting Advanced Level students and university attendees in the Technology stream. This chatbot employs

RASA NLU APIs and Slack as its chat platform, with data stored in an SQLite database.

In the agricultural domain, AgriCom is a multi-agent system facilitating communication among farmers, buyers, sellers, and instructors using the 'MaSMT' multi-agent development architecture [7]. Ekanayake et al. [8] extended this concept by developing an intelligent chatbot and chat room specifically for farming-related issues, utilizing Artificial Intelligence Markup Language (AIML) and a cloud platform to handle resource demands. The E-agro system supports Sinhala language interactions, with Google translation APIs translating between English and Sinhala, although its data collection is limited to two districts.

In India, the FarmChat mobile app [9] was developed to assist potato farmers by using Google Translate for Hindi-English translation and IBM Watson for identifying intents and entities. Similarly, Reddy et al. [10] developed an IoT-based system where data captured via sensors at crop sites is fed into a cloud database and used by a bot to provide seasonally and environmentally tailored advice.

Chatbots employing Natural Language Processing (NLP) techniques, such as the talking chatbot described in [11], use neural networks to understand user queries and provide voice-based responses via speech synthesis Web APIs. However, these systems typically operate in English. Kaviya et al.

[12] designed an AI-based farmer assistant chatbot using the Na¨ıve Bayes algorithm for query analysis, allowing voice and text interactions in English, while Agribot [13] focuses on agriculture-specific questions using the Word2Vec model but excludes weather-related queries and operates in English.

Sahana et al. [14] has implemented a chatbot called Farmerbot, a machine learning-based virtual assistant for Indian farmers, providing insights on technologies and market trends. It uses speech-to-text, language translation, and text-to-speech synthesis. The system employs Long Short-Term Memory (LSTM) neural networks to process speech data and ensure accurate translation. Farmerbot utilizes high-level audio data representations for both source and target languages to recommend the best crops for optimal yield based on given conditions.

Agriculture Helper Chatbot is designed and developed by Anitha et al. [15] based on deep learning, computer vision, and natural language processing. User input is processed by deep learning models, including LSTMs, which then recommend crops. Computer vision is used in this project for detecting diseases in crops based on the image recognition techniques, while natural language processing can be used for queries in natural language. The system integrates weather APIs and farming practices expert knowledge. It features a straightforward interface for both text and image inputs. Based on user feedback, the bot keeps improving and increasing its recommendations and accuracy.

Crop prediction and price forecasting have also been explored, with Selvanayagam et al. [16] developing 'Agro-Genius,' a mobile application that uses the Autoregressive Integrated Moving Average (ARIMA) model to predict crop prices by harvest time based on ten years of district-specific data. Yield prediction models have employed various algorithms, with Random Forest showing the highest accuracy in studies such as those by [17], [18], and [19], outperforming Logistic Regression and Naive Bayes in predicting crop yield. Chaithanya et al. [20] further advanced this field by applying Recurrent Neural Networks (RNNs) to predict rice crop yields in Karnataka, achieving high accuracy in their predictions.

Research indicates that while conversational AIs and chat bots have been explored in agriculture, gaps remain, particularly in Sri Lanka. There is a notable absence of Sinhala language conversational AI, despite the global development of agricultural chatbots. Additionally, while the Department of Export Agriculture regularly publishes producer prices for export crops, there is limited research on price and yield predictions for these crops, and little effort has been made to integrate these predictions into conversational AI.

Through our work, we hope to address the above questions and implement a solution based on conversational AI that uses the Sinhala language to help farmers. In summary, the main goals of this research are to develop an integrated model of a Sinhala chatbot that integrates price and yield prediction models with advisory support enabling farmers by responding effectively to queries on cinnamon and coffee-related issues. Regarding such main objectives for both crops, it is expected to predict correctly the producer price in a kilogram for the following week and estimate the expected yield with respect to the extent of cultivation provided by the user. It also attempted to improve access to agricultural knowledge for non-English-speaking farmers by creating a user-friendly, accessible Sinhala chatbot. In addressing these objectives, the research attempts to empower farmers with relevant and timely information that would help them make informed decisions and improve efficiency in farming activities.

The study underscores the importance of agriculture in Sri Lanka, where over 30% of the population is employed, and the challenges farmers face due to environmental and regional factors. Introducing a Sinhala language chatbot enhances accessibility, allowing farmers to access agricultural information in their native language, thereby overcoming communication barriers. The chatbot specifically addresses export crops like Coffee and Cinnamon, offering real-time predictions for crop prices and yields to help farmers make informed decisions. Additionally, it provides a user-friendly interface and leverages historical data to bridge the gap between data collection and practical application in agriculture.

This study focuses on Cinnamon and Coffee, two key export crops in Sri Lanka, selected based on their relationship with producer prices and dates. The chatbot, which operates in Sinhala, addresses inquiries related to fertilization, crop management, and establishment while also providing predictions on the next producer's price per kilogram. The yield predictions were not accurate, therefore in the final development, yield prediction models are not included. However, due to limited recorded data on yields, these predictions may vary from actual outcomes.

There is a significant gap in integrating predictive modeling and chatbot technology for the agricultural sector,

especially in the Sinhala language. Most of the related research done to date has focused either on predictive analytics or on chatbot implementations as a standalone solution and has given little attention to how these two technologies can be integrated.

Moreover, the inability of the Sinhala-language chatbots already available to respond to agriculture-based questions has resulted in barriers to the much-needed access to information by farmers and other stakeholders who do not understand English. This hinders effective decision-making and knowledge dissemination with implications for agricultural practices.

The novelty of this study is that it develops a Sinhala-language chatbot for cinnamon and coffee farmers by integrating a weekly producer price prediction model with interactive advisory support. While the existing systems address only one aspect-either providing general information or predictive insight-this research will offer a unified solution that will deliver real-time responses to user inquiries, incorporating data-driven forecasts.

This chatbot is facilitated with a seamless interaction in Sinhala, enhancing accessibility and usability by empowering farmers with localized and actionable insights. This is a different approach toward agricultural support right at the seam between predictive analytics and conversational AI for cinnamon and coffee farming in Sri Lanka.

III. Materials and Methods

1) Dataset collection

Price Data Collection: The Department of Export Agriculture website releases weekly producer's prices for export crops. Figure 1 displays the producer's prices for 1 kg of Cinnamon crop across different districts in a specific week. Weeks are shown on the Department of Export Agriculture web page. Inside of a week producer's prices are mentioned according to the crop and the district.

District	Alba (Highest Price)	Alba (Average Price)	C-5 Sp (Highest Price)	C-5 Sp (Average Price)	C-5 (Highest Price)	C-5 (Average Price)	C-4 (Highest Price)	C-4 (Average Price)
Ratnapura	2,100.00	2,066.00	1,750.00	1,716.00	1,650.00	1,616.00	1,550.00	1,533.00
Badulla	-	-	-	-	1,700.00	1,700.00	1,500.00	1,500.00
Kurunegala	-	-	-	-	-	-	-	-
Colombo	-	-	1,800.00	1,800.00	1,700.00	1,700.00	1,650.00	1,650.00
Gampaha	-	-	-	-	-	-	1,800.00	1,600.00
Kalutara	2,000.00	2,000.00	1,700.00	1,650.00	1,600.00	1,550.00	1,500.00	1,430.00
Galle	2,100.00	2,000.00	1,800.00	1,780.00	1,650.00	1,630.00	1,630.00	1,620.00
Matara	2,150.00	2,117.00	1,750.00	1,750.00	1,700.00	1,683.00	1,680.00	1,653.00
Hambantota	-	-	-	-	1,675.00	1,662.00	1,650.00	1,650.00
Monaragala	-	-	-	-	-	-	-	-
National	2,150.00	2,045.75	1,800.00	1,739.20	1,700.00	1,648.71	1,800.00	1,579.50

Figure.1. Producer's price table of Cinnamon for different districts on August 2023

The table includes districts as rows and displays the average, highest, and national producer prices for Cinnamon species. For the study, producer prices for selected crops were recorded in .csv files, with data spanning from January 2016 to November 2022. Each crop's data was organized into individual .csv files by district, focusing on those with consistently available price data.

Figure 2 shows the availability of weekly producer's price data of export crops in different districts according to the Department of Export Agriculture website.

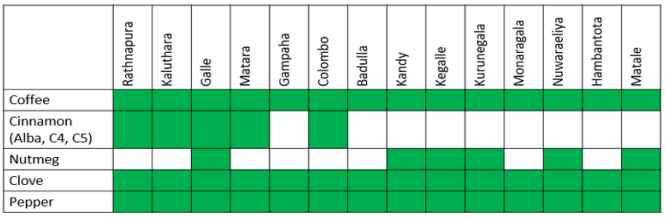

Figure. 2. District-wise Crop Data: Green Highlights Illuminate Availability

The Department of Export Agriculture publishes weekly producer prices, valid for that specific week, typically released every Tuesday by district. Table 1 shows the number of data points collected for coffee and cinnamon across various districts.

Crop	District	Number of Producer's Prices Data Points
Cinnamon	Kalutara	374
	Galle	380
	Rathnapura	373
Coffee	Kalutara	335
	Colombo	287
	Gampaha	325
	Hambantota	340
	Rathnapura	329

Table.1. Number of Producer's Prices Data Points by Crop and District.

2) Data/Text Cleaning:

Price Data and Harvest Data: In the data cleaning process for producer's price and harvest data, the dataset is first loaded using Pandas, and the 'Average Price' column is converted to integers. Rows with missing or erroneous values, like '-' or '#REF', are then removed to improve data reliability.

3) Data/Text Pre-processing:

Inquiries from Export Crop Farmers: The collected queries are structured in columns denoting 'pattern', 'response', and 'tag'. The 'pattern' column is the questions themselves, the 'response' column provides corresponding answers, and the 'tag' column indicates the category to which the question belongs. This tabular arrangement aims to facilitate the process of predicting the appropriate tags for each question. There are a total of 24 tags, with each tag having 20 associated patterns and their corresponding responses. Pre-processing of inquiries is followed by a series of steps such as tokenization, removal of numbers and punctuations, stop.

Price Data and Harvest Data: The 'Average Price' values are scaled using 'MinMaxScaler' to normalize them between 0 and 1, ensuring balanced feature contribution during LSTM model training. This process ensures data quality and prepares it for effective machine learning application.

4) Prediction Models

A. Price Prediction Model: The LSTM model for predicting producer's prices starts with data pre-processing. The dataset is split into 20% training and 80% testing sets, using the past 60 days' prices to predict the next week's price. The LSTM architecture is built with Keras Sequential API, consisting of three LSTM layers with different units for each crop, each followed by dropout layers to prevent overfitting, and a final Dense layer to generate the predicted producer's price.

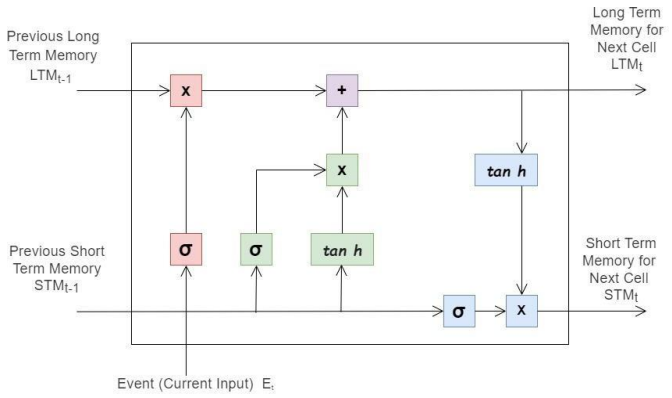

Figure.3. LSTM Architecture

After compiling the model with the 'adamax' optimizer and mean squared error loss function, it undergoes training using the training data with specific batch sizes and epochs as shown in Table 2. For the LSTM models predicting producer's price for coffee, three layers were employed in the models for Kalutara, Hambantota, Colombo, Rathnapura, and Gampaha districts. The layer units were configured as 128, 64, and 64. Similarly, for the cinnamon models in Kalutara, Galle, and Rathnapura, the layer units were set at 128, 64, and 64. In the coffee models across Kalutara, Hambantota, Colombo, Rathnapura, and Gampaha districts, the batch sizes were 2, 20, 8, 10, and 5 respectively. The epochs for these districts were set at 100, 1760, 70, 200, and 200 correspondingly. Additionally, for cinnamon models, the epochs for Kalutara, Galle, and Rathnapura were 50, 50, and 200 respectively. Following training, the model's performance is evaluated using the testing dataset, calculating the Root Mean Squared Error (RMSE) between the predicted and actual producer's prices.

The model's performance is visually assessed by plotting the training data, actual prices, and predicted producer's prices using Matplotlib. This helps evaluate how well the model fits the data and predicts future prices. For future predictions, the model forecasts the next week's producer price based on the last 60 days' data, with the predicted price scaled back to its original range using inverse transformation, and the next week's date determined from the last date in the dataset.

Crop	Coffee					Cinnamon		
District	Kalutara	Hambantota	Colombo	Rathnapura	Gampaha	Kalutara	Galle	Rathnapura
Layer 1 Units	128	128	128	128	128	128	128	128
Layer 2 Units	64	64	64	64	64	64	64	64
Layer 3 Units	64	64	64	64	64	64	64	64
Batch Size	2	20	8	10	5	20	20	15
Epochs	100	1760	70	200	200	50	50	200

Table.2. Summary of Producer's Price Prediction LSTM Model Implementation Details

B. Harvest Prediction Model: The LSTM model for crop harvest prediction begins with importing the dataset using pandas and cleaning it by removing rows with missing or invalid values like '#REF'. The 'harvest' column is converted into a NumPy array for further processing, and normalization is applied to standardize the harvest values, enhancing the model's learning and convergence. The data is then split into 80% training and 20% testing sets to allow the model to learn from historical harvest records. The LSTM model is built using the Keras Sequential API, with multiple LSTM layers to capture temporal patterns, and dropout layers to prevent overfitting. All of the harvest prediction models are implemented using LSTM neural network architecture. The models are as follows:

- Coffee harvest prediction models for Kalutara, Hambantota, Colombo, Rathnapura and Gampaha districts.
- Cinnamon harvest prediction models for Kalutara, Galle and Rathnapura districts.

Manual hyperparameter tuning, employing the trial-and-error method, is utilized for experimenting with various sets of hyperparameters. Therefore, number of units in each layer of the LSTM models, batch size, number of epochs, dropout rate, activation function and optimizer for each model are determined using the trial-and-error method. Hyperparameter values are selected based on the evaluation of the root mean square error, with the aim of minimizing this metric.

The LSTM models for Coffee in Kalutara, Hambantota, and Colombo districts used a batch size of 40, while those for Rathnapura and Gampaha had a batch size of 36. The LSTM models trained on coffee data from Kalutara, Hambantota, and Colombo districts all featured the same layer units configuration, with 32 units in the first two layers and 24 units in the third layer. Additionally, two dropout layers were incorporated into each model, with dropout rates set at 0.3 and 0.2 respectively.

For the LSTM models for Coffee in Rathnapura and Gampaha districts, the layer units were configured as 32, 24, and 24. Additionally, two dropout layers were incorporated into each model, with dropout rates set at 0.3 and 0.2 respectively. The batch size for both districts remained at 36.

The cinnamon LSTM model that was built for the Kalutara district included two dropout layers with dropout rates of 0.3 and 0.2, respectively, in addition to layer units that were 64, 64, and 32. For this district, the batch size was fixed at 8.

The layer units of the LSTM model created for Cinnamon in the Galle district were 64, 32, and 16, and two dropout layers were incorporated, with dropout rates of 0.3 and 0.2, respectively. This district's batch size was changed to 10. Layer units of 16, 16, and 8 were used in the cinnamon models that were tuned for the Rathnapura district. There were two dropout layers present, each with a dropout rate of 0.2 and 0.3. For this district, the batch size was established at 40.

Crop	Coffee					Cinnamon		
District	Kalutara	Hambantota	Colombo	Rathnapura	Gampaha	Kalutara	Galle	Rathnapura
Layer 1 Units	32	32	32	32	32	64	64	16
Layer 2 Units	32	32	32	24	24	64	32	16
Layer 3 Units	24	24	24	24	24	32	16	8
Batch Size	40	40	40	36	36	8	10	40
Epochs	100	100	100	100	100	20	100	100

Table.3. Summary of Harvest Prediction LSTM Model Implementation Details

Table 3 provides a summary of the hyperparameter values utilized in the LSTM models for both Cinnamon and Coffee crops across various districts. Once the training process is complete, the model's performance is assessed using the testing dataset, with the root mean squared error providing insights into its predictive accuracy. Comparing the model's predictions with the actual harvest data allows for an evaluation of its accuracy.

C. Chatbot Development:

The chatbot is trained on a dataset comprising a CSV file with three columns, namely Pattern, Response, and Tags, which correspond to the categories. In total, it contains 605 questions dealing with fertilization, crop management, and crop establishment of cinnamon and coffee. This includes the text being transformed into lowercase, tokenization of text into words, removal of common stop words (such as 'the', 'is' etc.), stemming of words to their base forms. These preprocessing steps aim at standardizing the data by reducing unnecessary variability within the text, so it can focus on core elements of each query.

After the preprocessing of data, the next step involves the feature extraction from the text. This is done using the TF-IDF (Term Frequency-Inverse Document Frequency) vectorizer. The vectorizer then converts these text patterns into numerical vectors that depict the importance of each term in the context of the overall dataset. First, the TF-IDF Vectorizer is fitted on the training data and then applied to both training and testing sets. Then, the sparse matrices are representing the text in the form understandable for the machine learning algorithms.

The Support Vector Machine, with a sigmoid kernel, is the machine learning model of the chatbot. Generally, SVM is a powerful classifier which finds its wide applications in text classification because it is capable of handling high-dimensional data successfully. The model is trained with TF-IDF features of training data and related labels representing tags associated with the text patterns. The model predicts the expected tag for the chatbot, from these incoming input data, in order to come up with correct responses in concert with the expected tag.

The dataset will be divided into an 80-20 split for training and testing. The random value is fixed for the same splitting every time, to ensure consistent results each time the process is repeated. Now, the model was tested on the testing data for which the chatbot was supposed to predict the tag (category) of user queries. Based on these predictions, the SVM classifier get the chatbot to choose a response corresponding to what tag comes out.

In this methodology, the expected and actual labels of the test data are compared to measure the accuracy. The ratio of correctly classified instances to all instances in the test set computes the accuracy score. This metric gives the percentage of correct predictions the classifier made, thus providing a complete picture of how well the model is doing. The result is given as the higher the percentage, the better the classification accuracy.

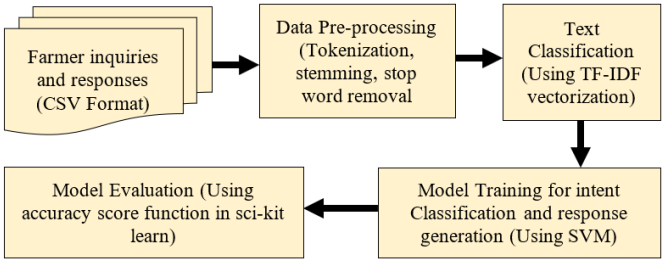

Figure.4. Chatbot model architecture

The web application provides a user interface to interact with the chatbot and the different price prediction models. The web application was developed using Flask python web framework. The chatbot model uses a Support Vector Machine for intent identification. The Sinhala user inputs undergo a natural language preprocessing pipeline including tokenization, stop word removal, stemming, translation to English using googletrans API. This pre-processed input feed goes into the SVM model, which predicts the intent from the pre-trained responses. The selected response from the predicted intent selects the response from a predefined dataset. If the intent selection is pointing towards price-related inquiries, then this chatbot invokes the corresponding LSTM-based price prediction model that generates responses dynamically.

The price prediction model will forecast the weekly producer's price of cinnamon and coffee crops by using historical data. This LSTM model will take in scaled and reshaped input data using MinMaxScaler, representing the last 60 days of producer's price information to predict the future producer's price. Predictions will then be transformed back into their original scale for user interpretation.

To allow for multiple crops based on the locations, the integration pipeline has different models for each crop and location, making sure the predictions are appropriate. The Flask application coordinates the selection of the appropriate model based on the user's intent.

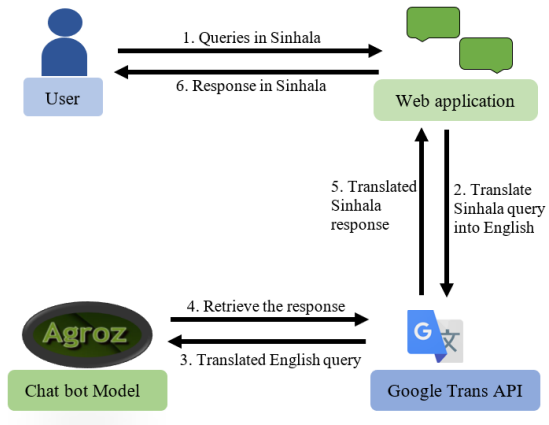

Figure.5. Chatbot model work flow.

IV. RESULTS AND DISCUSSION

1) Data Analysis

A. Correlation Analysis between Price data and Date: To reveal significant relationships between producer's prices and the start date of the week, data from selected crops in specific districts was analysed. Dates were converted to datetime format for easier manipulation and then transformed into numeric values representing nanoseconds since the Unix epoch. The Pearson correlation coefficient was calculated between these numeric date values and the corresponding average prices. A summary of the resulting Pearson coefficient values for each crop across districts is depicted in Table 4.

Price prediction models for cinnamon and coffee crops were implemented for districts that are selected based on calculated Pearson coefficient values. According to the results of calculated Pearson coefficient values the crops and districts shown in Table 5 were selected. The criteria for selecting the districts for each crop are as follows.

- Cinnamon Alba – The three districts that result in the highest Pearson coefficient values.
- Coffee – The five districts that result in the highest Pearson coefficient values.

Crop	Districts
Cinnamon Alba	Rathnapura, Kaluthara, Galle
Coffee	Rathnapura, Hambantota, Gampaha, Kaluthara, Colombo
Pepper	-
Nutmeg	-
Clove	-

Table.4. Selected Districts for Each Crop

Crop\ District	Rathnapura	Nuwara Eliya	Monaragala	Matara	Matale	Kurunegala	Kegalle	Kandy	Kaluthara	Hambantota	Gampaha	Galle	Colombo	Badulla
Cinnamon Alba	0.72	-	-	0.67	-	-	-	-	0.77	-	-	0.78	-	-
Cinnamon C4	0.73	-	-	0.71	-	-	-	-	0.70	-	-	0.72	0.73	-
Cinnamon C5	0.76	-	-	0.72	-	-	-	-	0.71	-	-	0.72	0.73	-
Coffee	0.88	0.85	0.85	0.87	0.85	0.87	0.86	0.86	0.89	0.88	0.88	0.85	0.89	0.87
Pepper	0.19	0.21	0.17	0.16	0.18	0.17	0.15	0.18	0.16	0.19	0.2	0.16	0.19	0.17
Nutmeg	-	0.75	-	-	0.75	0.71	0.74	0.75	-	-	-	0.56	-	-
Clove	0.55	0.56	0.46	0.52	0.59	0.53	0.53	0.57	0.52	0.51	0.56	0.52	0.62	0.5

Table.5. A summary of the resulting Pearson coefficient values for each crop across districts

B. Correlation Analysis between Extent and Harvest of the Crop Harvests: A Pearson coefficient has been calculated for each crop, determining the correlation between the extent and the harvest in selected districts for each crop spanning from 2000 to 2022. Pearson coefficient values depict the correlation between the extent and harvest of cinnamon across various districts. The relationship between extent and harvest in Galle district is notably positive, with a resulting value of 0.54. Hambantota exhibits a significant correlation between extent and harvest, while Gampaha shows the lowest correlation among the districts studied.

Model	RMSE Value
Cinnamon Kaluthara	330.9
Cinnamon Galle	433.6
Cinnamon Rathnapura	614.9
Coffee Rathnapura	64.7
Coffee Kaluthara	60.7
Coffee Gampaha	50.4
Coffee Colombo	55.4
Coffee Hambantota	60.7

Table.6. RMSE values for cinnamon and coffee price prediction models in different districts

C. Price Prediction Models: The performance of each LSTM-based price prediction models is assessed using the Root Mean Squared Error (RMSE) metric. RMSE provides a measure of the average deviation between the predicted and actual producer's prices, offering insights into the accuracy of each model. The RMSE values for each model are summarized in Table 6. Figures 6 to 12 show visual representations of the Cinnamon and Coffee producer's price model predictions next to the real producer's pricing.

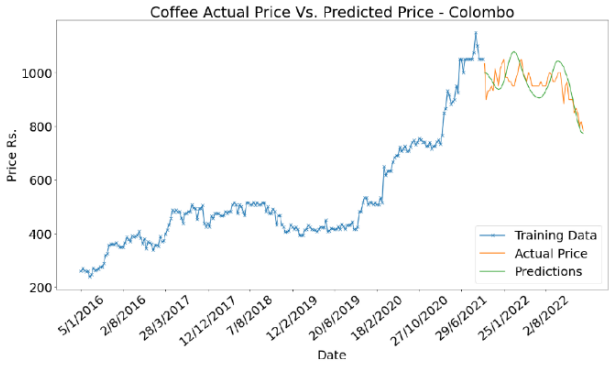

Figure.6. Actual Producer's Price and the Predicted Producer's price of Coffee in Colombo District

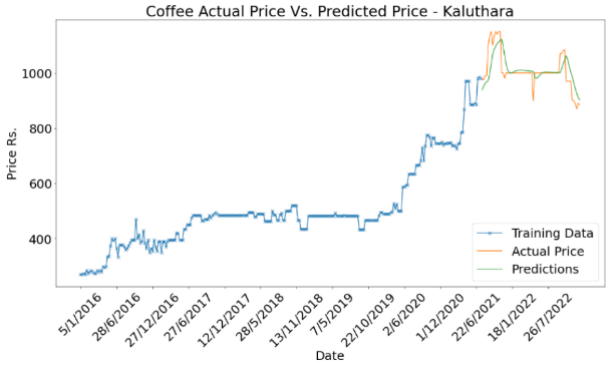

Figure.7. Actual Producer's Price and the Predicted Producer's price of Coffee in Kalutara District

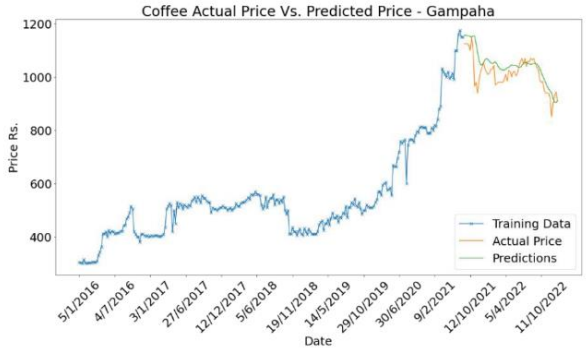

Figure.8. Actual Producer's Price and the Predicted Producer's price of Coffee in Gampaha District

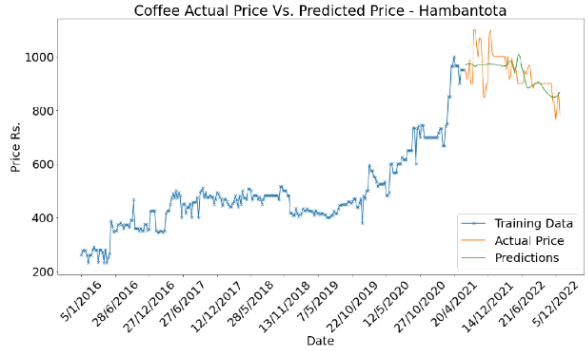

Figure.9. Actual Producer's Price and the Predicted Producer's price of Coffee in Hambantota District

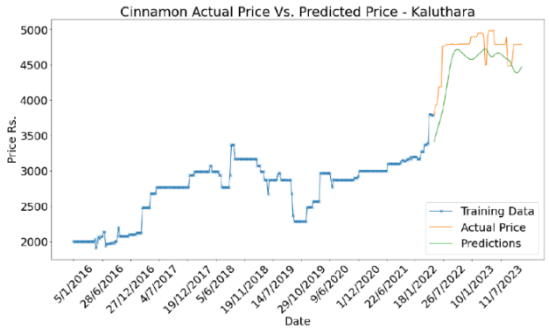

Figure.10. Actual Producer's Price and the Predicted Producer's price of Cinnamon in Kalutara District

Figure.11. Actual Producer's Price and the Predicted Producer's price of Cinnamon in Galle District

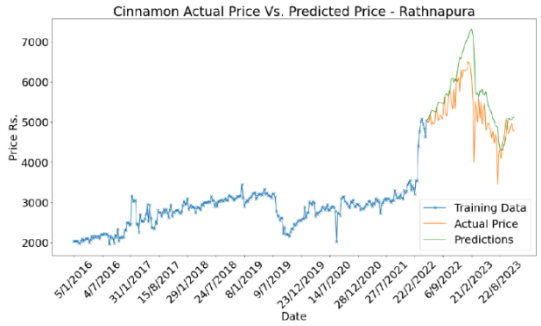

Figure.12. Actual Producer's Price and the Predicted Producer's price of Cinnamon in Rathnapura District

Kalutara shows the lowest RMSE for cinnamon price predictions, indicating relatively accurate predictions in that district. Rathnapura, however, has the highest RMSE at 614.9, reflecting less precise predictions. For coffee, Gampaha demonstrates the best performance with an RMSE of 50.4, suggesting more accurate predictions, while Rathnapura again records the highest RMSE at 64.7, showing lower accuracy. Overall, Kalutara and Gampaha have stronger predictive accuracy in cinnamon and coffee, respectively, while Rathnapura consistently exhibits higher RMSE values across both crops.

D. Harvest Prediction Models: The predictive performance of the chatbot's harvest prediction models was evaluated using the Root Mean Squared Error (RMSE) metric, providing insights into the accuracy of their predictions (Table 7). These

RMSE estimates represent the average difference between actual and expected harvest values. Higher RMSE indicates a larger gap between expected and actual prices, while lower RMSE suggests better forecasting accuracy. The model with the lowest RMSE is considered the most accurate for harvest forecasts within the chatbot.

Model	RMSE Value
Cinnamon Kalutara	144.2
Cinnamon Galle	492.4
Cinnamon Rathnapura	131.7
Coffee Rathnapura	20.1
Coffee Kalutara	17.1
Coffee Gampaha	432.3
Coffee Colombo	6.4
Coffee Hambantota	8.1

Table.7. RMSE values for cinnamon and coffee harvest prediction models in different districts

Figures 13 to 16 show visual representations of the Coffee and Cinnamon harvest model predictions next to the real harvests.to For each model, a comparison of the current and projected harvests for coffee and cinnamon in the chosen districts is shown in each figure. The extent in hectare is shown on the x-axis, while the harvest in metric ton (MT) is shown on the y-axis.

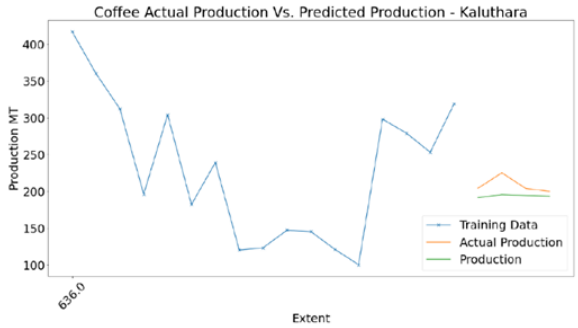

Figure.13. Actual Harvest and the Predicted Harvest of Coffee in Kalutara District

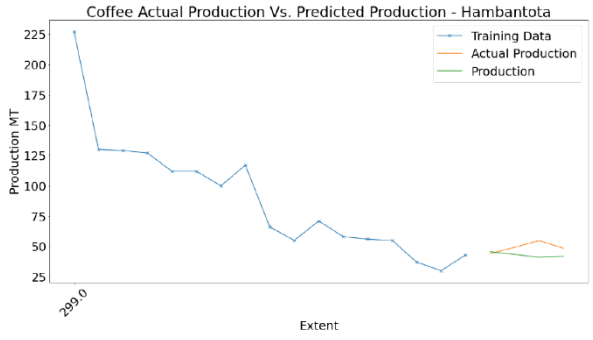

Figure.14. Actual Harvest and the Predicted Harvest of Coffee in Hambantota District

The RMSE values highlight the predictive accuracy of cinnamon and coffee models across districts. Galle has the highest RMSE for cinnamon at 492.4, indicating less accurate predictions, while Kalutara shows the lowest RMSE at 144.2, reflecting better accuracy. For coffee, Gampaha has the highest RMSE at 432.3, suggesting less reliable predictions, whereas Colombo has the lowest at 6.4. Overall, Kalutara consistently achieves lower RMSE values, indicating more accurate predictions, while Galle and Gampaha show higher RMSE values.

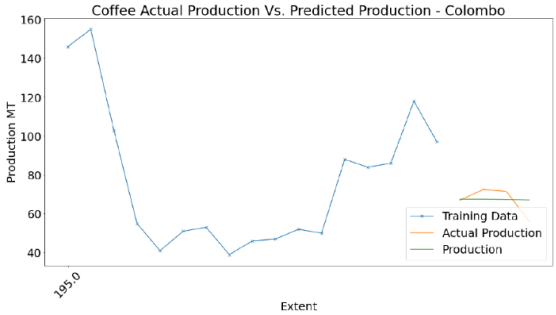

Figure.15. Actual Harvest and the Predicted Harvest of Coffee in Colombo District

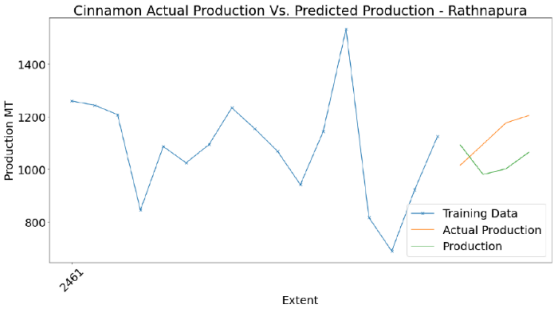

Figure.16. Actual Harvest and the Predicted Harvest of Cinnamon in Rathnapura District

By carefully examining the RMSE values for price and harvest prediction models of cinnamon and coffee across some districts, along with Pearson correlation coefficients that assess the relationships between extent and harvest, as well as between producer's prices and their respective validity dates, of cinnamon and coffee, some insightful information can be derived.

The RMSE values for coffee price prediction models are lower, and their Pearson coefficients show a stronger positive relationship between the date and the producer's price. Cinnamon has lower Pearson coefficients and higher RMSE values, indicating less accurate predictions. A stronger relationship between the date and producer's price leads to lower RMSE values and better prediction accuracy. However, cinnamon harvest prediction models show significantly higher RMSE values and weaker or negative relationships between extent and harvest, while coffee models display better accuracy. Despite low RMSE values, the overall prediction accuracy for cinnamon and coffee harvests is low due to insufficient data. Consequently, the harvest prediction model was not integrated into the chatbot, as it relied heavily on limited extent data.

E. Chatbot Development: The chatbot's price prediction models are integrated into its responses, achieving 70% accuracy in handling user inquiries related to coffee and cinnamon as shown in the Figure 17. This accuracy was determined by training an SVM classifier and evaluating its

performance on a test set. The scikit-learn accuracy score function calculated the accuracy by comparing the true labels with the predicted labels, resulting in a score of 0.70, meaning the chatbot correctly classified 70% of the test instances.

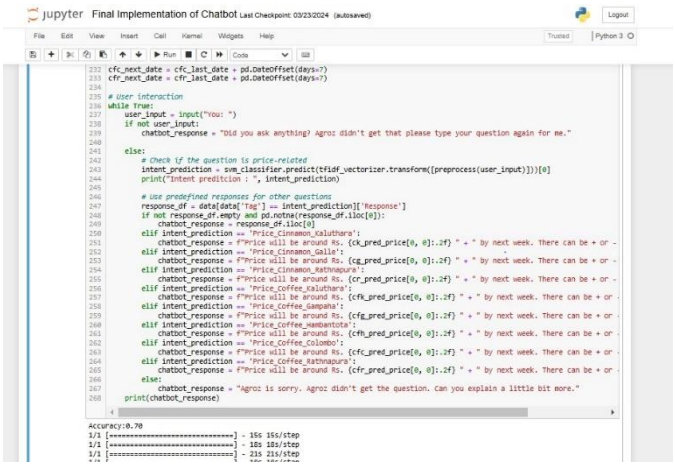

Figure.17. Accuracy of the chatbot model after integration with producer's price prediction models

Figure 18 and Figure 19 give the interaction of the user and chatbot, where the language used by them is Sinhalese. From this chatbot, the prediction price in various districts regarding coffee and cinnamon can be done. Some of the responses depicted in Figure 19 appear to be incorrect. If a valid response cannot be generated, then the chatbot would greet the user again. Since some of those responses are found as improper, the efficiency of the accuracy of the entire chatbot remained at 70%.

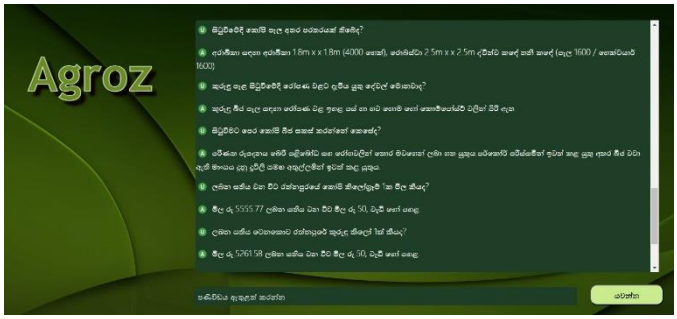

Figure.18. Chat interface displaying user interactions and chatbot responses

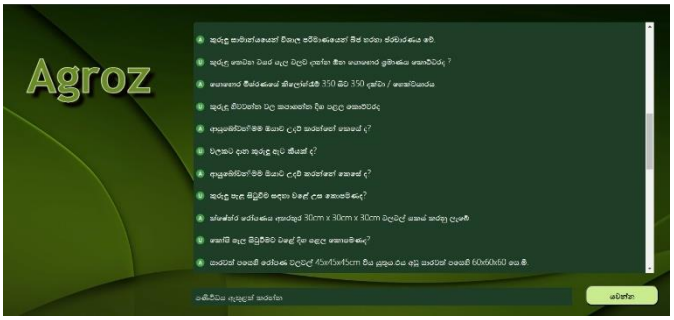

Figure.19. Chat interface displaying user interactions and incorrect chatbot responses

V. CONCLUSION

The project successfully developed a Sinhala-capable chatbot that integrated models for the prediction of prices and yields of coffee and cinnamon, hence fulfilling the research objectives: enabling farmers to access relevant agricultural information. In such a manner, the chatbot was able to respond to most user queries regarding management and establishment questions about the particular crops, with 70% accuracy of the responses. The price prediction models had a higher degree of accuracy on some district-like Kalutara for coffee and Rathnapura for cinnamon, where the RMSE values were found to be lesser. But the harvest prediction models could not perform well due to insufficiency in data, which gave unreliable yield forecasts. In spite of these challenges, the chatbot achieved its objective of enhancing the access of agricultural knowledge to non-English-speaking farmers by providing a user-friendly interface in Sinhala. This could be further improved in the future by expanding data collection to obtain more accurate price and yield predictions, factoring in other variables such as meteorological conditions. This study therefore indicates that a Sinhala chatbot is in a better position to provide answers to agricultural questions and further study to refine the predictive models is required in order to help farmers.

VI. REFERENCES

[1] D. of Agriculture. (2024, August) Department of agriculture. Accessed: 2024-08-23. [Online]. Available: https://doa.gov.lk/

[2] N. A. Information and C. Centre. (2024, August) National agriculture information and communication centre. Accessed: 2024-08-23. [Online]. Available: https://doa.gov.lk/naicc-home/

[3] H. Agriculture. (2024, August) Hayles agriculture. Accessed: 2024- 08-23. [Online]. Available: https://www.hayleysagriculture.com/ agvize- agriculture-advisory-sri-lanka

[4] Dialog. (2024, August) Govi mithuru. Accessed: 2024-08-23. [Online].
Available: https://www.dialog.lk/govi-mithuru/

[5] B. Hettige and A. S. Karunananda, "First sinhala chatbot in action," Proceedings of the 3rd Annual Sessions of Sri Lanka Association for Artificial Intelligence (SLAAI), University of Moratuwa, vol. 13, 2006.

[6] U. Kumanayake, "A sinhala chatbot for user inquiries regarding degree programs at university of ruhuna," Ph.D. dissertation, 2021.

[7] H. Jayarathna and B. Hettige, "Agricom: A communication platform for agriculture sector," IEEE 8th international Conference on industrial and information systems, pp. 439–444, 2013.

[8] J. Ekanayake and L. Saputhanthri, "E-agro: Intelligent chat-bot. iot and artificial intelligence to enhance farming industry," Agris on-line Papers in Economics and Informatics, vol. 12, no. 1, pp. 15–21, 2020.

[9] M. Jain, P. Kumar, I. Bhansali, Q. V. Liao, K. Truong, and S. Patel, “Farmchat: a conversational agent to answer farmer queries,” Proceed- ings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, vol. 2, no. 4, pp. 1–22, 2018.

[10] P. Venkata Reddy, K. Nandini Prasad, and C. Puttamadappa, “Farmer’s friend: Conversational ai bot for smart agriculture,” Journal of Positive School Psychology, vol. 6, no. 2, pp. 2541–2549, 2022.

[11] J. Vijayalakshmi and K. Pandimeena, ‘Agriculture talkbot using AI’, Int. J. Recent Technol. Eng, vol. 8, pp. 186–190, 2019.

[12] P. Kaviya, M. Bhavyashree, M. D. Krishnan, and M. Sugacini, “Artificial intelligence based farmer assistant chatbot,” International Journal of Research in Engineering, Science and Management, vol. 4, no. 4, p. 26–29, 2021.

[13] N. Jain, P. Jain, P. Kayal, J. Sahit, S. Pachpande, J. Choudhari et al., “Agribot: agriculture-specific question answer system,” 2019.

[14] L. B. Sahana, B. Anjali, S. Naik, P. M. Shreenidhi, “ “Farmerbot”- An Interactive And Assistive Interface For Farmers,” International Journal of Creative Research Thoughts (IJCRT), vol. 10, no. 6, pp. 2320 – 2882, 2022

[15] M. Anitha, C. H. Satyanarayana Reddy, and C. H. Deepika, "Agriculture Helper Chatbot using deep learning," Int. Res. J. Mod. Eng. Technol. Sci., vol. 5, no. 7, pp. 1518, 2023.

[16] A. Gamage and D. Kasthurirathna, “Agro-genius: crop prediction using machine learning,” 2019.

[17] A. Nigam, S. Garg, A. Agrawal, and P. Agrawal, “Crop yield prediction using machine learning algorithms,” in 2019 Fifth International Conference on Image Information Processing (ICIIP), pp. 125–130, 2019.

[18] A. Chlingaryan, S. Sukkarieh, and B. Whelan, “Machine learning approaches for crop yield prediction and nitrogen status estimation in precision agriculture: A review,” Computers and electronics in agriculture, vol. 151, pp. 61–69, 2018.

[19] S. G. Sangeeta, “Design and implementation of crop yield prediction model in agriculture,” International Journal of Scientific & Technology Research, vol. 8, no. 1, pp. 544–549, 2020.

[20] S. Chaithanya, A. P. Raj, N. Rajeshrahul, H. Sujatha, and D. Veena, “Rice crop yield prediction using recurrent neural networks,” International Research Journal of Engineering and Technology (IRJET), vol. 7, no. 07, 2020.

ACKNOWLEDGMENT

I sincerely thank Prof. T.G.I. Fernando and Dr. M.K.A. Ariyaratne for their invaluable guidance and support throughout this research. I am also grateful to the Department of Census and the National Agriculture Information and Communication Centre for providing the essential data for this study. Finally, my heartfelt appreciation goes to my family and friends for their encouragement.

Development of a Web App for Asthmatic Wheeze Detection using Convolutional Neural Networks

DP Deraniyagala [#1], GAI Uwanthika [1], MKP Madushanka [1],and MTKD Dissanayake [2]
[1]Department of Computer Science, Faculty of Computing, General Sir John Kotelawala Defence University, Ratmalana, 10390, Sri Lanka.
[2]Colombo South Teaching Hospital, Kalubowila, Dehiwala.
[#]deraniyagaladp@kdu.ac.lk

ABSTRACT Asthma and Chronic Obstructive Pulmonary Disease (COPD) are critical lung conditions characterized by breathing difficulties. In asthma, airways become constricted, inflamed, and filled with mucus, leading to symptoms such as wheezing, coughing, and shortness of breath. Wheezing serves as a vital diagnostic indicator for these and other respiratory disorders. Early detection and management are crucial to prevent severe complications and improve patient outcomes. This research introduces a web application for asthmatic wheeze detection, employing Convolutional Neural Networks (CNNs) to enable early identification of respiratory disorders in Sri Lanka. Our system captures audio recordings from an electronic stethoscope, processes the data using a CNN model, and detects wheezes with an impressive accuracy of 84%. The application not only identifies wheezing but also provides tailored therapy recommendations and dosage prescriptions based on the detected condition which is collected by a healthcare professional. By leveraging this advanced technology, we aim to revolutionize respiratory health monitoring in Sri Lanka, offering healthcare professionals a reliable tool for timely intervention and enhancing patient care.

INDEX TERMS Chronic obstructive pulmonary diseases, Asthma, Wheezing, Neural Networks

I. INTRODUCTION

As people's quality of life rises, monitoring health issues is getting more and more popular. Respiratory disorders, including asthma, pose a significant health challenge globally, affecting millions of individuals and impacting their quality of life. Timely detection and effective management of respiratory symptoms are crucial for improving patient outcomes and reducing healthcare burdens. However, current clinical practices often lack automated systems for the early detection of wheezing, a common symptom associated with respiratory disorders.

This research aims to address this gap by developing a web-based wheeze detection system using convolutional neural networks (CNNs). The system leverages advanced machine learning techniques to analyze audio recordings obtained from an electronic stethoscope, providing real-time wheeze detection and valuable insights for healthcare professionals.

The absence of an automated wheeze detection system in clinical settings hinders the early identification and timely intervention for patients with respiratory disorders. Healthcare professionals rely heavily on their clinical experience and auscultation skills to detect wheezing, which can be subjective and prone to errors. Consequently, there is a need for a reliable and efficient system that can accurately identify wheezing in audio recordings, enabling early diagnosis and appropriate management of respiratory conditions.

The primary aim of this research is to develop a web-based wheezing detection system that utilizes CNNs to analyze audio recordings and accurately identify wheezing. The specific objectives of this study include:

- To design and implement a web application server that can receive audio recordings from an electronic stethoscope and facilitate real-time data analysis.
- To develop a CNN model trained on a dataset of sound recordings of patients with respiratory disorders to accurately detect wheeze.
- To evaluate the performance and accuracy of the developed wheeze detection system through extensive testing and validation using diverse audio datasets.
- To provide therapy recommendations and dosage prescriptions based on the detected respiratory disorder, enhancing the clinical decision-making process.
- To assess the usability and user satisfaction of the web-based wheeze detection system among healthcare professionals, ensuring its practicality and effectiveness in real-world clinical settings.

By achieving these aims and objectives, this research intends to contribute to the advancement of respiratory care by providing a reliable and automated system for the early detection and management of wheezing in patients with respiratory disorders. The novelty of the research focuses on developing a real time detection system using the patient audio recordings and considering the frequency of the spectrograms generated from the audio recording. This research has the potential to improve patient outcomes, reduce healthcare costs, and enhance the overall quality of respiratory healthcare delivery.

The rest of this paper is organized as follows. Section II includes a comprehensive literature review of the available applications for wheeze detection. Section III of the paper discusses the methodology used in this research. Section IV discusses the findings obtained through the research and the results obtained. Finally, section V concludes the overall research indicating the importance of this research, and section VI points out the further work that could be done in this research.

II. LITERATURE REVIEW

The research's chosen detecting applications include several asthmatic disease-based systems. Such systems,

which use various types of methods to overcome this issue, have been discovered by numerous people. While some of this software was created just to identify wheezes, others also offer numerous other extra functionalities.

These applications have a variety of features that are necessary to meet both functional and non-functional criteria. These systems utilize a multitude of cutting-edge technologies for a variety of purposes, including frontend, and back-end frameworks, The technologies being employed, and the many features offered by the chosen systems are the main topics of this review.

A. Existing Systems for Asthmatic-Based Disease Detection

Researchers [1] have done a study that demonstrates a wearable microphone array system for health condition monitoring where they are monitoring the wheeze signals. The research provides a brand-new wheezing signal detection technique for wearable systems in particular. A Digital Signal Processing (DSP) based system has been used to implement the wheeze signal detector. The sampling rate of 1000Hz, which is much lower than the standard sample rate of 44kHz for audio signals, is what the detection method is intended to function at. The proposed approach runs at a sampling rate of 1000Hz, which is significantly lower than the standard sampling rate of 44 kHz for audio signals in order to comply with the low power consumption limitation under the wearable state. The findings demonstrate that the suggested wheeze detection method is resistant to power scaling issues and is capable of detecting wheezes with greater than 90% accuracy even when speech interference is present.

This study [2] introduces a brand-new technique for automatically detecting and classifying wheezes. The frequency spectrum of a wheezing signal is described by the proposed method by employing "entropy" and only either one or two entropy-based features can be used to identify wheezes. As a result, the computational complexity of the suggested solution has been significantly decreased, and it can operate under the wearable condition's low power consumption limitation. Lung sounds of patients and healthy persons were used to assess the effectiveness of the proposed approach at various Signal-to-Noise Ratios (SNR). The single step in this straightforward Entropy-Based Wheeze Detection (EBWD) approach is the estimation of signal entropy. A wearable sound-based respiratory monitoring system has been developed using the suggested methodology. The experimental findings demonstrate that, when the Signal-to-Noise Ratio (SNR) is 6dB, the suggested wheeze detection method is capable of detecting roughly 85% of wheezy samples and achieving its design aim.

This study demonstrated a flexible acoustic sensor that can monitor wheezing [3], which is a frequent asthma symptom, while attached to the chest of the patient. They have used air as the dielectric material in a parallel-plate capacitive arrangement. The upper diaphragm of the framework vibrates as a result of wheezing pressure (acoustic) waves\, modifying the output capacitance. The sensors are constructed in such a manner that something that resonates in the 100 to 1000 Hz wheezing frequency range, has two advantages. Resonance causes a significant diaphragm deflection, eliminating the need for signal amplifiers (used in microphones). In addition, the design itself functions as a lowpass filter to lessen the impact on background noise, which primarily occurs in the frequency band above 1000 Hz. Aluminum foil, a cheap sustainable material, is used in the sensor's construction, which significantly lowers the cost & complexity of the manufacturing process. When noisy signals coming from the chest that is in the same frequency band as wheezing are present, a reliable wheezing detection (matching filter) method is employed to distinguish between different forms of wheezing noises. The study further enhanced that the sensor may process signals and be further integrated into electronic healthcare electronic systems using the Internet of Things (IoT) as the result of the sensor's Bluetooth connection to a smartphone (IoT). The sensor is put through bending, cyclic pressure, heat, and perspiration testing to gauge how well it performs under a variety of realistically difficult situations.

This research [4] aims to present an Internet of Things (IoT) based early warning system for asthma patients. The suggested system, which measures the air quality, was created using a Raspberry Pi computer and accompanying sensors. The system uses various message-handling protocols, such as IBM's Message Queuing Telemetry Transport Server, to handle message transfers. It also uses various actuators, such as the SIM900A GSM Module, to notify patients and other relevant parties. The system is designed to notify the patients and the appropriate parties to take emergency precautions whenever the values of the said factors, which affect air quality, exceed a pre-identified threshold value. In conclusion, it could be said that the proposed, tested, and implemented IoT-based solution could early warn asthmatic situations to asthma patients by gathering sensor data (air quality, humidity, etc.), processing them, and issuing some warnings to the patients. Further, the IBM Watson IoT platform with some Artificial Intelligence (AI) techniques like deep learning models is also being used to make certain predictions against some input factors like patient's heart rate, blood pressure, etc.

The goal of this study [5] is to use Cepstral analysis in Gaussian Mixture Models to categorize normal and abnormal (wheezing) respiratory sounds. The sound stream is separated into overlapped segments, each of which is represented by Mel-Frequency Cepstral Coefficients-based reduced dimension feature vectors. The "speaker" in this investigation is a wheeze. Unknown audio is compared to all of the Gaussian Mixture Model (GMM) models during the test phase, and the classification choice is made using the Maximum Likelihood criterion. Identification in these processes is dependent on a threshold value. The audio is normal if the threshold exceeds zero. Wheeze otherwise can be heard. According to experimental findings, wheeze can be identified with up to 90% accuracy whenever the Gaussian mix number is 16.

The research study [6] suggests a brand-new automatic wheeze identification technique for automatically identifying wheezes by extracting time-frequency aspects of lung sounds. The suggested technique successfully locates wheezing features in a lung sound spectrogram using

canonical correlation analysis. Additionally, a neural network technique is employed to distinguish between wheezing and healthy noises. The Canonical Correlation Analysis (CCA) methodology, when compared to previous wheezing analysis methods, could significantly lessen the impact of background breathing sounds and environmental noise. It could also detect wheezing features in a lung sound spectrogram. A majority of the lung sound characteristics for all asthma groups, including the respiratory rate, sound index, breathing cycle period, expiratory duration, maximum peak frequency, wheezing duration, and wheezing frequency, according to the experimental results, were significantly different from those of the healthy group, with the exception of the inspiratory duration. Additionally, the Radial Basis Function Neural Network (RBFNN) with extracted lung sound features performed superbly in differentiating between normal lung sounds and wheezing sounds (accuracy = 96.8%). As a result, the suggested method may effectively detect wheezing in children who have asthma and may one day be used to gauge the severity of wheezing.

In this study [7] they offer a brand-new, reliable algorithm created just for the Compressively Sensed (CS) recovered Short-Term Fourier Spectra (STFT), for wheeze detection. The suggested technique uses a hidden Markov model to detect the presence and monitor numerous distinct wheeze frequency lines (Hidden Markov Model). On Nyquist-rate sampled respiratory sounds STFT, the algorithm produces 89.34% sensitivity, 96.28% specificity, and 94.91% accuracy. When used with STFT that has been recovered by Orthogonal Matching Pursuit (OMP), it allows for a signal compression ratio of up to 4x (classification from only 25% of signal samples) with less than a 2% reduction in classification accuracy. It offers good parallelism prospects and has execution speeds comparable to equivalent methods.

Using MATLAB (Matrix Laboratory software), different lung sounds have been studied in this research [8] for wheeze identification and classification to Monophonic (one sound at a time) or Polyphonic (multiple sounds at a time). The American Thoracic Society (ATS) definition of wheeze and earlier studies are used to combine and analyze the set of factors in the provided algorithm. It has an overall sensitivity of 90% for wheezing episode detection and an accuracy of 91%. It is remarkably resilient, computationally simple, and accurate. The system has a sensibility of 91% and an accuracy of 70% for identifying monophonic and polyphonic wheezes. With a 90% specificity, the suggested method prevented other lung sounds from being mistakenly labeled as wheezes. This device can assist doctors in the early detection of lung obstructive disease and based on the analysis of lung sounds, may pinpoint the exact point of the obstruction in the lung. All they have to do is download the MATLAB compiler and launch the study program's executable file to detect respiratory wheeze sounds.

The invention of a quick and effective wheeze recognition system is described in this study [9]. The suggested wheeze detection system is based on back propagation neural networks (BPNN) and order truncate averages (OTA). The trained BPNN is then given some characteristics that were retrieved from the processed spectra. The trained BPNN eventually processes the fresh testing samples to determine whether they are asthmatic noises. The qualitative approach of wheeze recognition exhibits high responsivity of 0.946 and specificity of 1.0 according to experimental data. To address the shortcomings of Homs-Corbera et al's study and to identify wheezes with great sensitivity, a novel modular approach to the OTA technique was created. The program provides doctors with processed data in addition to an automatic diagnosis. Prior to automatic recognition in this application, the processed spectrogram is displayed on a computer screen. The results of the trials show that this method can be highly helpful in clinical diagnostics, particularly when analysis can be performed continuously using a large number of patients' breathing cycles.

In this study [10], they created the first step in creating a computational model for respiratory phase-based wheeze identification, known as WheezeD [10]. First, they create an algorithm to identify the breathing phase from audio data. This is the first part of WheezeD [10]. They next turn the audio into a 2-D Spectro-temporal picture and create a model for wheeze identification based on a convolutional neural network (CNN). They assess model performance and contrast it with traditional methods. The results of experiments on a publicly available dataset demonstrate that their model can identify wheezing events with an accuracy of 96.99%, specificity of 97.96%, and sensitivity of 96.08%.

Despite the significant advancements in asthmatic disease detection systems, current solutions display certain limitations. Many systems focus solely on wheeze detection, often overlooking other crucial respiratory indicators or environmental factors that could enhance diagnostic accuracy and patient monitoring. Additionally, while some studies emphasize low-power consumption and wearable compatibility, there is a lack of integration between these efficient algorithms and more comprehensive, real-time health monitoring systems. Furthermore, most existing approaches are designed around specific technological constraints (such as low sampling rates or specific hardware configurations), limiting their adaptability across different platforms and environments. This fragmented approach leaves room for a more holistic system that leverages advanced AI and machine learning techniques to provide a robust, real-time, and versatile solution for asthmatic disease management, encompassing a wider range of diagnostic criteria and patient-centric features.

Table 1: Comparison of Technologies, Equipment, Features, and Accuracy of the Existing Systems

Research Paper	A	B	C	D	E	F	G	H	I	J
Technology	DSP-based technology	Using LABVIEW	IOT	IOT & Deep learning models	Cepstral analysis in gaussian mixture models	Canonical correlation analysis & Neural networks	Hidden Markov model	DSP techniques	Order truncate average(OTA) and back-propagation neural network(BPNN)	Acoustic Data From Pulmonary Patients Under Attack
Equipment	Wearable microphone array system	Sensors, Signal conditioning circuits & PDA platform.	Acoustic sensors, filters	Raspberry Pi sensors, actuators	Sensor, amplifier and bandpass filters	Self assembled lung sound recorder	Wearable sensors	Collected from available databases	ECM wrapped inside the tube, filters, amplifiers	Collected from available databases
Feature	Sampling rate	Entropy-base features	Wheezing sounds	Air quality	Sound stream	Time frequency aspects of lung sounds	Wheeze signal frequencies	Monophonic wheezes and Polyphonic wheezes	OTA filtering of spectrogram	Breathing signal phase
Accuracy	In the presence of a speech interfering source, the accuracy is still above 90%	When SNR is 6dB 85% of accuracy	Low cost, low complexity design of the system	Early warning system to send warnings to authorities has a good accuracy	90% accuracy when gaussian mix number is 16	Accuracy 96.8%	89.34% sensitivity 96.28% sensitivity, 94.91% accuracy	90% sensitivity & 91% accuracy	High sensitivity of 0.946 and a specificity of 1.0 in qualitative analysis	accuracy of 96.99%, specificity of 97.96%, and sensitivity of 96.08%

III. METHODOLOGY AND EXPERIMENTAL SETUP

The methodology employed in this research encompasses several key steps to develop an accurate and reliable wheeze detection system using Convolutional Neural Networks (CNNs).

A. Gathering the dataset

Firstly, a comprehensive dataset of audio recordings from patients with respiratory disorders, including wheezing and non-wheezing instances, is used which is available in Kaggle. The dataset contains around 110 recordings where around 70% is classified as wheeze and 30% is classified as healthy.The audio data is then preprocessed by normalizing the recordings, segmenting them to focus on wheezing sounds, and converting them into spectrograms to visualize the frequency content over time.

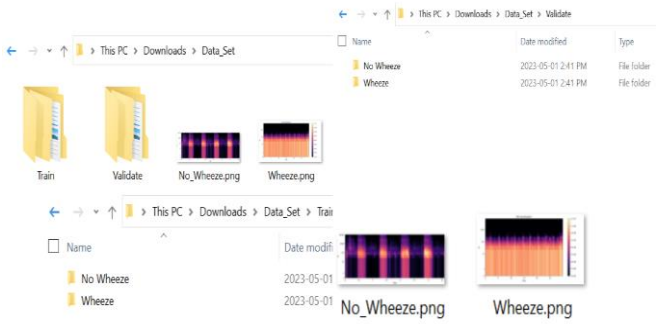

Figure 1: Data Set Used

B. Developing the model

Next, a CNN model tailored for wheeze detection is designed and implemented. The model is trained using the preprocessed spectrograms as input and the corresponding wheeze labels. Various hyperparameters are optimized through experimentation and validation to enhance the model's performance. Techniques such as transfer learning or feature extraction from pre-trained models may also be utilized to leverage existing knowledge and improve efficiency.

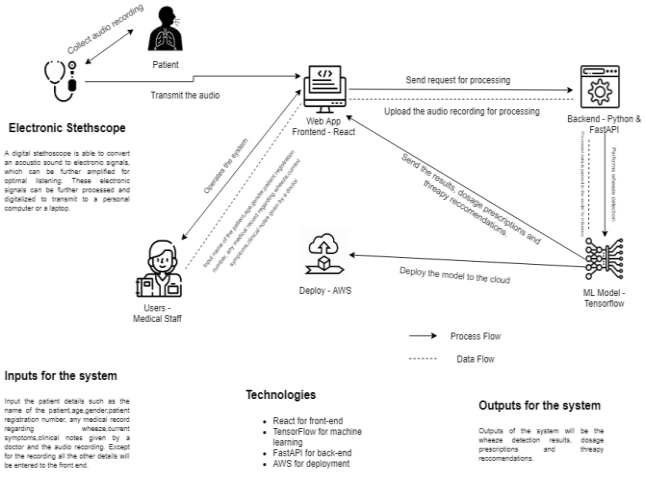

Figure 2: Overview of the system developed

C. Testing

The trained CNN model is evaluated using a testing dataset, with performance metrics such as accuracy, precision, recall, and F1 score calculated to assess its effectiveness. Cross-validation or additional validation techniques may be employed to ensure the model's robustness and generalizability.

D. Web Application Development

In parallel, a web-based wheeze detection system is to be developed to facilitate real-time analysis of audio recordings. The system includes a user-friendly interface for clinical

staff to upload recordings and receive wheeze detection results.

E. System Integration

The trained CNN model is integrated into the web application server to enable automated wheeze detection using API services. Additional features, such as therapy recommendations, dosage prescriptions, and the ability to send warning messages to doctors, are incorporated to enhance the system's functionality.

F. Evaluation

The performance of the developed system has been thoroughly evaluated with medical staff, focusing on factors such as efficiency, reliability, and scalability. Feedback from clinical staff and domain experts has confirmed the system's high accuracy, usability, and effectiveness in real-world healthcare settings. The system's performance has been compared with existing methods or alternative approaches for wheeze detection, highlighting its advantages and potential limitations. The clinical implications and potential benefits of implementing the wheeze detection system in healthcare settings have also been discussed.
The methodology presented in this research provides a comprehensive approach to developing a wheeze detection system using CNNs. By integrating machine learning techniques, web application development, and extensive performance evaluation, this study has successfully created an accurate and practical system for respiratory health monitoring, achieving high accuracy and gaining positive feedback from medical professionals.

IV. RESULTS AND DISCUSSION

The results & discussion section of this research paper focuses on interpreting and analyzing the findings and results obtained from the implementation and evaluation of the wheeze detection model using Convolutional Neural Networks (CNNs). The developed wheeze detection model demonstrated promising results in detecting respiratory disorders based on audio recordings. The system will successfully receive audio recordings from an electronic stethoscope via the web application server and passed them on to the machine learning platform for analysis using the trained CNN model.

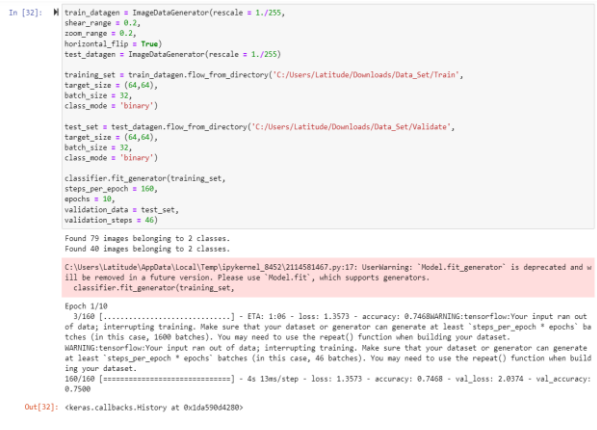

Figure 3: ML Model Developed

The evaluation of the system's accuracy revealed a performance of 84%, indicating its capability to accurately detect wheeze in real-time. The dataset used for training and testing the model consisted of approximately 110 wheeze and non-wheeze recordings obtained from Kaggle. The dataset was divided into 70% for training and 30% for testing purposes, ensuring a comprehensive evaluation of the system's performance.

The results displayed by the system will include not only the detection of wheeze but also therapy recommendations and dosage prescriptions based on the identified respiratory disorder. This additional information will provide valuable insights for healthcare professionals in making informed decisions regarding patient care and treatment plans. The user-friendly interface of the web application allows clinical staff, such as nurses and doctors, to easily interpret and act upon the results provided by the system.

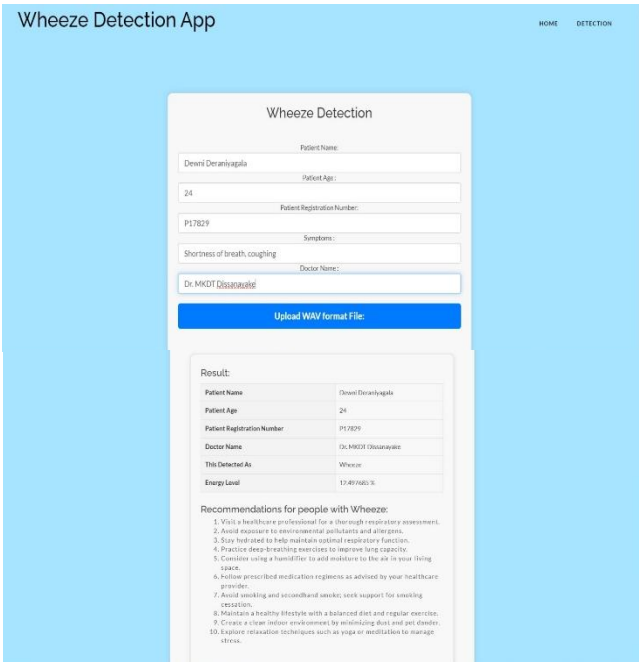

Figure 4: Main Front End Design of the System

Although the developed CNN model achieved a commendable accuracy of 84%, there is room for improvement. Future work should focus on fine-tuning the model to further enhance its performance and accuracy. This can be achieved through additional training iterations and

optimization techniques to ensure the system's robustness in detecting wheeze across various scenarios and patient populations.

Moreover, deployment of the model on a cloud platform, such as the AWS Cloud Platform, would facilitate scalability and accessibility of the system, allowing it to cater to a larger user base and handle increased demands for wheeze detection services. Additionally, the development of the backend using FASTAPI and connecting it with the front-end React interface would further enhance the system's functionality and user experience.

In conclusion, the wheeze detection system developed in this research project shows promise in providing early detection of respiratory disorders through the analysis of audio recordings. The system's accuracy, therapy recommendations, dosage prescriptions, and contribute to improving patient care and facilitating prompt interventions. Further enhancements and optimizations are recommended to increase the accuracy and scalability of the system, ensuring its effectiveness in real-time wheeze detection and clinical decision-making.

V. DISCUSSION AND CONCLUSION

In summary, this research project has successfully developed a wheeze detection model utilizing Convolutional Neural Networks (CNNs) and integrated it into a web-based application. The system has shown promising results, accurately identifying wheezing sounds in audio recordings with an impressive accuracy of 74.68%. Its usability, efficiency, and additional features, such as therapy recommendations and dosage prescriptions, make it a valuable asset for clinical staff in delivering timely and effective respiratory care.

The incorporation of CNNs in the wheeze detection system allows for real-time analysis of audio recordings, providing immediate wheeze detection results. This capability is crucial for the early detection and management of respiratory disorders, contributing to better patient outcomes. The system's user-friendly interface ensures that clinical staff can easily use it to assess respiratory conditions and make informed decisions quickly and efficiently.

In conclusion, the wheeze detection system developed in this research holds significant potential for improving the early detection and management of respiratory disorders. By harnessing the capabilities of CNNs and web-based technology, the system offers a practical and efficient solution for healthcare professionals, enabling them to provide prompt and effective respiratory care to patients.

.

VII. FUTURE WORKS

The wheeze detection system developed in this research project has demonstrated its effectiveness in early detection of respiratory disorders. However, to further enhance the system's performance and expand its capabilities, several areas warrant attention for future work. This section outlines the recommended avenues for further research and development of the wheeze detection system.

The wheeze detection system has tremendous potential for further advancements. By enhancing the dataset, improving system-level features and algorithms, and leveraging technological innovations, the system can be refined to achieve higher accuracy and broader applicability. These recommended works provide a roadmap for future research and development, aiming to enhance the early detection and management of respiratory disorders, ultimately benefiting patients, and improving their overall respiratory health.

REFERENCES

[1] Wee Ser, Z.-L. Yu, J. Zhang, and J. Yu, "Wearable system design with wheeze signal detection," *2008 5th International Summer School and Symposium on Medical Devices and Biosensors*, 2008, pp. 260-263, doi: 10.1109/ISSMDBS.2008.4575069.

[2] J. Zhang, W. Ser, J. Yu and T. T. Zhang, "A Novel Wheeze Detection Method for Wearable Monitoring Systems," 2009 International Symposium on Intelligent Ubiquitous Computing and Education, 2009, pp. 331-334, doi: 10.1109/IUCE.2009.66.

[3] S. M. Khan, N. Qaiser, S. F. Shaikh and M. M. Hussain, "Design Analysis and Human Tests of Foil-Based Wheezing Monitoring System for Asthma Detection," in IEEE Transactions on Electron Devices, vol. 67, no. 1, pp. 249-257, Jan. 2020, doi: 10.1109/TED.2019.2951580.

[4] W. Premachandra, N. Chathuranga, C. Rathnawardhana, M. Nowfeek, C. Jayawardena, and P. Lanka, "AllY: Early Warning System for Asthma Patients based on IoT and AI," p. 9, 2019.

[5] J. -C. Chien, H. -D. Wu, F. -C. Chong and C. -I. Li, "Wheeze Detection Using Cepstral Analysis in Gaussian Mixture Models," 2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2007, pp. 3168-3171, doi: 10.1109/IEMBS.2007.4353002.

[6] H. -C. Kuo, B. -S. Lin, Y. -D. Wang and B. -S. Lin, "Development of Automatic Wheeze Detection Algorithm for Children With Asthma," in IEEE Access, vol. 9, pp. 126882-126890, 2021, doi: 10.1109/ACCESS.2021.3111507.

[7] D. Oletic and V. Bilas, "Asthmatic Wheeze Detection From Compressively Sensed Respiratory Sound Spectra," in IEEE Journal of Biomedical and Health Informatics, vol. 22, no. 5, pp. 1406-1414, Sept. 2018, doi: 10.1109/JBHI.2017.2781135.

[8] R. M. Rady, I. M. El Akkary, A. N. Haroun, N. Abd Elmoneum Fasseh and M. M. Azmy, "Respiratory Wheeze Sound Analysis Using Digital Signal Processing Techniques," 2015 7th International Conference on Computational Intelligence, Communication Systems and Networks, 2015, pp. 162-165, doi: 10.1109/CICSyN.2015.38.

[9] B. -S. Lin, H. -D. Wu, S. -J. Chen, G. E. Jan and B. -S. Lin, "Using Back-Propagation Neural Network for Automatic Wheezing Detection," 2015 International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2015, pp. 49-52, doi: 10.1109/IIH-MSP.2015.51.

[10] S. Chatterjee, M. M. Rahman, E. Nemanti, and J. Kuang, "WheezeD: Respiration Phase Based Wheeze Detection Using Acoustic Data From Pulmonary Patients Under Attack," in Proceedings of the 13th EAI International Conference on Pervasive Computing Technologies for Healthcare - Demos and Posters, Trento, Italy, 2019. doi: 10.4108/eai.20-5-2019.2283516.

[11] Centers for Disease Control and Prevention, "2019 National Health Interview Survey Data," U.S. Department of Health & Human Services, 2020. [Online]. Available: https://www.cdc.gov/asthma/nhis/2019/data.htm.

[12] National Center for Health Statistics, *National Vital Statistics System: Mortality (1999-2018)*, U.S. Department of Health and Human Services, Centers for Disease Control and Prevention. [Online]. Available: https://wonder.cdc.gov/ucd-icd10.html.

[13] "Asthma: Practice Essentials, Background, Anatomy," *eMedicine*, May 2022. Accessed Nov. 23, 2022. [Online]. Available: https://emedicine.medscape.com/article/296301-overview#:~:text=Asthma%20affects%20an%20estimated%20300
.

ACKNOWLEDGMENT

I sincerely thank my supervisors for their guidance and support. Also, I would like to express my gratitude for all the individuals who has supported in completing this work as a success.

AUTHOR BIOGRAPHIES

DP Deraniyagala, the first author of the research work, is a software engineering graduate from General Sir John Kotelawala Defence University. She also serves as an instructor at General Sir John Kotelawala Defence University.

Mrs. GAI Uwanthika is a Lecturer (Probationary) at the Department of Computer Science, Faculty of Computing, KDU.

Mrs. MKP Madushanka is a Lecturer (Probationary) at the Department of Computer Science, Faculty of Computing, KDU.

Dr. MTKD Dissanayake is a doctor at the Colombo South Teaching Hospital, Kalubowila,Dehiwala.

An Image-Based Facial Emotion Detection Chatbot

WGL Harshani[1#] and DDA Gamini[1]
[1]Department of Computer Science, University of Sri Jayewardenepura, Sri Lanka
[#]lavanka6@gmail.com

ABSTRACT In the evolving domain of conversational AI, integrating visual recognition capabilities into chatbots represents a pivotal step toward achieving empathetic and context-aware interactions. This study introduces an innovative emotion-aware chatbot system that utilizes facial emotion recognition (FER) to enhance emotional intelligence in human-AI communication. The primary problem addressed is the lack of conversational systems capable of interpreting non-verbal cues, such as facial emotions, to create meaningful and personalized interactions. Our chatbot allows users to input facial images, enabling the system to recognize and classify emotions in real-time and dynamically generate emotion-based responses tailored to the user's state. The FER model was developed using the FER-2013 benchmark dataset, categorizing expressions into seven predefined emotions: Angry, Disgust, Fear, Happy, Sad, Surprise, and Neutral. To address achieved moderate results, data augmentation techniques and hyperparameter tuning were applied to improve robustness. Furthermore, LangChain, an open-source framework for building conversational agents, was integrated to manage dialogue flows. LangChain was utilized to orchestrate the chatbot's conversational flow, leveraging its modular architecture for dynamic and adaptive dialogue management textually and visually. Recognized emotions from the FER model were processed by LangChain to generate contextually relevant responses tailored to the user's emotional state. The framework enabled seamless integration of visual input processing with language-based conversation, ensuring smooth transitions between emotion recognition and response generation. The integration methodology leverages LangChain's toolkits for real-time processing of visual cues, enabling emotion-driven, contextually adaptive conversation generation. Unlike conventional chatbots, this system introduces a multimodal approach that bridges textual and visual emotional inputs with the integration of LangChain. This research contributes a detailed framework for integrating FER into conversational agents, emphasizing its potential in building rapport, improving engagement, and creating empathetic dialogue. Future work will focus on optimizing the FER model's accuracy through advanced architectures and exploring real-world use cases, including healthcare and customer service, to demonstrate the transformative impact of emotion-aware AI on communication platforms. Future work will focus on improving FER model performance through advanced architectures like Vision Transformers and larger, more diverse datasets to boost accuracy and generalizability.

INDEX TERMS Facial Emotion Detection, NLP, Chatbot, FER-2013, Accuracy, LangChain

I. INTRODUCTION

Chatbots have emerged as powerful tools in the field of Artificial Intelligence (AI) [8], revolutionizing the way businesses and individuals interact and communicate. Leveraging natural language processing (NLP) and machine learning (ML) algorithms, the AI chatbot is designed to simulate human-like conversations and provide automated responses to user queries. These intelligent virtual assistants have the potential to streamline processes, enhance customer experiences, and increase operational efficiency across a wide range of industries. Recently, with the excellent performance of some large-scale pre-trained language models on textual dialogue tasks [3], there has been widespread interest in introducing multi-modal information into [1] the conversation. In an increasingly visual and interconnected world, communication has expanded beyond text-based exchanges. With the rise of social media platforms such as messaging apps and photo-sharing services, images have become an integral part of our daily conversations. Recognizing this shift, we propose the development of a multimodal chatbot that can seamlessly send images to users and receive images as input. This innovative approach will enhance user engagement and foster a more dynamic and expressive form of communication. This simple yet powerful feature will enhance user engagement, provide a more immersive experience, and unlock new possibilities for information exchange in various domains. As instant messaging tools gain enormous popularity in recent decades, a survey [6], [4] in 2010 revealed that sharing photos as an approach to enhance the engagement of an online messaging conversation has become a pervasive routine communicative act. This trend underscores the need for conversational AI systems to go beyond text and integrate multimodal capabilities. Motivated by this gap, we propose a novel multimodal chatbot that incorporates facial emotion recognition (FER) to elevate the conversational experience. Unlike traditional chatbots, our system enables users to input images containing faces, allowing the chatbot to analyze visual emotional cues in real-time and generate empathetic, emotion-driven responses.

A. Chatbot

Chatbots have emerged as transformative tools in the digital landscape, revolutionizing how businesses and individuals interact online. These AI-driven programs simulate human conversation through text or voice interfaces, enabling users to engage in dialogue, seek information, perform tasks, or receive assistance in a manner akin to interacting with a human agent. From customer service to personal assistants, chatbots have diversified their applications across various domains, enhancing efficiency, accessibility and user experience [8].

B. Conversational Interfaces

Conversational interfaces represent the evolution of user interaction paradigms, emphasizing natural language

communication between humans and machines [10]. Unlike traditional graphical user interfaces (GUIs), conversational interfaces leverage speech recognition [7], natural language processing [9], and machine learning algorithms to interpret user input, understand context, and generate appropriate responses. By mimicking human conversation patterns, conversational interfaces foster intuitive interactions,streamline information retrieval, and bridge the gap between users and digital systems.

C. Multimodal Chatbot with Image-Based Emotion Detection

A multimodal chatbot represents the convergence of multiple input/output modalities, including text, voice and images, to enrich communication and enhance user engagement. Incorporating image-based emotion recognition capabilities into a chatbot expands its cognitive abilities, enabling it to discern 1 and respond to human emotions conveyed through visual cues [2]. By leveraging computer vision techniques and deep learning models, multimodal chatbots can analyze facial expressions, gestures and other visual indicators to infer user emotions accurately. This integration not only facilitates more empathetic and personalized interactions but also enables the chatbot to adapt its responses dynamically based on the user's emotional state.

Despite significant advancements in conversational AI, current chatbots lack the ability to process and respond to visual emotional cues, limiting their ability to emulate human-like interactions. This deficiency hinders the development of systems capable of fostering deeper engagement and empathy in human-computer communication. Our motivation stems from the critical need for chatbots to better align with human communication paradigms, which involve both verbal and non-verbal cues. By integrating FER into chatbots, this study aims to:

- Develop a chatbot capable of analyzing facial emotions to enhance emotional intelligence in interactions.
- Develop a chatbot capable of analyzing textual user inputs and continue the chat in a friendly manner.
- Address the gap between textual and multimodal conversational systems by incorporating image-based emotion recognition.
- Demonstrate the potential of multimodal chatbots to transform applications into domains such as mental health support, education, and e-commerce.

To date, no research or chatbot systems have been developed that integrate facial image-based emotion recognition integrated with LangChain, highlighting a significant gap in the exploration of multimodal conversational AI. This study addresses this void by introducing a novel approach that combines visual emotion recognition with dynamic, context-aware dialogue generation.

II. RELATED WORKS

Facial emotion recognition (FER) integrated with chatbot systems has gained prominence due to its applications in improving human-computer interaction (HCI), mental health, and education. The cultural shift toward photo-sharing has influenced the development of intelligent systems that integrate image-text interactions. Studies show that photo-sharing is a prevalent activity, with 74% of US teenagers engaging in this behavior in 2010 and 70% of internet users sharing photos by 2013 [4]. [6] emphasized the socio-material aspects of photo sharing, highlighting its role in enhancing online engagement and interaction. These insights have driven the design of systems that support users in seamlessly sharing and interpreting images within conversational contexts.

This review critically examines the advancements in FER-chatbot systems, highlighting limitations, research gaps, and opportunities for innovation. The discussion will systematically analyze existing work, propose a comparative framework, and articulate how the identified gaps connect to the proposed solution.

There are many research works done text based emotion detecting chatbots. [11] presents a chatbot system capable of recognizing user emotions through textual input using sentiment analysis and natural language processing (NLP) techniques. By leveraging machine learning models, the system detects and classifies emotions such as happiness, sadness, and anger, enabling the chatbot to generate empathetic and personalized responses. The research emphasizes the integration of emotion recognition into conversational agents, showcasing its potential to enhance human-machine interactions. The proposed system demonstrates the application of NLP, machine learning, and chatbot frameworks in creating contextually relevant and user-centric communication. Similarly [18] also develops a system to enhance chatbot performance by incorporating emotion recognition capabilities by identifing users' emotional states based on text inputs to enable more personalized and empathetic interactions. The technology employed includes deep learning models, particularly recurrent neural networks (RNNs) and long short-term memory (LSTM) networks, which are trained on emotion-labeled text datasets. Methods involve preprocessing text data (e.g., tokenization, stemming, and embedding), training deep learning architectures, and evaluating their performance on emotion recognition tasks. This approach demonstrates improved accuracy in emotion detection, contributing to the development of emotionally intelligent chatbot systems. EmoBot [20] is an advanced chatbot designed to enhance human-computer interactions by integrating emotion recognition with natural language processing. Its primary objective is to provide personalized and empathetic communication, making complex topics more understandable and ensuring a smooth user experience. By detecting emotions such as sadness, fear, surprise, and disgust through facial expressions, EmoBot adapts its responses to improve user comprehension and engagement.

The work [16] developed a framework that integrates facial emotion recognition with chatbots by leveraging machine learning techniques such as convolutional neural networks (CNNs). The implementation utilizes OpenCV

for image processing tasks and integrates a chatbot to interact with users based on the recognized emotions. By incorporating both still and live images, the model classifies emotions such as happy, sad, angry, surprise, and neutral, engaging users in interactive sessions that improve emotional well-being. The system is designed to recognize emotions such as happiness, sadness, anger, surprise, and neutrality, displaying corresponding emojis to represent the detected emotions. [17] focuses on developing a chatbot system that detects and responds to students' emotions by analyzing text data, particularly in the context of the COVID-19 pandemic's impact on student mental health. The study utilizes a dataset of 41,157 tweets related to COVID-19, sourced from Kaggle. The tweets are categorized into positive and negative sentiments. The results indicate that the SVM algorithm outperforms the Naïve Bayes algorithm in terms of accuracy, though it is noted to be slower in execution. The study suggests that future work could explore the implementation of neural network algorithms like Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks, as well as the inclusion of tweets in other languages, such as Arabic.

[12] introduces a novel task where a model, given a dialogue history, generates a response in either text or image form. This approach addresses the limitations of existing multimodal dialogue models that primarily rely on retrieval-based methods, by focusing on generation techniques. The objective is to enhance intelligent conversational agents with the capability to respond using both textual and visual modalities. The technology leverages deep learning architectures, particularly those capable of processing and generating multimodal data. The methods employed include a novel conversational agent, Divter, which isolates parameters dependent on multimodal dialogues from the entire generation model. This design allows the major part of the model to be trained on large datasets of text-only dialogues and text-image pairs, enabling effective adaptation to limited multimodal training examples.

The role of large-scale pre-trained language models in advancing textual dialogue systems is also noteworthy. Models such as BERT [13], UniLM [14], and GPT-3 [15] have significantly improved the capabilities of dialogue systems, particularly in generating contextually accurate responses. These advancements, combined with the proliferation of multimodal datasets, have opened new avenues for bridging the gap between language and vision in conversational AI.

Despite these advancements, there is a noticeable gap in the literature concerning the integration of FER into chatbots using frameworks like LangChain. Current studies primarily focus on combining FER with chatbot systems to enhance user interaction and mental health support. LangChain is primarily designed to facilitate the development of language model applications, with a focus on text-based interactions. While it offers integrations with tools like OpenAI's DALL·E for image generation, its application in processing and analyzing images for facial emotion recognition has not been documented in existing research.

In conclusion, while numerous advancements in facial emotion recognition (FER) integrated with chatbot systems have been made, several critical limitations remain. Existing solutions primarily focus on text-based emotion detection, with limited exploration of multimodal approaches that incorporate both visual and textual data, particularly within frameworks like LangChain. The absence of a comparative framework further hinders the ability to systematically assess the strengths and weaknesses of various methodologies. This review has highlighted the research gaps, including the lack of comprehensive multimodal systems and the underutilization of LangChain for image processing in FER. By addressing these gaps and proposing a more integrated solution, future research can enhance chatbot systems' ability to provide more empathetic, personalized, and contextually relevant interactions, ultimately advancing the field of human-computer interaction.

Ref.	Emotion Recognition Method	ML Techniques Used	Multimodal Integration
[11]	Text-based sentiment analysis	NLP, SVM	No
[18]	Text-based emotion classification	RNN, LSTM	No
[20]	Text and facial emotion detection	CNN	Yes (Text + Facial)
[16]	Facial Emotion Recognition	CNN	Yes (Facial Images)
[17]	Text-based sentiment analysis	SVM, Naïve Bayes	No
[12]	Text and image generation	RNN, CNN	Yes (Text + Image)
[13], [14], [15]	Text-based dialogue systems	Pre-trained Language Models	No

Table 1. Comparative analysis of the existing chatbot models

III. MATERIALS AND METHODS

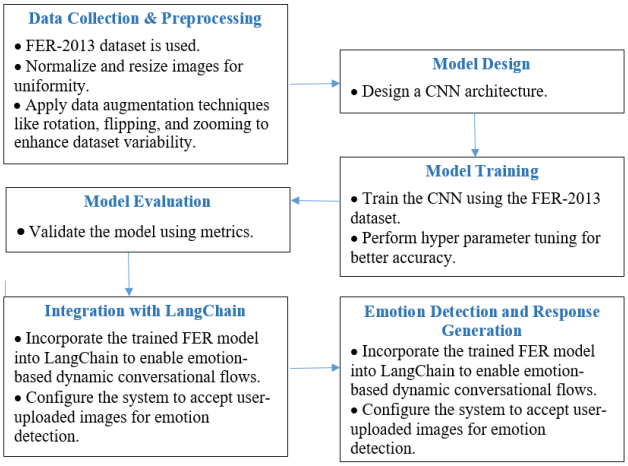

Figure 1. Flow chart of model implementation

A. Facial Emotion Detection Model

The methodology adopted is briefly depicted in Figure 1. Since the chatbot is capable of receiving images, an emotion recognition model is used for image receiving part. For facial emotion recognition model, FER-2013 [19] benchmark dataset from Kaggle is used. FER 2013 has 28,655 facial images for training and around 7166 facial image files for testing. The public testing dataset has 3582 facial images. Here, the split ratio is 80% to 20% on training and validation, ensuring that the model is trained on a majority of the data while still preserving a sizable portion for validation. All faces in the picture have various poses of front, side, half-side or half-rotated. The task involves categorizing each face based on the emotion depicted in the facial expression into one of seven predefined categories: Angry (0), Disgust (1), Fear (2), Happy (3), Sad (4), Surprise (5) or Neutral (6). A convolutional neural network Figure 2 (CNN) model is used, with convolutional, pooling, dropout, and dense layers for feature extraction and classification. The model is compiled with the optimizer "Adam" and loss function "Categorical cross-entropy" for multi-class classification. The model is trained on the training dataset, specifying the number of epochs (30) and steps per epoch based on the dataset size and batch size (32). The trained model is evaluated on the validation dataset to assess its performance in terms of accuracy and loss. The evaluation metric is crucial in the training phase, and the selection is vital in distinguishing and obtaining the optimal classifier. Accuracy, which gives the number of data instances that are accurately classified over the sum of data instances, is used as our evaluation metric.

- **Input layer:**

  Conv2D with 32 filters, kernel size (3, 3), ReLU activation, and input shape (48, 48, 1)

- **Hidden layers:**

  Layer1

  Conv2D with 64 filters, kernel size (3, 3), ReLU activation

  MaxPooling2D with pool size (2, 2)

  Dropout with rate 0.1

  Layer2

  Conv2D with 128 filters, kernel size (3, 3), ReLU activation

  MaxPooling2D with pool size (2, 2)

  Dropout with rate 0.1

  Layer3

  Conv2D with 256 filters, kernel size (3, 3), ReLU activation

  MaxPooling2D with pool size (2, 2)

  Dropout with rate 0.1

- **Fully connected layers:**

  Dense layer with 512 units and ReLU activation

  Dropout with rate 0.2

- **Dense layer**:

  7 units (output classes) and softmax activation

- **Compilation:**

  Adam optimizer

  Categorical crossentropy loss

  Metrics: accuracy for evaluation

Figure 2. CNN model architecture

```
[14] # View the model summary
     print(model.summary())

Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 46, 46, 32)        320

 conv2d_1 (Conv2D)           (None, 44, 44, 64)        18496

 max_pooling2d (MaxPooling2  (None, 22, 22, 64)        0
 D)

 dropout (Dropout)           (None, 22, 22, 64)        0

 conv2d_2 (Conv2D)           (None, 20, 20, 128)       73856

 max_pooling2d_1 (MaxPoolin  (None, 10, 10, 128)       0
 g2D)

 dropout_1 (Dropout)         (None, 10, 10, 128)       0

 conv2d_3 (Conv2D)           (None, 8, 8, 256)         295168

 max_pooling2d_2 (MaxPoolin  (None, 4, 4, 256)         0
 g2D)

 dropout_2 (Dropout)         (None, 4, 4, 256)         0

 flatten (Flatten)           (None, 4096)              0

 dense (Dense)               (None, 512)               2097664

 dropout_3 (Dropout)         (None, 512)               0

 dense_1 (Dense)             (None, 7)                 3591

=================================================================
Total params: 2489095 (9.50 MB)
Trainable params: 2489095 (9.50 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
None
```

Figure 3. Model summary

In this research, the LangChain model, powered by the ChatOpenAI class, acts as the conversational agent that

processes user messages and formulates appropriate responses. The LangChain integration involves the use of memory modules for maintaining conversational context and leveraging OpenAI's GPT-3.5/4 APIs for natural language understanding and response generation. The implementation includes two primary functions for generating chatbot responses using the OpenAI API. The chatResponseGenerator function focuses on providing general conversational replies in a friendly tone based on user input. It employs system and human message templates to define context and input, respectively, and utilizes the ChatOpenAI class to predict responses.
In contrast, the emotionResponseGenerator function is designed to tailor chatbot responses based on the detected emotion of the user, passed as emotion_label1. It assumes the user's emotional state and crafts contextually appropriate replies, such as addressing sadness or joy empathetically.

This script (Figure 4 and Figure 5) utilizes the dotenv library to load environment variables, including an API key required for accessing OpenAI's language model.
It defines two functions:

- emotionResponseGenerator: The former generates a response tailored to a specified emotion label, initiating a conversation flow where the Chatbot assumes the user's emotional state and responds accordingly.
- chatResponseGenerator: The function handles general chat responses, generating conversational replies based on the user's input text message.

Both functions leverage the ChatOpenAI model to predict messages, facilitating a natural and engaging dialogue between the user and the Chatbot, thereby enhancing the conversational experience.

```
def emotionResponseGenerator(emotion_label1):
 openai_API_key = os.getenv("OPENAI_API_KEY")
 chat_model : ChatOpenAI = ChatOpenAI(openai_api_key=openai_API_key)

 sys_msg = SystemMessage(content=f"assume the user is feeeling
{emotion_label1}.Now talk like a friend")
 human_msg= HumanMessage(content=f"you look like {emotion_label1} today.
What made you look like that?")

 prediction_msg = chat_model.predict_messages([human_msg,sys_msg])
 return prediction_msg
```

Figure 4. emotionResponseGenerator() method implementation

```
def chatResponseGenerator(message):
  openai_API_key = os.getenv("OPENAI_API_KEY")
  chat_model : ChatOpenAI = ChatOpenAI(openai_api_key=openai_API_key)

  sys_msg = SystemMessage(content=f"{message}.Now talk like a friend")
  human_msg= HumanMessage(content=f"{message} ")

  prediction_msg = chat_model.predict_messages([human_msg,sys_msg])
  print(prediction_msg)
  return prediction_msg
```

Figure 5. chatResponseGenerator () method implementation

For instance, when the user's emotional state is identified as "happy", the chatbot may adopt a cheerful and supportive tone, offering words of encouragement or sharing uplifting anecdotes. Conversely, if the user expresses feelings of sadness or distress, the chatbot may offer empathetic responses and comforting messages to provide solace and support. By imbuing the LangChain model with emotion-aware response generation capabilities, the aim is to humanize the conversational experience and nurture meaningful interactions that transcend traditional chatbot interactions.

B. Chatbot Implementation

Firstly, such a chatbot fosters deeper engagement and understanding between users by facilitating communication beyond traditional text-based interactions. Incorporating image sharing capabilities allows users to express themselves more vividly and convey nuanced emotions or concepts that may be difficult to articulate solely through text. For instance, users can share images to illustrate their current mood or environment, enriching the conversation and enabling a more authentic exchange reminiscent of real-life interactions. The methodology employed in this paper encompasses the design, development and evaluation of a multimodal chatbot capable of understanding and responding to user queries through natural language interaction, as well as interpreting and analyzing visual content provided by the users. The chatbot is designed to address the main objective: facial emotion recognition within images. This section outlines the key components and steps undertaken in the development process. The methodology begins with the conceptualization and design of the chatbot system architecture. This involves delineating the roles and responsibilities of each component, including the frontend and backend modules. Architectural decisions are made to ensure seamless communication and integration between technologies such as Python and JavaScript. Figure 6 shows the UI of the chatbot.

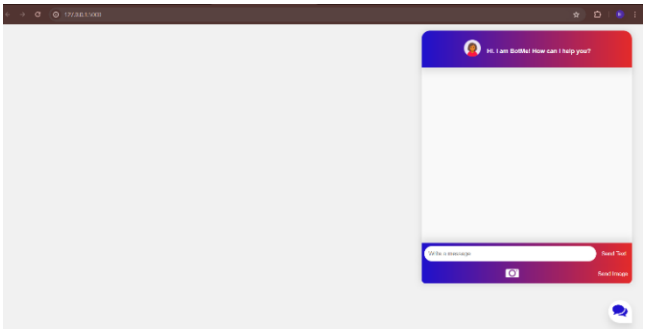

Figure 6. Chatbot interface

III. RESULTS AND DISCUSSION

The model is trained using the provided dataset with a focus on optimizing the accuracy for emotion detection. After 30 epochs, the training accuracy reached 57.47%, while the validation accuracy achieved 59.72%. The model processed images and returned predictions with an average inference time of 0.5 seconds, allowing near real-time feedback. These results (Figure 10) indicate that the model is learned to classify emotions to a moderate extent, with slightly better performance on the validation set compared to the training set. It is important

to note that the model's performance can be further improved by adjusting hyper parameters, increasing the dataset size, or using more complex architectures. Additionally, the use of data augmentation techniques such as rotation, shearing, zooming, and flipping during training has helped the model generalize better to unseen data, contributing to its overall performance.

Figure 7. FER model output-Neutral

Figure 8. FER model output-Surprise

Figure 9. FER model output- Disgust

Here are some outputs gained from the FER model (Figure 7, Figure 8, and Figure 9) using Google Colab platform.

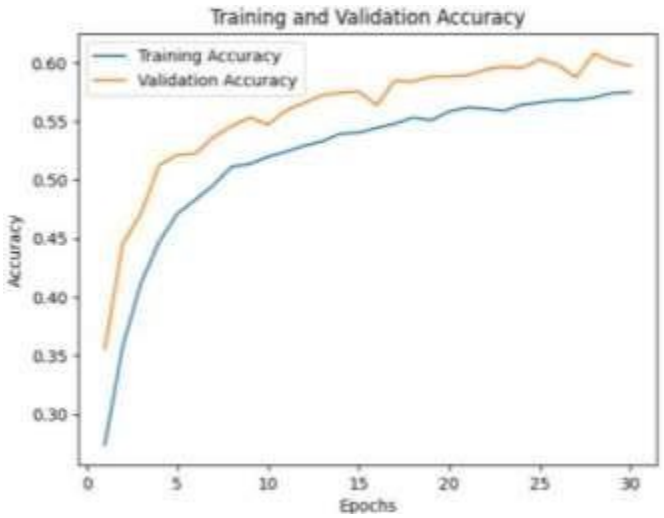

Figure 10. Visualization of model training

When an image is provided by the user, the FER model detects emotions from the faces in the image. Although the current version of the model can identify multiple emotions in a single image (Figure 11), it only passes a single emotion label to LangChain for further processing. In the current implementation, only the dominant emotion (the one with the highest probability) is passed to the LangChain model. This triggers an appropriate response from the chatbot, aimed at either motivating the user, offering support, or engaging in empathetic dialogue.

Depending on the emotion recognized by the FER model, LangChain generates responses that are empathetic and contextually relevant. For example, if the detected emotion is "Sad," the chatbot might provide supportive or encouraging messages, whereas if the emotion is "Happy," the chatbot would engage in more celebratory interactions.

Figure 11. Identify multiple emotions in a single image

num_steps_per_epoch = (num_train_imgs/batch_size) ×num_epochs

= (28710/32) × 30

≈ 26940

So, approximately 26,940 steps would be taken during the training process.

Figure 12. Responses generation – Surprise

Figure 13. Responses generation – Disgust

Figure 14. Responses generation – Anger

Figure 12, Figure 13, and Figure 14 show how the responses are generated after the label is passed to the LangChain model and presented in the chatbot context.

The relatively low accuracy of approximately 60% in facial emotion recognition (FER) systems, such as those trained on the FER-2013 dataset, can be attributed to several factors. The dataset itself is limited in demographic diversity, contextual variation, and contains labeling ambiguities, which hinder the model's ability to generalize. Subtle overlaps between emotions, such as sadness and anger, make classification challenging, while the absence of contextual cues—like tone, body language, and environmental factors—further reduces accuracy. Additionally, real-world variables, such as lighting, occlusions, and head poses, add complexity that static image-based models fail to handle effectively. To improve the performance of our FER systems, several measures in future will be adopted. Such as; using more diverse and inclusive datasets that capture a wide range of demographic and contextual variations, utilizing transfer learning/transformer-based architectures with pre-trained models to leverage knowledge from larger, more diverse datasets, optimizimg the model architecture to balance complexity and generalizability, avoiding overfitting while ensuring robustness.

III. CONCLUSION

In today's chat applications, integrating image-sharing functionalities has become an indispensable feature, enriching conversations with visual context and enhancing user engagement. This paper advances chatbot capabilities by incorporating seamless image-sharing functionalities, leveraging facial emotion recognition (FER) technology to analyze user-submitted images in real-time. By discerning users' emotional states through images, the chatbot responds empathetically with tailored feedback, such as motivational support or attentive listening, making conversations more personalized and emotionally aware. However, despite these advancements, there remain critical limitations and opportunities for improvement.

One of the key limitations is the reliance on the FER-2013 dataset, which lacks demographic and contextual diversity. This limitation contributes to the model's relatively low accuracy (~60%), as the dataset fails to account for variations in lighting, cultural expression, and complex emotional overlaps. Future work will involve experimenting with larger, more diverse datasets that encompass varied demographics, facial expressions, and environmental conditions. These improvements would not only enhance model robustness but also mitigate biases inherent in the current dataset. Furthermore, technical novelty could be introduced by adopting advanced machine learning techniques such as transfer learning, transformer-based architectures, and hybrid approaches that combine convolutional neural networks (CNNs) with attention mechanisms to better capture subtle emotional nuances. Data augmentation strategies, including the use of generative adversarial networks (GANs), could further enlarge and diversify the training dataset, leading to higher generalization capabilities.

Another limitation is the model's inability to handle multiple emotions from group images effectively. The current system processes only a single emotion label, which is then passed to

the LangChain model for generating responses. This simplistic approach limits the chatbot's applicability in dynamic, group-based scenarios. Future implementations will address this by enabling multi-emotion detection, ensuring that each individual's emotional state is recognized and appropriately processed. This would make the chatbot more adept at handling complex interpersonal interactions.

In addition, to make interactions more human-like, future iterations of the system will integrate multimodal inputs, including voice, video, and text, allowing the chatbot to leverage a combination of cues for a deeper understanding of user emotions. This multimodal approach will further bridge the gap between human and machine interaction, enabling the chatbot to provide richer and more context-aware responses.

Despite its contributions, the technical novelty of the current system is limited. The conclusions drawn are not fully supported by the results, as the performance metrics leave significant room for improvement. To address this, the research will focus on systematically benchmarking the proposed system against state-of-the-art solutions, enabling a clearer comparison of its effectiveness. This will include implementing and evaluating transformer-based architectures like Vision Transformers (ViT) and fine-tuning pre-trained models to boost performance. Additionally, the chatbot's impact has not been thoroughly demonstrated. Future research will incorporate user studies to evaluate its real-world applicability and effectiveness in enhancing engagement and emotional well-being.

In conclusion, while the integration of FER technology into chatbots marks a significant step forward in creating empathetic and responsive AI systems, substantial challenges remain. By addressing the limitations of datasets, expanding to multimodal inputs, improving emotion recognition through advanced machine learning techniques, and incorporating multi-emotion handling, future research aims to create a system that is not only more accurate but also more human-like and context-aware. The development of emotionally intelligent chatbots promises to transform human-computer interaction, making it more engaging, inclusive, and impactful in diverse applications such as mental health, education, and customer service.

REFERENCES

[1] M. Y. Lee, "Building multimodal ai chatbots," arXiv preprint arXiv:2305.03512, 2023.

[2] J. Martınez-Miranda and A. Aldea, "Emotions in human and artificial intelligence," Computers in Human Behavior, vol. 21, no. 2, pp. 323–341, 2005.

[3] J. Li, Z. Zhang, H. Zhao, X. Zhou, and X. Zhou, "Task-specific objectives of pre-trained language models for dialogue adaptation," arXiv preprint arXiv:2009.04984, 2020.

[4] W. H. Dutton, G. Blank, and D. Groselj, Cultures of the internet: the internet in Britain: Oxford Internet Survey 2013 Report. Oxford Internet Institute, 2013.

[5] N. Ma and R. Khynevych, "Modern trends in intelligent chatbot digital visual image," Art and Design, no. 3, pp. 35–44, 2023.

[6] K. Lobinger, "Photographs as things–photographs of things. a textomaterial perspective on photo-sharing practices," Information, Communication & Society, vol. 19, no. 4, pp. 475–488, 2016.

[7] E. M. Johnson and E. W. Healy, "The optimal speech-to-background ratio for balancing speech recognition with environmental sound recognition," Ear and Hearing, pp. 10–1097, 2024.

[8] G. A. Godghase, R. Agrawal, T. Obili, and M. Stamp, "Distinguishing chatbot from human," arXiv preprint arXiv:2408.0464, 2024.

[9] K. Chowdhary and K. Chowdhary, "Natural language processing," Fundamentals of artificial intelligence, pp. 603–649, 2020.

[10] M. K. Dobbala and M. S. S. Lingolu, "Conversational ai and chatbots: Enhancing user experience on websites," American Journal of Computer
Science and Technology, vol. 11, no. 1, pp. 62–70, 2024.

[11] S. Pophale, H. Gandhi, and A. K. Gupta, "Emotion recognition using chatbot system," in Proceedings of International Conference on Recent Trends in Machine Learning, IoT, Smart Cities and Applications:
ICMISC 2020. Springer, 2021, pp. 579–587.

[12] Q. Sun, Y. Wang, C. Xu, K. Zheng, Y. Yang, H. Hu, F. Xu, J. Zhang, X. Geng, and D. Jiang, "Multimodal dialogue response generation," arXiv preprint arXiv:2110.08515, 2021.

[13] J. D. M.-W. C. Kenton and L. K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in Proceedings of naacL-HLT, vol. 1, no. 2. Minneapolis, Minnesota, 2019.

[14] L. Dong, N. Yang, W. Wang, F. Wei, X. Liu, Y. Wang, J. Gao, M. Zhou, and H.-W. Hon, "Unified language model pre-training for natural language understanding and generation," Advances in neural information processing systems, vol. 32, 2019.

[15] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell et al., "Language models are few-shot learners," Advances in neural information processing systems, vol. 33, pp. 1877–1901, 2020.

[16] T. Majumder, S. SR et al., "Machine learning based method to design a facial emotion detection and chatbot system." Journal of Advanced Zoology, vol. 44, 2023.

[17] S. Assayed, K. Shaalan, S. Al-Sayed, and M. Alkhatib, "Psychological emotion recognition of students using machine learning based chatbot," International Journal of Artificial Intelligence and Applications (IJAIA), vol. 14, no. 2, 2023.

[18] M. Karna, D. S. Juliet, and R. C. Joy, "Deep learning based text emotion recognition for chatbot applications," in 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184). IEEE, 2020, pp. 988–993.

[19] L. Zahara, P. Musa, E. P. Wibowo, I. Karim, and S. B. Musa, "The facial emotion recognition (fer-2013) dataset for prediction system of micro-expressions face using the convolutional neural network (cnn) algorithm based raspberry pi," in 2020 Fifth international conference on informatics and computing (ICIC). IEEE, 2020, pp. 1–9.

[20] S. Eswar Sudhan, V. K. R. Garapati, M. R. G. Nadella, G. Jangala, and P. Aswathy, "Enhancing chatbot interaction with emotion detection for improved understanding: Emobot," in Congress on Intelligent Systems. Springer, 2023, pp. 131–140.

Faces Unveiled: A Deep Dive into Modern Face Detection and Recognition Techniques

DAA Deepal[1], MKA Ariyaratne[2], PR De Silva[2] and TGI Fernando[2]

[1]Faculty of Graduate Studies, University of Sri Jayewardenepura.
[2] Department of Computer Science, Faculty of Applied Sciences, University of Sri Jayewardenepura.
Email: mkanuradha@sjp.ac.lk

ABSTRACT

This paper provides a comprehensive overview of contemporary research in face detection, facial feature detection, and face recognition, categorizing methodologies into four primary types: knowledge-based, template matching, feature-based, and appearance-based. Analysis reveals a predominant focus on appearance-based techniques, particularly in recent studies. Literature showcases the increasing utilization of deep learning algorithms, such as CNN, DCNN, and Faster RCNN, to address challenges in face detection and recognition. Notably, these algorithms demonstrate high accuracy in complex scenarios, including variations in pose, scale, and occlusion. The overview highlights the effectiveness of knowledge-based methods in detecting facial features with low computational requirements, albeit with limited accuracy in complex situations. Appearance-based methods, particularly those employing deep learning, emerge as highly successful in face detection and recognition, achieving accuracy rates exceeding 99%. The integration of one-stage and two-stage algorithms, coupled with traditional classifiers, underscores their efficacy. Researchers enhance accuracy through data augmentation, multi-task learning, and network acceleration techniques. The paper concludes that deep learning algorithms significantly impact face detection, recognition, and feature extraction, reflecting their pivotal role in advancing computer vision. The comprehensive review of 28 selected papers emphasizes the importance of continued research to further enhance these essential aspects of object detection.

INDEX TERMS : Face Detection, Facial Feature Detection, Deep Learning, Review

I. INTRODUCTION

Detecting and classifying objects in digital images and videos is a critical task in computer vision, with face detection and recognition being among its most significant applications. Object detection techniques have evolved over the past two decades, transitioning from rule-based methods [1], feature-based methods [2], and region-based methods [3] to modern deep learning algorithms after 2020. While these advancements have significantly improved accuracy and computational efficiency, several **challenges** persist, particularly in handling diverse environmental conditions, real-time processing, and scalability across datasets.

Despite the success of deep learning methods such as Convolutional Neural Networks (CNNs), Single Shot Detector (SSD) [14], Faster R-CNN [15], and YOLO [16], gaps remain in comprehensively comparing these techniques, especially in their suitability for specific tasks like real-time face recognition or handling occlusions. Moreover, there is a lack of systematic reviews that delve into the underlying methods, datasets, and metrics in a way that bridges the gap between theoretical advancements and practical applications.

Problem Statement:

Traditional face detection methods often struggle with low accuracy in challenging scenarios, such as poor lighting, extreme poses, or occlusions. Although deep learning-based techniques have addressed many of these issues, there is a need for a **comprehensive review** that categorizes and evaluates these methods based on key factors such as:

- Robustness to environmental changes.
- Dataset requirements and performance evaluation.
- Computational efficiency and real-time feasibility.

Objectives:

The primary objective of this survey is to provide a systematic and comprehensive review of face detection and recognition techniques, with a focus on:

1. Categorizing the techniques into knowledge-based, feature-based, template-matching, and appearance-based methods.
2. Evaluating the performance of state-of-the-art algorithms, such as CNNs, Faster R-CNN, YOLO, and SSD, across various tasks (e.g., front face detection, feature point detection).

3. Discussing the datasets and evaluation metrics used for training and testing these algorithms.
4. Highlighting the strengths, limitations, and suitability of each algorithm for specific applications.

Rationale:
While several surveys have been conducted on object detection [22–25] and face recognition [26–27], they often lack a nuanced comparison of emerging algorithms and their applications. This survey aims to address these gaps by:

- Examining the evolution of face detection and recognition techniques, from traditional methods to deep learning-based approaches.
- Providing practical insights into dataset selection, algorithm suitability, and performance metrics.
- Offering a resource for researchers and practitioners to navigate the complexities of modern face recognition systems.

The remainder of this paper is structured as follows: Section 2 outlines the methodology used to conduct this systematic review. Section 3 discusses the applications, datasets, and evaluation metrics in face detection. Section 4 categorizes object detection and recognition techniques based on their underlying methods. Section 5 explores the categories of algorithms specifically used for face detection and recognition. Finally, Section 6 concludes with a summary and discussion.

II. MATERIALS AND METHODS FOR THE SYSTEMATIC REVIEW

To conduct this systematic review, we adhered to the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) guidelines to ensure transparency and reproducibility of the research process [28].

Research Objectives and Scope

The primary objective of this review was to comprehensively analyze advancements in **face detection** and **recognition technologies**, identify **challenges** in the field, and assess **state-of-the-art techniques**. Key focus areas included:

- The datasets used for training and testing algorithms.
- The reported accuracy of algorithms.
- Drawbacks and challenges associated with various techniques.
- Key methods employed to improve algorithm accuracy.

Search Strategy and Data Sources

Our systematic approach began with the identification of relevant research articles using a predefined set of keywords, including "face detection," "face recognition," "knowledge-based face detection," "feature-based face detection," "template matching," "appearance-based methods," "convolutional neural networks," "Faster R-CNN," "YOLO," and "deep learning-based approaches." To maximize the scope of our review, we searched popular repositories such as **Google Scholar**, **Scopus**, and **Semantic Scholar** for literature published between **1990 and September 2023**.

Inclusion and Exclusion Criteria

We adopted the following criteria to systematically include and exclude studies:

- **Inclusion Criteria**:
    - Peer-reviewed research articles, review articles, book chapters, and conference proceedings written in English.
    - Studies addressing face detection, recognition, facial feature extraction, or algorithms trained/tested on relevant datasets.
    - Publications discussing single or multiple face scenarios in real-time or static images.
- **Exclusion Criteria**:
    - Non-English studies without available translations.
    - Duplicates or redundant studies identified during the screening process.

- Articles unrelated to face detection or recognition, such as studies focusing solely on non-facial object detection.

Screening and Selection Process

Through an extensive web search, **37 articles** were initially identified. After screening for duplicates and applying the inclusion and exclusion criteria, **31 articles** were selected for detailed review. An additional **6 articles** were identified through references within the selected papers. Ultimately, **2 articles** were excluded due to redundancy, lack of translation, or irrelevance. This process is summarized in **Figure 2**.

Data Extraction and Analysis

Each selected article was carefully analyzed and categorized into four primary methods:

1. **Knowledge-based methods**.
2. **Template matching methods**.
3. **Feature-based methods**.
4. **Appearance-based methods**.

The analysis focused on:

- The datasets used to train and test algorithms.
- The reported accuracy and performance metrics.
- Challenges and limitations of the proposed approaches.
- Strategies for improving algorithm accuracy.

This systematic categorization and analysis provide insights into the evolution and trends within the field, highlighting advancements, challenges, and solutions that shape the landscape of facial recognition technology.

, further summarizes the selection of the articles.

Table 1 Summary of the search process

Duration of the search	Used research repositories	Keywords	Type of research work
March 2022 to September 2023	Google Scholar, Scopus, Semantic Scholar	face detection face recognition Knowledge-based face detection and recognition Feature-based face detection and recognition Template Matching face detection and recognition Appearance-based face detection and recognition Convolutional Neural Networks Faster R-CNN YOLO Deep learning-based approaches for face detection and recognition	research articles, review articles, book chapters, conference materials

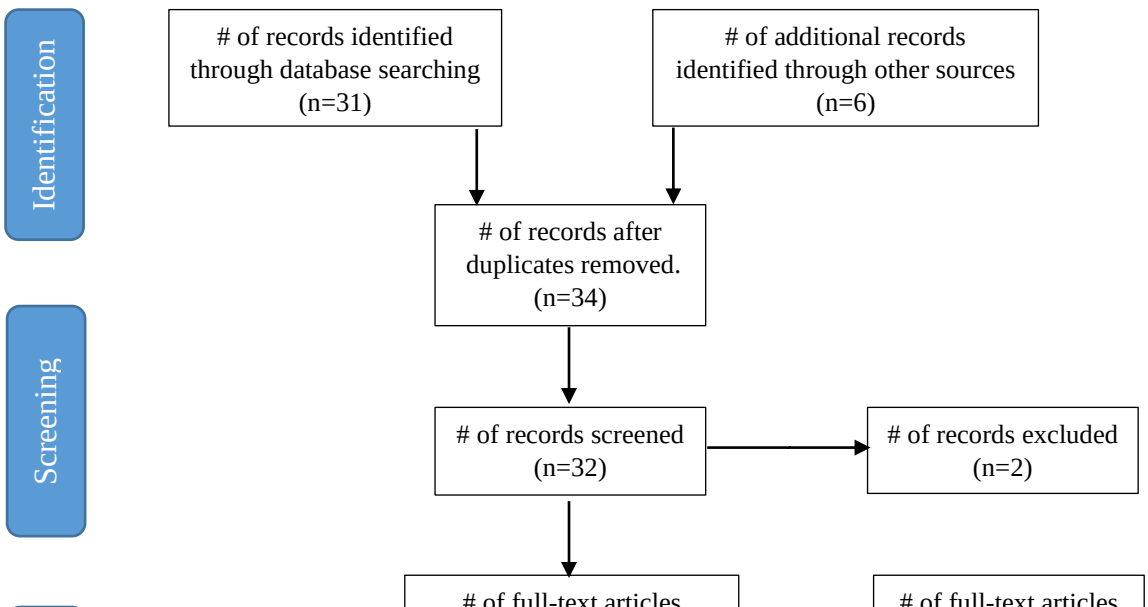

Figure 1 Flow chart of the comprehensive review process based on the PRISMA.

III. APPLICATIONS, RESOURCES, AND MEASURING METHODS FOR FACE DETECTION USING MACHINE LEARNING ALGORITHMS

Before a discussion of the previous studies, this section aims to provide a detailed description of the applications where machine learning-based face detection has been used and the importance, available resources used in research literature for such research, and, what kind of approaches have been taken to measure the quality of the algorithms.

1) Applications

Over the years, face detection and recognition technologies have been used in several applications across various industries. It is commonly used in security and surveillance systems for authentication and tracking purposes [29], [30], [31] [32]. Biometric identification also uses face detection to unlock devices and grant access to secure systems and attendance systems [33] [34] [35]. Social media platforms and photo-editing software also use this technology to identify faces in images, tag individuals, and apply filters and effects [36] [37] [38]. Gaming applications use face detection to capture facial expressions and movements of players for controlling characters or game elements [39] [40]. Marketing and advertising industries use this technology to target ads based on facial expressions and emotional responses [41] [42]. Additionally, face detection is used in medical diagnosis to identify and track facial features, such as skin lesions, and to analyze changes over time [43] [44].

Face detection technology can also be used for gender classification and age estimation [45] [46]. This technology can accurately detect facial features to determine a person's gender and estimate their age. Face detection and recognition are important in the field of social robotics for effective human-robot interaction [47] [48]. Gender estimation and facial expression analysis also play important roles in attracting user attention and providing personalized services, such as adjusting lighting and temperature based on an individual's facial features [49]. The large application domain further proves the importance of face detection and the need for a good survey encapsulating most of such recent advancements.

2) Available Datasets/ Resources /Programs/Software/etc.

At present, face detection applications can be implemented either using pre-trained models or by training customized models. There are several resources available to train new models, including face detection datasets like the WIDER FACE and CelebA datasets, as well as open-source libraries such as OpenCV and TensorFlow. This section briefly discusses the available resources, datasets, software, etc.

2.1 Datasets

2.1.1 DigiFace-1M

The DigiFace-1M dataset, introduced by Bae et al. [50] in 2023, contains over one million diverse synthetic face images for face recognition. It includes 720,000 images with 10,000 unique identities, each with 72 images generated using four sets of accessories. Additionally, it has 500,000 images with 100,000 unique identities, each with five images generated using one set of accessories. The dataset outperforms SynFace, a leading method for synthetic face recognition [50], demonstrating its robustness across various datasets. Fine-tuning with a smaller set of real-face images further enhances model accuracy.

2.1.2 FDDB: Face detection data set and benchmark

The FDDB dataset, introduced by Jain and Learned-Miller [51]at the University of Massachusetts Amherst, is a

collection of 2,845 labeled images featuring 5,171 faces for face detection. The images include varying resolutions, orientations, lighting conditions, out-of-focus faces, difficult poses, and low-resolution images. The FDDB is widely used in research, including by Zhang et al. [52], who trained and evaluated their MTCNN framework on this dataset, achieving superior accuracy in face detection and alignment. Sun X et al. [53] and Jiang H et al. [54] also utilized the FDDB dataset to test and fine-tune their respective Faster R-CNN and enhanced Faster RCNN models for face detection.

2.1.3 WIDER FACE

The WIDER FACE dataset, presented by Yang et al. [55] in 2016, includes 32,203 images and 393,703 labeled faces, selected to represent a diverse range of scales, poses, and occlusions. Wang J et al. [56] trained and evaluated the Face Attention Network (FAN) on this dataset, achieving an average precision (AP) of 94.6% (easy) and 88.5% (hard). Additionally, Sun X et al. [53] trained their Enhanced Faster RCNN approach on the WIDER FACE dataset, demonstrating its adaptability and effectiveness by generating hard negatives during testing.

2.1.4 CelebFaces Attributes Dataset (CelebA)

The CelebA dataset, introduced by Liu et al. [57] in 2015, is a large-scale face attributes dataset comprising over 200,000 images of famous persons. Each image contains 5 landmark locations and 40 binary attribute annotations, covering a wide range of pose variations and background clutter. With over 10,000 unique identities and 202,599 face images, CelebA is suitable for tasks such as face detection, recognition, landmark localization, and attribute recognition. Ranjan et al. [58] introduced the Deep Pyramid Single Shot Face Detector (DPSSD), a high-speed model effective for detecting faces with significant scale variations, including tiny faces. Their deep learning pipeline showed exceptional performance in face identification and verification across various benchmarks, including CelebA. Yang et al. [59] proposed a deep convolutional neural network (CNN) for face detection with facial attributes-based supervision. They enhanced their attribute-aware networks using the CelebA dataset for training, fine-tuning the model with a substantial number of face and non-face images from CelebA.

2.1.5 VGG Face2

The Visual Geometry Group introduced the VGG Face2 dataset [60] in 2018 at the University of Oxford. It is one of the largest datasets for face recognition, containing over 3.3 million face images and 9,131 unique identities. Cao et al. [60] trained ResNet-50 Convolutional Neural Networks, both with and without Squeeze-and-Excitation blocks, on VGGFace2, demonstrating that training on VGGFace2 significantly enhances recognition performance, especially with pose and age variations. Aghdam et al. [61] investigated factors affecting the identification performance of deep face recognition models under low-resolution and mismatched conditions, using models trained on MS-Celeb-1M and fine-tuned on VGGFace2. This approach achieved state-of-the-art accuracies on the SCFace and ICB-RW benchmarks, highlighting VGGFace2's effectiveness in addressing challenges related to appearance variety and low-resolution face recognition.

2.1.6 Flickr-Faces-HQ Dataset (FFHQ)

The Flickr-Faces-HQ Dataset (FFHQ), created by researchers at NVIDIA [62], contains 70,000 high-resolution human face images sourced from Flickr. It offers a varied set of images in terms of poses, ages, and ethnicities, making it ideal for training and evaluating face-related computer vision models. Karras et al. [63] redesigned the generator architecture for generative adversarial networks (GANs) and conducted extensive experiments using the FFHQ dataset. Their method significantly improved control over image synthesis, achieving state-of-the-art results in high-quality image generation. Bencheriet et al. [64] introduced a robust approach to fake face detection using a Deep CNN architecture with three distinct discriminators, each trained differently to enhance performance. They trained and evaluated their system on a dataset combining authentic faces from the FFHQ dataset and 70,000 synthetic faces generated with Nvidia's StyleGAN. Their system achieved impressive accuracy rates of 96% for detecting fake faces and 98% for identifying real faces.

2.1.7 Tufts Face Dataset

The Tufts Face Dataset [65] is a unique and extensive dataset offering seven distinct image modalities for face recognition: thermal, near-infrared, visible, LYTRO, computerized sketch, 3D images, and recorded video. It includes 10,000 images of 112 individuals (38 males and 74 females) from over 15 countries, aged 4 to 70 years. This dataset is valuable for benchmarking and evaluating algorithms across multiple modalities, such as thermal, sketches, heterogamous face recognition, and 3D face recognition. Martins et al. [66] used the Tufts dataset to develop a multi-spectral face recognition system. They tested three SSD-based methods: the S3FD algorithm, the facial detection deep neural network of OpenCV, and the DSFD algorithm. The system achieved impressive Rank-1 scores of 99.5% for pose variations and 99.6% for expression variations in the Tufts database.

2.1.8 Labeled Faces in the Wild (LFW) Dataset

The LFW Dataset [67] is a vital resource for researchers in unconstrained face recognition, specifically designed for studying face verification or "pair matching." It contains over 13,000 face images gathered from the Internet, featuring diverse faces with varying sizes, poses, and illumination conditions, and has a total size of 173 megabytes. Almabdy et al. [68] conducted a comprehensive study on face recognition, evaluating the performance of three CNN-based methods across various image databases, including LFW. Sun et al. [69] proposed a hybrid model for face verification in unconstrained conditions, combining convolutional networks (ConvNets) with Restricted Boltzmann Machines (RBMs), and tested their method on the LFW dataset. Additionally, Sanchez et al. [70] presented an efficient facial recognition

system for unconstrained environments, achieving an impressive accuracy of 99.7% on the LFW dataset.

2.1.9 UTKFace

The UTKFace dataset [71] is a versatile resource for various computer vision tasks related to faces, such as age estimation, gender classification, and facial landmark localization. It comprises over 20,000 face images spanning diverse ages, ethnicities, and variations in pose, illumination, and expression. Additionally, the dataset provides aligned and cropped faces along with landmark annotations for 68 points, making it suitable for training and evaluating machine learning models in facial recognition research. Nugroho et al. [72] conducted a significant study analyzing the impact of different color spaces on face detection accuracy and efficiency. They validated their method using the UTKFace dataset. Devaraj et al. [73] focused on effectively classifying individuals by ethnicity, employing datasets including UTKFace for training their Convolutional Neural Network (CNN) model. Their methodology, involving image preprocessing and CNN utilization, achieved an average accuracy of 88% upon rigorous evaluation.

2.1.10 Google Facial Expression Comparison Dataset

Introduced by Vemulapalli and Agarwala in 2018, the Google Facial Expression Comparison (FEC) dataset [74] contains approximately 87,517 unique photos with around 500K facial image triplets. Each triplet is annotated by humans to identify the two faces most similar in terms of facial expression. Vemulapalli et al. [75] proposed a novel approach for efficiently capturing similarities in facial expressions, benefiting applications such as expression retrieval, photo album summarization, and emotion recognition. Their trained network achieves an impressive 81.8% accuracy in predicting the most similar pair within a triplet from the FEC dataset.

2.1.11 YouTube Faces Dataset with Facial Keypoints

The YouTube Faces Dataset [76] comprises short videos of celebrities sourced from YouTube, with faces cropped and multiple frames extracted to create a collection of face images. This processed version also includes facial keypoint annotations, facilitating detailed analysis of facial expressions and movements. Almabdy et al. [68] conducted a thorough study on face recognition using three distinct CNN-based methods, evaluating their performance on various image databases, including the YouTube Faces dataset. FaceNet, introduced by Schroff et al. [77], capable of face verification, recognition, and clustering using deep convolutional networks, achieved a high accuracy of 95.12% on the YouTube Faces DB, among other datasets.

2.2 Other datasets

In the literature, we have identified a series of face databases that have been used to evaluate the performance of models. The Yale Face Database [78] contains grayscale images capturing individuals under different lighting conditions, pioneering research in illumination-invariant face recognition. The European ACTS M2VTS [79] dataset offers video sequences depicting individuals in different poses and lighting conditions. The Cohn-Kanade database [80] provides facial expression sequences of varying intensities. The AR face database [81] features frontal face images under diverse illumination and expressions. For large-scale research, the PubFig face verification datasets [69] offer celebrity face images, while the CUFS/CUFSF [5] datasets capture student faces with pose, illumination, and expression variations. The CASIA NIR-VIS 2.0 [5] dataset aids in cross-spectrum face recognition research by providing near-infrared and visible-light images of the same individuals. GTAV Face [68] offers real-world complexity by extracting faces from Grand Theft Auto V. AFW [82] focuses on facial attribute analysis, providing gender, age, and facial hair annotations.

Other datasets address specific challenges like pose and illumination invariance (MIT-CBCL, CMU PIE [83]), aging effects (CAS-PEAL-R1 [82]), and real-world complexities such as low resolution and occlusion (VOC2012 [82], IJB-A [54]). MAFA [56] and MALF [58] explore subtle facial movements via facial action coding, showcasing the diverse research enabled by these resources. These datasets, each with unique strengths and limitations, emphasize the multifaceted nature of facial analysis research, offering rich opportunities for exploration and refinement in this dynamic field.

2.3 Resources

2.3.1 TensorFlow.js

TensorFlow.js, a JavaScript library developed by Google [84], brings machine learning models directly to the browser or Node.js environment. It offers various pre-trained models for tasks like image classification, object detection, semantic segmentation, face detection and recognition, face landmarks detection, pose detection, body segmentation, hand pose detection, natural language processing, and speech recognition. Moreover, developers can fine-tune existing models with custom datasets and build/train models directly in JavaScript using flexible APIs. Face-api.js [85], built on TensorFlow.js, provides a comprehensive JavaScript module for face-related tasks like detection, recognition, landmark detection, expression recognition, age estimation, and gender recognition. It enables seamless integration of advanced facial recognition functionalities into web applications and Node.js projects. Additionally, the @tensorflow-models/blazeface [86] repository enhances this ecosystem with pre-trained models tailored for TensorFlow.js, easily accessible via NPM or unpkg. These models expand developers' capabilities, allowing effortless integration of state-of-the-art facial detection and recognition mechanisms.

2.3.2 OpenVINO

OpenVINO, an open-source toolkit developed by Intel [87], optimizes deep learning models to run efficiently on Intel CPUs, GPUs, and accelerators like FPGAs. It offers libraries and tools for developing, optimizing, and deploying models for tasks like image classification, object detection, face recognition, and segmentation. The Open Model Zoo provides free, pre-trained models and demo applications usable with Python, C++, or OpenCV Graph API (G-API). Wang and Hu

[88] utilize OpenVINO to optimize CNN inference speed in their intelligent lecture recording system, enhancing face detection performance, especially with MobileNet-SSD. This optimization enables efficient lecturer tracking even during rapid movements. Dane Brown's [89] study explores mobile attendance systems using face detection and recognition, powered by OpenVINO on a Raspberry Pi platform. Despite positioning constraints, the system achieves remarkable recognition accuracy and processing speed, demonstrating OpenVINO's versatility and efficacy in real-world applications.

2.3.3 Face-api.js

Face-api.js [85] is a JavaScript module for face detection, recognition, landmark detection, expression recognition, age estimation, and gender recognition in the browser and Node.js. It leverages TensorFlow.js and provides high accuracy and speed using pre-trained machine-learning models. The library supports various input sources such as image files, video streams, and webcams, making it versatile for different use cases. Being open-source and actively maintained, it has a growing community of contributors and users continuously improving its capabilities. Basurah et al. [90] highlight the importance of liveness detection in thwarting spoofing attempts in facial recognition systems. Using TensorFlow.js and face-api.js, they implement a method for detecting facial movements, achieving an impressive 85% accuracy for face recognition. Yadav et al. [91] explore an In-Browser Attendance System showcasing the versatility of face-api.js in real-world applications. Powered by serverless edge computing, their system seamlessly integrates face detection and recognition functionalities into web browsers, enhancing efficiency and eliminating backend latency issues.

2.3.4 OpenCV

OpenCV (Open Source Computer Vision Library) [92] is a widely used open-source software library for computer vision and machine learning. It offers a broad range of algorithms and tools for image and video processing, including filtering, feature detection, object detection and recognition, segmentation, and camera calibration. With support for various programming languages like C++, Python, Java, and MATLAB, OpenCV is accessible to a wide range of developers and researchers. Its large and active community constantly contributes to its development and provides support for users. OpenCV finds applications in robotics, surveillance, augmented reality, and medical imaging.

OpenCV has been instrumental in various research endeavors across diverse domains [93]. Hoque et al. [94] developed an Autonomous Face Detection System for real-time security intelligence. Similarly, during the COVID-19 pandemic, Das et al. [95] created a Face Mask Detection system using TensorFlow, Keras, and OpenCV, showcasing OpenCV's versatility in addressing contemporary challenges. In education, Mehariya et al. [96] used OpenCV to develop a method for Counting Students in classrooms, integrated with Firebase for efficient classroom allocation. Moreover, Soomro et al. [97] applied OpenCV in a real-time Electronic Voting System, where face recognition ensures secure and transparent electoral processes. These projects highlight OpenCV's adaptability and significance in fostering innovation across security, healthcare, education, and governance domains.

3) Tutorials

Tutorials serve as invaluable resources, especially for beginners, to grasp the fundamentals of face detection and related concepts. El Bruno's tutorial [98] provides a comprehensive walkthrough of face detection using deep neural networks in a Windows environment based on .NET and OpenCV. Utilizing the res10_300x300_ssd_iter_140000_fp16.caffemodel and interfacing with the camera feed via OpenCV, the tutorial demonstrates the real-time processing of video frames. Written in C# within a Windows Forms application, El Bruno's tutorial offers a reliable source for face detection using C#. In Akhtar Jamil's tutorial [99], viewers are guided through facial landmarks detection using Emgu CV 4.4 and C# within a Windows Forms application. The tutorial covers loading pre-trained models into the Emgu CV framework, essential for accurate facial feature identification. Practical implementation includes detecting faces in images or video streams and marking critical facial landmarks like eyes, nose, and mouth. Emgu CV 4.4, a .NET wrapper for OpenCV, ensures seamless integration of computer vision capabilities into C#. By utilizing a Windows Forms application, the tutorial enhances user interaction for intuitive exploration of detected facial landmarks. This comprehensive guide caters to beginners and enthusiasts, offering insights into leveraging Emgu CV for effective facial landmarks detection and visualization.

4) Measurement Methods in Model Evaluation

Accuracy is the basic measure of correctness for machine learning models that calculates the proportion of correct predictions out of all predictions made. However, when dealing with imbalanced datasets, accuracy alone can be misleading. In such cases, other measures such as **Precision**, **Recall**, and **F1-score** should be used. From the literature, we have found a series of measurements that have been used along with the accuracy to evaluate the performance and correctness of models.

- Precision [100] [46] [58] [82]: Examines the accuracy of positive predictions, providing insights into the proportion of correctly identified faces among the total predicted positive cases.
- Average Precision [5], [101] [56]: Measures the area under the precision-recall curve, providing a consolidated evaluation of a model's precision at various recall levels and offering a comprehensive assessment of its performance in ranking and classification tasks

- Mean Average Precision (mAP) [100] [56] [102]: Evaluates the precision-recall curve across multiple thresholds, providing a comprehensive measure of model performance.
- Recall (Sensitivity) [100] [46] [58] [82] [101]: Evaluates the model's ability to capture all relevant instances, focusing on the ratio of correctly identified faces to the total actual positive cases.
- F1 Score [102] [103]: Represents the harmonic mean of precision and recall, offering a balanced metric that considers both false positives and false negatives.
- Receiver Operating Characteristic (ROC) Curve [82] [53] [54]: illustrates the trade-off between sensitivity (true positive rate) and specificity (true negative rate) at various thresholds and Area Under ROC Curve provides a quantitative measure of a model's ability to distinguish between positive and negative instances, summarizing the overall discriminatory performance.
- Intersection over Union (IoU) [5] [101] [104] [54]: Quantifies the overlap between the predicted bounding box and the ground truth bounding box, commonly used in object detection tasks.
- Confusion Matrix: Summarizes the model's performance by presenting the counts of true positives [58], true negatives, false positives [102], and false negatives [102].
- False Positive Rate (FPR) [58] [82] [54] and False Negative Rate (FNR): Express the proportion of incorrect positive and negative predictions, respectively.
- Top-1 accuracy, Top-k accuracy [105]: Top-1 accuracy is the measure used in classification problems, which calculates the percentage of cases where the model's top prediction matches the actual label. Top-k accuracy, on the other hand, allows for more flexibility in the model's predictions. Instead of requiring that the top choice exactly matches the actual label, top-k accuracy counts a prediction as correct if the actual label is among the top-k predictions made by the model.
- Accuracy [104] [68]: The ratio of correctly predicted instances (both true positives and true negatives) to the total number of instances in the dataset. It assesses the overall correctness of the model's predictions.

In this section, the focus is on exploring the applications, resources, and measuring methods related to face detection using machine learning algorithms. The diverse applications of face detection are highlighted. Next, the focus was on an extensive array of datasets, resources, programs, and software available for face detection applications. Furthermore, we explained essential resources like TensorFlow.js, OpenVINO, Face-api.js, and OpenCV, providing an overview of their capabilities and applications, and making it a comprehensive guide for researchers and practitioners. Finally, about the measurement methods employed in model evaluation for face detection. It emphasizes the importance of metrics beyond accuracy. The section equips readers with the knowledge to assess and understand the performance of face detection models effectively.

IV. DIFFERENT MACHINE LEARNING ALGORITHMS FOR FACE DETECTION AND FACE RECOGNITION

This section provides a comprehensive exploration of four key methodologies employed in this domain: Knowledge-Based, Feature-Based, Template Matching, and Appearance-Based methods. From the manipulation of human knowledge to the integration of sophisticated statistical analysis and deep learning, these approaches represent a spectrum of techniques designed to address the intricate challenges posed by varying lighting conditions, facial expressions, and complex scenes. As we examine deeper into each method, we uncover the nuances of their design, their strengths, and the continuous efforts to enhance the accuracy and robustness of these methods.

1) Knowledge-Based methods

Knowledge-based methods offer the advantage of simplicity and computational efficiency. These methods rely on a set of rules designed using human knowledge, such as the distance between facial features like eyes, nose, and mouth, to detect faces. However, the accuracy of these methods depends on the quality of the designed rules, and designing appropriate rules can be challenging. Too many or too detailed rules may reduce accuracy, and these methods may struggle with multiple face detections. Translating human knowledge into well-defined rules is also difficult, and if a face does not meet at least one rule, it may not be detected.

Despite their limitations, knowledge-based methods can provide accurate localization of facial landmarks in applications like facial landmark detection. However, they may struggle with variations in lighting, pose, and expression, and may not be suitable for complex scenes or images with multiple faces. To address these challenges, knowledge-based methods are often combined with other approaches, such as deep learning, to improve accuracy and robustness [17], [19], [20].

2) Feature-Based methods

The feature-based method utilizes a pre-trained classifier to distinguish facial and non-facial regions in an image, relying on structural features extracted from a face. Unlike the Knowledge-Based method, which relies on human-defined rules, this method extracts facial features like eyes, eyebrows, and mouth using edge detectors and statistical models. The presence of a face is then verified using the classifier [17], achieving a reported success rate of 94% even with images containing multiple faces [19], [20]. However, challenges such as illumination, noise, and occlusion can affect its

performance, causing blurring of feature boundaries and shadow interference [17].

Despite these challenges, the feature-based method remains widely used in face recognition systems due to its high accuracy and robustness. Researchers have proposed various feature extraction techniques to enhance its performance, including hybrid methods that combine multiple approaches. For example, integrating the feature-based method with graph-based techniques or pose normalization has shown promising results in handling occlusions and improving recognition rates under varying poses.

While the feature-based method is powerful, its effectiveness relies on the quality of extracted features and classifier robustness. Thus, ongoing research focuses on developing new techniques and algorithms to enhance its performance in challenging conditions.

3) Template Matching

Template matching is a popular technique for face detection that involves the use of predefined or parameterized face templates. The correlation between these templates and input images is used to locate or detect faces. One approach involves dividing a human face into different parts such as the eyes, face delineation, nose, and mouth. By using the edge detection method, a prototype face can be created. However, this approach is not sufficient for accurate face detection in complex situations.

To address this limitation, deformable templates have been proposed to deal with problems such as occlusion and variations in pose and expression. These templates allow the shape of the template to vary to better match the input image. Although template matching is relatively easy to implement, its effectiveness is limited by the quality of the templates used and the degree of variability in the target faces. Despite these challenges, template matching remains a valuable technique for face detection, and researchers continue to explore ways to improve its accuracy and robustness.

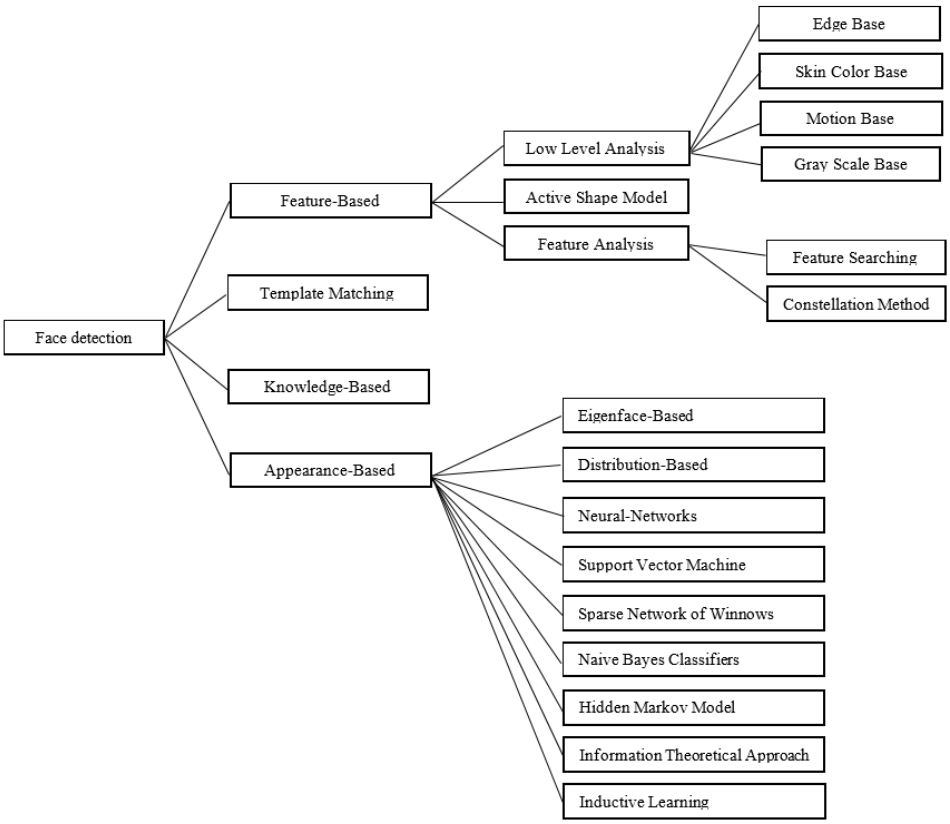

Figure 2 primary categories of face detection methods

4) Appearance-based methods

The appearance-based method uses statistical information in pixel values of facial images to extract features for face recognition [106]. It leverages machine learning and statistical techniques to identify facial characteristics, delivering superior performance. This method integrates various algorithms, including Support Vector Machines, Neural Networks, Eigenface-Based methods, Distribution-Based

methods, Hidden Markov Models, Naive Bayes Classifiers, Information Theoretical Approaches, and Inductive Learning [14],[16],[17]. These sub-methods offer diverse approaches for extracting and analyzing facial features, making the appearance-based method versatile and effective.

Convolutional neural networks (CNN), such as Multi-task Cascaded Convolutional Networks (MTCNN), have significantly advanced tasks like object and face detection and emotion recognition [5]. Rowley et al. [83] developed an early successful neural network-based face detection system in 1998, using a two-stage approach with coarse scanning and refinement stages. This method, evaluated on the MIT-CBCL [107] and CMU PIE [108], [109] datasets became a common paradigm in face detection systems.

Researchers have developed algorithms categorized into two types: two-stage and one-stage object detection algorithms [110]. Two-stage detectors, like Faster RCNN, RCNN, Fast RCNN, Mask R-CNN, SPPNet, Pyramid Networks, and G-RCNN, first identify regions of interest and then classify them. One-stage detectors, such as RetinaNet, YOLO, YOLOR, SSD, YOLOv3, and YOLOv4, directly predict bounding boxes without a separate region proposal step.

One-stage algorithms like YOLO [16] and SSD [14], use dense grids of bounding boxes with various aspect ratios and scales, making them computationally efficient and faster. However, they are less accurate in detecting small or complex-shaped objects and often suffer from false positives.

Two-stage algorithms, such as Faster R-CNN [15] and Mask R-CNN [111], use a region proposal network (RPN) to generate candidate object regions, which are then classified and refined by a separate network. These are more accurate and robust in detecting small and complex-shaped objects but are computationally more expensive and slower. Two-stage methods typically achieve the highest detection accuracy but are slower, while one-stage detectors are faster but struggle with irregularly shaped objects or small groups of objects [110], [112].

In summary, the exploration of face detection and recognition methodologies reveals a landscape rich with diversity and innovation. Knowledge-based methods, although simple, showcase computational efficiency, particularly in tasks such as facial landmark detection. Feature-based approaches, overcoming the limitations of rule-based methods, provide high accuracy in recognizing faces even in images with multiple subjects. Template Matching, utilizing predefined templates, remains a valuable technique, while Appearance-Based methods, integrating statistical information, emerge as powerful tools with superior performance.

V. FACE DETECTION FEATURE EXTRACTION AND FACE RECOGNITION ALGORITHMS

Face detection and recognition algorithms can be broadly categorized into knowledge-based, feature-based, template-matching, and appearance-based approaches. Knowledge-based methods rely on predefined rules and expert knowledge of facial anatomy to identify features. Feature-based algorithms extract specific facial components for representation and analysis. Template matching involves comparing a facial template with the target image for similarity. Appearance-based approaches consider overall facial appearance, utilizing statistical models or machine learning to capture holistic representations. These categories encompass diverse techniques, each with its strengths, ranging from rule-based systems to advanced machine learning methods, collectively contributing to the evolution of face detection and recognition technologies for a variety of applications. Here, we present literature relevant to each category.

1) knowledge-Based

Zhang et al. [78] developed a knowledge-based eye detection algorithm for human faces, combining image processing techniques with knowledge-based methods. The algorithm comprises two stages: first, locating the face and eye regions using histogram thresholding and smoothing procedures. Then, the eye region coordinates are identified, and the eye region is extracted. In the second stage, edge enhancement via the Laplacian operator and a knowledge-guided eye contour searching method are employed. This method leverages knowledge of the eye's shape and location to improve detection accuracy. Evaluation on the Yale Face Database using 320*243 face images and 200*102 eye region images demonstrated effective face and eye region localization, with successful eye contour extraction. The approach offers a balance of accuracy and computational efficiency, making it suitable for eye contour searching applications.

Yang G et al. [113] proposed a hierarchical knowledge-based approach for detecting human faces in complex backgrounds. The algorithm includes three levels of rules based on mosaic images of varying resolutions. Facial features are identified using an improved edge detection method at lower levels. At the highest level, a window scans the input image to identify potential face candidates, applying rules at each location. Higher-level rules define general facial appearance, while lower-level rules define specific features. The algorithm proceeds by dividing the image into small regions and processing each with an edge detector to detect edge pixels. These pixels are grouped into connected segments, which are then clustered into larger ones. These segments are analyzed to determine if they represent facial features, and if so, the algorithm combines them to form a complete face. Testing on images with complex backgrounds yielded promising results.

Kotropoulos et al. [79] introduced a rule-based method for frontal face detection using heuristic rules derived from domain knowledge. The algorithm begins with pre-processing steps like noise reduction and image segmentation, followed by applying rules to identify potential face regions. By averaging pixel intensities, the algorithm obtains horizontal and vertical profiles, detecting local minima to locate facial features such as hair, eyebrows, eyes, mouth, and chin. It

employs nostril, nose, eyebrow, eye, and mouth detection rules to validate these features. Using edge detection and the Hough transform, the algorithm identifies line segments corresponding to facial features, generating candidate facial feature combinations that meet geometric constraints. These candidates are assessed against heuristic rules to select potential face regions, verified by template-based classifiers. Evaluated on 90 frontal face images, the algorithm achieved a 97.77% detection rate with a 2.2% false positive rate. However, it was limited to detecting single faces against a uniform background from the European ACTS M2VTS dataset. Despite this, the method demonstrated robustness to occlusion, illumination, and expression variations. Primarily designed for frontal face detection, the method may not effectively detect non-frontal or profile views. Nevertheless, it showed high accuracy across different lighting conditions and outperformed existing methods at the time of publication.

2) Feature-Based

Chan et al. [114] present a face detection system using feature-based chrominance color information from a single indoor face image with a non-uniform background. The method employs the Modified Golden Ratio (MGR), Adapted Chain Code (ACC), and eye detection for accuracy. The algorithm starts with skin color segmentation to identify potential facial regions, then uses ACC to estimate the face boundary and predict eye candidates. To reduce computational complexity, estimated eye candidates and face boundary images are downsampled to 128x128 pixels using wavelet transform. The algorithm approximates eye positions, performs eye detection, and extracts key facial features such as eyes, brows, mouth, and nose based on the detected eyes and refined boundary. The method focuses on estimating eye candidates and analyzing chrominance in skin color segmentation. Chrominance, representing color purity and saturation, consists of Cr (red difference chroma) and Cb (blue difference chroma), capturing color information independent of brightness. Each pixel in a color image has YCbCr values containing luminance (Y) and chrominance (Cr and Cb). The method empirically determines Cr and Cb ranges from 16 skin regions for segmentation. For eye candidate estimation, an equation reduces the computational complexity of the eye map used by Hsu et al. [115]. To avoid misclassifying noise as eyes, the method introduces multilevel thresholding with 3-level priority. Evaluated 80 face images, including faces with and without spectacles and headscarves,

Table 2 Summary of the literature, sorted according to the Face detection– Knowledge-Based

Year	Authors	Detections	Key methods	Dataset	Accuracy and Advantages	Drawback
2000	Zhang et al. [78]	Locating both the face and eye regions	• Histogram thresholding • Histogram smoothing • Automatic thresholding • Laplacian operator • Knowledge-guided eye contour searching	Yale Face Database	Low computational cost	• Accurate for frontal-view face images with a plain background • Not suitable for complex images and multi faces images.
1997	Kotropoulos et al. [79]	Detect frontal faces	• Heuristic rules • Noise reduction • Image segmentation • Horizontal and vertical profiles • Eyebrow, eye, nostril, nose, and mouth detection rules • Edge detection • Hough transform • Template-based classifiers	90 frontal face images from the European ACTS M2VTS dataset	The overall detection rate of 97.77% with a false positive rate of 2.2%	• Limited to detecting only one face in a uniform background • Not suitable for detecting faces in profile or other non-frontal views
1994	Yang G et al. [113]	Locate human faces	• Improved edge detection method, • Edge detector, • Rule-based connected edge segments • Clustering algorithm	set of images with complex backgrounds	Accurate for complex background	• Accurate only for black-and-white pictures

the algorithm achieved 91.25% accuracy for detecting near-frontal faces, indicating its potential for practical applications.

Viola and Jones et al. [116] introduced the Viola-Jones algorithm for object detection using a boosted cascade of simple Haar-like features. The algorithm involves training a cascade of weak classifiers to efficiently classify image regions as containing or not containing an object of interest, using the AdaBoost machine learning algorithm to iteratively select the most informative features. Haar-like features, which are rectangular regions with varying intensities (e.g., edges or contrasts), represent the object of interest. The algorithm applies a sliding window technique, moving a fixed-size rectangular window across the image and computing Haar-like features at each location. AdaBoost evaluates these features, selecting the most informative ones to build a strong classifier for accurate object detection. The algorithm uses a cascade of classifiers, where each stage comprises multiple weak classifiers. Each stage quickly rejects regions that do not contain the object, reducing false positives and improving efficiency. The algorithm was trained and tested on a manually labeled dataset of face and non-face images, with 4,000 positive (face) and 8,000 negative (non-face) images in the training set, and 5,000 positive and 10,000 negative images in the testing set. The Viola-Jones algorithm employs Haar-like features and AdaBoost to train a cascade of weak classifiers for efficient object detection. By selecting informative features and using a classifier cascade, the algorithm achieves high accuracy and fast detection times. It demonstrated high accuracy on both training and testing datasets, making it a powerful tool for object detection widely used in various applications.

Fasel et al. [117] present a novel real-time eye detection method combining generative and discriminative models using the AdaBoost algorithm. They train a cascade of weak classifiers with features selected by AdaBoost, supplementing the training data with synthetic examples from a generative model, addressing limited training data, and enhancing discriminative model performance. A new feature representation for eye detection incorporates the spatial relationship between the eyes and other facial features, forming a feature vector used as input for the weak classifiers in the boosting cascade. This enables accurate, efficient eye tracking in real-time video streams. The system includes two types of eye detectors: one for general illumination and background conditions, and another for higher accuracy by leveraging contextual information. The algorithm first identifies potential face regions using a likelihood-ratio model trained on web images from Compaq Research Laboratories, detecting faces as small as 24x24 pixels. Efficiency is improved by using a sequence of smaller classifiers to evaluate wavelets and make early decisions, reducing unnecessary processing. Inspired by Viola and Jones [116], the system scales larger image patches to 24x24 pixels and applies the likelihood ratio model. It scans the entire image plane for patches with high likelihood ratios, identifying probable eye locations, and forwards these patches to a blink detection module for further analysis. Treating each frame independently, the system works for both static images and videos, encoding eye location and behavior for multiple faces appearing and disappearing randomly. By utilizing both discriminative and generative models, the algorithm achieves high accuracy and robustness in real-time eye coding applications. The approach is demonstrated to be effective on a dataset of video sequences, outperforming other state-of-the-art methods.

Vukadinovic et al. [80] introduced an automated technique to detect 20 facial feature points in expressionless face images using boosted classifiers based on Gabor features. Their approach improves on the original Viola-Jones face detector by Fasel et al. [117], offering a fast and reliable face detection process. The detected face region is divided into 20 regions of interest, where feature points are predicted using "GentleBoost templates" derived from gray-level intensities and Gabor wavelet features. The method employs an enhanced Viola-Jones face detection algorithm using GentleBoost instead of AdaBoost to detect the face region initially. It then refines feature selection with new filters, training on 5,000 faces and millions of non-face patches, achieving a 100% detection rate on 422 images. The approach automates the detection of the medial point of the mouth and irises to determine regions of interest, dividing the face into upper (eyes) and lower (mouth) sections. It uses horizontal and vertical histogram analysis to locate the irises and calculates the angle between the irises and the horizontal plane, rotating the image if needed. The algorithm identifies the medial point of the mouth within a defined region based on the distance between the irises (ED). It positions the region top at 0.85×ED and height at 0.65×ED, determining the vertical position by analyzing the mouth region's vertical histogram. This method achieves a 100% detection rate for both the irises and the medial point of the mouth. The algorithm demonstrated a 93% average recognition rate on the Cohn-Kanade database for facial expression analysis. It was trained on 5,000 faces and numerous non-face patches from about 8,000 web images and evaluated on 422 Cohn-Kanade images, achieving a 100% detection rate.

Cox et al. [118] presented a feature-based face recognition approach using mixture-distance. They modeled the training data as a combination of normal densities, projecting local second-order statistics onto it. They used 35 manually identified facial features from each face to create a 30-dimensional feature vector. Testing on a database of 685 individuals, they achieved a 95% recognition rate. Duplicate images from 95 individuals were used as queries to measure performance. The approach involved mixture-distance functions to measure distance, encountering two model selection challenges: determining the number of mixture elements and selecting between first and second-order statistics for each Gaussian component. They addressed these with a flat prior approach, yielding results comparable to the

best individual model. The experimental results confirmed the method's effectiveness, achieving the highest recognition rate among feature-based systems for a database of this size.

Manjunath et al. [119] proposed a face recognition approach divided into three stages: feature detection and localization, graph representation of the face, and recognition. Feature detection is based on the end-habilitation property model, using local scale interactions between oriented features. It involves extracting oriented feature information at different scales using the Gabor wavelet transform and then interacting these features across scales to achieve the end-habilitation effect. The method can be executed in parallel, making it suitable for real-time applications. It employs topological graphs to represent feature relationships and uses a deterministic graph marching schema to recognize faces from a database. This approach represents an early use of graph-based methods for face recognition, which has since become popular in the field.

3) Template Matching

Najat et al. [120] proposed a human face detection method in crowded images using template-matching techniques. The method starts by converting target and template images to grayscale. The template image is divided into a grid, and each cell is matched against the corresponding target image cell using Two-Dimensional Normalized Cross-Correlation (2D-NCC) to measure similarity and find maximum correlation. If the correlation between a template grid cell and the corresponding target cell exceeds a threshold, it is considered a match, and the location is recorded. The process involves comparing each template cell to face and non-face templates, repeating for all cells to identify matches, which are then combined for the final detection results. The algorithm can detect faces in low-resolution images with varying lighting conditions and expressions but may struggle with occlusion and rotation. This straightforward approach is promising for real-time applications.

Bose et al. [121] proposed a technique for detecting facial parts using the normalized cross-correlation (NCC) template matching method. They created a template database with images of different facial features (nose, eyes, mouth, and face), rotated at various angles to handle pose variations and improve the matching likelihood. To detect facial parts, the NCC algorithm calculates a correlation map between the input and template images at each position, identifying the best match by the highest correlation value. The method uses NCC values to detect the face and its parts. If the NCC value exceeds a threshold, the corresponding facial part is considered detected. The authors claimed their method can accurately detect facial parts in real-time with high accuracy but noted potential limitations due to facial expressions, occlusion, lighting variations, and rotation.

Chai et al. [81] presented a face detection method using a skin color model based on the r, g color space. The model was created using the equations: $r = R / (R + G + B)$ and $g = G / (R + G + B)$, where R, G, and B are the red, green, and blue color components of each pixel. Pixels representing skin color were selected if their color values, $v = (r, g)$, were above a specific threshold. The mean and covariance matrix of these selected pixels were used to create a Gaussian distribution model with the probability density function. After identifying the face region using the skin color model, the image was converted to grayscale, and a grayscale closing operation was applied. Morphological operations such as dilation and erosion were performed to refine the face region by filling gaps and removing small holes, smoothing the edges for easier feature extraction. For iris detection, the authors applied image processing techniques including illumination normalization and light spot deletion to enhance iris features. They reported a 93.06% accuracy rate for iris detection. The mouth region was detected using the location of the irises, applying a color space method, and the SUSAN [122] corner detector to refine the mouth corners and exact location, achieving a 95.83% accuracy rate. After feature extraction and refinement, template matching was used to classify the extracted features and recognize faces. Experiments on the AR face database, comprising 72 color images of 12 male and 12 female subjects with variations in pose, head orientation, facial expression, and other factors, yielded a recognition rate of 86.11%.

Bose et al. [121] proposed a technique for detecting facial parts using the normalized cross-correlation (NCC) template matching method. They created a template database with images of different facial features (nose, eyes, mouth, and face), rotated at various angles to handle pose variations and improve the matching likelihood. To detect facial parts, the NCC algorithm calculates a correlation map between the input and template images at each position, identifying the best match by the highest correlation value. The method uses NCC values to detect the face and its parts. If the NCC value exceeds a threshold, the corresponding facial part is considered detected. The authors claimed their method can accurately detect facial parts in real-time with high accuracy but noted potential limitations due to facial expressions, occlusion, lighting variations, and rotation.

Chai et al. [81] presented a face detection method using a skin color model based on the r, g color space. The model was created using the equations: $r = R / (R + G + B)$ and $g = G / (R + G + B)$, where R, G, and B are the red, green, and blue color components of each pixel. Pixels representing skin color were selected if their color values, $v = (r, g)$, were above a specific threshold. The mean and covariance matrix of these selected pixels were used to create a Gaussian distribution model with the probability density function. After identifying the face region using the skin color model, the image was converted to grayscale, and a grayscale closing operation was applied. Morphological operations such as dilation and erosion were performed to refine the face region by filling gaps and removing small holes, smoothing the edges for easier feature

Table 3 Summary of the literature, sorted according to the Face detection– Feature-Based

Year	Authors	Detections	Key methods	Dataset	Accuracy and Advantages	Drawbacks
2005	Fasel et al. [117]	location of a person's eyes in a video stream, real-time eye coding	• AdaBoost algorithm • sequence of smaller classifiers feature vector • likelihood-ratio model	dataset of web images provided by Compaq Research Laboratories	• address the problem of insufficient training data • simultaneous encoding of eye position and behavior in multiple faces • achieve high accuracy and robustness in real-time eye-coding applications	• Detectors that provide high accuracy within the face may exhibit high false-alarm rates outside the face
2005	Vukadinovic et al. [80]	detect 20 facial feature points in expressionless face images	• Gabor feature-based boosted classifiers • GentleBoost templates • individual feature patch templates • vertical and horizontal histogram • horizontal and vertical thresholded edges	Cohn-Kanade database	• fast and robust face detection algorithm • achieved a detection rate of 100% for the irises and the medial point of the mouth • average recognition rate of 93%	• Limited Training Data Tested on Grayscale Images • Limited Variation in Inter-ocular Distance and Sensitivity to Inter-ocular Distance
2004	Chan et al. [114]	one face in an indoor environment with a non-uniform background	• Adapted Chain Code (ACC) • Modified Golden Ratio (MGR) • chrominance color information • skin color segmentation • face boundary estimation • eyes candidate estimation	80 face images consisting of faces with and without spectacles, wearing a headscarf and without wearing a headscarf	achieved 91.25% accuracy in detecting a near-frontal face	Accurate for one face in an indoor environment.

			• wavelet transform • Multilevel thresholding with 3-level priority			
2001	Viola and Jones et al. [116]	object detection, face detection	• Haar-like features • AdaBoost algorithm • machine learning algorithm • sliding window technique • classifiers	manually labeled face and non-face images	achieving high accuracy and fast detection times	• Deeper classifiers in the cascade exhibit higher false positive rates.
1996	Cox et al. [118]	face recognition	• mixture-distance • 30-dimensional feature vector • mixture-distance functions • first and second-order statistics	a database of 685 individuals	• recognition rate of 95%	• Challenges in Gaussian mixture model selection impact recognizer performance due to the difficulty in determining the optimal number of mixtures. • Challenge in optimal model selection
1992	Manjunath et al. [119]	face recognition	• Gabor wavelet • topological graphs • simple deterministic graph marching schema	Face images of 86 persons with two or four images per person, taken with different facial expressions	• approach can be implemented in parallel. • Ability to use real-time face recognition applications. • The recognition accuracy 86%	• Not suitable for complex images and multi faces • Sensitivity to variations in lighting conditions, facial expressions, pose, scale, and occlusions

extraction. For iris detection, the authors applied image processing techniques including illumination normalization and light spot deletion to enhance iris features. They reported a 93.06% accuracy rate for iris detection. The mouth region was detected using the location of the irises, applying a color space method, and the SUSAN [122] corner detector to refine the mouth corners and exact location, achieving a 95.83% accuracy rate. After feature extraction and refinement, template matching was used to classify the extracted features and recognize faces. Experiments on the AR face database, comprising 72 color images of 12 male and 12 female subjects with variations in pose, head orientation, facial expression, and other factors, yielded a recognition rate of 86.11%.

4) *Appearance-Based*

Yang et al. [5] introduced a method for detecting heterogeneous facial features using a Multi-Task Cascaded Convolutional Neural Network (MTCNN), consisting of three sub-networks: Proposal Network (P-Net), Refine Network (R-Net), and Output Network (O-Net). The first step scales the image to various scales to construct an image pyramid. Candidate face windows are obtained using the P-Net, a fully convolutional attention network. Highly overlapping windows are adjusted by a bounding box regression vector and combined using non-maximum suppression (NMS). In the second step, all candidate windows are processed by the R-Net, which refines initial candidate windows and eliminates most non-face candidates using border regression and facial keypoint localization. NMS is applied again to combine intersecting windows. Finally, the O-Net recognizes the facial area, outputting coordinates for the upper left and lower right corners of the face and locations of five facial landmarks (two eyes, nose, and mouth corners). The MTCNN algorithm effectively detects faces with significant variations in pose, scale, and occlusion by leveraging the three networks. It achieved an average accuracy of 96.95% on the CASIA NIR-VIS 2.0, CUFS, and CUFSF datasets, outperforming traditional face detection algorithms and achieving state-of-the-art performance on various benchmarks.

Rowley et al. [83] introduced a face detection system using neural networks, tested with 130 images, achieving a detection rate between 78.9% and 90.5%. The approach has two main stages: coarse scanning and refinement. In the coarse scanning stage, neural network-based filters scan the image at various scales and locations to generate candidate face regions. In the refinement stage, a detailed neural network evaluates these candidate regions to confirm face presence, merging detections and removing overlaps to reduce errors. The method was evaluated on the MIT-CBCL and CMU PIE datasets, comparing its performance with traditional feature-based and other neural network-based methods. The evaluation metric was the detection rate at a false positive rate of 10%. Results showed superior performance, with detection rates of 93.5% on the MIT-CBCL dataset and 96.7% on the CMU PIE dataset, outperforming state-of-the-art methods at the time.

Almabdy et al. [68] conducted a study on face recognition using three CNN-based methods, evaluated on databases including GTAV face, LFW, YouTube face, FEI faces, ORL, F_LFW, and Georgia Tech face. In the first method, they used the pre-trained AlexNet model combined with an SVM classifier for classification. In the second method, they employed a pre-trained ResNet-50 model, also using an SVM for classification. In the third method, they modified AlexNet by removing the last three layers and adding a new fully connected layer for fine-tuning. The accuracy of these models ranged between 94% and 100\%, with the third method achieving 100% accuracy on the GTAV face dataset and outperforming the other methods in accuracy and computational efficiency. Their methods achieved comparable or superior results to state-of-the-art methods on most datasets. They found that increasing the number of training samples improved accuracy. LDA-based dimensionality reduction improved model accuracy compared to PCA, and higher-resolution images generally led to better performance.

Sun et al. [69] proposed a face verification method in unconstrained conditions using a hybrid model combining convolutional networks (ConvNets) with Restricted Boltzmann Machines (RBMs). They used multiple ConvNets to capture high-level and global facial characteristics. The framework integrates deep belief networks, ConvNets, and deep Boltzmann machines in three stages: pre-training, fine-tuning, and feature extraction. During pre-training, deep Boltzmann machines and deep belief networks were trained layer-by-layer. In fine-tuning, the pre-trained models were refined using labeled face verification data. Finally, in feature extraction, the learned representations were used for face verification. The method was evaluated on the LFW [67] and PubFig [123] datasets, using the receiver operating characteristic curve (ROC) and verification rate at a false positive rate of 0.1% as metrics. The hybrid approach outperformed state-of-the-art methods, achieving a verification rate of 99.52% on LFW and 91.54% on PubFig. Experiments showed that each component of the approach contributed to its overall performance.

Schroff et al. [77] proposed FaceNet, a face recognition system that performs face verification, recognition, and clustering using a deep convolutional network. The system learns a 128-D Euclidean embedding per image, representing key facial features with L2 normalization. They employed triplet loss to train the network, ensuring faces with similar features are close in the embedding space, while dissimilar faces are farther apart. Triplet loss minimizes the distance between an anchor image and a positive image (same person) and maximizes the distance to a negative image (different person). A triplet dataset of anchor, positive, and negative images was used for training, with stochastic gradient descent

Table 4 Summary of the literature, sorted according to the Face detection– Template Matching

Year	Authors	Detections	Key methods	Dataset	Accuracy and Advantages	Drawbacks
2022	Najat et al. [120]	Face recognition	• Two-Dimensional Normalized Cross-Correlation (2D-NCC) technique	not mentioned	• detect faces in low-resolution images with varying lighting conditions and facial expressions	• may face limitations when dealing with occlusion and rotation • certain image processing techniques may still require
2020	Bose et al. [121]	detecting facial parts of the human face	• normalized cross-correlation template matching • Template database of facial parts images • correlation maps	more than 100 images were used to verify the algorithm	• detect facial parts in real-time and with high accuracy	• May suffer from limitations such as variations in lighting and facial expressions, occlusion, and rotation
2009	Chai et al. [81]	face detection, iris detection, mouth detection	• skin color model • Gaussian distribution model • probability density function • grayscale closing method • image processing techniques like illumination normalization, • light spot deletion • SUSAN corner detector • color space method • template matching	AR face database	• accuracy rate of 93.06% for iris detection • accuracy rate of 95.83% for mouth detection • accuracy rate of 86.11% for template matching	• Accuracy limited to head-shoulder images with plain backgrounds • Lower intensity differences in iris and facial expressions like frowning can cause iris detection failures. • Failure of the projection method in images with thick facial hair or 'mouth-opened' conditions.

minimizing the triplet loss. In testing, the network produced embeddings compared using a distance metric to verify identity. The model was evaluated on YouTube Faces DB and Labeled Faces in the Wild (LFW), achieving 99.63% accuracy on LFW and 95.12% on YouTube Faces DB, outperforming existing methods and demonstrating the effectiveness of deep learning for face recognition.

Jiang H et al. [54] proposed using the Faster R-CNN framework for face detection, combining a Region-based Convolutional Neural Network (R-CNN) and a Region Proposal Network (RPN). They trained the model using the WIDER face dataset, containing 12,880 images and 159,424 faces. VGG16, pre-trained on ImageNet, was used as the face detection model, and its performance was evaluated on FDDB and IJB-A benchmark datasets. For optimization, they resized input images to 600 pixels on the longer side while maintaining the aspect ratio. They adjusted anchor scales to [16, 32, 64, 128, 256] pixels and reduced the number of proposed regions in the RPN from 3,000 to 600 per image for improved speed and accuracy. A specialized loss function enhanced face detection accuracy. The RPN was the most effective module in the Faster R-CNN model. Their approach achieved state-of-the-art performance on the WIDER face detection dataset, with a mean Average Precision (mAP) score of 94.9%, making the Faster R-CNN framework a popular choice for face detection due to its high accuracy and efficiency.

Sun X et al. [53] proposed an enhanced Faster RCNN approach for face detection using deep learning techniques. They improved the Faster RCNN framework with several strategies: feature concatenation (combining features from different layers), hard negative mining (identifying and adding false negatives to the training set), and multi-scale training (assigning random scales to images for improved scale invariance). The CNN network was trained on the WIDER FACE dataset and tested on the same dataset to generate hard negatives, which were then incorporated into the training process. The model was fine-tuned using the FDDB. They also converted the rectangular detection bounding boxes into ellipses to better match the human face shape. The proposed algorithm demonstrated state-of-the-art results and top rankings among published methods, attributed to the combination of multi-scale training, feature concatenation, model pre-training, hard negative mining, and proper calibration of key parameters.

Guo G et al. [82] proposed a fast face detection algorithm using discriminative complete features (DCFs) from a deep convolutional neural network. The method includes two main components: sparse discriminative features and a nonlinear mapping function. The nonlinear mapping function uses convolutional and max-pooling layers in the CNN to project raw data onto the Euclidean space. Sparseness is achieved using rectified linear units (ReLU), generating a sparse feature space. Once the sparse feature space is created, a nearest neighbor interpolation technique resizes the multi-scale features, enhancing feature extraction at various scales to detect faces of different sizes. The authors propose a fast method to obtain features based on complete feature maps for each window in an input image, extracting desired features before the fully connected layer. The proposed model was trained on the CAS-PEAL-R1 [124] and VOC2012 [125] datasets, and evaluated on the AFW AFW [126] and FDDB [51] datasets. The DCFs-based face detection approach was compared to several state-of-the-art methods, including RCNN, fast RCNN, faster RCNN, DeepIR, and YOLO. A sliding window approach was also used to detect small faces. The proposed method demonstrated significant improvements in face detection performance on benchmark datasets, achieving high accuracy and fast detection speed, making it suitable for real-time applications like surveillance systems and facial recognition technology.

Zhang et al. [101] introduced AInnoFace, a face detection method based on the RetinaNet [127] approach, incorporating modern techniques like Selective Two-step Regression (STR), Intersection over Union (IoU) loss function, Two-step Classification (STC), data augmentation, and max-out operation to reduce false positives. They utilized a multi-scale testing strategy to enhance detection accuracy. AInnoFace employs ResNet-152 with a 6-level feature pyramid structure as the backbone network, enabling feature extraction at multiple scales. It achieves state-of-the-art performance on the WIDER FACE dataset, attributed to the combination of modern techniques and the feature pyramid structure of ResNet-152, which collectively contribute to its high accuracy and fast speed. Experiments on the WIDER FACE dataset compared AInnoFace to YOLO, Faster R-CNN, and SSD detectors using average precision (AP) and average recall (AR) at different IoU thresholds. AInnoFace achieved an AP of 95.8% and an AR of 95.6% at an IoU threshold of 0.5, significantly outperforming other detectors. AInnoFace demonstrated high detection accuracy across different scales of faces, crucial for real-world applications. Its multi-scale testing strategy further contributed to its performance. Overall, AInnoFace exhibits state-of-the-art performance on the challenging WIDER FACE dataset.

Wang J et al. [56] proposed the Face Attention Network (FAN) for detecting partially occluded faces using a deep convolutional neural network with a novel face attention module. This module filters out irrelevant regions to focus on facial features. They introduced an anchor-level attention mechanism to enhance facial parts and improve detection accuracy. The method uses a feature pyramid network to handle faces of different scales, similar to RetinaNet. The FAN method consists of five detector layers, each linked to a specific scale anchor with aspect ratios of 1 and 1.5, covering areas from 162 to 4,062 on pyramid levels. Data augmentation techniques, such as random cropping, generated a large

Table 5 Summary of the literature, sorted according to the Face detection– Appearance-Based

Year	Authors	Detections	Key methods	Dataset	Accuracy and Advantages
2023	Sun et al. [69]	face verification	• hybrid convolutional network (ConvNet) • Restricted Boltzmann Machine (RBM) • multiple groups of ConvNets • Deep Boltzmann machines • Deep belief networks • convolutional neural networks	• LFW • PubFig face verification datasets	• achieve strong characterization of face similarities from different features • verification rate of 99.52% on LFW and 91.54% on PubFig, which outperformed the state-of-the-art methods
2022	Yang et al. [5]	upper left corner coordinates and lower right corner coordinates of the face area and positions of the five feature points, including two eyes, a nose, and two corners of the mouth	• MTCNN • P-Net • R-Net • O-Net • image pyramid • estimated bounding box regression vector • non-maximum suppression (NMS)	• CUFS • CUFSF • CASIA NIR-VIS 2.0	• capable of detecting faces with large variations in pose, scale, and occlusion • achieved an average accuracy of 96.95 %
2022	Cao et al. [103]	detecting face mask-wearing	• YoloMask model • convolutional neural network (CNN) • YOLOX model • CSP-DarkNet • feature pyramid network (FPN) • path aggregation network (PAN) • "Decoupled Head" technique • alpha-CIoU loss function	• Diverse Masked Faces	• authors introduce a new dataset called Diverse Masked Faces • higher detection performance

			• data augmentation techniques such as Mosaic and MixUp • multi-positives trick • OTA • dynamic top-k strategy		
2021	Sanchez et al. [70]	face recognition	• YOLO-Face • image processing techniques • FaceNet+SVM • FaceNet+KNN • FaceNet+RF	• LFW dataset	• FaceNet+SVM model achieves an accuracy of 99.7% • FaceNet+KNN model achieves an accuracy of 99.5% • FaceNet+RF model achieves an accuracy of 85.1% • recognition accuracy of 99.1% and operates in 49 ms
2021	Ali-Gombe et al. [102]	face detection	• 10m-YOLO model • 5m-YOLO model • 2m-YOLO model	• WIDER face • FDDB	• three models were significantly smaller than the original 33m-YOLO model • reduce the model's size • suitable for deployment in a resource-limited environment
2019	Almabdy et al. [68]	face recognition	• Deep Convolutional Neural Network • pre-trained AlexNet + SVM • pre-trained ResNet-50 + SVM • modified AlexNet	• GTAV face • YouTube face • labeled faces in the wild (LFW) • ORL • Frontalized labeled faces in the wild (F_LFW) • Georgia	• 100% accuracy achieved on the GTAV face dataset with modified AlexNet. • modified AlexNet outperformed the other two methods in terms of accuracy and computational efficiency • higher resolution images generally led to better performance

				Tech face • FEI faces	
2019	Zhang et al. [101]	face detection	• AInnoFace detector which is based on the RetinaNet approach • Intersection over Union (IoU) loss function • Two-step Classification (STC) • Selective Two-step Regression (STR) • data augmentation • max-out operation • multi-scale testing strategy • ResNet-152 with a 6-level feature pyramid structure	• WIDER face	• high-performance face detector • achieved high detection accuracy across different scales of faces • achieves state-of-the-art average precision performance results with an AP of 95.8% and an AR of 95.6% • efficiency in detecting faces in complex scenes • high accuracy and fast speed
2019	Zeng et al. [128]	face detection	• cascade face detector • CNN • multi-task learning • network acceleration techniques • multi-scale face proposals • bounding box and facial landmark regression • NMS • multi-layer merging • knowledge distilling • down-sampling • batch normalization (BN) • data augmentations • online and offline hard sample mining • novel multi-layer merging technique	• FDDB	• fast and accurate multi-scale face detection • focus on computational efficiency and accuracy performance • achieving comparable results with state-of-the-art methods at a speed of 165 frames per second on Titan GPU
2018	Sun X et al. [53]	face detection	• improved Faster RCNN	• WIDER face	• achieved state-of-the-art results and ranked the best among all the published

			• feature concatenation • hard negative mining • multi-scale training • converted the detection bounding boxes into ellipses	• FDDB	methods
2018	Guo G et al. [82]	face detection	• discriminative complete features (DCFs) • Deep convolutional network • nonlinear mapping function • sparse discriminative features • Euclidean space • sparse feature space • nearest neighbor interpolation method • sliding window approach	• CAS-PEAL-R1 • VOC2012 • FDDB • AFW	• better feature extraction at different scales • detecting faces of varying sizes • detect small-sized faces • achieves high accuracy and fast detection speed • suitable for real-time applications
2018	Garg et al. [104]	face detection	• YOLO • NMS technique • gradient descent optimizer algorithm	• FDDB	• The proposed approach achieved a high accuracy of 92.2% while maintaining real-time performance
2017	Jiang H et al. [54]	face detection	• Faster R-CNN • Region Proposal Network (RPN) • Region-based Convolutional Neural Network (R-CNN) • pre-trained ImageNet model- VGG16	• WIDER face • FDDB • IJB-A	• mean Average Precision (mAP) score of 94.9% • become a popular choice for face detection due to its high accuracy and efficiency
2017	Wang J et al. [56]	detecting partially occluded faces	• Face Attention Network (FAN) • Deep convolutional neural network with a novel face attention module • new anchor-level attention mechanism • feature pyramid network • data augmentation techniques	• WIDER face • MAFA	• achieved an average precision (AP) of 94.6% (easy) and 88.5% (hard) on the WIDER FACE dataset • achieved an accuracy of 88.3 % on the MAFA dataset • effective solution for detecting partially occluded faces in images

2015	Schroff et al. [77]	face verification, recognition, and clustering	• deep convolutional network called FaceNet • Euclidean embedding • 128-D embedding • L2 normalization • triplet-based loss function • Large Margin Nearest Neighbor (LMNN)	• Labeled Faces in the Wild (LFW) • YouTube Faces DB	• high accuracy of 99.63% for the LFW dataset and 95.12% for the YouTube Faces DB
2015	Ranjan et al. [58]	face detection	• DP2MFD algorithm • Deformable Part Models (DPM) • Deep convolutional neural network • deep pyramidal features • a seven-level normalized deep feature pyramid • sliding window approach • SVM • z-score normalization • 10-fold cross-validation approach • non-maximum suppression and bounding box regression	• AFW • FDDB • MALF • IJB-A	• designed to detect faces of various sizes and poses in unconstrained conditions • achieved new state-of-the-art detection performances • able to detect profile faces as well as different size faces in images with a cluttered background
1998	Rowley et al. [83]	face detection	• neural network-based filters	• MIT-CBCL • CMU PIE	• achieved detection rate between 78.9% and 90.5% for face detection • one of the earliest successful approaches using deep learning techniques in 1998 • The coarse scanning stage and a refinement stage, have become a common paradigm in many faces detection systems

number of occluded faces for training. The authors trained and evaluated FAN on datasets including WIDER FACE and MAFA [129]. FAN achieved an average precision (AP) of 94.6% (easy) and 88.5% (hard) on WIDER FACE, and 88.3% accuracy on MAFA, demonstrating its effectiveness. The FAN method outperformed YOLO, Faster R-CNN, and SSD in accuracy and robustness to occlusion. This research highlights the potential of attention-based methods for face detection and provides an effective solution for detecting partially occluded faces.

Cao et al. [103] propose a novel method for detecting face mask-wearing using the YOLOX [130] model, a state-of-the-art CNN for object detection. They introduce the Diverse Masked Faces dataset, containing five mask-wearing classes: normal, irregular, chin, nose-only, and spoofing. The YoloMask architecture comprises four main components: input, neck, backbone, and predictor. The input is a fixed-size 416x416 RGB image. The backbone is CSP-DarkNet, an improved version of DarkNet inspired by CSPNet. The neck uses Path Aggregation Network (PAN) and Feature Pyramid Network (FPN) techniques to fuse features at different scales. During prediction, the "Decoupled Head" technique splits localization and classification into parallel branches, enhancing accuracy and speed. Experimental results show that YoloMask outperforms state-of-the-art models on popular face detection datasets. A novel composite loss function, alpha-CIoU, is introduced, merging CIoU and alpha-IoU losses to improve performance. The CIoU loss considers the distance between midpoints, width-to-height ratio, and IoU, while the alpha-IoU loss considers the confidence score of the predicted box. The YoloMask model is trained using data augmentation techniques like Mosaic [131] and MixUp [132], and it uses the multi-positives trick, designating the center 3x3 area as positives. The Optimal Transport Assignment (OTA) technique is employed for label assignment, approximating solutions effectively. Evaluated on the Diverse Masked Faces dataset, YoloMask outperforms other state-of-the-art methods. It effectively detects different types of mask-wearing, making it suitable for monitoring proper mask usage in public places, especially important during the COVID-19 pandemic.

Sanchez et al. [70] presented an efficient facial recognition system for real-time operation in unconstrained environments. They combined deep learning techniques, such as FaceNet, with traditional classifiers like K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Random Forest (RF). The system includes face detection, preprocessing, feature extraction, and comparison stages. Using YOLO-Face, a real-time face detector based on YOLOv3, the system detects faces with over 89.6% accuracy on the Honda/UCSD dataset [133], handling partial occlusion, pose variations, and small faces. The preprocessing stage enhances image quality, while the feature extraction stage employs FaceNet and a supervised learning algorithm. FaceNet+SVM achieves 99.7% accuracy on the LFW dataset, with FaceNet+RF and FaceNet+KNN achieving 85.1% and 99.5% respectively. The system reaches 99.1% person identification accuracy by comparing extracted features to enrolled users' known features. The combined face detection and classification stages operate in 49 milliseconds, demonstrating high performance on unconstrained datasets.

Ali-Gombe et al. [102] proposed a technique for face detection using YOLO on edge devices. YOLO, a well-known object detection algorithm, divides an image into grids of varying sizes, such as 52×52, 26×26, and 13×13, and outputs bounding box coordinates for detected objects. YOLO employs a convolutional neural network backbone with multiple convolution layers and a fully connected layer for prediction. The YOLO v3-tiny model, a simplified version of YOLO v3, uses smaller grid sizes (26×26 and 13×13) and only nine convolution layers for faster processing but lower accuracy. The 33m-YOLO model, based on YOLO v3-tiny, includes ten convolution layers, resulting in a 34.8 MB model with 8,676,244 parameters and 5.448 BFLOPS. The 10m-YOLO model, a smaller version, removes layers seven to nine and the final max-pooling layer, resulting in a 10.1 MB model with 2,508,692 parameters and 3.365 BFLOPS. The 5m-YOLO model further reduces the sixth layer's filters from 512 to 256, resulting in a 5.7 MB model with 1,388,948 parameters and 2.987 BFLOPS. The 2m-YOLO model uses 1x1 filters in all convolutions in the second part, creating a 2.7 MB model with 602,516 parameters and 1.924 BFLOPS. Experiments tested the 10m-YOLO, 5m-YOLO, and 2m-YOLO models. While smaller than the 33m-YOLO model, they had a slight reduction in performance. On the WiderFaces dataset, the 2m-YOLO model, 16 times smaller than 33m-YOLO, had a mean Average Precision (mAP) of only 4 points lower. On the FDDB dataset, the smaller models had higher F1 scores than the 33m-YOLO model but also a higher false positive rate, attributed to the dataset. Regarding speed, the 2m-YOLO model was the fastest, achieving 25.9 FPS when tested on a core i5 MacBook Pro-2019 running a pre-recorded one-minute video.

Based on the YOLO architecture, Garg et al. [104] suggested a deep-learning technique for face detection. The study utilized the FDDB dataset, comprising 2,845 images and 5,171 faces, to train and test the proposed face detection model. The model architecture includes seven convolutional layers, a 2x2 max pooling layer, and three fully connected layers. The output layer uses the Non-Maximum Suppression (NMS) technique to predict bounding box coordinates and class probabilities. After tuning various performance parameters, the model was optimized by selecting the best values. The training process was carried out for 25 epochs using a gradient descent optimizer with a learning rate of 0.0001. The suggested approach achieved a high accuracy of 92.2% on the dataset called the FDDB dataset.

Ranjan et al. [58] introduced DP2MFD, a face detection algorithm combining deep pyramidal features and Deformable

Part Models (DPM) to address unconstrained conditions. DP2MFD includes a normalization layer in its deep convolutional neural network (CNN) to mitigate bias towards specific face sizes in deep feature representations. DP2MFD comprises two modules. The first module generates a normalized deep feature pyramid with seven levels for any input image size using a sliding window approach. Fixed-length features are extracted from each pyramid location. The second module employs a linear support vector machine (SVM) to classify pyramid locations as face or non-face based on their scores. Training DP2MFD involved using the FDDB dataset and Caffe framework to train both 1-component (DP2MFD-1c) and 2-component (DP2MFD-2c) DPMs. Positive and negative training samples were collected directly from the deep feature pyramid, and z-score normalization was applied to max5 features at each level to reduce size bias. DP2MFD was evaluated on Face Detection Dataset and Benchmark (FDDB), IARPA Janus Benchmark A (IJB-A), Annotated Face in-the-Wild (AFW), and the Multi-Attribute Labelled Faces (MALF) datasets, achieving new state-of-the-art performance in unconstrained face detection. It detects profile faces and faces of various sizes in cluttered backgrounds. Non-maximum suppression and bounding box regression techniques were employed to improve bounding box accuracy, contributing to DP2MFD's overall high performance.

Zeng et al. [128] proposed a fast and accurate multi-scale face detection method using multi-task learning and network acceleration techniques in a CNN cascade face detector. The method consists of three stages: the first stage is a fully convolutional network with a pyramid architecture that generates multi-scale face proposals with minimal image resizing, detecting faces in a 12x12 window and processing larger windows in subsequent branches. The second and third stages refine face proposals with bounding box and facial landmark regression, using non-maximum suppression (NMS) to eliminate overlaps. To enhance the method, extensive data augmentations, online and offline hard sample mining, and a novel multi-layer merging technique were employed. Network compression and acceleration techniques, including knowledge distilling and merging batch normalization layers with neighboring convolutions, improved inference speed. The network was designed to balance computational efficiency and accuracy, using increased convolutional strides instead of pooling layers for down-sampling. The face detector was tested on the FDDB benchmark, achieving competitive results with state-of-the-art methods and operating at 165 frames per second on a Titan GPU. Key contributions include the novel pyramid network, hard sample mining techniques, and performance improvements via knowledge distilling and multi-layer merging.

VI. RESULTS

This study categorizes the tasks of face detection, facial feature detection, and face recognition into four primary approaches: knowledge-based, template-matching, feature-based, and appearance-based methods. Through a systematic review of 28 research papers spanning from 1991 to 2023, key findings and advancements in face detection and recognition were identified, along with insights into their limitations and applications.

1. Knowledge-Based Methods:

These methods rely on predefined rules and simple image processing techniques such as histogram thresholding, edge detection, and noise reduction. They exhibit high accuracy (over 95%) for detecting frontal faces in plain backgrounds but lack robustness in complex images or multi-face detection scenarios. Computationally efficient, these methods are unsuitable for real-time or multi-face detection tasks due to their limited adaptability.

2. Feature-Based Methods:

Widely used in real-time applications, these methods employ techniques like skin color segmentation, Haar-like features, AdaBoost, and thresholding. Feature-based approaches achieved high accuracy (over 90%) in face detection and recognition, with the ability to detect multiple faces and facial features efficiently. While effective, these methods are less accurate than appearance-based approaches in handling pose variations and occlusions.

3. Template-Matching Methods:

Template-based methods compare predefined templates with detected regions in the image, offering limited flexibility. Although effective in controlled environments, their computational cost and inability to adapt to variations in scale, pose, and lighting conditions make them less suitable for modern real-world applications.

4. Appearance-Based Methods:

The most widely used category, leveraging advanced machine learning and deep learning techniques such as CNN, Faster R-CNN, YOLO, and hybrid convolutional networks. Achieved significant advancements in accuracy (over 99%) and robustness, effectively handling challenges like pose, scale, and occlusions. Techniques such as data augmentation, multi-task learning, and batch normalization have improved detection speed and performance, making these methods highly suitable for real-time applications.

5. Deep Learning Algorithms:

Algorithms like MTCNN, DCNN, and YOLO-Face models dominate the field due to their scalability, accuracy, and efficiency. The integration of traditional classifiers (e.g., SVM, KNN, Random Forest) with deep learning models has further enhanced their performance. These approaches have been applied extensively in applications like biometric systems, driver drowsiness detection, and surveillance.

6. Trends Over Time:

From 1991 to 2010, research focused primarily on knowledge-based and feature-based methods with limited capabilities.

After 2010, deep learning techniques revolutionized the field, enabling significant improvements in accuracy, adaptability, and speed.

Key Outcomes:

Accuracy: Deep learning methods consistently outperform traditional techniques, achieving over 99% accuracy in face detection and recognition tasks.

Applications: Modern algorithms are robust against variations in environmental conditions, enabling their use in real-world scenarios such as surveillance, healthcare, and automotive safety.

Challenges: Despite advancements, further research is needed to address issues such as computational overhead, dataset bias, and performance in extreme environmental conditions.

This systematic review highlights the evolution of face detection and recognition methods and underscores the dominance of deep learning approaches in addressing real-world challenges while identifying areas for future research.

VII. DISCUSSION AND CONCLUSION

The literature discussed in this paper categorizes the tasks of face detection, facial feature detection, and face recognition into four primary categories: knowledge-based, template matching, feature-based, and appearance-based. As shown in

Figure 3, the majority of research has focused on face detection. Moreover, most of the algorithms for face detection and recognition have been developed using the Appearance-Based technique, as depicted in

Figure 4.

Figure 4 provides an overview of the research conducted in face detection and recognition using knowledge-based and

feature-based approaches, indicating that there have been no recent studies in these categories. However, three research studies were identified from 1991 to the present.

In recent studies, deep learning algorithms have been frequently employed to address the challenges of face detection and recognition. The literature highlights the use of machine learning algorithms such as Haar Cascade Classifiers, AdaBoost, CNN, DCNN, and Faster RCNN in various face detection and recognition approaches. Among these algorithms, CNN, DCNN, and Faster RCNN have gained popularity among researchers for their effectiveness in face detection and recognition tasks. Additionally, SVM classifiers are commonly utilized for classification tasks in this domain.

According to Table 2, Knowledge-based methods are used for detecting a human face and facial features like eyes, nose, and mouth. There are several image processing techniques like histogram thresholding, noise reduction, edge detection methods, and edge segments were used to enhance the accuracy of these algorithms. In general, these methods require low computational power, and some algorithms have achieved over 95% accuracy in face detection. According to the literature discussed in this paper, most of the algorithms can only detect one face and are not suitable for complex images and multi-face detection. Also, these algorithms are only accurate for frontal view with plain background images.

In **Error! Reference source not found.**, Feature-based m ethods find extensive application in real-time scenarios involving face detection, facial feature detection, and face recognition. Various techniques including skin color segmentation, face boundary estimation, eyes candidate estimation, thresholding, Haar-like features, classifiers, feature patch templates, and the AdaBoost algorithm, are employed to implement these algorithms. The literature shows that these algorithms achieved high accuracy and successfully detected multiple faces and eye coordinates in real-time. Moreover, these algorithms achieved over 90% accuracy in face recognition and fast detection time.

According to the literature, these algorithms exhibit high accuracy in detecting faces with significant variations in pose, scale, and occlusion. They can achieve accurate results even in complex backgrounds and images containing multiple faces.

Table 5 provides a summary of research conducted on Appearance-Based methods for face detection and recognition, emphasizing their notable success in these areas as well as facial feature detection. Researchers have developed various one-stage and two-stage algorithms, which are extensively utilized in real-time applications.

Two-stage object detectors such as CNN, DCNN, Fast RCNN, and Faster RCNN, as well as one-stage object detectors like YOLO-Face, 10m-YOLO model, 5m-YOLO model, and 2m-YOLO model, have been employed to achieve accurate and efficient face detection and recognition. The literature reveals that deep learning techniques, including MTCNN, DCNN, YOLO, and hybrid convolutional networks like ConvNet, have been combined with traditional classifiers such as K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Random Forest (RF).

Table 5 provides a summary of research conducted on Appearance-Based methods for face detection and recognition, emphasizing their notable success in these areas as well as facial feature detection. Researchers have developed various one-stage and two-stage algorithms, which are extensively utilized in real-time applications.

Two-stage object detectors such as CNN, DCNN, Fast RCNN, and Faster RCNN, as well as one-stage object detectors like YOLO-Face, 10m-YOLO model, 5m-YOLO model, and 2m-YOLO model, have been employed to achieve accurate and

efficient face detection and recognition. The literature reveals that deep learning techniques, including MTCNN, DCNN, YOLO, and hybrid convolutional networks like ConvNet, have been combined with traditional classifiers such as K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Random Forest (RF).

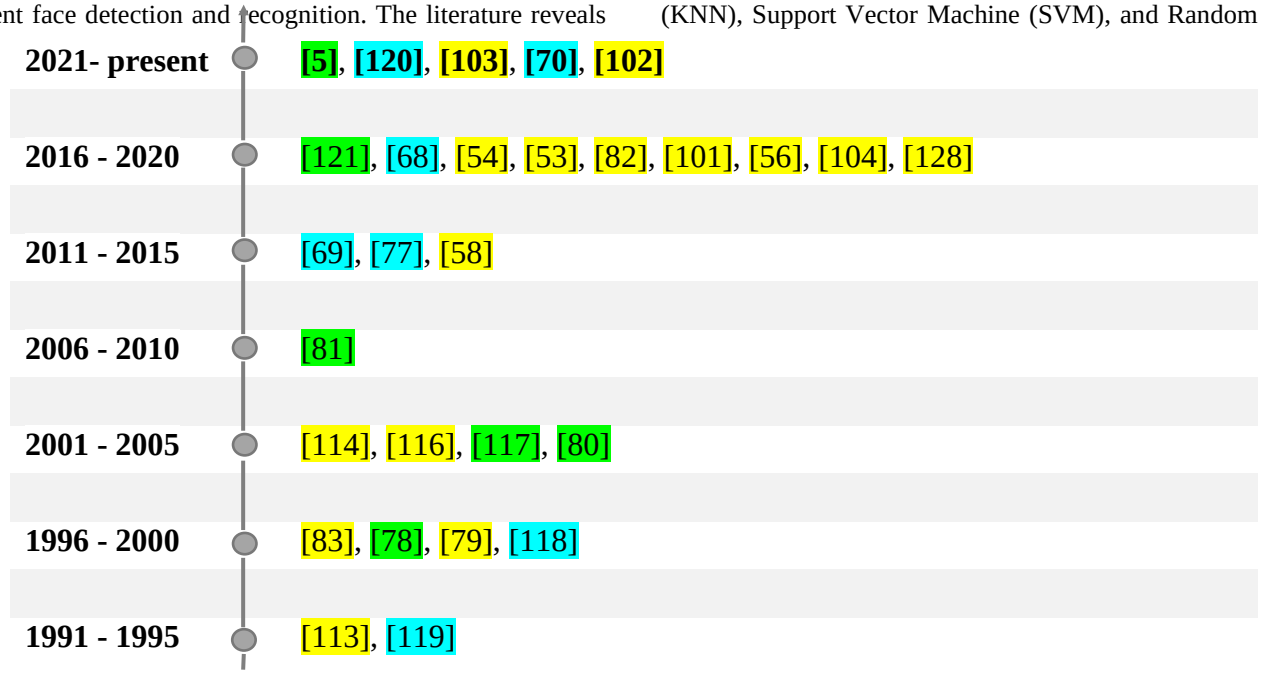

Figure 3 Timeline of research where face detection, facial feature detection, and face recognition method were used; Background colors represent different tasks of problems: face detection, facial feature detection, face recognitio

To enhance the accuracy of face detection and address challenges like pose, scale, and occlusion variations, researchers have employed a variety of training strategies, including data augmentation techniques, multi-task learning, network acceleration techniques, multi-layer merging, down-sampling, and batch normalization. These techniques have enabled the algorithms to achieve accuracy rates of over 99\% in face detection. This demonstrates that these algorithms surpass previous categories in terms of accuracy, making them highly capable of face detection and recognition.

Researchers commonly employed Deep Learning algorithms, highlighting their significant impact on face detection,

recognition, and feature extraction. This observation indicates the remarkable position attained by Deep Learning algorithms in these fields. Due to the increasing complexity of real-world problems, fast and accurate face detection, recognition, and feature extraction have become widely discussed topics in computer vision and object detection. Consequently, significant amounts of research have been conducted to find effective algorithms for these tasks. This paper aims to provide an overview of state-of-the-art research and various mechanisms used in face detection, recognition, and feature extraction.

A total of twenty-eight papers were selected for this study. Beginning with a comprehensive introduction to the topic, we discuss modern algorithms utilized in face detection, recognition, and feature extraction. Additionally, we explore different problem-generation mechanisms employed in various research studies related to these tasks. The content of the paper highlights the fact that face detection, recognition, and feature extraction are considered crucial aspects of object detection, and numerous solutions have been proposed. However, there is still ample room for further research to develop mechanisms that enhance face detection, recognition, and feature extraction, thus making significant contributions to the field of computer vision.

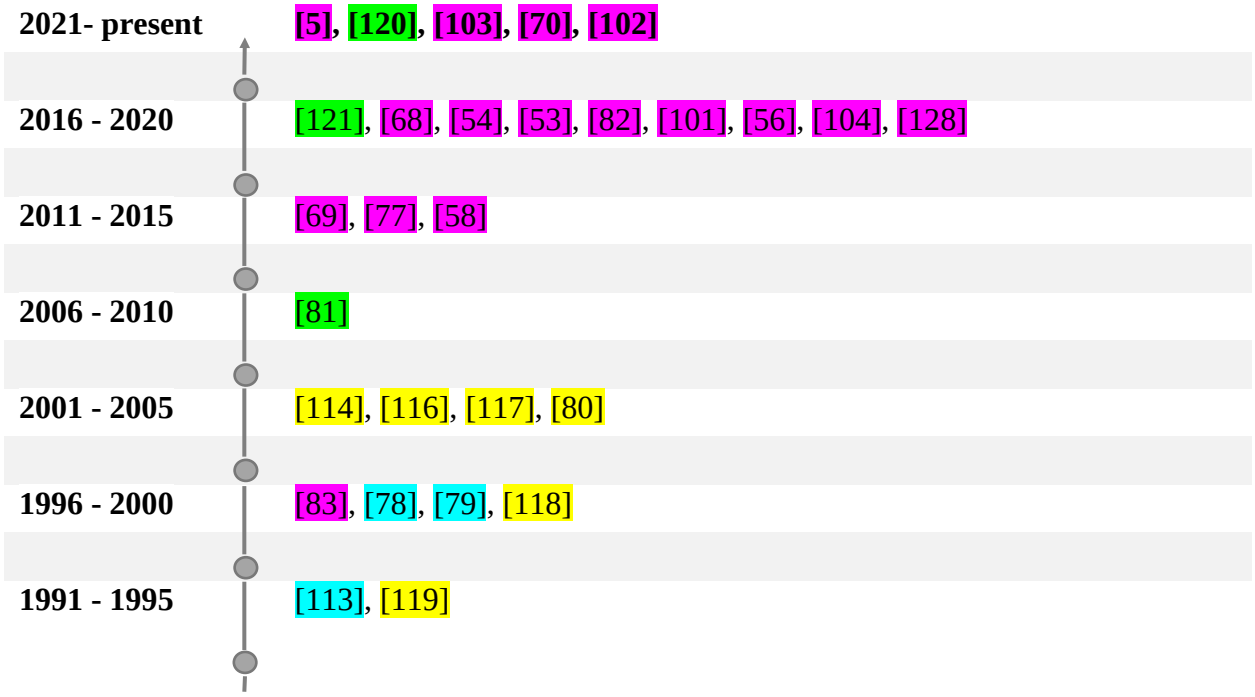

Figure 4 Timeline of research where face detection and recognition method were used; Background colors represent different categories of problems: Knowledge-Based, Feature-Based, Template Matching, Appearance-Based

REFERENCES

[1] Y.-Q. Wang, "An Analysis of the Viola-Jones Face Detection Algorithm," Image Process. Line, vol. 4, pp. 128–148, Jun. 2014, doi: 10.5201/ipol.2014.104.

[2] G. Lowe, "Sift-the scale invariant feature transform," Int J, vol. 2, no. 91–110, p. 2, 2004.

[3] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," IEEE Trans. Pattern Anal. Mach. Intell., vol. 38, no. 1, pp. 142–158, 2015.

[4] S. E. B. D. P. Ltd.23/4736, A. Road, and D. Delhi 110 002, "Multimedia Image and Video Processing," Routledge & CRC Press. Accessed: Mar. 16, 2023. [Online]. Available: https://www.routledge.com/Multimedia-Image-and-Video-Processing/Guan-He-Kung/p/book/9781138072534

[5] X. Yang and W. Zhang, "Heterogeneous face detection based on multi-task cascaded convolutional neural network," IET Image Process., vol. 16, Jan. 2022, doi: 10.1049/ipr2.12344.

[6] T. S. Srinivas, T. Goutham, and D. M. S. Kumaran, “Face Recognition based Smart Attendance System Using IoT,” vol. 09, no. 03, 2022.

[7] A. A. Alsanabani, M. A. Ahmed, and A. M. Al Smadi, “Vehicle Counting Using Detecting-Tracking Combinations: A Comparative Analysis,” in Proceedings of the 2020 4th International Conference on Video and Image Processing, in ICVIP ’20. New York, NY, USA: Association for Computing Machinery, Apr. 2021, pp. 48–54. doi: 10.1145/3447450.3447458.

[8] D. A. A. Deepal and T. G. I. Fernando, “Convolutional Neural Network Approach for the Detection of Lung Cancers in Chest X-Ray Images,” in Deep Learning for Cancer Diagnosis, U. Kose and J. Alzubi, Eds., in Studies in Computational Intelligence. , Singapore: Springer, 2021, pp. 203–226. doi: 10.1007/978-981-15-6321-8_12.

[9] J. Wu, A. Osuntogun, T. Choudhury, M. Philipose, and J. Rehg, “A Scalable Approach to Activity Recognition based on Object Use,” Nov. 2007, pp. 1–8. doi: 10.1109/ICCV.2007.4408865.

[10] M. A. Berbar, H. M. Kelash, and A. A. Kandeel, “Faces and Facial Features Detection in Color Images,” in Geometric Modeling and Imaging–New Trends (GMAI’06), Jul. 2006, pp. 209–214. doi: 10.1109/GMAI.2006.18.

[11] H. Hatem, Z. Beiji, and R. Majeed, “A Survey of Feature Base Methods for Human Face Detection,” Int. J. Control Autom., vol. 8, no. 5, pp. 61–78, May 2015, doi: 10.14257/ijca.2015.8.5.07.

[12] J. Chatrath, P. Gupta, P. Ahuja, A. Goel, and S. M. Arora, “Real time human face detection and tracking,” 2014 Int. Conf. Signal Process. Integr. Netw. SPIN, pp. 705–710, Feb. 2014, doi: 10.1109/SPIN.2014.6777046.

[13] I. R. Tsang, J. P. Magalhaes, and G. D. C. Cavalcanti, “Combined AdaBoost and gradientfaces for face detection under illumination problems,” in 2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Oct. 2012, pp. 2354–2358. doi: 10.1109/ICSMC.2012.6378094.

[14] W. Liu et al., “SSD: Single Shot MultiBox Detector,” vol. 9905, 2016, pp. 21–37. doi: 10.1007/978-3-319-46448-0_2.

[15] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” Jan. 06, 2016, arXiv: arXiv:1506.01497. Accessed: Mar. 16, 2023. [Online]. Available: http://arxiv.org/abs/1506.01497

[16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA: IEEE, Jun. 2016, pp. 779–788. doi: 10.1109/CVPR.2016.91.

[17] D. Q. Rizvi, “A Review on Face Detection Methods,” J. Manag. Dev. Inf. Technol., vol. 11, Feb. 2011.

[18] M. C. P. Archana, C. K. Nitish, and S. Harikumar, “Real time Face Detection and Optimal Face Mapping for Online Classes,” J. Phys. Conf. Ser., vol. 2161, no. 1, p. 012063, Jan. 2022, doi: 10.1088/1742-6596/2161/1/012063.

[19] Sciforce, “Face Detection Explained: State-of-the-Art Methods and Best Tools,” Sciforce. Accessed: Mar. 16, 2023. [Online]. Available: https://medium.com/sciforce/face-detection-explained-state-of-the-art-methods-and-best-tools-f730fca16294

[20] D. Dwivedi, “Face Detection For Beginners,” Medium. Accessed: Mar. 16, 2023. [Online]. Available: https://towardsdatascience.com/face-detection-for-beginners-e58e8f21aad9

[21] S. Harikumar and R. Ramachandran, “Hybridized fragmentation of very large databases using clustering,” in 2015 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES), Feb. 2015, pp. 1–5. doi: 10.1109/SPICES.2015.7091488.

[22] R. Kaur and S. Singh, “A comprehensive review of object detection with deep learning,” Digit. Signal Process., vol. 132, p. 103812, Jan. 2023, doi: 10.1016/j.dsp.2022.103812.

[23] Y. Xiao et al., “A review of object detection based on deep learning,” Multimed. Tools Appl., vol. 79, no. 33, pp. 23729–23791, Sep. 2020, doi: 10.1007/s11042-020-08976-6.

[24] “A Detailed Review on Object Detection Algorithms | IEEE Conference Publication | IEEE Xplore.” Accessed: Dec. 30, 2023. [Online]. Available: https://ieeexplore.ieee.org/document/10125764

[25] Z.-Q. Zhao, P. Zheng, S. Xu, and X. Wu, “Object Detection with Deep Learning: A Review,” Apr. 16, 2019, arXiv: arXiv:1807.05511. Accessed: Dec. 30, 2023. [Online]. Available: http://arxiv.org/abs/1807.05511

[26] “A Survey of Face Recognition Techniques.” Accessed: Jun. 30, 2023. [Online]. Available: https://www.researchgate.net/publication/220635738_A_Survey_of_Face_Recognition_Techniques

[27] M. Lal, K. Kumar, R. Hussain, A. Maitlo, S. Ali, and H. Shaikh, “Study of Face Recognition Techniques: A Survey,” Int. J. Adv. Comput. Sci. Appl., vol. 9, no. 6, 2018, doi: 10.14569/IJACSA.2018.090606.

[28] D. Moher, A. Liberati, J. Tetzlaff, and D. G. Altman, “Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement,” BMJ, vol. 339, p. b2535, Jul. 2009, doi: 10.1136/bmj.b2535.

[29] “Don’t Take Facial Recognition at Face Value: Application Examples Across Different Industries,” AnyConnect. Accessed: Jan. 16, 2024. [Online]. Available: https://anyconnect.com/blog/facial-recognition-applications/

[30] “The Uses of Facial Recognition Across Industries | Mindy Support Outsourcing.” Accessed: Jan. 16, 2024. [Online]. Available: https://mindy-support.com/news-post/the-uses-of-facial-recognition-across-industries/

[31] “Facial Recognition (Updated with Examples).” Accessed: Jan. 16, 2024. [Online]. Available: https://www.thalesgroup.com/en/markets/digital-identity-and-security/government/biometrics/facial-recognition

[32] “Sensors | Free Full-Text | Face Recognition Systems: A Survey.” Accessed: Jan. 16, 2024. [Online]. Available: https://www.mdpi.com/1424-8220/20/2/342

[33] “About Face ID advanced technology,” Apple Support. Accessed: Jan. 16, 2024. [Online]. Available: https://support.apple.com/en-us/102381

[34] “What is biometric unlock technology? Face recognition history | Blackview Blog.” Accessed: Jan. 16, 2024. [Online]. Available: https://www.blackview.hk/blog/tech-news/what-is-biometric-unlock

[35] E. Czerwonka, “5 Best Attendance Systems with Face Recognition.” Accessed: Jan. 16, 2024. [Online]. Available: https://buddypunch.com/blog/attendance-system-face-recognition/

[36] “Top 10 Photo Manager Software with Facial Recognition: A Comprehensive Guide | Daminion Blog.” Accessed: Jan. 16, 2024. [Online]. Available: https://daminion.net/articles/tools/photo-management-software-with-facial-recognition/

[37] E. Fatekhov, “Top 12 Photo Managers with Face Recognition.” Accessed: Jan. 16, 2024. [Online]. Available: https://tonfotos.com/articles/best-face-recognition-software/

[38] “Make Your Own Face Filters in PictoBlox Using the Face Detection,” STEMpedia Education. Accessed: Jan. 16, 2024. [Online]. Available: https://ai.thestempedia.com/project/make-your-own-face-filters-in-pictoblox-using-the-face-detection/

[39] M. Ilves, Y. Gizatdinova, V. Surakka, and E. Vankka, “Head movement and facial expressions as game input,” Entertain. Comput., vol. 5, no. 3, pp. 147–156, Aug. 2014, doi: 10.1016/j.entcom.2014.04.005.

[40] C. Zhan, W. Li, P. Ogunbona, and F. Safaei, “A Real-Time Facial Expression Recognition System for Online Games,” Int. J. Comput. Games Technol., vol. 2008, p. e542918, Mar. 2008, doi: 10.1155/2008/542918.

[41] “Emotional response evoked by viewing facial expression pictures leads to higher temporal resolution - Misa Kobayashi, Makoto Ichikawa, 2023.” Accessed: Jan. 16, 2024. [Online]. Available: https://journals.sagepub.com/doi/10.1177/20416695231152144

[42] “How Emotion-Detection Technology Will Change Marketing.” Accessed: Jan. 16, 2024. [Online]. Available: https://blog.hubspot.com/marketing/emotion-detection-technology-marketing

[43] X. Kong, Z. Wang, J. Sun, X. Qi, and Q. Qiu, “Facial Recognition for Disease Diagnosis Using a Deep Learning Convolutional Neural Network: A systematic Review and Meta-Analysis”.

[44] J. Qiang, D. Wu, H. Du, H. Zhu, S. Chen, and H. Pan, “Review on Facial-Recognition-Based Applications in Disease Diagnosis,” Bioengineering, vol. 9, no. 7, p. 273, Jun. 2022, doi: 10.3390/bioengineering9070273.

[45] B. Abirami, T. S. Subashini, and V. Mahavaishnavi, “Gender and age prediction from real time facial images using CNN,” Mater. Today Proc., vol. 33, pp. 4708–4712, Jan. 2020, doi: 10.1016/j.matpr.2020.08.350.

[46] S. Haseena, S. Saroja, R. Madavan, A. Karthick, B. Pant, and M. Kifetew, “Prediction of the Age and Gender Based on Human Face Images Based on Deep Learning Algorithm,” Comput. Math. Methods Med., vol. 2022, p. e1413597, Aug. 2022, doi: 10.1155/2022/1413597.

[47] W. S. M. Sanjaya, D. Anggraeni, K. Zakaria, A. Juwardi, and M. Munawwaroh, “The design of face recognition and tracking for human-robot interaction,” Nov. 2017, pp. 315–320. doi: 10.1109/ICITISEE.2017.8285519.

[48] Y. Wang, J. Shen, S. Petridis, and M. Pantic, “A real-time and unsupervised face re-identification system for human-robot interaction,” Pattern Recognit. Lett., vol. 128, pp. 559–568, Dec. 2019, doi: 10.1016/j.patrec.2018.04.009.

[49] “AI Facial Recognition with Temperature Measurement | Solution - GIGABYTE Global,” GIGABYTE. Accessed: Jan. 16, 2024. [Online]. Available: https://www.gigabyte.com/Solutions/facialntemp

[50] G. Bae et al., “DigiFace-1M: 1 Million Digital Face Images for Face Recognition,” presented at the Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2023, pp. 3526–3535. Accessed: Jan. 16, 2024. [Online]. Available: https://openaccess.thecvf.com/content/WACV2023/html/Bae_DigiFace-1M_1_Million_Digital_Face_Images_for_Face_Recognition_WACV_2023_paper.html

[51] “FDDB : Main.” Accessed: Apr. 17, 2023. [Online]. Available: http://vis-www.cs.umass.edu/fddb/index.html#explore

[52] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, “Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks,” IEEE Signal Process. Lett., vol. 23, no. 10, pp. 1499–1503, Oct. 2016, doi: 10.1109/LSP.2016.2603342.

[53] X. Sun, P. Wu, and S. C. H. Hoi, “Face detection using deep learning: An improved faster RCNN approach,” Neurocomputing, vol. 299, pp. 42–50, Jul. 2018, doi: 10.1016/j.neucom.2018.03.030.

[54] H. Jiang and E. Learned-Miller, “Face Detection with the Faster R-CNN,” in 2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017), May 2017, pp. 650–657. doi: 10.1109/FG.2017.82.

[55] “WIDER FACE: A Face Detection Benchmark.” Accessed: Apr. 17, 2023. [Online]. Available: http://shuoyang1213.me/WIDERFACE/

[56] J. Wang, Y. Yuan, and G. Yu, “Face Attention Network: An Effective Face Detector for the Occluded Faces,” Nov. 22, 2017, arXiv: arXiv:1711.07246. doi: 10.48550/arXiv.1711.07246.

[57] “CelebA Dataset.” Accessed: Apr. 17, 2023. [Online]. Available: https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html

[58] R. Ranjan, V. M. Patel, and R. Chellappa, “A Deep Pyramid Deformable Part Model for Face Detection,” Aug. 18, 2015, arXiv: arXiv:1508.04389. doi: 10.48550/arXiv.1508.04389.

[59] S. Yang, P. Luo, C. C. Loy, and X. Tang, “Faceness-Net: Face Detection through Deep Facial Part Responses,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 40, no. 8, pp. 1845–1859, Aug. 2018, doi: 10.1109/TPAMI.2017.2738644.

[60] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, “VGGFace2: A dataset for recognising faces across pose and age,”

May 13, 2018, arXiv: arXiv:1710.08092. Accessed: Apr. 17, 2023. [Online]. Available: http://arxiv.org/abs/1710.08092

[61] O. A. Aghdam, B. Bozorgtabar, H. K. Ekenel, and J.-P. Thiran, "Exploring Factors for Improving Low Resolution Face Recognition," in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Jun. 2019, pp. 2363–2370. doi: 10.1109/CVPRW.2019.00290.

[62] NVlabs/ffhq-dataset. (Apr. 17, 2023). Python. NVIDIA Research Projects. Accessed: Apr. 17, 2023. [Online]. Available: https://github.com/NVlabs/ffhq-dataset

[63] T. Karras, S. Laine, and T. Aila, "A Style-Based Generator Architecture for Generative Adversarial Networks," Mar. 29, 2019, arXiv: arXiv:1812.04948. doi: 10.48550/arXiv.1812.04948.

[64] C. E. Bencheriet, H. Abdelmoumène, A. Sebbagh, A. Yahiyaoui, and Z. Taba, "Fake face detection based on a multi discriminator deep CNN architecture (MDD-CNN)," 2023, doi: 10.14311/AP.2023.63.0305.

[65] "Tufts Face Database." Accessed: Apr. 17, 2023. [Online]. Available: https://www.kaggle.com/datasets/kpvisionlab/tufts-face-database

[66] P. Martins, J. Silva, and A. Bernardino, "Multispectral Facial Recognition in the Wild," Sensors, vol. 22, p. 4219, Jun. 2022, doi: 10.3390/s22114219.

[67] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, "Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments".

[68] S. Almabdy and L. Elrefaei, "Deep Convolutional Neural Network-Based Approaches for Face Recognition," Appl. Sci., vol. 9, no. 20, Art. no. 20, Jan. 2019, doi: 10.3390/app9204397.

[69] Y. Sun, X. Wang, and X. Tang, "Hybrid Deep Learning for Face Verification," presented at the Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 1489–1496. Accessed: Mar. 18, 2023. [Online]. Available: https://openaccess.thecvf.com/content_iccv_2013/html/Sun_Hybrid_Deep_Learning_2013_ICCV_paper.html

[70] A. S. Sanchez-Moreno, J. Olivares-Mercado, A. Hernandez-Suarez, K. Toscano-Medina, G. Sanchez-Perez, and G. Benitez-Garcia, "Efficient Face Recognition System for Operating in Unconstrained Environments," J. Imaging, vol. 7, no. 9, Art. no. 9, Sep. 2021, doi: 10.3390/jimaging7090161.

[71] "UTKFace," UTKFace. Accessed: Apr. 17, 2023. [Online]. Available: https://susanqq.github.io/UTKFace/

[72] H. A. Nugroho, R. D. Goratama, and E. L. Frannita, "Face recognition in four types of colour space: a performance analysis," IOP Conf. Ser. Mater. Sci. Eng., vol. 1088, no. 1, p. 012010, Feb. 2021, doi: 10.1088/1757-899X/1088/1/012010.

[73] S. J. Devaraj, R. Catherine Joy, I. Santhosh, and I. C. Kevin, "Deep Learning Based Facial Feature Detection for Ethnicity Recognition," in Smart Computing Techniques and Applications, S. C. Satapathy, V. Bhateja, M. N. Favorskaya, and T. Adilakshmi, Eds., in Smart Innovation, Systems and Technologies. Singapore: Springer, 2021, pp. 527–534. doi: 10.1007/978-981-16-1502-3_52.

[74] "Google facial expression comparison dataset," Google Research. Accessed: Apr. 18, 2023. [Online]. Available: https://research.google/resources/datasets/google-facial-expression/

[75] R. Vemulapalli and A. Agarwala, "A Compact Embedding for Facial Expression Similarity," in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA: IEEE, Jun. 2019, pp. 5676–5685. doi: 10.1109/CVPR.2019.00583.

[76] "YouTube Faces With Facial Keypoints." Accessed: Apr. 18, 2023. [Online]. Available: https://www.kaggle.com/datasets/selfishgene/youtube-faces-with-facial-keypoints

[77] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2015, pp. 815–823. doi: 10.1109/CVPR.2015.7298682.

[78] L. Zhang and P. Lenders, "Knowledge-based eye detection for human face recognition," in KES'2000. Fourth International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies. Proceedings (Cat. No. 00TH8516), IEEE, 2000, pp. 117–120.

[79] C. Kotropoulos and I. Pitas, "Rule-based face detection in frontal views," 1997 IEEE Int. Conf. Acoust. Speech Signal Process., vol. 4, pp. 2537–2540, 1997, doi: 10.1109/ICASSP.1997.595305.

[80] D. Vukadinovic and M. Pantic, "Fully automatic facial feature point detection using gabor feature based boosted classifiers," in 2005 IEEE International Conference on Systems, Man and Cybernetics, IEEE, 2005, pp. 1692–1698.

[81] T.-Y. Chai, R. M, W. San, and T. Seong, "Facial Features for Template Matching Based Face Recognition," Am. J. Appl. Sci., vol. 6, Nov. 2009, doi: 10.3844/ajassp.2009.1897.1901.

[82] G. Guo, H. Wang, Y. Yan, J. Zheng, and B. Li, "A Fast Face Detection Method via Convolutional Neural Network," Mar. 27, 2018, arXiv: arXiv:1803.10103. doi: 10.48550/arXiv.1803.10103.

[83] H. A. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," IEEE Trans. Pattern Anal. Mach. Intell., vol. 20, no. 1, pp. 23–38, Jan. 1998, doi: 10.1109/34.655647.

[84] "TensorFlow.js | Machine Learning for JavaScript Developers," TensorFlow. Accessed: May 01, 2023. [Online]. Available: https://www.tensorflow.org/js

[85] "face-api.js." Accessed: May 13, 2023. [Online]. Available: https://justadudewhohacks.github.io/face-api.js/docs/index.html

[86] "@tensorflow-models/blazeface," npm. Accessed: Feb. 05, 2024. [Online]. Available: https://www.npmjs.com/package/@tensorflow-models/blazeface

[87] "Overview — OpenVINOTM documentation." Accessed: May 13, 2023. [Online]. Available: https://docs.openvino.ai/latest/home.html

[88] H. Wang and J. Hu, "Intelligent lecture recording system based on coordination of face-detection and pedestrian dead reckoning," PeerJ Comput. Sci., vol. 8, p. e971, May 2022, doi: 10.7717/peerj-cs.971.

[89] D. Brown, Mobile Attendance based on Face Detection and Recognition using OpenVINO. 2021, p. 1157. doi: 10.1109/ICAIS50930.2021.9395836.

[90] M. Basurah, W. Swastika, and O. H. Kelana, "IMPLEMENTATION OF FACE RECOGNITION AND LIVENESS DETECTION SYSTEM USING TENSORFLOW.JS," J. Inform. Polinema, vol. 9, no. 4, Art. no. 4, Aug. 2023, doi: 10.33795/jip.v9i4.1332.

[91] D. Yadav, S. Maniar, K. Sukhani, and K. Devadkar, "In-Browser Attendance System using Face Recognition and Serverless Edge Computing," in 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), Jul. 2021, pp. 01–06. doi: 10.1109/ICCCNT51525.2021.9580042.

[92] "GitHub - opencv/opencv at master," GitHub. Accessed: May 13, 2023. [Online]. Available: https://github.com/opencv/opencv

[93] R. T. Hasan and A. B. Sallow, "Face Detection and Recognition Using OpenCV," J. Soft Comput. Data Min., vol. 2, no. 2, Art. no. 2, Oct. 2021.

[94] M. A. Hoque, T. Islam, T. Ahmed, and A. Amin, "Autonomous Face Detection System from Real-time Video Streaming for Ensuring the Intelligence Security System," 2020 6th Int. Conf. Adv. Comput. Commun. Syst. ICACCS, pp. 261–265, Mar. 2020, doi: 10.1109/ICACCS48705.2020.9074260.

[95] A. Das, M. Wasif Ansari, and R. Basak, "Covid-19 Face Mask Detection Using TensorFlow, Keras and OpenCV," in 2020 IEEE 17th India Council International Conference (INDICON), Dec. 2020, pp. 1–5. doi: 10.1109/INDICON49873.2020.9342585.

[96] J. Mehariya, C. Gupta, N. Pai, S. Koul, and P. Gadakh, "Counting Students using OpenCV and Integration with Firebase for Classroom Allocation," Jul. 2020, pp. 624–629. doi: 10.1109/ICESC48915.2020.9155825.

[97] Z. Soomro, T. Memon, F. Naz, and A. Ali, "FPGA Based Real-Time Face Authorization System for Electronic Voting System," Jan. 2020, pp. 1–6. doi: 10.1109/iCoMET48670.2020.9073880.

[98] "#dotnet – Detecting Faces using DNN from the camera feed in a WinForm using #OpenCV and #net5 – El Bruno." Accessed: May 13, 2023. [Online]. Available: https://elbruno.com/2020/11/18/dotnet-detecting-faces-using-dnn-from-the-%F0%9F%8E%A6-camera-feed-in-a-winform-using-opencv-and-net5/

[99] "EmguCV #62: Face Landmark Detection from Images - YouTube." Accessed: May 13, 2023. [Online]. Available: https://www.youtube.com/watch?v=ZOt-A7-Ehq0

[100] Q. Xu, Z. Zhu, H. Ge, Z. Zhang, and X. Zang, "Effective Face Detector Based on YOLOv5 and Superresolution Reconstruction," Comput. Math. Methods Med., vol. 2021, p. e7748350, Nov. 2021, doi: 10.1155/2021/7748350.

[101] F. Zhang, X. Fan, G. Ai, J. Song, Y. Qin, and J. Wu, "Accurate face detection for high performance," ArXiv Prepr. ArXiv190501585, 2019.

[102] A. Ali-Gombe, E. Elyan, and J. Zwiegelaar, "Face detection with YOLO on edge.," Jul. 2021, doi: 10.1007/978-3-030-80568-5_24.

[103] Z. Cao, W. Li, H. Zhao, and L. Pang, "YoloMask: An Enhanced YOLO Model for Detection of Face Mask Wearing Normality, Irregularity and Spoofing," Nov. 2022, pp. 205–213. doi: 10.1007/978-3-031-20233-9_21.

[104] D. Garg, P. Goel, S. Pandya, A. Ganatra, and K. Kotecha, "A Deep Learning Approach for Face Detection using YOLO," in 2018 IEEE Punecon, Nov. 2018, pp. 1–4. doi: 10.1109/PUNECON.2018.8745376.

[105] S. Ding, L. Lin, G. Wang, and H. Chao, "Deep feature learning with relative distance comparison for person re-identification," Pattern Recognit., vol. 48, no. 10, Art. no. 10, Oct. 2015, doi: 10.1016/j.patcog.2015.04.005.

[106] L. Wolf, "Face Recognition, Geometric vs. Appearance-Based," in Encyclopedia of Biometrics, S. Z. Li and A. Jain, Eds., Boston, MA: Springer US, 2009, pp. 347–352. doi: 10.1007/978-0-387-73003-5_92.

[107] "CBCL FACE RECOGNITION DATABASE." Accessed: Mar. 15, 2024. [Online]. Available: http://cbcl.mit.edu/software-datasets/heisele/facerecognition-database.html

[108] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker, "Multi-PIE," in 2008 8th IEEE International Conference on Automatic Face & Gesture Recognition, Sep. 2008, pp. 1–8. doi: 10.1109/AFGR.2008.4813399.

[109] "The CMU Multi-PIE Face Database." Accessed: Mar. 15, 2024. [Online]. Available: https://www.cs.cmu.edu/afs/cs/project/PIE/MultiPie/Multi-Pie/Home.html

[110] G. Boesch, "Object Detection in 2023: The Definitive Guide," viso.ai. Accessed: Mar. 16, 2023. [Online]. Available: https://viso.ai/deep-learning/object-detection/

[111] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in 2017 IEEE International Conference on Computer Vision (ICCV), Oct. 2017, pp. 2980–2988. doi: 10.1109/ICCV.2017.322.

[112] "Object Detection: Models, Architectures & Tutorial [2023]." Accessed: Mar. 16, 2023. [Online]. Available: https://www.v7labs.com/blog/object-detection-guide#h2

[113] G. Yang and T. S. Huang, "Human face detection in a complex background," Pattern Recognit., vol. 27, no. 1, pp. 53–63, Jan. 1994, doi: 10.1016/0031-3203(94)90017-5.

[114] Y. H. Chan and S. A. R. Abu-Bakar, "Face detection system based on feature-based chrominance colour information," in Proceedings. International Conference on Computer Graphics, Imaging and Visualization, 2004. CGIV 2004., IEEE, 2004, pp. 153–158.

[115] R.-L. Hsu, M. Abdel-Mottaleb, and A. K. Jain, "Face detection in color images," IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 5, pp. 696–706, May 2002, doi: 10.1109/34.1000242.

[116] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern

Recognition. CVPR 2001, Kauai, HI, USA: IEEE Comput. Soc, 2001, p. I-511-I–518. doi: 10.1109/CVPR.2001.990517.

[117] I. R. Fasel, B. Fortenberry, and J. R. Movellan, “GBoost: A generative framework for boosting with applications to realtime eye coding,” Comput. Vis. Image Underst., vol. 98, no. 1, pp. 182–210, 2005.

[118] I. J. Cox, J. Ghosn, and P. N. Yianilos, “Feature-based face recognition using mixture-distance,” in Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA: IEEE, 1996, pp. 209–216. doi: 10.1109/CVPR.1996.517076.

[119] B. S. Manjunath, R. Chellappa, and C. von der Malsburg, “A feature based approach to face recognition,” in Proceedings 1992 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Jun. 1992, pp. 373–378. doi: 10.1109/CVPR.1992.223162.

[120] N. Abdulsada and S. Ali, “Human face detection in a crowd image based on template matching technique,” presented at the AIP Conference Proceedings, Aug. 2022, p. 020033. doi: 10.1063/5.0093156.

[121] P. Bose and S. Bandyopadhyay, “Human Face and Facial Parts Detection using Template Matching Technique,” Int. J. Eng. Adv. Technol., vol. 9, pp. 2249–8958, May 2020, doi: 10.35940/ijeat.D6689.049420.

[122] S. M. Smith and J. M. Brady, “SUSAN—A New Approach to Low Level Image Processing,” Int. J. Comput. Vis., vol. 23, no. 1, pp. 45–78, May 1997, doi: 10.1023/A:1007963824710.

[123] “Pubfig: Public Figures Face Database.” Accessed: Mar. 15, 2024. [Online]. Available: https://www.cs.columbia.edu/CAVE/databases/pubfig/

[124] “The PEAL Face Database.” Accessed: Mar. 15, 2024. [Online]. Available: http://www.jdl.link/peal/index.html

[125] “The PASCAL Visual Object Classes Challenge 2012 (VOC2012).” Accessed: Mar. 15, 2024. [Online]. Available: http://host.robots.ox.ac.uk/pascal/VOC/voc2012/

[126] X. Zhu and D. Ramanan, “Face detection, pose estimation, and landmark localization in the wild,” in 2012 IEEE Conference on Computer Vision and Pattern Recognition, Jun. 2012, pp. 2879–2886. doi: 10.1109/CVPR.2012.6248014.

[127] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal Loss for Dense Object Detection,” in 2017 IEEE International Conference on Computer Vision (ICCV), Oct. 2017, pp. 2999–3007. doi: 10.1109/ICCV.2017.324.

[128] D. Zeng, F. Zhao, S. Ge, and W. Shen, “Fast cascade face detection with pyramid network,” Pattern Recognit. Lett., vol. 119, pp. 180–186, Mar. 2019, doi: 10.1016/j.patrec.2018.05.024.

[129] “Masked Face Analysis.” Accessed: Mar. 15, 2024. [Online]. Available: https://imsg.ac.cn/research/maskedface.html

[130] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, “YOLOX: Exceeding YOLO Series in 2021,” Aug. 05, 2021, arXiv: arXiv:2107.08430. doi: 10.48550/arXiv.2107.08430.

[131] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “Yolov4: Optimal speed and accuracy of object detection,” ArXiv Prepr. ArXiv200410934, 2020.

[132] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “mixup: Beyond Empirical Risk Minimization,” Apr. 27, 2018, arXiv: arXiv:1710.09412. doi: 10.48550/arXiv.1710.09412.

[133] K.-C. Lee, J. Ho, M.-H. Yang, and D. Kriegman, “Visual tracking and recognition using probabilistic appearance manifolds,” Comput. Vis. Image Underst., vol. 99, no. 3, pp. 303–331, Sep. 2005, doi: 10.1016/j.cviu.2005.02.002.

Journal website: