Thesis_Final_Draft - Content.pdf

Delhi Technological University

Document Details

Submission ID
trn:oid:::27535:129277583

Submission Date
Feb 26, 2026, 3:05 PM GMT+5:30

Download Date
Feb 26, 2026, 3:09 PM GMT+5:30

File Name
Thesis_Final_Draft - Content.pdf

File Size
10.8 MB

132 Pages

30,611 Words

172,661 Characters

4% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography

Exclusions

- 2 Excluded Sources
- 4 Excluded Matches

Match Groups

80 Not Cited or Quoted 4%
Matches with neither in-text citation nor quotation marks

11 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

0% Internet sources
3% Publications
2% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

- **80** Not Cited or Quoted 4%
  Matches with neither in-text citation nor quotation marks
- **11** Missing Quotations 0%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

- 0% Internet sources
- 3% Publications
- 2% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Submitted works	Staffordshire University on 2025-02-04 <1%
2	Publication	Paolo Ferro, Harinadh Vemanaboina, Chander Prakash. "Computational Techniqu... <1%
3	Submitted works	University of Wollongong on 2018-08-16 <1%
4	Publication	"Advanced Concepts for Intelligent Vision Systems", Springer Science and Busines... <1%
5	Publication	"Biometric Recognition", Springer Science and Business Media LLC, 2016 <1%
6	Submitted works	University of Information Technology, Yangon on 2020-08-21 <1%
7	Publication	Pushpa Choudhary, Sambit Satpathy, Arvind Dagur, Dhirendra Kumar Shukla. "Re... <1%
8	Publication	"Computer Vision – ECCV 2016", Springer Nature, 2016 <1%
9	Submitted works	University of Wollongong on 2023-07-14 <1%
10	Publication	"Intelligent Computing", Springer Science and Business Media LLC, 2019 <1%

11	Internet	
ebin.pub		<1%
12	Internet	
www.ijimai.org		<1%
13	Internet	
eprints.qut.edu.au		<1%
14	Publication	
Ongkittikul, Surachai. "Hand Tracking with Parametric Skin Modelling Using Parti...		<1%
15	Submitted works	
University of Monastir on 2023-12-31		<1%
16	Submitted works	
Brunel University on 2013-04-05		<1%
17	Publication	
Vishakha Sood, Arun Lal Srivastav, Ravneet Kaur, Neha Bhati. "Generative AI for R...		<1%
18	Publication	
"Advances in Face Detection and Facial Image Analysis", Springer Nature, 2016		<1%
19	Publication	
松井 淳. "Bayesian on-line learning for video face recognition with applications in ...		<1%
20	Publication	
Lecture Notes in Computer Science, 2016.		<1%
21	Publication	
"Neural Information Processing", Springer Science and Business Media LLC, 2017		<1%
22	Publication	
Frank Y. Shih. "AI Deep Learning in Image Processing", CRC Press, 2025		<1%
23	Publication	
Lecture Notes in Computer Science, 2006.		<1%
24	Publication	
Arvind Dagur, Karan Singh, Pawan Singh Mehra, Dhirendra Kumar Shukla. "Artific...		<1%

25 Publication

Jin Liu, Jianxin Wang, Yi Pan. "AI in MRI-based Brain Disease Prediction", CRC Pres... <1%

26 Publication

Lecture Notes in Computer Science, 2015. <1%

27 Publication

Loubrys Lázaro Rojas Reinoso. "Um sistema de análise facial em tempo real para ... <1%

28 Internet

link.springer.com <1%

29 Publication

João Manuel R. S. Tavares, R. M. Natal Jorge. "Computational Vision and Medical I... <1%

30 Publication

Li, Pei. "Studying Unconstrained Degraded Face Recognition and Redaction With ... <1%

31 Publication

Siddhartha Roy, Soumya Sen, Agostino Cortesi. "Intelligent Systems - Emerging Tr... <1%

Chapter 1

INTRODUCTION

In the rapidly evolving field of computer vision, face detection and tracking have become essential technologies with a wide range of applications, including real-time video analysis, security systems, and human-computer interaction. These techniques are fundamental to various systems, such as facial recognition for security purposes, augmented reality experiences, and live surveillance. However, the increasing demand for high-quality video content present key challenges for existing face detection models. High-resolution video streams generate substantial amount of data, which can lead to a strain on computational resources utilization. This thesis explores these challenges and offers some optimized solutions designed to enhance the efficiency of face detection and tracking algorithms, ensuring they continue to work effectively even when handling high-quality video streams.

1.1 *Background of the Study*

Face detection and tracking have been proven to be essential components of real-time video processing. The traditional algorithms that form the foundation of this domain, such as the Viola-Jones framework [2], have provided real-time results in low-quality video streams but are inadequate when faced with the complexity of high-definition (HD) video feeds. This inadequacy stems from the substantial increase in data volume, leading to slower processing speeds and lower computational efficiency. With the exponential growth in the use of high-definition

cameras for applications ranging from entertainment to surveillance, there is an increasing need for algorithms capable of maintaining accuracy and speed.

Figure 1.1 illustrates the flowchart depicting the steps involved in a typical Face Detection and Tracking system.

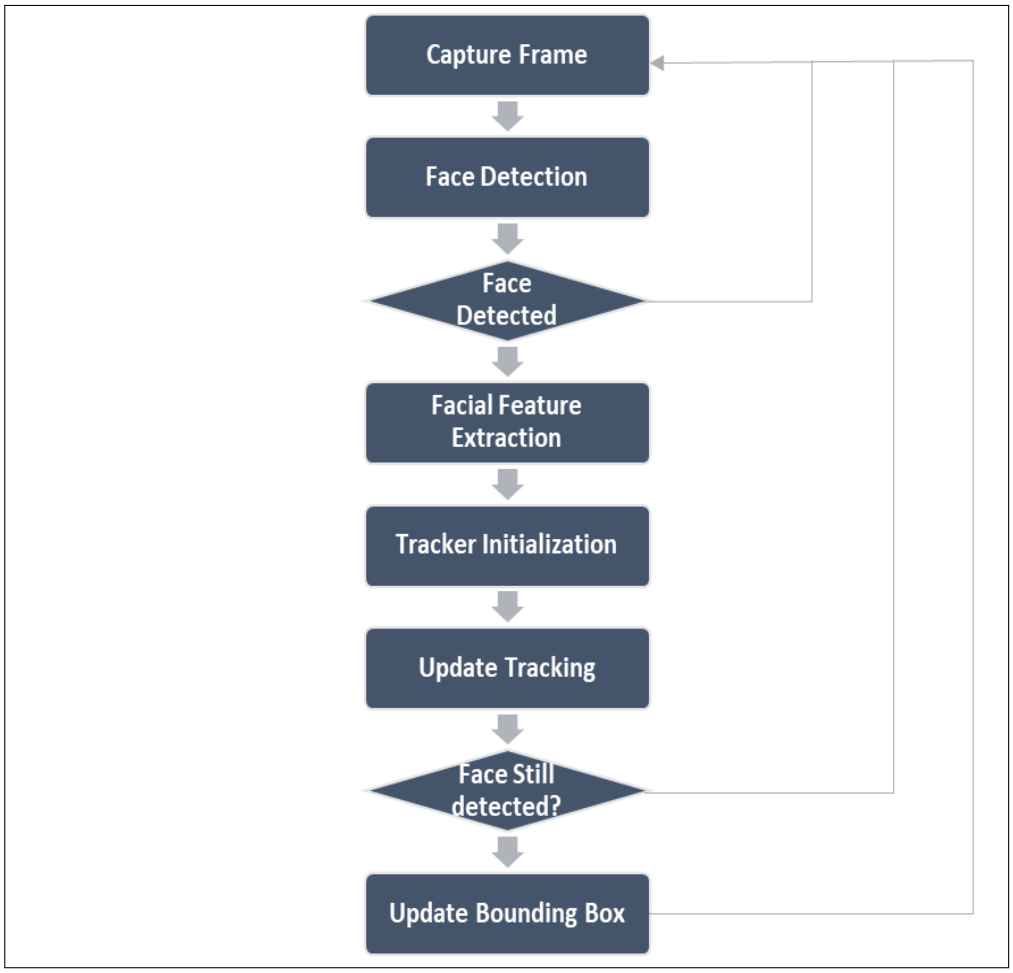


Figure 1.1: Flowchart depicting a typical Face Detection and Tracking system

Recent advancements in machine learning, particularly in convolutional neural networks (CNNs), have introduced highly accurate detection algorithms, including MTCNN and YOLO. These algorithms are renowned for their rapid detection capabilities in real-time scenarios, but even they struggle when handling HD content, where data per frame is significantly higher. Consequently, there is a pressing need for an optimized face detection system that can balance speed and accuracy while reducing computational overhead, particularly in high-quality

video environments.

1.2 *Problem Statement*

As the demand for high-quality video processing rises, existing face detection algorithms encounter several challenges. The large amount of data in each frame, characteristic of high-resolution videos, increases the computational cost of detection algorithms, thereby slowing the entire process. In real-time systems, such as autonomous vehicles, security surveillance, and live event broadcasting, delays caused by inefficient processing can render these systems ineffective. Current models are optimized for low-resolution streams and cannot cope with the computational requirements imposed by high-quality content without a significant loss in performance.

This research aims to bridge this gap by proposing two optimized models: first model integrates the face detection with feature-tracking technique, and the second model integrates face detection with a Non-Neighbourhood Background Elimination (NNBE) technique. These methods focus computational efforts on regions of interest within the video frames, reducing the amount of unnecessary data processed in each frame. By tracking only the detected facial landmark points across frames, and limiting the detection area to the previously detected face region, these models significantly reduce computational demands without compromising the accuracy of face detection and tracking.

1.3 *Objectives of the Study*

The main objective of this thesis is to develop a model that optimizes face detection and tracking in high-quality videos, improving processing speed without compromising detection accuracy. To achieve this, the following specific objectives are defined:

1. To implement various traditional and State-of-the-Art methods of Face de-

tection and tracking.

2. To develop a novel efficient algorithm for face detection that can optimize the trade-off between feature detection and non-facial region elimination.

3. To develop a face detection model that gives high accuracy while increasing execution speed

4. To design a face tracking algorithm that tracks detected faces at high rate.

1.4 *Contributions of the Research*

This research introduces an optimized face detection model that addresses the existing limitations in processing high-quality video streams. It also introduces a Neighbourhood calculating technique, that facilitates in improving the processing speed of overall face detection system. The key contributions are as follows:

1. The proposed model integrates the use of Face Detection algorithms with Feature-tracking technique, which reduces the processing time by decreasing the required computations of face detection in every frame, making it significantly faster to process faces in High-quality video streams.

2. The work also contributes by introducing simple occlusion resolution methods using extrapolation techniques, improving the reliability of face tracking when parts of the face are obscured.

3. Another contribution of the work is by proposing a model that integrates an NNBE technique, which reduces the area of interest and limits unnecessary computations in subsequent frames, significantly improving the processing speed for HD video feeds.

4. This thesis provides a comprehensive evaluation of the proposed models across various frame rates and video qualities, demonstrating its superiority in terms of execution time and accuracy compared to current commercial models like YOLO and MTCNN.

1.5 *Thesis Outline*

The thesis is organized into several chapters as follows:

1. Chapter 2 reviews existing literature on face detection and tracking algorithms, with an emphasis on real-time processing and computational efficiency in high-quality video content.

2. Chapter 3 provides an analysis of traditional and modern face detection methods, highlighting their strengths, weaknesses, and suitability for HD video applications.

3. Chapter 4 details the datasets used for evaluating the proposed model, focusing on high-resolution video feeds and the challenges they present to detection algorithms.

4. Chapter 5 presents an optimized model, detailing the integration of the Face Detection technique with Feature tracking, and its impact on performance.

5. Chapter 6 discusses the issue of occlusion in face tracking, proposing methods for resolving partial and full occlusions in high-resolution videos.

6. Chapter 7 presents the NNBE component for the overall face detection and tracking systems, and evaluates the performance improvements of various Face Detection methods by its integration.

7. Chapter 8 integrates the previously discussed NNBE component to the Face Detection System, and evaluates the runtime performance of the proposed model, offering a detailed comparison with existing commercial face detection systems.

8. Chapter 9 summarizes the findings, conclusions, and potential future research directions in the field of face detection and tracking.

Chapter 2

LITERATURE REVIEW

This chapter presents an overview of recently published, highly cited research on face detection algorithms, and object and face tracking methods, and their integration in effective Face Detection and Tracking systems. It also includes a review of studies assessing the impact of traditional statistical and machine learning techniques on the evolution of face detection in the field of Computer Vision.

2.1 *Introduction*

With advancements in technology, research increasingly focused on improving the quality of life for humans as technology users. Many of these technologies required systems capable of detecting and tracking the presence of humans, particularly in video feeds. Among the key identifiers in such systems was the human face, which was commonly used to recognize and identify individuals.

As a result, extensive research had been conducted on detecting and identifying the presence of human faces in video streams. This information was then utilized to track the movements of identified individuals for various applications such as security, surveillance, gesture control, and emotion recognition. This research specifically targets the detection of human faces in high-quality video streams, and their tracking across subsequent frames.

2.2 *Reviews on different researches for Face Detection and Tracking schemes*

To identify the new field of research work and to understand the state-of-the-art environment, extensive literature is collected in all the diverse fields of face detection. The literature is research oriented, which explains only the concepts, and provides an introduction to the discussed techniques in this field for interested readers. To understand and interpret the previous works on different aspects related to the formulation of the present research problem, the literature available in various sub-areas of face detection that mainly includes facial landmark localisation, point and feature tracking, object detection in general, machine learning and deep learning insights into faces, and other specific literature are also considered. An effort is made to review the literature based on defined broad objectives as well as identified literature features related to proposed work. Identified categories are as:

- Foundational Studies
- Classical Face Detection Techniques
- Other Modern Face Detection Techniques
- Related Object Detection Schemes
- Face Tracking Methods
- Real-Time Processing and Computational Efficiency
- Human-Computer Interaction and Gesture Control
- Deep Learning Techniques and Facial Recognition

It is beyond the scope of this thesis to descriptively report all the references listed under the various headings. Therefore, an attempt is made to portray the entire literature in a format so that only a few research articles are covered in the explanation for each category as a representative work in that particular category. It helps in deriving important inferences regarding the trend and potential for further research in that field.

The literature has not been exhaustive owing to the human information-processing

limitations, and therefore, is only an indicative sample; nevertheless, it supports the development of the methodology for efficient processing of the high-quality facial content information carried out in the thesis. The following few sections discuss only the major influential articles in various categories which are very important to the basic formulation of the problem selected for the present research work.

2.2.1 Foundational Studies

Tomasi and Kanade (1991) [3] presented foundational techniques for detecting and tracking point features in images. Their work laid the groundwork for many modern computer vision algorithms by establishing robust methods for feature detection and tracking. These techniques greatly influenced the development of contemporary face detection and tracking systems, marking a pivotal contribution to the field.

The influential work by Shi and Tomasi (1994) [4] presented a method for identifying robust features that could be reliably tracked over time in computer vision applications. Their algorithm focused on selecting features that minimized optical flow estimation error, thus ensuring reliable tracking. The method's effectiveness was demonstrated in applications such as object recognition, motion analysis, and video surveillance. This paper was widely cited and significantly influenced subsequent research in feature detection and tracking.

Viola and Jones (2000) introduced a groundbreaking method for object detection, particularly face detection [5], using a boosted cascade of simple features. Their approach employed a series of classifiers trained with AdaBoost to quickly reject non-face regions in an image. By utilizing simple rectangular features, computed rapidly with an integral image, the cascade structure enabled real-time performance. This method became a standard in the field and was widely implemented in software frameworks like OpenCV.

Yang, Kriegman, and Ahuja (2002) provided a comprehensive survey of face detection techniques [6], categorizing them into knowledge-based methods, feature-

invariant approaches, template matching methods, and appearance-based methods. The survey discussed the strengths and weaknesses of each category and addressed challenges such as variations in pose, lighting, and facial expressions. This paper became a foundational reference for anyone researching face detection and offered valuable insights into the state of the field as of 2002.

Yang et al. (2002) [6] provided an extensive survey of face detection techniques, categorizing them into feature-based and image-based approaches. This comprehensive review offered valuable insights into the strengths, limitations, and applications of various methods, summarizing the evolution of face detection technologies and guiding future research directions.

Chao (2007) [7] provided a comprehensive overview of face recognition techniques, offering foundational knowledge and methodologies used in the field. The paper covered various aspects of face recognition, including theoretical foundations, algorithmic approaches, and practical applications. It reviewed both classical and modern methods, ranging from geometric feature-based approaches to machine learning and neural networks, serving as an educational resource for understanding the development and evolution of face recognition technology. Additionally, it discussed implementations in security, authentication, and human-computer interaction, making it valuable for students, researchers, and practitioners developing face recognition systems.

Seshadrinathan et al. (2010) [8] performed a comprehensive subjective study to evaluate video quality assessment (VQA) algorithms, incorporating human perception into the analysis. They collected ratings from 38 observers on 150 distorted video sequences, which included various compression artifacts and transmission errors. By using a single-stimulus paradigm with hidden reference removal, the study minimized bias and led to the creation of the LIVE Video Quality Database. This publicly available benchmark helped evaluate VQA algorithms against human perception, with the MOtion-based Video Integrity Evaluation (MOVIE) index emerging as a leading performer. This research bridged the gap between objective algorithms and human perception, providing a critical resource

for developing and validating future VQA systems.

Yang et al. (2014) [9] presented their research at the 2014 IEEE International Joint Conference on Biometrics, focusing on enhancing face detection accuracy in multi-view scenarios where faces were captured from diverse angles. Their proposed approach involved extracting features from multiple image color channels and combining them to achieve robust face detection across a wide range of head poses. This method addressed the challenge of detecting faces in varying orientations, a key issue in real-world applications.

Gupta (2014) [10] offered a critical analysis of various face detection methodologies, comparing their strengths and weaknesses in different application scenarios. This review provided a comprehensive comparison of existing techniques, serving as a valuable resource for researchers and practitioners. It helped in selecting appropriate face detection methods based on specific needs, ranging from security systems with high accuracy requirements to social media platforms that prioritize speed and efficiency.

Dehkordi et al. (2015) [11] explored the impact of frame rates on the quality and bitrate of 3D videos, providing valuable insights for optimizing video playback. The study examined how varying frame rates influenced the perceived quality and corresponding bitrate requirements of 3D video content. It utilized both subjective assessments from viewers and objective quality metrics to evaluate different frame rates. The research observed that higher frame rates generally enhanced perceived video quality but also increased bitrate requirements. The study identified optimal frame rates that balanced quality and bitrate, offering guidance for designing efficient 3D video systems and streaming protocols, ensuring high-quality playback while managing bandwidth effectively.

Marcomini and Cunha (2018) [12] presented a detailed comparison of background modeling techniques for vehicle segmentation, offering valuable insights into methods that could be adapted for background subtraction in facial analysis tasks. These background modeling strategies could prove beneficial for isolating facial regions from dynamic video environments, particularly in scenes with mov-

ing cameras. Meijering's (2002) [13] historical review of interpolation techniques shed light on image resolution enhancement strategies. While not directly linked to facial detection, the insights gained could be useful during the pre-processing of high-quality multimedia content within the model, potentially improving the precision of subsequent facial analysis.

Pairo et al. (2019) [14] focused on delay-free object tracking for robotics applications, emphasizing the critical role of real-time processing—a goal that aligned with the objectives of the proposed face detection model. Their work on optimizing real-time object tracking could inspire the development of efficient algorithms for real-time face detection within the system. Meanwhile, Park et al. (2011) [15] proposed a multiscale foreground extraction technique that could be applied to extracting facial regions from complex video backgrounds. Incorporating such foreground extraction methods could help the model achieve more reliable facial detection in busy or cluttered video environments.

2.2.2 Viola-Jones Variations and Optimizations

Viola and Jones (2001) [5] introduced a groundbreaking face detection technique combining integral images, AdaBoost, and cascaded classifiers. This innovation set a new standard for accuracy and speed in face detection, influencing subsequent research and applications. The method's robustness and efficiency had a profound impact on the field, making it a foundational reference for modern face detection systems.

Viola and Jones (2004) [16] expanded on their earlier work, providing a detailed description of their face detection framework. They elaborated on the training process, feature selection, and the construction of the classifier cascade. Extensive experiments demonstrated the algorithm's robustness under various conditions, including different lighting, poses, and occlusions. The practical implementation aspects discussed made this paper a valuable resource for both researchers and developers. The method's real-time performance on standard hardware led to its widespread adoption in various applications.

Wu et al. (2008) [17] presented advancements in cascade face detection through fast asymmetric learning techniques. They proposed an optimized training method for cascade classifiers, focusing on reducing false positives while maintaining high detection rates. The asymmetric boosting algorithm introduced enhanced the learning process, making the cascade more effective at distinguishing face from non-face regions. Experimental results indicated significant improvements in both speed and accuracy, contributing to the development of more efficient and robust real-time face detection systems.

Sundaraj (2008) [18] examined a real-time face detection approach that used dynamic background subtraction to handle fluctuating backgrounds, which are common in outdoor surveillance. The method's key innovation was its dynamic background model that adapted to changing conditions, ensuring robust face detection. This represented a significant improvement over static models, enhancing accuracy in dynamic environments.

Heijden et al. (2010) [19] investigated the two-stage process of visual information processing and perception, offering insights into how the human brain interprets visual data. The proposed theory outlined a two-stage model where the first stage involved basic processing of visual stimuli, including feature extraction and preliminary analysis. The second stage integrated this information with higher-level cognitive functions to achieve perception and understanding. This model aided in explaining phenomena such as object recognition and spatial awareness, providing a framework for developing computer vision algorithms that emulate human visual processing, thereby enhancing accuracy and efficiency in visual tasks.

Jin et al. (2014) [20] proposed a method for privacy-preserving face detection using a variation of the Viola-Jones algorithm. Their work focused on random base image representation, transforming images to obscure individual identities while allowing face detection. Balancing privacy and accuracy, they presented various transformation techniques and their impacts on detection performance. They discussed the security implications, emphasizing how the method protected

sensitive information from misuse while enabling effective surveillance. Experimental results demonstrated the efficacy of the privacy-preserving method compared to the standard approach, detailing the trade-offs involved.

Shamia and Chandy (2017) [21] conducted a detailed analysis of the Viola-Jones face detection algorithm using the LDHF (Large Diverse High-Fidelity) dataset, which included a wide range of images with varying lighting, expressions, occlusions, and backgrounds. Their evaluation assessed performance using metrics like detection rate, false positives, and processing time, and compared Viola-Jones with other face detection algorithms on the same dataset to highlight its relative strengths and weaknesses. The findings provided insights into the efficiency of Viola-Jones in handling real-world situations and identified areas where it could be improved.

Raya et al. (2017) [22] discussed implementing the Viola-Jones face detection method on an embedded system for CCTV cameras. The authors addressed embedded system constraints, such as limited processing power and memory, which required efficient algorithms. Real-time processing was essential for surveillance applications. They applied optimization techniques, including code optimization, parallel processing, and hardware acceleration, to enhance performance. Practical tests on CCTV setups demonstrated the system's effectiveness, highlighting detection accuracy and response times.

Alyushin and Lyubshov (2018) [23] proposed an enhanced Viola-Jones algorithm for face detection in the long-wave infrared (IR) spectrum, which can be useful in low-light conditions. Their work involved adapting Haar-like features to suit the thermal characteristics of IR images, fine-tuning algorithm parameters such as window size, scale factor, and minimum neighbors for IR data, and using an IR-specific dataset for training and evaluation to highlight the differences from visible spectrum data. The paper demonstrated improved performance in terms of detection accuracy and robustness under various IR conditions.

Huang et al. (2019) [24] explored various strategies to optimize the Viola-Jones face detection algorithm, a well-known method for real-time object detec-

tion. They focused on enhancing computational efficiency and detection accuracy by adjusting Haar-like features to better capture facial characteristics, tweaking the AdaBoost algorithm to reduce computational overhead while maintaining strong classifier performance, and improving the cascade structure to reject negative samples more efficiently, thus speeding up the detection process. The paper presented experiments that compared their proposed model with the original Viola-Jones method, showing improvements in detection rates and processing time.

Pairo, Loncomilla, and Solar (2019) [25] proposed facial parts detection using the Viola-Jones algorithm, focusing on applying the algorithm to detect specific facial features (eyes, nose, mouth) instead of just the whole face. They discussed the methodology for segmenting facial parts and outlined modifications to improve detection precision in detail. The paper presented experiments conducted under various conditions (lighting, orientation), demonstrating significant improvements in detection rates. It emphasized the algorithm's usefulness for applications requiring precise facial feature recognition, such as biometrics, emotion detection, and facial expression analysis.

2.2.3 Other Modern Face Detection techniques

Yang et al. (2014) [9] introduced a method for multi-view face detection using aggregate channel features (ACF). By integrating multiple image channels, such as gradient magnitude and orientation, ACF enhanced robustness across different face poses and angles. This method proved particularly effective in complex environments, making it suitable for diverse applications, including surveillance and security.

Derhalli et al. (2015) [26] presented an enhanced face detection technique that integrated boosting algorithms with histogram normalization. The approach aimed to improve the accuracy and efficiency of face detection systems by addressing issues such as varying lighting conditions and facial expressions. The study demonstrated the robust framework provided by these integrated tech-

niques through experimental results.

Putro et al. (2015) [27] focused on developing adult image classifiers using the Viola-Jones face detection method. The research highlighted the adaptation of the algorithm for identifying and classifying adult content images. It discussed the challenges and solutions associated with implementing face detection algorithms in content moderation systems, demonstrating the versatility of the Viola-Jones algorithm in various applications.

Dasan et al. (2015) [28] combined the traditional Viola-Jones method with neural networks to enhance face detection accuracy. By leveraging the strengths of both techniques, the hybrid approach improved detection rates and reduced false positives. The study addressed the limitations of the Viola-Jones algorithm, such as its dependency on predefined features and sensitivity to occlusions, by integrating neural networks into the detection process.

Illumination Invariant Face Detection Using the Viola-Jones Algorithm, introduced by Nehru et al. (2017) [29], addressed the challenge of detecting faces under varying lighting conditions, which can significantly impact accuracy. They proposed enhancements to the Viola-Jones algorithm to improve robustness against illumination changes. The method integrated preprocessing techniques, such as histogram equalization and adaptive gamma correction, to normalize lighting before applying face detection. The enhancements achieved improved detection accuracy, especially in uneven or poor lighting environments, which is crucial for real-world applications like security surveillance and smartphone facial recognition, where lighting cannot be controlled.

Ranftl et al. (2017) [30] introduced a real-time AdaBoost cascade face tracker based on likelihood maps and optical flow, presenting an advanced face tracking system that combined Viola-Jones, AdaBoost, likelihood maps, and optical flow. Designed for real-time operation, the system provided robust and accurate tracking in video sequences. It integrated AdaBoost for improved accuracy, likelihood maps for probabilistic face location estimation, and optical flow for tracking detected faces across video frames. The system maintained stable face tracking even

under challenging conditions, such as rapid movements and occlusions, making it relevant for video surveillance and human-computer interaction systems that require real-time responsiveness.

2.2.4 Related Object Detection Schemes

Redmon et al. (2018) [31] introduced a newer version of the popular object detection framework YOLO (You Only Look Once), called YOLOv3. This version featured a deeper network architecture and improved feature extraction techniques, which not only maintained high detection accuracy but also kept the processing in real-time. As a result, YOLOv3 proved highly effective for applications requiring both precision and efficiency, such as face detection, autonomous driving, and security systems.

Rungruangbaiyok et al. (2019) [32] addressed the challenge of static foreground elements in background subtraction for moving object detection. Their paper introduced a novel probabilistic approach to remove static foreground objects from the background model. This method enhanced the precision of background subtraction and improved the detection of moving objects, which is crucial for high-fidelity applications like surveillance and traffic monitoring.

Wang et al. (2021) [33] enhanced object detection with the CIoU loss function, which improved bounding box regression precision. By considering the overlap area, center point distance, and aspect ratio of predicted and actual boxes, CIoU provided significant advantages for face detection, where precise localization is crucial.

2.2.5 Face Tracking Methods

Bolme et al. (2010) [34] introduced adaptive correlation filters for real-time visual object tracking, capable of adjusting to changes in the appearance of tracked objects. Their primary aim was to develop a robust object tracking system that could handle variations in object appearance due to lighting changes, occlusions, and deformations. The system employed adaptive correlation filters that dynam-

ically updated the tracking model based on recent observations. By utilizing a correlation filter that adapted to visual changes, it maintained high tracking accuracy. The technique demonstrated significant improvements in tracking accuracy and robustness compared to static filter approaches, proving applicable in fields such as video surveillance, autonomous vehicles, and augmented reality.

Suleiman et al. (2014) [35] proposed an energy-efficient hardware architecture for real-time object detection in HD videos. Their system utilized Histogram of Oriented Gradients (HOG) features for object recognition and a Support Vector Machine (SVM) for classification, enabling the processing of 1080 HD video at 60 frames per second (fps) with multi-scale support. This design addressed the challenge of balancing power consumption with high-speed processing, making it ideal for applications in surveillance and video analytics. Further exploration could expand this architecture to incorporate other detection algorithms and more complex vision tasks.

Ranftl et al. (2017) [30] introduced a real-time AdaBoost cascade face tracker based on likelihood maps and optical flow. This advanced face tracking system combined the Viola-Jones algorithm, AdaBoost, likelihood maps, and optical flow, and was designed for real-time operation, providing robust and accurate tracking in video sequences. The integration of AdaBoost improved accuracy, while likelihood maps facilitated probabilistic face location estimation. Additionally, optical flow was employed to track detected faces across video frames, ensuring stable face tracking even under challenging conditions such as rapid movements and occlusions. This system proved relevant for video surveillance and human-computer interaction applications requiring real-time responsiveness.

Barquero et al. (2021) [36] proposed a rank-based verification approach aimed at improving the accuracy and robustness of long-term face tracking in crowded environments. Their work addressed challenges in face tracking within such scenes, including occlusions, changes in appearance, and interactions. The solution enhanced face tracking by prioritizing and verifying the most probable face matches over time. By integrating ranking mechanisms into the tracking

process, the method continuously updated and verified tracked faces against a database. The paper reported significant improvements in tracking performance in complex, crowded environments, making it suitable for applications like public surveillance and addressing the limitations of traditional face tracking methods in dynamic, densely populated settings.

Shen et al. (2022) [37] explored the application of knowledge distillation in Siamese networks for visual object tracking. Their method employed a teacher-student framework, wherein a complex teacher network trained a more efficient student network, achieving high tracking performance while reducing computational demands. This innovation is particularly valuable for real-time applications that require both accuracy and efficiency, contributing to more reliable face detection and tracking systems in dynamic environments.

2.2.6 Real-Time Processing and Computational Efficiency

Acasandrei et al. (2013) [38] present a hardware accelerator designed for the Viola-Jones face detection algorithm, implemented using the AMBA bus protocol. Their study emphasizes enhancing computational efficiency by translating the algorithm into a hardware format. The paper includes detailed technical specifications and performance benchmarks, showcasing significant speed and efficiency improvements achieved through this hardware-based solution.

Wai et al. (2015) [39] investigate the GPU acceleration of the Viola-Jones face detection algorithm. They focus on adapting the traditionally CPU-based algorithm to harness the parallel processing capabilities of GPUs, resulting in substantial reductions in processing time. Their approach achieves significant improvements in speed by parallelizing both the feature extraction and classification stages, making real-time face detection feasible even for high-resolution images and video streams. The study also addresses the technical challenges and solutions involved in implementing the algorithm on GPU hardware.

Wang et al. (2018) [40] explore the parallelization and optimization of the Viola-Jones face detection algorithm using OpenCL. Their study aims to enhance

the performance of face detection systems through the use of parallel computing techniques. They demonstrate significant improvements in processing speed and efficiency, providing valuable insights into the potential of parallel computing for real-time face detection applications.

Yu and Tao (2019) [41] and Zeng et al. (2019) [42] emphasize the crucial need for computational efficiency in face detection systems. Yu and Tao introduce an "anchor cascade" method that employs a progressive filtering approach. This technique refines detected faces while swiftly discarding non-facial regions during the initial detection stages, resulting in faster detection times. Meanwhile, Zeng et al. present a "pyramid network" architecture, which conducts image analysis at multiple resolutions, allowing for the detection of faces at various scales. Both approaches significantly enhance the speed and efficiency of face detection processes, making them more applicable for real-time scenarios.

Wu et al. (2023) [43] introduce YuNet, a lightweight face detection model that effectively balances model size with computational efficiency. YuNet achieves detection times in the millisecond range without compromising accuracy, positioning it as an ideal solution for real-time applications on resource-constrained platforms such as mobile devices and embedded systems. This development represents a notable advancement in high-performance face detection technology.

2.2.7 Human-Computer Interaction and Gesture Control

Lahiani et al. (2016) [44] propose a hand pose estimation system based on the Viola-Jones algorithm tailored for Android devices. This work extends the application of the Viola-Jones algorithm beyond face detection to accurately estimate hand positions and recognize gestures in real time using mobile device cameras. The authors adapt feature selection specifically for hand shapes and movements, achieving effective detection of various hand poses and gestures. This capability is essential for gesture-based user interfaces, highlighting the algorithm's versatility and potential integration into mobile applications, particularly in virtual and augmented reality interfaces.

Chib et al. (2023) [45] investigate the use of the VGG19 neural network for multimodal learning in surveillance applications, aiming to handle multiple data types to enhance monitoring capabilities. Their research enhances surveillance systems by integrating multimodal data, such as images and audio, through a calibrated VGG19 architecture. The methodology involves fine-tuning the network's layers and parameters to optimize performance with multimodal inputs, leading to significant improvements in surveillance accuracy and robustness. This work demonstrates how deep learning techniques can be leveraged to create more adaptable and precise surveillance systems, thereby providing a comprehensive monitoring solution.

2.2.8 Deep Learning Techniques and Facial Recognition

Zhang et al. (2016) [46] introduce the multitask cascaded convolutional network (MTCNN), which combines face detection and alignment into a cohesive framework. This network employs a three-stage process that progressively refines both tasks, significantly enhancing accuracy and robustness, especially in challenging scenarios such as varying lighting conditions and occlusions. The MTCNN's unified approach is vital for real-world applications where precise face detection and landmark localization are required.

Li et al. (2016) [47] propose an innovative method for face detection that incorporates depth information along with traditional features like Histogram of Oriented Gradients (HOG) and Local Binary Patterns (LBP). This fusion of depth data significantly improves detection accuracy by providing additional spatial context, making it particularly effective in complex scenes where standard 2D methods may fall short.

Similarly, Lu et al. (2022b) [48] explore object segmentation through relational visual data analysis, focusing on the interactions among elements within a scene. This approach could offer advantages for facial analysis in crowded or cluttered environments. Integrating depth information as an optional input stream in the proposed model or investigating methods to infer depth from regular RGB

video are promising strategies for enhancing robustness in challenging conditions.

Joseph et al. (2017) [49] examine the integration of Histogram of Oriented Gradients (HOG) features with Support Vector Machines (SVM) for effective object tracking. The proposed technique combines HOG for feature extraction with SVM for classification, enabling accurate tracking of objects within video sequences. HOG features offer robust representations of object appearance, while the SVM classifier effectively distinguishes the target object from other elements in the scene. This combined approach demonstrates high tracking accuracy and robustness, successfully addressing challenges such as changes in object appearance and occlusions. This advancement is particularly relevant for real-time object tracking systems utilized in surveillance and automation.

Zeng et al. (2018) [50] investigate facial expression recognition through the use of deep sparse autoencoders. Their methodology focuses on learning sparse representations of facial features, leading to improved accuracy in facial expression classification. This technique is particularly beneficial for applications in human-computer interaction and emotional analysis, contributing to enhanced precision in recognizing and interpreting facial expressions.

Yu and Tao (2019) [41] and Zeng et al. (2019) [42] highlight the critical importance of computational efficiency in face detection systems. Their work introduces techniques designed to optimize both execution speed and resource utilization. Yu and Tao propose an "anchor cascade" method, which is a progressive filtering approach that refines detected faces while quickly discarding non-facial regions in the initial stages, resulting in faster detection times. Zeng et al. introduce a "pyramid network" architecture that performs image analysis at multiple resolutions, allowing for the detection of faces at varying scales. Both methods significantly contribute to enhancing the speed and efficiency of face detection processes, making them more suitable for real-time applications.

Lu et al. (2020) [51], (2019) [52], and (2022a) [53] introduce substantial advancements in video object segmentation through the use of episodic graph memory networks and co-attention Siamese networks. These innovations offer superior

performance in maintaining temporal coherence across video frames, which is essential for long-term video analysis. This functionality is particularly relevant for facial tracking applications within this framework. Notably, the zero-shot video object segmentation capability developed by Lu et al. (2022a) expands the range of potential applications by enabling the segmentation of previously unseen objects, making it especially useful in real-world conditions where individuals may wear accessories like hats or glasses. By integrating or adapting these segmentation techniques, the framework could enhance facial tracking performance, particularly in extended video sequences or scenarios involving partial occlusions.

Goel et al. (2022) [54] explore the role of artificial neural networks (ANNs) and machine learning in processing spatial information, emphasizing their effectiveness across various tasks. The research investigates how ANNs and machine learning techniques are employed to analyze and interpret spatial data, discussing several models and neural network architectures specifically designed for this purpose. The study highlights applications in geographic information systems (GIS), urban planning, and environmental monitoring. The findings demonstrate that machine learning significantly improves both the accuracy and efficiency of spatial data analysis, underscoring the transformative impact of these technologies in fields that rely on spatial information.

This review integrates recent research efforts from Meijering (2002), Park et al. (2011), Li et al. (2016), Marcomini & Cunha (2018), Pairo et al. (2019), and Lu et al. (2020, 2019, 2022a, 2022b), among others, shedding light on advancements in facial detection, tracking, video object segmentation, and related techniques. Although the primary focus is on facial analysis, the exploration of these broader areas provides valuable insights and potential synergies for developing the proposed face detection framework. By incorporating the insights gleaned from this expansive literature review, a robust and efficient face detection framework tailored for high-quality multimedia content analysis is targeted for development. The comparative analysis conducted through this review not only informs the

design of a conventional facial detection framework but also highlights potential areas for future research that could further enhance its capabilities.

2.3 *Performance Evaluation Parameters*

State-of-the-Art and upcoming algorithms and models are evaluated on standard benchmark datasets based on a variety of performance measure metrics and parameters. Face detection and tracking systems are mostly evalulated on the following parameters:

Accuracy: Accuracy is all about how well the system can correctly identify faces and non-faces. It's a measure of overall correctness [55], calculated as:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

Here, TP stands for true positives (faces correctly detected), TN for true negatives (non-faces correctly identified), FP for false positives (incorrect face detections), and FN for false negatives (missed faces). It gives us a good general idea of how well the system is performing.

Precision: Precision tells us how accurate the system is when it says something is a face. It's calculated by [56]:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

If the system has high precision, it means that most of the faces it finds are actually faces, which is crucial for reducing false alarms.

Recall: Also known as sensitivity, recall is about how good the system is at finding all the faces in an image or video [56]. It's calculated with the given formula:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

High recall means the system is catching most of the faces, even if it sometimes makes a few mistakes.

F1 Score: The F1 Score balances precision and recall into a single number. It's especially useful when we want a good mix of both for assessing an algorithm [56]. It is calculated by:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

This score is a handy way to get a quick sense of the overall performance of the system.

Frame Rate: Frame rate refers to how many frames the system can process per second, measured in frames per second (FPS) [57]. A higher frame rate means smoother face tracking, which is key for real-time applications where everything needs to happen quickly.

Execution Time (CPU Clock Cycle): Execution time is the amount of time it takes for the system to do its job, measured in CPU clock cycles [58]. Faster execution times are important, especially for real-time tasks, because they ensure the system can handle a lot of data without slowing down.

Computational Efficiency: Computational efficiency is about how well the system uses its resources, like processing power and memory, to get the job done [57]. It's mainly about getting good results without using a lot of the computer's power.

Detection Rate: The detection rate tells us how many actual faces the system correctly identifies [59]. It's calculated as follows:

$$\text{Detection Rate} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

This metric helps us understand how reliable the system is at finding faces.

True Positive Rate (Hit Rate): The true positive rate, also called the hit rate, measures how often the system correctly finds real faces [60]. The formula

is:

$$\text{True Positive Rate} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

A high true positive rate is crucial, especially in systems where missing a face could lead to big problems, like in security systems.

False Positive Rate: The false positive rate shows how often the system mistakes something that isn't a face for a face [60]. It's calculated as:

$$\text{False Positive Rate} = \frac{\text{FP}}{\text{FP} + \text{TN}}$$

Reducing the false positive rate makes the system more reliable by decreasing wrong face detections.

False Negative Rate: The false negative rate tells us how often the system misses actual faces [60]. Below is the formula for how it's calculated:

$$\text{False Negative Rate} = \frac{\text{FN}}{\text{TP} + \text{FN}}$$

A lower false negative rate is better because it means the system isn't missing many faces.

16

False Alarm Rate: The false alarm rate measures how often the system wrongly detects faces when there aren't any [28]. It's calculated by:

$$\text{False Alarm Rate} = \frac{\text{FP}}{\text{Total Detections}}$$

Lowering this rate helps make sure the system doesn't give too many false warnings.

Intersection over Union (IoU): IoU evaluates how well the system's predicted face box matches the actual face box [61]. It's calculated like this:

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

Higher IoU values mean the system is doing a good job at pinpointing exactly where the faces are.

Area Under Receiver Operating Characteristic (AUROC): The AUROC measures how well the system can tell the difference between faces and non-faces at different threshold settings [60]. A higher AUROC value means the system is better at identifying faces while keeping mistakes low.

Error in Yaw: Yaw error looks at how accurately the system detects the horizontal angle of a face [62]. It's the difference between the detected yaw and the actual yaw:

$$\text{Yaw Error} = |\text{Detected Yaw} - \text{Actual Yaw}|$$

Minimizing yaw error helps ensure that the system accurately tracks faces even when they're turned sideways.

Tilt Error: Tilt error measures how accurately the system detects the vertical angle of a face [62]. It's calculated by:

$$\text{Tilt Error} = |\text{Detected Tilt} - \text{Actual Tilt}|$$

Lower tilt error means the system is better at tracking faces no matter how they're tilted.

Confidence Score: The confidence score tells us how likely it is that a detected face is actually a face [63]. This score ranges from 0 to 1, with higher scores meaning more certainty.

Pyramid Factor: Pyramid factor refers to the different scaling levels used in the system to detect faces of various sizes [64]. Adjusting the pyramid factor properly helps the system detect faces at different distances.

Pixel Representation: Pixel representation is about how the system processes image data to find faces [65]. The type of pixel representation (like grayscale or color) can affect how well the system identifies facial features.

Probability Score: The probability score indicates how likely it is that a particular area in the image contains a face [66]. This score usually ranges from 0 to 1, with higher numbers meaning the system is more confident that it has found a face.

The following table outlines the performance metrics used in evaluation of the researches that were studied as part of the literature review:

ID	Accuracy	Precision	Recall	F1 Score	Frame rate	Execution time\CPU Clock cycle	Computational efficiency	Detection Rate	True Positive Rate (Hit rate)	False Positive Rate	False Negative Rate	False alarm rate	IOU	AUROC	Error Yaw	Tilt	Confidence	Pyramid factor	Pixel Representation	Probability score
[67]								✓												
[68]								✓												
[69]					✓			✓				✓								
[70]						✓		✓												
[71]					✓		✓													
[72]	✓					✓				✓					✓					
[73]					✓										✓					
[74]	✓					✓										✓				
[75]						✓		✓				✓								
[76]								✓		✓										
[77]								✓												
[78]	✓																			
[79]	✓				✓															
[80]					✓			✓		✓										
[81]														✓						

ID	Accuracy	Precision	Recall	F1 Score	Frame rate	Execution time\CPU Clock cycle	Computational efficiency	Detection Rate	True Positive Rate (Hit rate)	False Positive Rate	False Negative Rate	False alarm rate	IOU	AUROC	Error Yaw	Tilt	Confidence	Pyramid factor	Pixel Representation	Probability score
[82]	✓					✓														
[83]		✓	✓																	
[84]					✓			✓		✓										
[85]						✓										✓				
[86]	✓																			
[87]															✓					
[88]	✓																			
[89]						✓														
[90]																		✓		
[91]								✓				✓								
[92]																			✓	
[93]	✓					✓			✓		✓									
[94]																✓				
[95]	✓																			
[96]						✓										✓				
[97]						✓		✓												
[98]															✓					
[99]								✓												
[100]	✓																			
[101]																		✓		
[102]								✓				✓								
[103]	✓																			

ID	Accuracy	Precision	Recall	F1 Score	Frame rate	Execution time\CPU Clock cycle	Computational efficiency	Detection Rate	True Positive Rate (Hit rate)	False Positive Rate	False Negative Rate	False alarm rate	IOU	AUROC	Error Yaw	Tilt	Confidence	Pyramid factor	Pixel Representation	Probability score
[104]						✓														
[105]	✓					✓														
[106]					✓			✓		✓										
[107]					✓			✓												
[108]									✓											
[109]	✓																			
[110]	✓																			
[111]					✓															
[112]	✓																			
[113]								✓				✓								
[114]	✓				✓															
[115]									✓	✓										
[116]																		✓		
[117]																			✓	
[118]								✓												
[119]	✓																			
[120]								✓												
[121]						✓														
[122]								✓												
[123]	✓																			
[124]								✓							✓					
[125]		✓	✓										✓							

ID	Accuracy	Precision	Recall	F1 Score	Frame rate	Execution time\CPU Clock cycle	Computational efficiency	Detection Rate	True Positive Rate (Hit rate)	False Positive Rate	False Negative Rate	False alarm rate	IOU	AUROC	Error Yaw	Tilt	Confidence	Pyramid factor	Pixel Representation	Probability score
[126]														✓						
[127]			✓																	
[128]									✓	✓										
[129]						✓		✓				✓								
[130]	✓				✓	✓														
[131]		✓	✓	✓																
[132]	✓								✓	✓	✓									
[133]		✓	✓	✓					✓	✓	✓									
[134]	✓													✓						
[135]		✓	✓																	
[136]	✓					✓														
[137]		✓																		
[138]		✓																		
[139]	✓																			
[140]	✓	✓	✓	✓		✓														
[141]	✓																			
[142]	✓					✓														
[143]	✓																			
[144]							✓	✓												
[145]								✓						✓						
[146]		✓																		
[147]						✓														

ID	Accuracy	Precision	Recall	F1 Score	Frame rate	Execution time\CPU Clock cycle	Computational efficiency	Detection Rate	True Positive Rate (Hit rate)	False Positive Rate	False Negative Rate	False alarm rate	IOU	AUROC	Error Yaw	Tilt	Confidence	Pyramid factor	Pixel Representation	Probability score
[148]	✓																			
[149]						✓		✓												
[150]								✓												
[151]	✓					✓														
[152]	✓	✓	✓	✓			✓													
[153]		✓	✓																	
[154]						✓														
[155]	✓					✓														
[156]								✓												
[157]						✓														
[158]						✓														
[159]								✓												
[160]						✓			✓	✓										
[161]														✓						
[162]	✓					✓														
[163]		✓	✓	✓																
[164]	✓	✓	✓							✓	✓									
[165]	✓																			
[166]	✓							✓				✓								
[167]	✓																			
[168]						✓		✓		✓										
[169]	✓						✓													

ID	Accuracy	Precision	Recall	F1 Score	Frame rate	Execution time\CPU Clock cycle	Computational efficiency	Detection Rate	True Positive Rate (Hit rate)	False Positive Rate	False Negative Rate	False alarm rate	IOU	AUROC	Error Yaw	Tilt	Confidence	Pyramid factor	Pixel Representation	Probability score
[170]	✓					✓														
[171]								✓												
[172]						✓		✓				✓								
[173]	✓						✓	✓												
[174]					✓	✓														
[175]						✓	✓													
[176]					✓															
[177]						✓	✓													
[178]	✓					✓	✓													
[179]																	✓			
[180]					✓					✓	✓									
[181]		✓	✓						✓	✓										
[182]		✓	✓	✓					✓	✓	✓									
[183]	✓					✓														
[184]						✓														
[185]	✓																			
[186]	✓	✓	✓	✓			✓													
[187]	✓							✓												
[188]									✓											
[189]	✓					✓														
[190]	✓					✓														
[191]		✓	✓	✓																

ID	Accuracy	Precision	Recall	F1 Score	Frame rate	Execution time\CPU Clock cycle	Computational efficiency	Detection Rate	True Positive Rate (Hit rate)	False Positive Rate	False Negative Rate	False alarm rate	IOU	AUROC	Error Yaw	Tilt	Confidence	Pyramid factor	Pixel Representation	Probability score
[192]									✓	✓	✓									
[193]	✓					✓														
[194]	✓																			
[195]								✓		✓										
[196]								✓												✓
[197]	✓	✓								✓	✓									
[198]	✓							✓												
[199]								✓												
[200]																	✓			
[201]	✓					✓		✓												
[202]	✓	✓																		
[203]	✓					✓														
[204]				✓				✓		✓	✓									
[205]									✓	✓	✓									
[206]		✓	✓	✓																
[207]								✓												
[208]						✓														
[209]	✓																			
[210]	✓						✓	✓												
[211]	✓																			
[212]	✓													✓						
[213]								✓												

ID	Accuracy	Precision	Recall	F1 Score	Frame rate	Execution time\CPU Clock cycle	Computational efficiency	Detection Rate	True Positive Rate (Hit rate)	False Positive Rate	False Negative Rate	False alarm rate	IOU	AUROC	Error Yaw	Tilt	Confidence	Pyramid factor	Pixel Representation	Probability score
[214]								✓												
[215]					✓			✓												
[216]					✓										✓					
[217]						✓		✓												
[218]	✓	✓								✓	✓									
[219]	✓																			
[220]	✓																			
[221]						✓		✓												
[222]								✓		✓										
[223]															✓					
[224]	✓																			
[225]						✓	✓													
[226]		✓	✓							✓										
[227]						✓														
[228]	✓					✓														
[229]								✓												
[230]	✓					✓		✓												
[231]	✓					✓														
[232]	✓																			
[233]	✓					✓														
[234]	✓	✓																		
[235]								✓							✓					

ID	Accuracy	Precision	Recall	F1 Score	Frame rate	Execution time\CPU Clock cycle	Computational efficiency	Detection Rate	True Positive Rate (Hit rate)	False Positive Rate	False Negative Rate	False alarm rate	IOU	AUROC	Error Yaw	Tilt	Confidence	Pyramid factor	Pixel Representation	Probability score
[236]														✓						
[237]	✓							✓				✓								
[238]																✓				
[239]	✓					✓														
[240]									✓	✓										
[241]	✓					✓														
[242]				✓				✓		✓	✓									
[243]									✓	✓	✓									
[244]						✓		✓												
[245]								✓												
[246]	✓																			
[247]								✓												
[248]								✓												
[249]						✓														
[250]						✓		✓												
[251]									✓											
[252]						✓	✓													
[253]								✓												
[254]								✓												
[255]						✓														
[256]	✓																			
[257]								✓												

ID	Accuracy	Precision	Recall	F1 Score	Frame rate	Execution time\CPU Clock cycle	Computational efficiency	Detection Rate	True Positive Rate (Hit rate)	False Positive Rate	False Negative Rate	False alarm rate	IOU	AUROC	Error Yaw	Tilt	Confidence	Pyramid factor	Pixel Representation	Probability score
[258]		✓	✓	✓																
[259]						✓		✓												
[260]	✓																			
[261]	✓	✓	✓	✓			✓													
[262]	✓	✓	✓	✓																
[263]						✓		✓												
[264]															✓					
[265]								✓												
[266]	✓																			
[267]					✓	✓														
[268]					✓															
[269]	✓					✓														
[270]	✓													✓						
[271]								✓				✓								
[272]								✓												
[273]	✓																			
[274]		✓	✓										✓							
[275]			✓																	
[276]									✓	✓										
[277]	✓				✓	✓														
[278]		✓	✓	✓																
[279]	✓													✓						

ID	Accuracy	Precision	Recall	F1 Score	Frame rate	Execution time\CPU Clock cycle	Computational efficiency	Detection Rate	True Positive Rate (Hit rate)	False Positive Rate	False Negative Rate	False alarm rate	IOU	AUROC	Error Yaw	Tilt	Confidence	Pyramid factor	Pixel Representation	Probability score
[280]		✓	✓																	
[281]	✓		✓																	
[282]		✓	✓	✓																
[283]	✓					✓														
[284]		✓																		
[285]		✓																		
[286]	✓																			
[287]	✓	✓	✓	✓		✓														
[288]	✓																			
[289]	✓	✓	✓	✓			✓													
[290]														✓						
[291]									✓											
[292]		✓	✓																	
[293]	✓																			
[294]	✓				✓	✓														
[295]	✓																			
[296]		✓	✓	✓																
[297]														✓						
[298]	✓													✓						
[299]	✓					✓														
[300]														✓						
[301]	✓																			

ID	Accuracy	Precision	Recall	F1 Score	Frame rate	Execution time\CPU Clock cycle	Computational efficiency	Detection Rate	True Positive Rate (Hit rate)	False Positive Rate	False Negative Rate	False alarm rate	IOU	AUROC	Error Yaw	Tilt	Confidence	Pyramid factor	Pixel Representation	Probability score
[302]	✓					✓														
[303]	✓					✓														
[304]	✓	✓	✓	✓		✓														
[305]															✓					
[306]		✓	✓																	
[307]	✓					✓														
[308]	✓							✓												
[309]								✓												
[310]	✓				✓															
Total	102	40	34	21	21	70	15	67	17	27	13	10	2	13	10	4	3	3	2	1

Table 2.1: Evaluation parameters used in various related researches

2.4 *Inferences of the Critical Review*

- Early researches like works done by Tomasi and Kanade (1991), and Viola and Jones (2001) provided a very strong foundation, but modern applications require more advanced approaches that can deal with real-world challenges such as lighting conditions, poses, and occlusions.

- Both traditional and modern approaches were gradually able to provide real-time performance, which is required for many practical applications. Deep learning models like Multitask Cascaded Convolutional Neural Networks (MTCNN) and

YOLO (You Only Look Once) have given promising results in improving accuracy and reliability. However, balancing computational speed with detection precision remains a challenge, especially for devices with limited processing power, such as mobile phones and embedded systems.

- The increasing use of high-resolution video content, along with higher frame rates, also puts pressure on even the most advanced algorithms. While solutions like hardware acceleration and parallel processing have shown promise in speeding things up, more research is needed to ensure these methods work well across different platforms and devices. Real-time face detection systems, particularly those used in data-heavy applications, must keep evolving to meet these growing demands.

- Additionally, face detection plays a key role in areas like human-computer interaction, gesture recognition, and facial expression analysis. This highlights the importance of developing systems that are not only efficient but also adaptive to different user needs, making technology smarter and more responsive.

- After studying and analysing the researches done specifically on face processing, the most widely used parameters have also been identified, that have been used in evaluating the various face processing (detection, tracking, recognition) algorithms. **Accuracy, Execution time** and **Detection rate** were the most commonly used parameters used in evaluating the algorithms, therefore they've been employed in this research as evaluation parameters upon which the proposed work is assessed.

- Therefore, while face detection and tracking technologies have progressed at a good rate, some challenges are yet to be dealt with. Improving the balance between accuracy and computational efficiency—especially for high-resolution and high-frame-rate video—remains a critical area for further development. Future research needs to focus on refining machine learning techniques and optimizing algorithms to better handle the complexities of real-time, high-quality video analysis. As these technologies continue to evolve, they will play an increasingly important role in a wide range of applications, ensuring they can keep up with the growing demand for smarter, faster, and more accurate systems.

Chapter 3

TECHNICAL ANALYSIS OF COMPARED FACE DETECTION ALGORITHMS

This chapter presents an overview of most widely used commercial face detection schemes, models and frameworks. It also presents an overview of identified shortcomings in those researches, and the corresponding objectives that have been worked on for this thesis. The subsequent sections discuss the researches done to assess the performance of the proposed works against them.

3.1 *Traditional Face Detection*

Applications of Face Detection in video streams have made it an integral part in the field of Computer Vision. Traditional face detection algorithms used to take more time to process each frame/image, so they were not useful in development of real-time applications. But the introduction of Viola-Jones algorithm [2] marked a breakthrough in the field because it gave decent-accuracy results in real-time.

Yang et. al [311] wrote a survey which classified traditional face detection into four subsets: Knowledge-based, feature-invariance, template-matching and appearance - based.

- Knowledge-based face detection techniques use the inherent knowledge of what comprises a face, and the relationship between those facial features.
- Feature-invariant techniques use the structural features of a face that remain the same under various environmental factors like occlusion or illumination, and extract those features from an image.

- Template-matching techniques compare the incoming input image with pre-processed, stored standard patterns of faces or facial features.
- Appearance-based techniques learn to detect a face using a set of training data that contains various deviations in appearance of faces. These classified methods worked well on still images to detect faces [312].

Viola-Jones algorithm, Face Detection using Linear Discriminant Analysis (LDA), Principle Component Analysis (PCA), Local Binary Pattern (LBP), SMQT Features and SNOW classifier methods [313], are some of the examples of the traditional face detection algorithms that have been very effective for many varieties of video input streams. However, their average performance is still lower compared to modern and more complex algorithms.

Figure 3.1 shows the flowchart of traditional Face Detection in Color images proposed by Hsu. et al. [1]

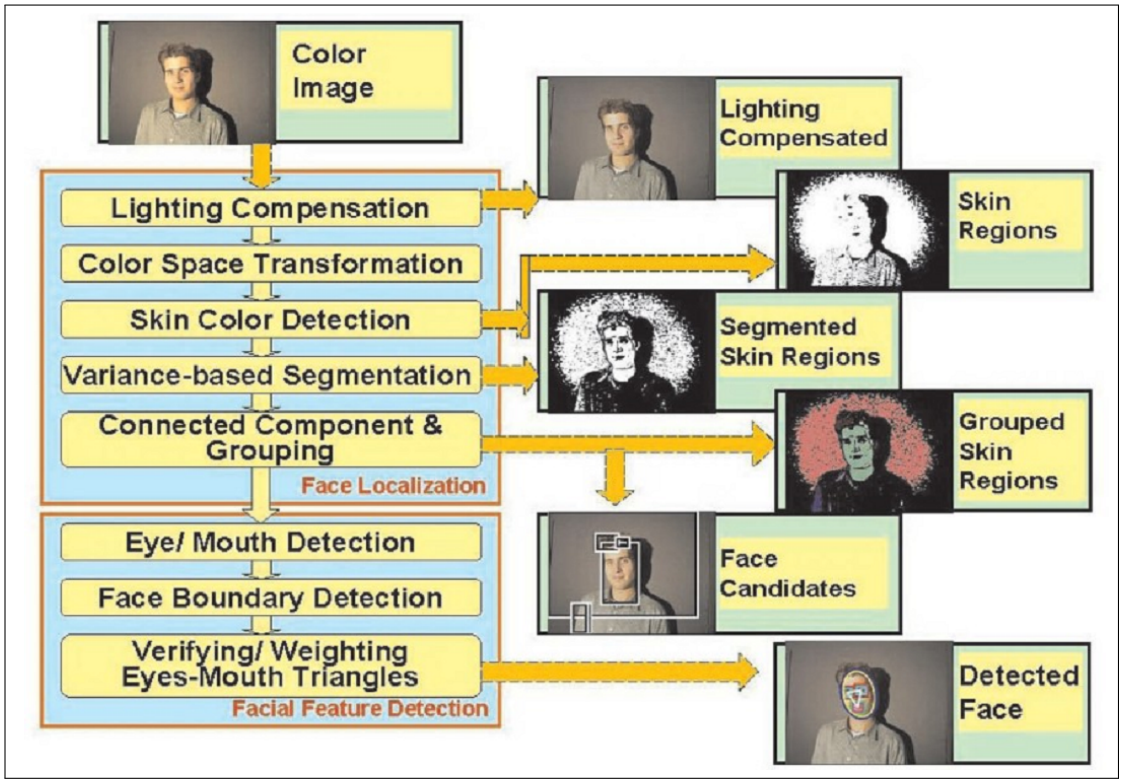


Figure 3.1: The flowchart of the face detection algorithm proposed by Hsu et al. [1]

3.2 *Modern Face Detection*

Modern Face detection algorithms provide high accuracy with real-time results useful for privacy and surveillance applications that use live video feeds. This section discusses the four face detection algorithms that have been used for comparison of the proposed works: FaceNet system, Face Detection using Histogram of Oriented Gradients (HOG) features [314], Face Detection and alignment using Multitask Cascaded Convolutional Neural Networks (MTCNN), and YuNet face detector.

- FaceNet system [315] works by utilizing the concept of Euclidean space. Input face images are directly mapped to a Euclidean space, that measure the degree of similarity to a face, or in case of face recognition, similarity to a particular identity of a face. 0.0 euclidean distance means an identical face, while 4.0 means entirely different person.

- HOG Face Detection uses HOG (Histogram of Oriented Gradients) features [316] to identify specialized objects in a detection window in an image. It does so by identifying and counting orientation of gradients, and by considering both their magnitude and angle in the window. HOG features work well in Computer Vision because they focus on structure and shape of the objects to be detected in an image.

- MTCNN (Multitask Cascaded Convolution Neural Network) [317] is a very fast and accurate face detection algorithm that utilizes three-stage neural networks framework. The input image is resized both upscale and downscale a few times so as to detect faces of different sizes. Then these boxes are fed to the three neural networks - P-net, R-Net, O-Net, in that order. They keep decreasing the amount of false positives after each net's output, until only true positive faces remain in the image.

- YuNet [318] is one of the recent lightweight face detectors that is based on Convolutional Neural Networks. The main components of YuNet detector include a small and efficient feature extractor along with a simple and manageable fusion of pyramid features. It has been commercially available since 2019 and is powerful enough to be used many devices, mainly edge devices. YuNet is commercially

used because it can deliver high detection accuracy with high processing speeds.

There are many state-of-the-art algorithms available for face detection. These algorithms perform poorly on videos with high content density, where data to be processed is more than in real-time videos. Techniques like [319] use a delayed object detection framework with feature-tracking KLT algorithm [320], that projects detection box to the frame being processed. Such frameworks provide better efficiency for even high-quality videos, so they can be a good alternative to be used for the processing of those videos, and their further analysis.

Figure 3.2 shows the working of Face Detection using SVM classifier.

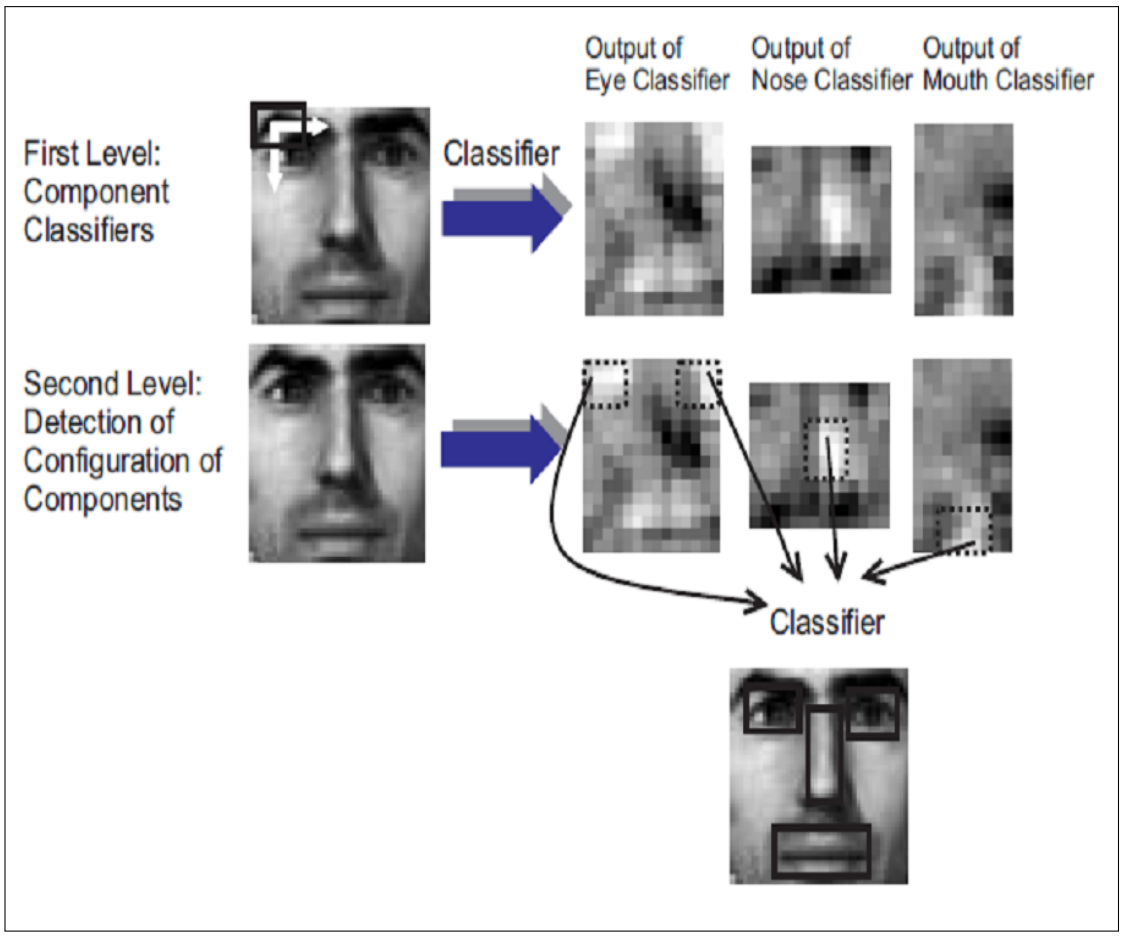


Figure 3.2: The system overview of the component-based SVM classifier using four components

3.3 *Real-time AdaBoost cascade face tracker*

Viola-Jones algorithm [321] is a benchmark face detection algorithm that was introduced in 2004. It achieved fast face detection speeds with decent accuracy of detected faces. It proved to be such a huge leap forward that it is still considered a benchmark against which the efficiency of newer algorithms is measured. Execution performance of the algorithm is appreciable in real-time when compared against other state-of-the-art algorithms. However, its main drawback is that its accuracy is fairly low, at about 60-70% detection of the target object. Modern applications need high accuracy in detection of faces, so the VJ algorithm cannot be used in such settings.

Ranftl [30] tried to tackle the problem of implementing the main face detecting algorithm of Viola-Jones on each and every frame in a video. An optical flow based face tracker was suggested in their research, that used likelihood map to predict where the face would be in the next few frames, after it was detected in the previous frames. Viola-Jones algorithm is essentially a static algorithm that does not incorporate the temporal aspect of a detected face in a video. It also does not pay any significance to near-positive windows, i.e, those intermediate stage windows that pass a significant number of classification cascade levels.

Ranftl tried to use the optical flow between frames for prediction of faces for the next frames. These optical flow calculations were then used to construct a likelihood map using interpolation techniques against time. The significance of the likelihood map was to determine the belonging-ness of a particular pixel as part of a face. It is constructed by incorporating the number of stages of classification cascade a particular window has passed through, even if the resultant window is not a face at the end stage. This modified version of Viola-Jones would be called after every n frames, and used likelihood map interpolated with optical flow calculations for intermediary frames. This algorithm was comparatively faster than traditional Viola-Jones algorithm, and provided better accuracy. The likelihood map takes care of partial or total occlusions between frames, making use of the likelihood map and interpolation.

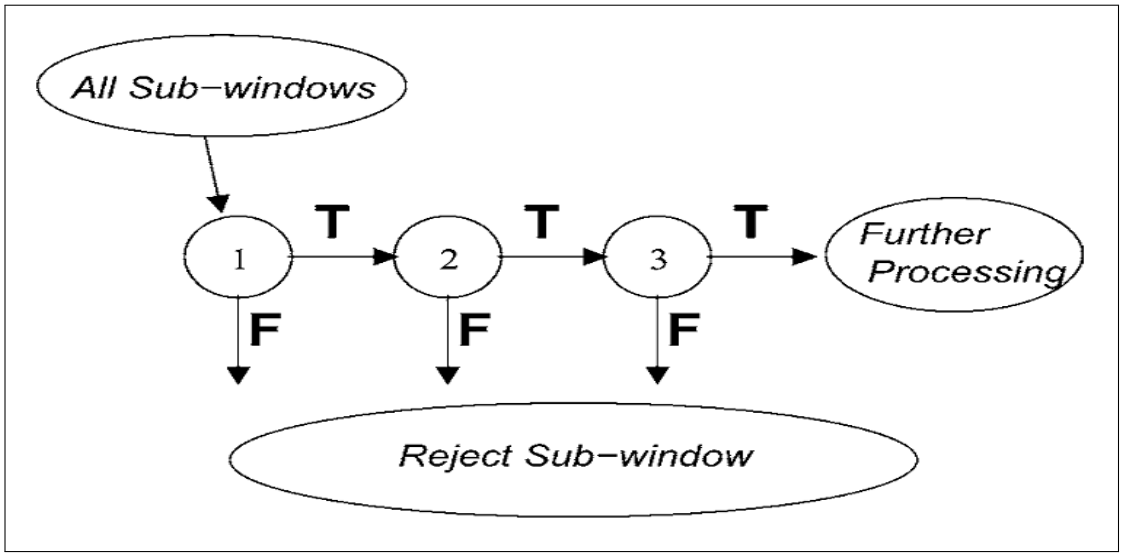


Figure 3.3: Schematic depiction of Detection cascade

3.4 *Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks (MTCNN)*

This algorithm is one of the modern face detecting algorithms [322] [323] that is more accurate than Viola-Jones, while also providing fast performance in real time. Zhang, Zhang and Li [324] developed this framework in 2016, that tried to tackle the challenges of pose variations, illumination variations and occlusion problems. The framework used a separate CNN (Convolutional Neural Network) for each of the three stages involved in it:

1. The first stage results in various potential face windows in the whole frame using a shallow CNN, called Proposal Network (P-net). These potential face windows are treated with Non-Maximum Suppression (NMS) so that all the overlapping windows could be merged.

2. The resultant candidate face windows are refined and reduced in the next stage using a complex CNN, called Refine Network (R-Net). Its task is to reject even more false candidate windows, to calibrate them using Bounding Box Regression, and to conduct NMS again.

3. The final stage (O-net) uses more supervision to reduce and finalize the previous

stage candidate windows into the optimal face windows, and outputs five locational landmarks in them: two eye markers, one nose marker, and two lips-ends markers.

Figure 3.4 shows the pipeline of the MTCNN face detection framework.

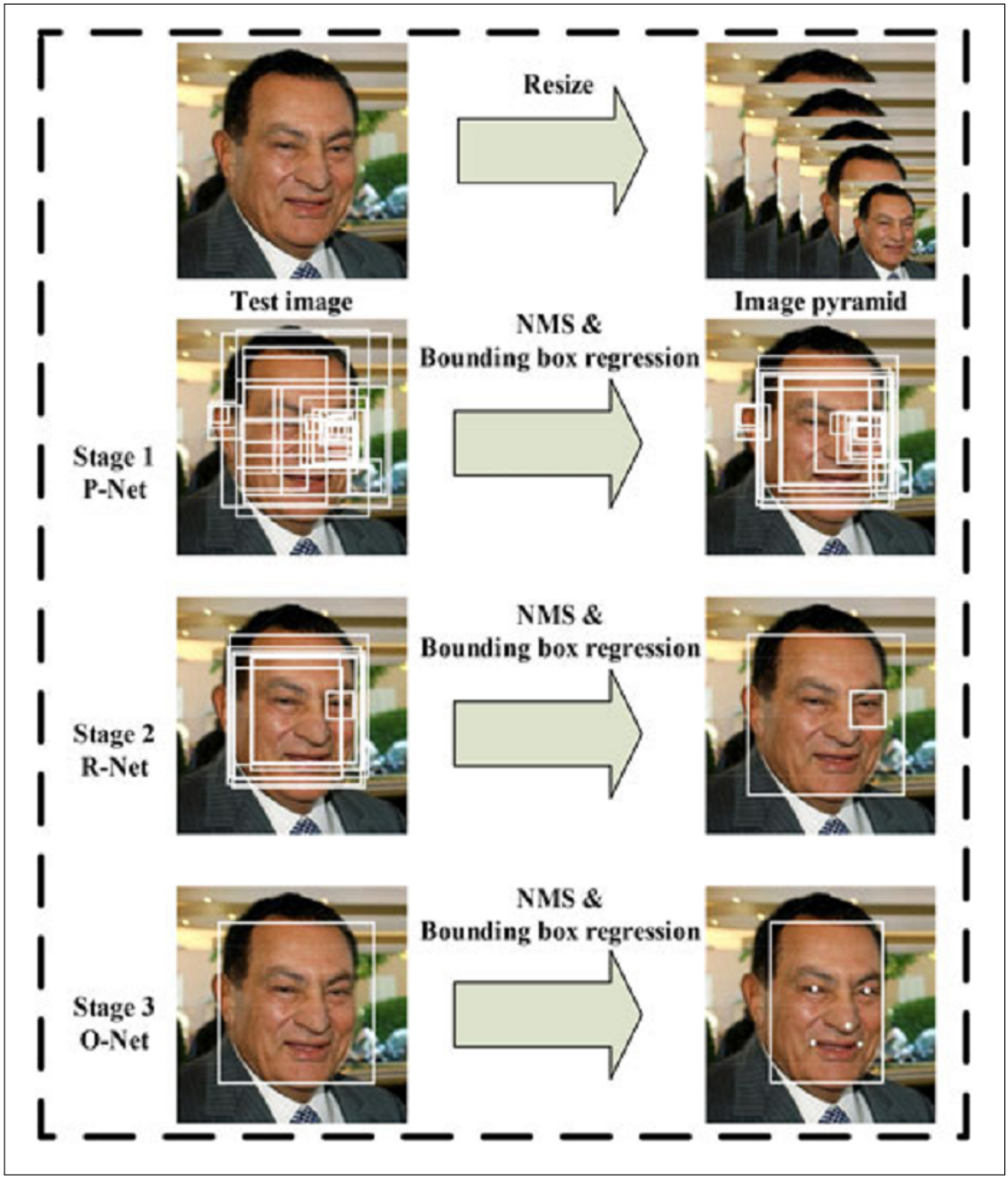


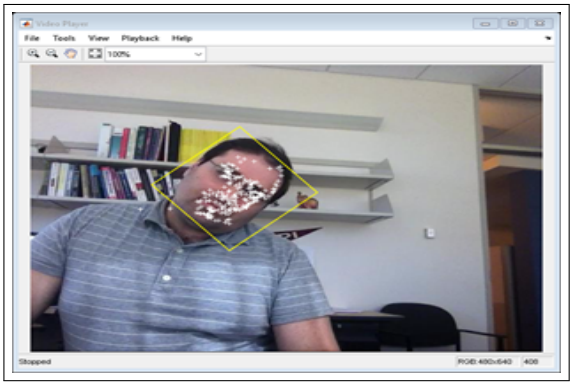
Figure 3.4: Pipeline of MTCNN framework

3.5 *Feature-Points tracking using KLT algorithm*

KLT algorithm [325] is named after the three researchers: Kanade-Lucas-Tomasi, who developed the method of tracking motion features across frames. The method first identifies important feature points to track. Then it identifies fixed-sized feature-point windows, and tries to minimize the sum of squared intensity differences between consecutive frames, (the past frame and the current frame). A window is chosen over individual pixels to track because a single pixel cannot be tracked unless it's intensity has much difference with respect to it's neighboring pixels. Therefore, a window of pixels is chosen that has "sufficient texture" required for feature-point tracking reliably.

Essentially, for each point(s) in the previous frame, the point tracker attempts to find the corresponding point(s) in the current frame. Then the translation, rotation, and scaling between the old points and the new points is estimated. This transformation is applied to the bounding box around the face.

(a) Identifying good features to track



(b) Tracking the identified features

Figure 3.5: KLT feature tracker first identifies the good features to track in a face, and then it tracks those features in the subsequent frame.

3.6 *Conclusions*

After implementing and reviewing these State-Of-The-Art models, we came across a few problems in the existing works:

- Existing face detection algorithms are not very effective in detecting faces under problems such as occlusion, illumination, reflectance, frame speed etc, as compared to the ideal condition face detection framework. Therefore, there is a need for some better algorithms to address these shortcomings.

- The Face detection algorithms that are presently in use have been extended to work on other types of images, such as infrared images, thermal images etc. However, they have not been able to achieve comparable accuracy and speed when compared to the black and white or color image face detection algorithms. So, performance of the algorithms needs to be increased for other types of images types as well.

- Viola Jones method in Face detection is a very tedious and computationally ineffective process. Although the speed of the method has been drastically increased over the recent years, the method can yet be improved, in terms of reduction of Haar features to determine non-faces, and of reduction of stages to determine a face. The basic algorithm can be modified to an extent to address these issues.

- Inclusion of various digital image processing schemes such as edge detection, histogram equalization etc have not made much significant contributions in increasing the performance of the existing face detection algorithms. They can be made more efficient by applying such schemes effectively.

Chapter 4

FACE STUDY DATASETS AND THEIR ANALYSIS

This chapter introduces and analyzes the two datasets used for developing and analysing our research works. Since the efficiency of any Machine Learning model is largely dependent of the data it is trained on, therefore it is very important to understand the datasets used in its development and progress. The datasets used for our work are: 300-VW Dataset, and YouTube Faces Dataset. These datasets are selected for training and analysing our models since these are the most cited freely available datasets used in the field of face processing in Computer Vision.

4.1 *300-Videos-in-the-Wild Dataset*

The 300-VW dataset [326] [327] [328] is a very helpful resource in the field of computer vision, that is specially used in assessing face detection models, facial landmark tracking, and alignment algorithms. It consists of highly deep annotations of the constituting videos which serve as a basis on which various algorithms can be evaluated and developed for a variety of challenging real-world scenarios.

Dataset Overview: The dataset consists of various videos that have been recorded under natural, uncontrolled conditions. The videos include large number of scenarios, with different lighting conditions, facial expressions, head poses, and occlusions, which can be used to test facial landmark tracking algorithms. Different number of videos with varying frame rates and resolutions are provided in the dataset so that it can cover a lot of different video qualities, which is important to assess the performance of

models under scrutiny for as many diverse situations as possible.

Frame Rate (fps)	Number of Videos
50	75
48	75
30	100
25	50
24	75
15	75

Table 4.1: Dataset description based on Frame Rate

Resolution	Number of Videos
320x240	75
640x480	100
1280x720	50
1920x1080	75

Table 4.2: Dataset description based on Video Resolution

4.1.1 Methodology and Experimental Design

10-Fold Cross-Validation: The experiments in the proposed research with the 300-VW dataset are performed through an exhaustive 10-fold cross-validation approach. This approach partitions the dataset into ten parts, such that nine parts are used for training the model, and the remaining one part is used to conduct experiments and tests for judging the algorithms' performance efficiency on unknown inputs. This process is repeated ten times, so that every part is tested on the model. This way, a comprehensive and complete assessment of the algorithms is guaranteed, so that the efficiency of their performance can be reported reliably.

Parameter averaging is also done to ensure reliable and dependable results. The experiments compute the average values of the defined parameters, which ensures low

biases that could emerge from dataset variations.

4.1.2 Classification of Videos:

Low-Quality Videos: Videos under this classification are assumed to be characterized by frame rates of 30 fps or less and resolutions of 640x480 or lower, to conduct experiments for the proposed research. These videos usually face challenges such as motion blur and low visual details. Advanced preprocessing of these videos and other analysis techniques are essential to achieve more accurate facial landmark detection and tracking.

High-Quality Videos: High-quality videos are defined by frame rates exceeding 30 fps and resolutions greater than 640x480 for the experiments. These videos are characterized by superior clarity and intricate details,thereby providing enhanced quality supports more precise and reliable facial analysis.

The 300-VW dataset acts as a benchmark dataset for a large number of applications and research fields, offering a rich and variety of face scenario variations that supports the development of robust and effective facial analysis technologies.

4.1.3 Applications:

Research and Development:

Various researches utilize the 300-VW dataset to test their facial landmark tracking algorithms in exhaustive almost-real scenarios. It covers a lot of different lighting conditions, facial expressions, and head poses that ensure a in-depth assessment of the performance of those algorithms on different metrics.

The dataset is beneficial repository for training machine learning models, providing a broad range of annotated data that represents diverse facial appearances and environmental conditions. This variation in videos in the dataset helps in developing models that are not only accurate but also adaptable to different situations.

Technological Advancements:

Improvements in facial landmarks tracking is essential in development of advanced facial recognition systems, which use the field of face detection and tracking as prerequisite. Facial recognition systems are widely used in various applications in security,

authentication, and surveillance.

The dataset can also be used to develop systems that can detect and analyze human facial expressions and emotions. These type of systems are very valuable in improving human life quality with their involvement in mental health monitoring, customer satisfaction analysis, and user experience research.

The utility of applications involving Human-Computer Interaction can also be increased by utilising better facial tracking algorithms. Virtual assistants, augmented reality (AR) systems, and other interactive technologies, that use these types of algorithms, can enhance their viability by efficient usage of facial recognition and tracking schemes.

In Conclusion, the 300-VW dataset proves to be a very practical data repository in improvement of the field of facial landmark tracking and recognition. Different quality videos of numerous types with different face alignments of front-facing faces are present in the dataset, that are useful for aspiring developers and researchers in evolving state-of-the-art technologies. This makes the dataset a vital resource for both academic research and industry applications, ensuring that systems are reliable and effective in a wide range of real-world scenarios.

4.2 *YouTube Faces Dataset*

The YouTube Faces (YTF) dataset [329] is a very important resource repository in computer vision, specifically used for testing face verification and recognition algorithms. This dataset is made up of a large collection of videos from YouTube showing faces in many different real-world situations, thereby providing a thorough and challenging collection of faces for developing and testing face detection and recognition models.

Dataset Overview:

The main goal of the YTF dataset is to serve as a solid benchmark for face detection and recognition systems. It offers a wide variety of videos that mimic real-life conditions, allowing for a detailed evaluation of how well face recognition algorithms perform. Collected from YouTube, the dataset includes videos taken in uncontrolled environments, making it highly relevant and practical for real-world uses.

4.2.1 Video Specifications:

The YTF dataset contains 3,425 videos, with each video averaging 181.3 frames.

Individuals Represented: These videos feature 1,595 different people, providing a broad range of facial appearances and conditions.

Annotations: Every video is clearly labeled with the person's identity, ensuring accurate data for training and testing face recognition models.

Resolution	Number of Videos
640x480 pixels (VGA)	3,000
320x240 pixels (QVGA)	425

Table 4.3: Number of Videos by Resolution in the YouTube Faces Dataset

Frame Rate (FPS)	Number of Videos
30 fps	2,600
25 fps	700
15 fps	100

Table 4.4: Number of Videos by Resolution in the YouTube Faces Dataset

4.2.2 Experimental Framework:

For face verification applications, the dataset includes pairs of videos to check for the appearance of the same person in both videos. This double labelling helps in testing the efficiency of face verification algorithms.

10-fold Cross validation: The dataset is divided into 10 parts, each containing different pairs of videos. This division supports cross-validation, which helps in verifying how well the algorithms can generalize to new data.

Lighting Conditions: The videos show faces in different lighting, from bright light to dim conditions. This variety helps test how well the algorithms can keep working accurately when the lighting changes.

Facial Expressions: The dataset also includes a wide range of facial expressions, challenging the algorithms to recognize faces even when they look different.

Head Poses and Movements: The videos capture faces from different angles and with various movements, similar to how people move in real life, making the tests more realistic.

Occlusions: Some videos have faces partly covered by things like glasses or hands, which adds another layer of testing for the face recognition systems.

4.2.3 Applications and Relevance

The YTF dataset is another benchmark resource repository (like 300-VW dataset) in the field of computer vision, that helps in testing and analysis of face detection and recognition algorithms.

Research and Development: Researchers make use of the YTF dataset to carefully test and evaluate face recognition and verification algorithms. The variations in the dataset ensure that these algorithms are able to handle real-world challenges.

Training Robust Models: The wide range of videos in the dataset helps in training models that are strong and work well in different situations, making them more useful in everyday life.

Technological Advancements: Algorithms improved with the YTF dataset make security and authentication systems much more reliable and accurate.

Surveillance and Monitoring: The dataset helps create advanced surveillance systems that can recognize people in different conditions, making them more effective in real-world situations.

Human-Computer Interaction: Improved face recognition technology makes human-computer interaction applications, like virtual assistants and user authentication systems, work better by ensuring accurate and reliable recognition.

In Conclusion, the YouTube Faces dataset is a crucial resource in advancing face recognition technology. Its wide range of challenging videos provides a solid foundation for developing and testing face recognition and verification algorithms. By covering a broad spectrum of real-world conditions, the YTF dataset ensures that algorithms are both accurate and practical for everyday use. This makes the dataset a valuable tool for both academic research and industry, driving the progress of facial recognition technology in many different areas.

4.3 *Conclusions*

This chapter focused on the analysis of two significant datasets, the 300-VW Dataset and the YouTube Faces Dataset, which are widely used in the evaluation of face detection and recognition algorithms. These datasets play a crucial role in developing and validating facial processing models, underscoring the necessity of high-quality data for training machine learning models that are effective in real-world applications.

The 300-VW Dataset is a comprehensive resource consisting of videos captured in uncontrolled environments. It is instrumental for testing facial landmark tracking and alignment algorithms. The dataset's diversity in terms of lighting conditions, facial expressions, head poses, and video quality allows for the development of robust models capable of adapting to various real-world scenarios. The application of 10-fold cross-validation across low- and high-quality videos provided a thorough and systematic evaluation of the algorithms, with findings indicating the importance of advanced preprocessing, particularly for lower-quality videos, to ensure accurate facial landmark detection and tracking.

The YouTube Faces (YTF) Dataset is another critical benchmark for the assessment of face verification and recognition systems. Its wide variety of facial conditions—including different lighting scenarios, head movements, and occlusions—presents real-world challenges that make it an ideal resource for developing reliable and robust face recognition algorithms. The dataset's use of cross-validation ensures a rigorous evaluation of model performance across various conditions, reinforcing the importance of comprehensive testing to improve system reliability in applications such as security, surveillance, and human-computer interaction.

In conclusion, the analysis of these datasets emphasizes their value in advancing facial recognition technologies. Their diversity and real-world applicability ensure that models trained and tested using these datasets are both accurate and versatile. This study highlights the critical role of well-structured and diverse datasets in enhancing the reliability, efficiency, and adaptability of face detection and recognition systems, thus contributing to further advancements in both research and industrial applications.

Chapter 5

OPTIMIZATION LEVER AND ITS IMPACT IN FACE DETECTION

This section provides methodology for the proposed efficient face detector and tracker. The model is designed to use the face detection algorithm in an efficient manner for high definition videos, without compromising it's integrity, while maintaining it's speed at par with live, lower quality videos.

5.1 *Overall Model*

The proposed model comprises of three sub-frameworks, each having its own function, and they work together over a stream of images (videos) to detect and track faces. The flowchart for the overall model of the proposed method is shown in Figure 5.1.

The process begins with the first sub-framework, which takes the initial frame from the input video. The MTCNN face detection algorithm is applied to detect faces in the frame, characterized by five feature points: two eye points, one nose point, and two mouth endpoints. The output of the algorithm is these five feature points, along with a rectangular box around the detected face. This output is then fed into the second sub-framework.

The second sub-framework uses the KLT feature tracking algorithm to track the five feature points across n subsequent frames, where n represents the refresh rate, after which the MTCNN algorithm is applied again for face detection. The value of n is chosen appropriately based on the video, ensuring that the model operates faster while maintaining accuracy.

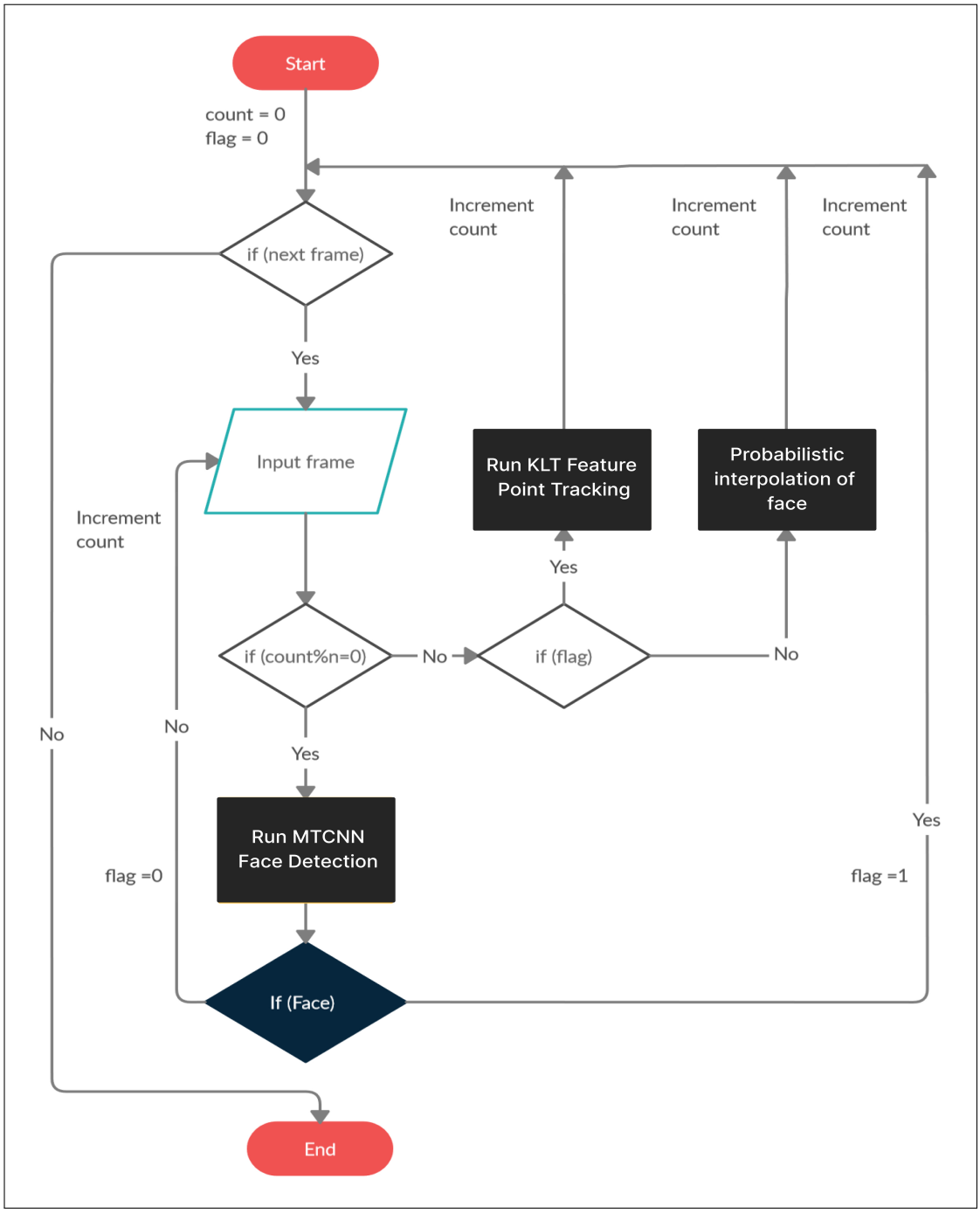


Figure 5.1: Flowchart of the overall model

When an occlusion occurs in the frame, the third sub-framework is called. It predicts the location of the face in the occluded region based on the positions of detected faces in previous frames. This process continues for n frames, after which the MTCNN algorithm is called again to confirm if the face has reappeared.

5.2 *Integrating Face detecting with Feature points tracking*

As discussed earlier, experimental results show that modern algorithms like MTCNN achieve highly accurate results with respectable runtime performance for live video feeds, which are generally of lower quality. However, these algorithms tend to slow down significantly when the video quality increases. Factors such as frame rate, bit depth, pixel aspect ratio, and interlacing contribute to the decline in performance. This work aims to accelerate the speed of the MTCNN algorithm while preserving its high accuracy.

The primary output of the MTCNN face detection algorithm consists of five facial landmark points: two eye points, one nose point, and two mouth-end points. These points are critical for identifying the location of a face in a frame. However, since MTCNN is computationally heavy and time-consuming for high-resolution frames, it is inefficient to apply the algorithm to every frame in the video.

To address this, the KLT feature-tracking algorithm is used. KLT is a fast and efficient method for tracking feature points across frames. While it has been applied in face tracking, it is not entirely reliable on its own because it may identify other objects in the frame as "good features" and track them, thus defeating the purpose of focusing solely on faces. To resolve this issue, we discard the portion of the algorithm that identifies new features to track and instead use only the part that tracks already-detected points. Since the MTCNN algorithm already provides the most important feature points to track, these points are passed to the KLT tracker, which tracks only the facial points across frames.

The main benefits of integrating these two algorithms are:

1. The output of the MTCNN face detection algorithm directly serves as the input for the KLT feature-tracking algorithm.

2. Using the KLT tracker reduces the computational cost significantly, while maintaining the high accuracy of the MTCNN algorithm.

3. By bypassing the feature-identification stage of the KLT algorithm and tracking only five key points, the computational cost is reduced even further.

A potential drawback of relying solely on the KLT feature tracker over many frames is that the properties of the points and their corresponding windows may change over time. As a result, the face may shift to a different location than indicated by the tracker, leading to reduced accuracy. To mitigate this, a variable n is introduced, which acts as a lever between accuracy and speed. The MTCNN algorithm is executed every n frames to refresh the face detection, ensuring accurate tracking over time. The value of n can be adjusted by the programmer. A lower n increases accuracy but reduces speed, while a higher n boosts speed but may slightly compromise accuracy. For optimal performance, the value of n should be greater than 8 to meet the human visual system's requirement of 24 frames per second for perceiving smooth video. The upper limit of n can be set based on the frame rate of the input video.

Table 5.1: Speed comparison of the model against other models, in Frames per Second (FPS), for 300-VW Dataset

Model	**Frame size**	**Speed (in FPS)**
MTCNN	1280 x 720	3.852317191 (i5-8300H)
HOG	1280 x 720	4.904332105 (i5-8300H)
FaceNet	1280 x 720	5.699034533 (i5-8300H)
DPM	1280 x 720	1.698523433 (i5-8300H)
Proposed	1280 x 720	12.12239514 (i5-8300H)

Algorithm 1: Face Detection and Tracking Algorithm

```
count = 0;
flag = 0;
while True do
    if next frame exists then
        Input frame;
        count = count + 1;
        if count % n == 0 then
            frame = Face_Detection(frame);
            if face.exists(frame) then
                flag = 1;
                break;
            end
            else
                flag = 0;
            end
        end
        else
            if flag == 1 then
                track_features = KLT(det_frame);
            end
            else
                track_features = Probabilistic_Interpolation(det_frame);
                flag = 1;
            end
        end
    end
end
return;
```

5.3 *Evaluating Runtime efficiency*

Table 5.1 and Table 5.2 show the speed of the proposed model compared to other models for the chosen video samples. It is clear from Table 5.1 that the proposed model processes more frames per second (FPS) than other models when dealing with higher data per frame.

Figure 5.2 illustrates the time taken by the processor to process each frame for different models. Figure 5.2(a) shows that for the proposed model, there are spikes in the processing time at regular intervals, determined by the value of n, which corresponds to when the MTCNN face detection algorithm is executed. However, the time taken across n frames is the lowest for the proposed model, indicating that over a continuous sequence of frames, the proposed model works faster on average compared to other algorithms. This efficiency becomes especially valuable in high-quality videos, where each frame contains more data, causing slower processing in other models.

As shown in Figure 5.2(b), when video quality is lower, algorithms like MTCNN and the proposed model outperform other techniques in terms of both speed and efficiency. This confirms that the performance of state-of-the-art algorithms is influenced by video quality; higher quality results in slower execution speeds.

The speed improvements of the proposed model are largely due to the performance of the KLT feature tracking algorithm. Let the image intensities (brightness) of a particular region or window be denoted by $B(x,y,t)$. The motion pattern from one frame to another can be represented as:

$$B(x, y, t + t') = B(x - x', y - y', t) \tag{5.1}$$

The displacement from point $p=(x,y)$ between time t and t' is denoted by $d=(x',y')$.

Let the local model of image to track the feature points is denoted by $R(p)$. So, initially, $R(p)=B\ (x,\ y,\ t\ +\ t'\ )$. Then,

$$R(p) = B(p - d) + n(p) \tag{5.2}$$

where noise is denoted by n. Now, for the residue error to be minimized, the displace-

ment vector is chosen appropriately using the following integral:

$$\epsilon = \int_r [B(p-d) - R(p)]^2 w dp \tag{5.3}$$

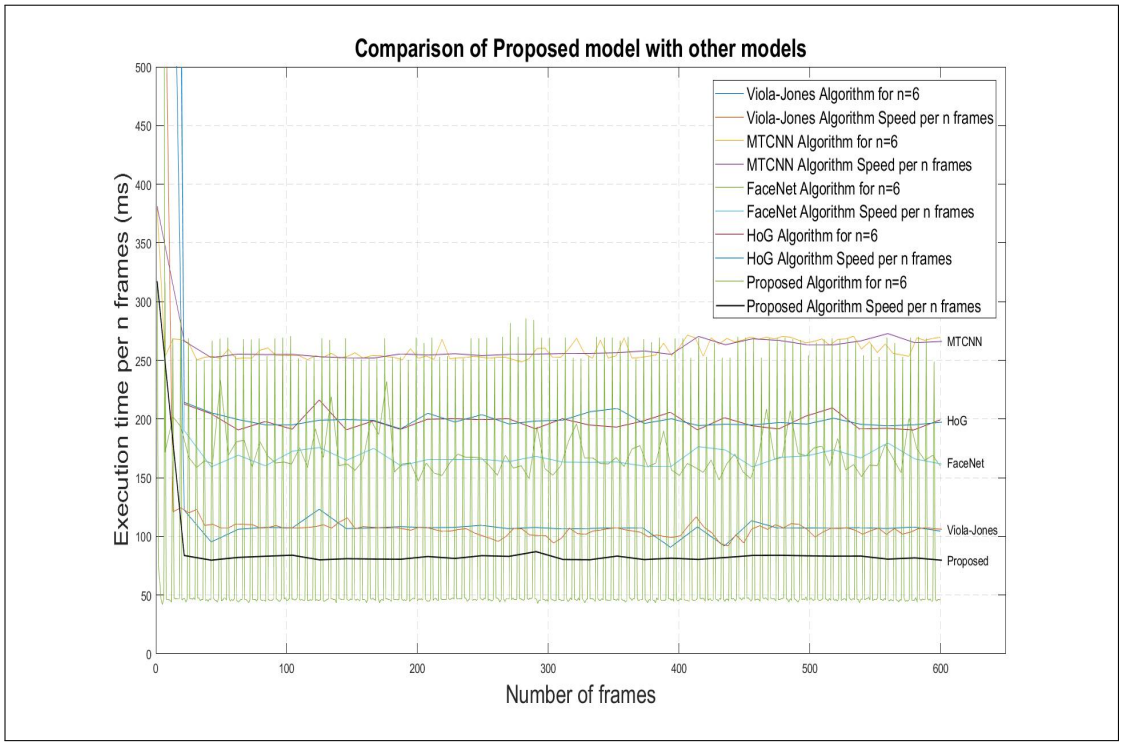
where r is the given displacement region, w is the assigned weight, and ϵ is the error to be minimized.

Table 5.2: Speed comparison of the model against other models, in Frames per Second (FPS), for YouTube Faces Dataset

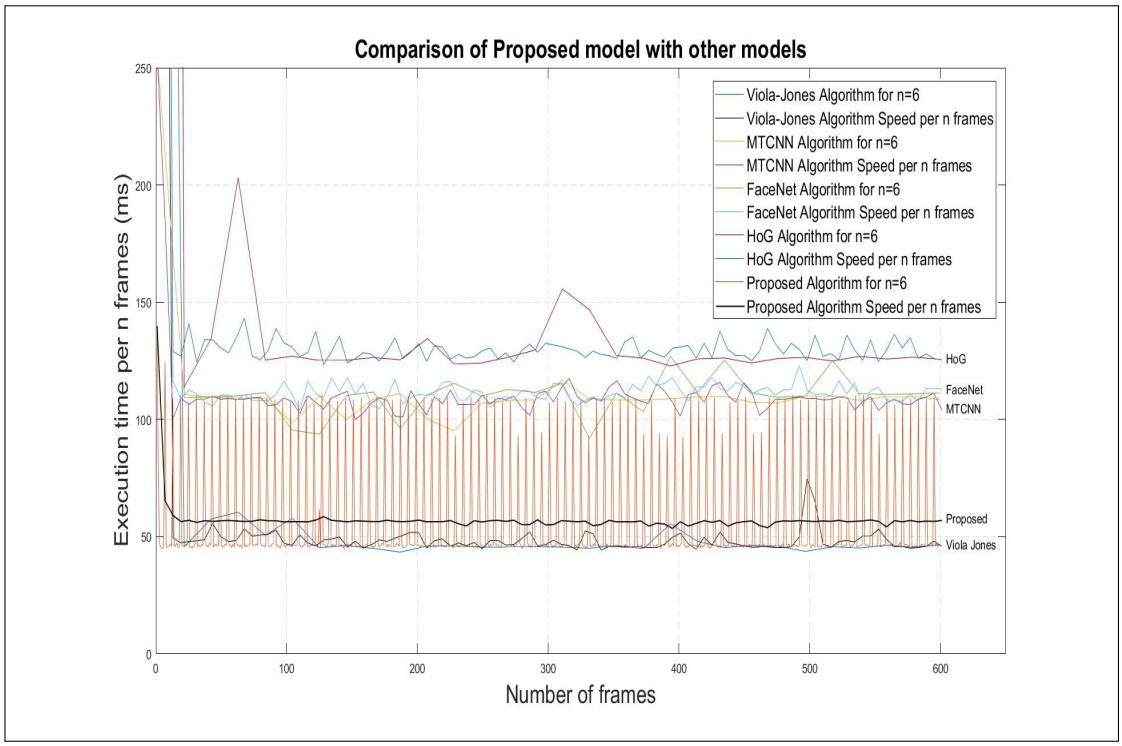
Model	Frame size	Speed (in FPS)
MTCNN	480 x 360	9.619947924 (i5-8300H)
HoG	480 x 360	7.973284182 (i5-8300H)
FaceNet	480 x 360	8.852338567 (i5-8300H)
DPM	480 x 360	3.976199327 (i5-8300H)
Proposed	480 x 360	17.57805462 (i5-8300H)

In Table 5.1 and 5.2, the proposed model's runtime performance is compared against traditional algorithm of DPM and HOG, as well as the modern algorithms of FaceNet and MTCNN face detection. The performance of Viola-Jones face detection algorithm is very fast when compared to others, which is partly why it is considered to be a benchmark for comparing the performance of other algorithms. However, it hasn't been included in this comparison, due to its low accuracy. Despite the relatively slower performance of MTCNN, it is selected as the base algorithm for the proposed model due to its high accuracy. Figure 5.2 demonstrates that on average, the proposed model performs better than other algorithms over continuous bursts of frames.

Figure 5.3 further illustrates the runtime performance of the proposed model under varying refresh rates. Increase in refresh rate n corresponds to a greater number of frames over which the detected facial location landmarks are tracked, thereby improving the overall speed of the proposed model. However, if n is set too low, the expensive face detection algorithm is executed more frequently, leading to slower performance.

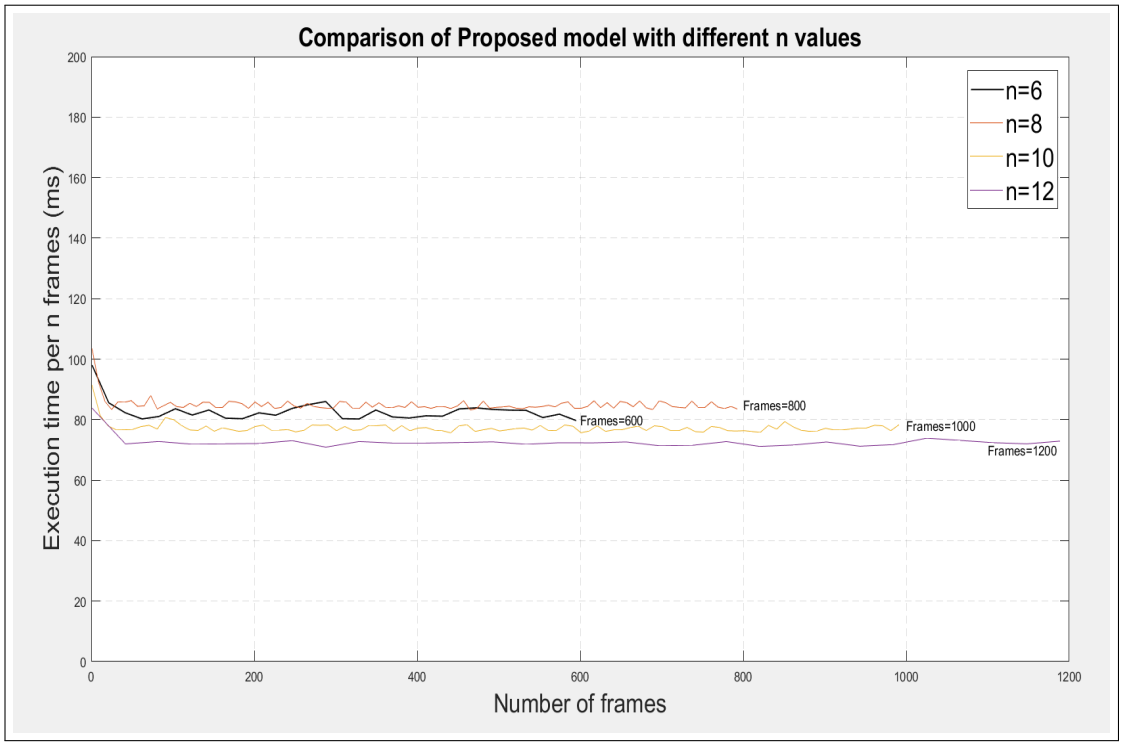


(a) For 300-VW dataset

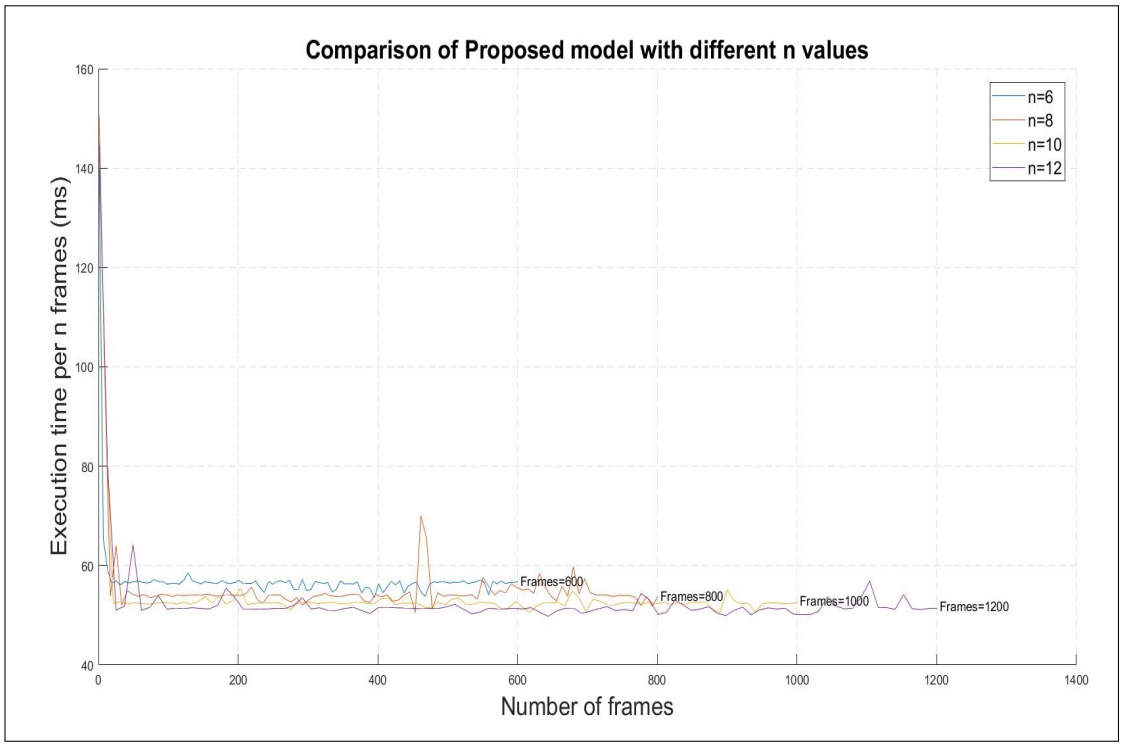


(b) For YouTube Faces Dataset

Figure 5.2: The Runtime performance of proposed model on 300-VW dataset and YouTube Faces dataset compared against other models. It is seen that the proposed model is faster than other models for the given value of n; where n is the refresh rate of the Face Detection algorithm.



(a) For 300-VW dataset



(b) For YouTube Faces dataset

Figure 5.3: The Runtime performance of proposed model on 300-VW dataset and YouTube Faces dataset for different values of n. It is seen that the proposed model gets faster as the value of n increases; where n is the refresh rate of the Face Detection algorithm.

5.4 *Evaluating accuracy of the proposed method*

Table 5.3 and Table 5.4 show the accuracy comparisons of the proposed model with other face detection models. It is evident from the tables that the accuracy of the proposed model is significantly higher than that of the traditional Viola-Jones algorithm. This explains why modern algorithms like MTCNN are generally preferred over Viola-Jones in recent times.

As the tables indicate, the MTCNN algorithm exhibits higher accuracy in detecting faces compared to both the proposed model and Viola-Jones. Nevertheless, the proposed model still demonstrates a competitive level of accuracy, especially considering that it is based on MTCNN. Table 5.4 further supports the observation that accuracy tends to increase as video quality decreases. This is consistent with the fact that lower video quality simplifies the detection task, often leading to improved accuracy.

The accuracy, α is calculated as follows:

$$\alpha = \frac{Number\ of\ positive\ frames}{Total\ number\ of\ frames\ with\ face} \tag{5.4}$$

Table 5.3: Comparing the accuracy of the proposed model with other models for 300-VW Dataset

Model	No. of positive frames	Total no. of frames	% accuracy
Viola-Jones	343	600	57.166667
MTCNN	592	600	98.666667
HoG	545	600	90.833333
FaceNet	550	600	91.666666
DPM	548	600	91.333333
Proposed model for n=6	555	600	92.5

Table 5.4: Comparing the accuracy of the proposed model with other models for YouTube Faces Dataset

Model	No. of positive frames	Total no. of frames	% accuracy
Viola-Jones	385	600	64.16666667
MTCNN	595	600	99.16666667
HoG	558	600	93
FaceNet	554	600	92.33333333
DPM	542	600	90.33333333
Proposed model for n=6	571	600	95.16666667

Table 5.5: Comparing the effect of n on accuracy of the proposed model for 300-VW Dataset

Model	No. of positive frames	Total no. of frames	% accuracy
n=6	555	600	92.5
n=8	710	800	88.75
n=10	870	1000	87
n=12	1029	1200	85.75

Table 5.5 and Table 5.6 illustrate the effect of the value of n on the accuracy of the proposed model. As n increases, the model's accuracy decreases. This highlights that, to achieve higher accuracy, a lower value of n is preferable. A lower n allows the detected facial landmarks to be refreshed more frequently, resulting in improved accuracy. However, if n is set too high, many frames may pass without updating the facial landmarks, which compromises accuracy.

Tables 5.7 and 5.8 compare the runtime and accuracy of the proposed model for different values of n. Although some algorithms may be faster or more accurate than our model, no algorithm excels in both aspects. The proposed model offers a balanced

Table 5.6: Comparing the effect of n on accuracy of the proposed model for YouTube Faces Dataset

Model	No. of positive frames	Total no. of frames	% accuracy
n=6	571	600	95.16666667
n=8	750	800	93.75
n=10	912	1000	91.2
n=12	1068	1200	89

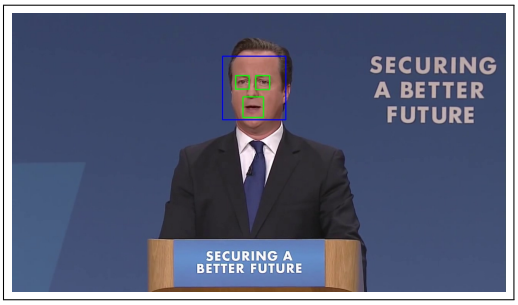
Table 5.7: Runtime and accuracy comparisons for different values of n for 300-VW Dataset

Proposed Model for	Frame Size	FPS Speed	% accuracy
n=6	1280 x 720	10.122395	92.5
n=8	1280 x 720	11.772989	88.75
n=10	1280 x 720	12.940982	87
n=12	1280 x 720	13.80364	85.75

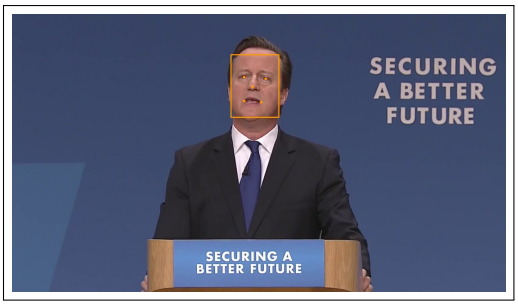
Table 5.8: Runtime and accuracy comparisons for different values of n for YouTube Faces Dataset

Proposed Model for	Frame Size	FPS Speed	% accuracy
n=6	480 x 360	17.72069687	95.16666667
n=8	480 x 360	18.21308608	93.75
n=10	480 x 360	18.93633601	91.2
n=12	480 x 360	19.24842618	89

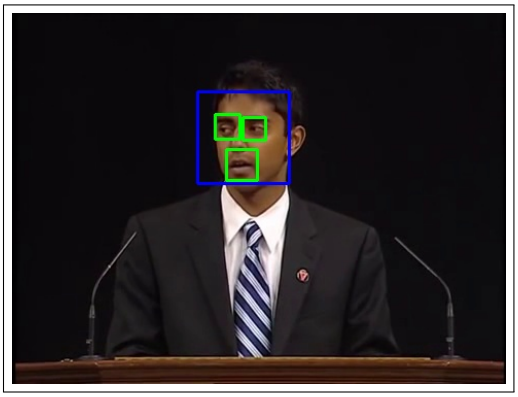
approach by providing both fast and accurate face detection for high-definition, high-quality videos. The tables also show how adjusting the value of n optimizes the model according to the specific requirements of an application.



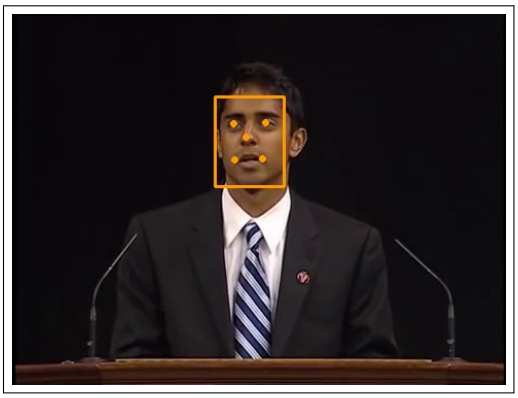
(a)



(b)

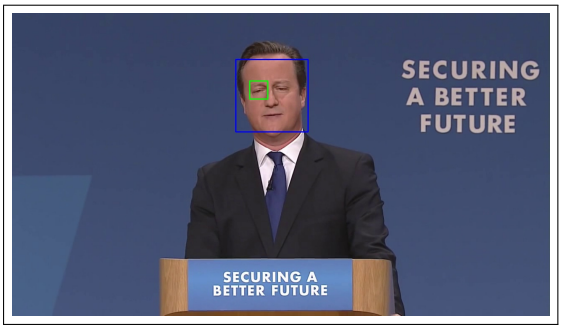


(c)

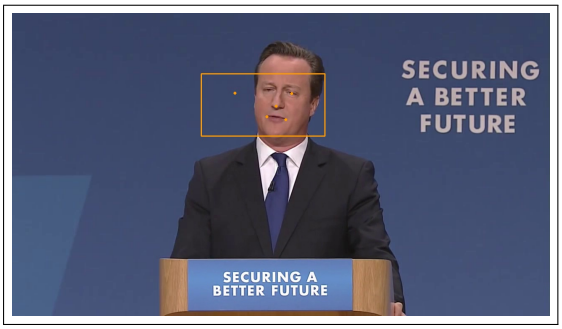


(d)

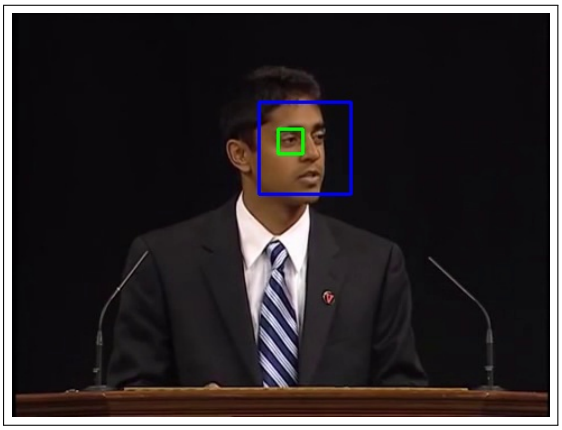
Figure 5.4: Positive face frames



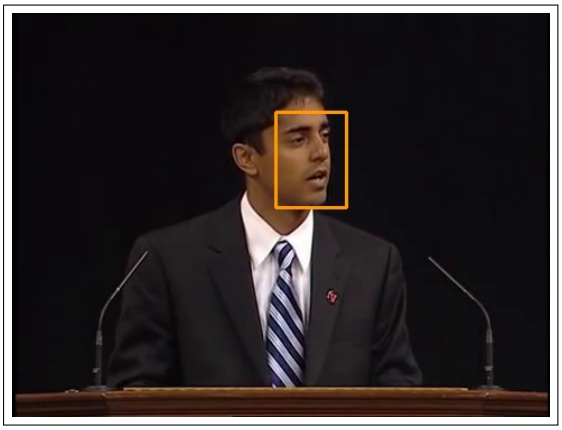
(a)



(b)



(c)



(d)

Figure 5.5: Negative face frames

5.5 *Occlusion Resolution in the integrated model*

This section provides an occlusion resolution component that works as a part of the whole Face Detection and Tracking system. This component is responsible for tracking the detected faces even after their partial or full occlusion occurs.

5.5.1 Extrapolation functions for Occlusion Resolution

The third sub-framework of the model addresses the challenge of occlusions in face detection. Occlusions occur when part or all of a face is blocked in a frame, which has been a persistent issue in digital image processing. Researchers have explored various methods to resolve this problem, and in this work, we apply mathematical techniques like interpolation and extrapolation to estimate the position of the face in occluded regions.

The three interpolation/extrapolation functions used in the proposed model for predicting approximate face positions during occlusion are:

1. **Linear 1-D Interpolation/Extrapolation**: This approach assumes that the movement of the five facial landmark points follows the path of a one-dimensional function. First, a linear polynomial function is estimated to fit the previous locations of the points. Using this function, the position of the points in the next frame is predicted through interpolation [330].

2. **Polynomial 1-D Interpolation/Extrapolation**: Similar to the linear method, but here the estimated function is not restricted to being linear. Any polynomial that closely fits the trajectory of the points from the previous frames is used to predict their next location [331].

3. **Spline Interpolation/Extrapolation**: In this method, the points are mapped to a piece-wise polynomial called a spline. The interpolation between each pair of previous and current frame points is done using polynomials $y = p(i),\ i = 0,1,2....$ The curvature k of the curve $y = f(x)$ is given by:

$$k = \frac{y''}{(1 + y'^2)^{\frac{3}{2}}} \tag{5.5}$$

To ensure y' and y'' are continuous, it is mathematically required that

$$p_i'(x_i) = p_{i+1}'(x_i) \tag{5.6}$$

$$p_i''(x_i) = p_{i+1}''(x_i) \tag{5.7}$$

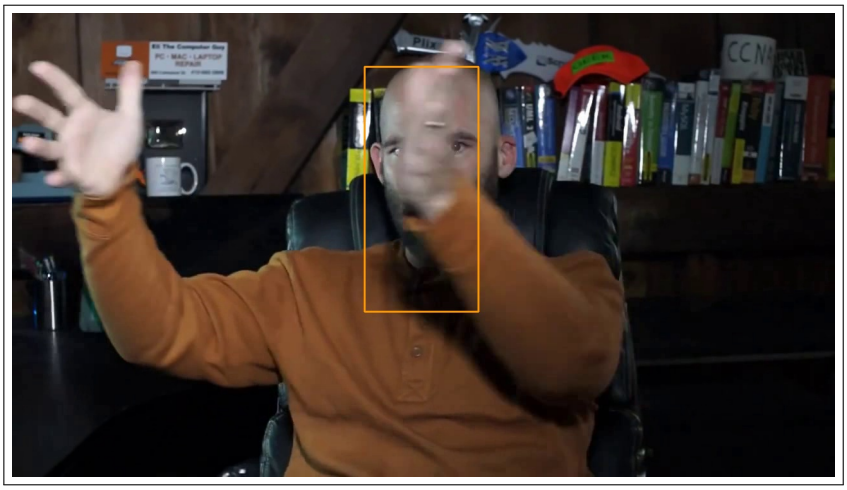
The above conditions are possible only in polynomials of degree 3 or above [332].

5.5.2 Occlusion resolution by Interpolation/Extrapolation

Figures 5.6a, 5.6b, and 5.6c show how interpolation and extrapolation methods assist in predicting the position of potential facial regions where a face could be located during occlusion. The three methods used produce different bounding boxes for the same facial region.

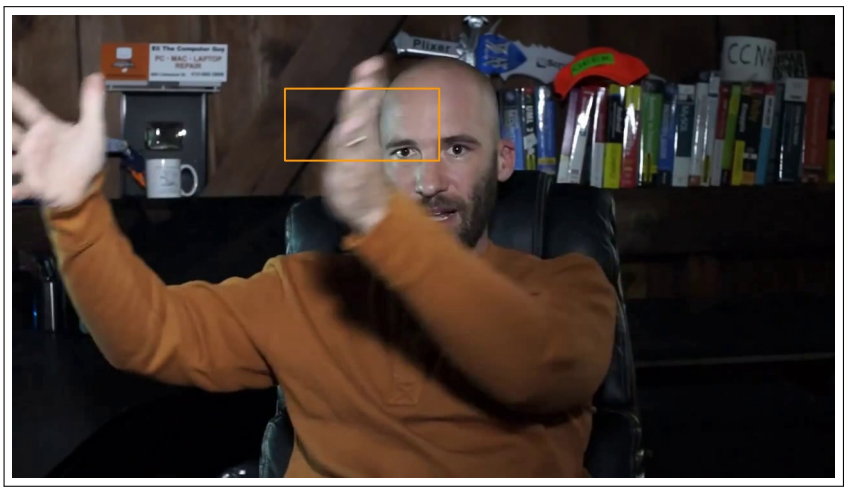
It is important to note that the three interpolation/extrapolation methods yield different bounding boxes because each method uses a different mathematical approach to estimate the movement of facial landmarks. Spline interpolation/extrapolation is generally more effective in most cases because it attempts to fit a minimum oscillating function to the data points (facial landmarks), resulting in a smoother prediction. In contrast, other methods tend to oscillate more between data points, causing the predicted face window to shift more sharply.

However, in certain cases, all three methods can produce the same bounding box for a particular frame. This happens because interpolation/extrapolation works by fitting preceding points to the closest function for each method. Since the same frame points are used across all three methods, it is possible for them to predict the same facial window in these special instances.



(a) Linear function

(b) Polynomial function



(c) Spline function

Figure 5.6: Occlusion Resolution using different Interpolation/Extrapolation functions

5.6 *Conclusions*

A modified face detection and tracking technique for high-quality, high-data videos is introduced in this chapter to maintain a balance between the speed of the model and the accuracy of the detected faces. This approach addresses the well-known issue that most existing face detection techniques perform well in terms of speed and accuracy for live feeds of low quality. However, when it comes to high-definition and high-quality videos, the performance of these models decreases significantly.

These algorithms fail to deliver fast results because the amount of data per frame is much higher in high-resolution videos, making it difficult for complex models to process the frames efficiently. Achieving high accuracy at the expense of very slow speeds is not ideal for real-time applications.

A sub-component of occlusion resolution is also introduced in this chapter, which uses three different interpolation/extrapolation functions to deal with it. This ensures that the model doesn't stop working when it can't detect any face in the frame. Instead, it uses the past information of the detected faces in previous frames to statistically predict where the face might be present in the current frame and continues this process until it can successfully detect a face using the face detection algorithm.

The occlusion resolution methods discussed through the three interpolation/extrapolation techniques achieve satisfactory results, as evidenced by the figures shown.

The proposed model accelerates the face detection process, reaching up to 19 FPS (a substantial improvement), while still maintaining accuracy above 90%, which makes it suitable for real-time processing.

Chapter 6

INTEGRATION OF NON-NEIGHBOURHOOD BACKGROUND ELIMINATION COMPONENT

The proposed model about Background elimination component is discussed in this chapter. The model maintains the accuracy and integrity of the base algorithm used, while reducing the processing time per frame, to make them more efficient to be used for high quality videos.

6.1 *Non-Neighbourhood Background Elimination (NNBE)*

Non-Neighbourhood Background Elimination (NNBE) is proposed in this work as part of a face detection and tracking system. The goal of NNBE is to reduce the area of interest that the detection algorithm scans in each frame, thereby decreasing the processing time required to detect the presence of a face. This is particularly important for high-quality videos, where large amounts of data can reduce the efficiency of detection algorithms.

Experimental results show that different face detection algorithms exhibit varying performance when video quality parameters, such as frame rate, bit depth, interlace, and PixelAspectRatio, are considered [333]. Processing large amounts of data per second decreases the efficiency of many state-of-the-art detection algorithms, which typically scan each frame window-by-window, increasing the number of computations and, consequently, the processing time.

NNBE uses frame rate as a parameter to estimate a probabilistic neighbourhood where a detected face in one frame is expected to appear in the next. Frame rate is a crucial factor in determining the Quality of Experience (QoE) for video, influencing how smoothly the human eye perceives the video. Scientific studies indicate that a minimum of 24 frames per second (fps) is required for the human eye to perceive a stream of images as continuous motion. Higher frame rates — such as 30, 48, 50, or 100 fps — yield smoother transitions between frames but also result in more data to process per second, slowing pre-processing times.

NNBE uses the information of the frame rate to estimate how far the detected face is likely to move between frames. Lower frame rates lead to larger displacements, while higher frame rates mean less displacement, since more frames are processed per second. NNBE uses the coordinates of the detected target and the frame rate to approximate a neighbourhood region to scan for the target in the next frame. The neighbourhood is calculated as a grid, with the grid size based on the frame rate of the video stream.

For the neighbourhood function, it is assumed that the minimum frame rate of 24 FPS generates the largest neighbourhood to be scanned. This neighbourhood spans 3 times the side of the detected square facial region in all directions—up, down, left, and right. This assumption is based on the logical assumption that the largest neighbourhood and face displacement across subsequent frames are a function of the frame rate. Similarly, for videos with a frame rate of 48 FPS, the neighbourhood area is assumed to be 2 times the side of the detected square facial region.

Using these assumptions, the neighbourhood function derived from graphical data is:

$$y = 14x^2 - 94x + 180 \tag{6.1}$$

where y is the frame rate, and x is the neighbourhood factor α.

Calculating x (neighbourhood factor α) from the above equation, we get:

$$x = \frac{1}{14}(47 - \sqrt{14y - 311}) \tag{6.2}$$

This function is used to find the coordinates of the neighbourhood window using the frame rate of the input video. The equation was derived by observing the movement

of objects across frames for different frame rates.

When the size of the detected face exceeds 13-15% of the frame size, the equation produces saturated results, resulting in the entire frame being treated as the neighbourhood window. This defeats the purpose of NNBE. To resolve this, the neighbourhood function is optimized as:

$$y = 56x^2 - 244x + 288 \tag{6.3}$$

Using the optimized equation, calculating x (neighbourhood factor α) results in:

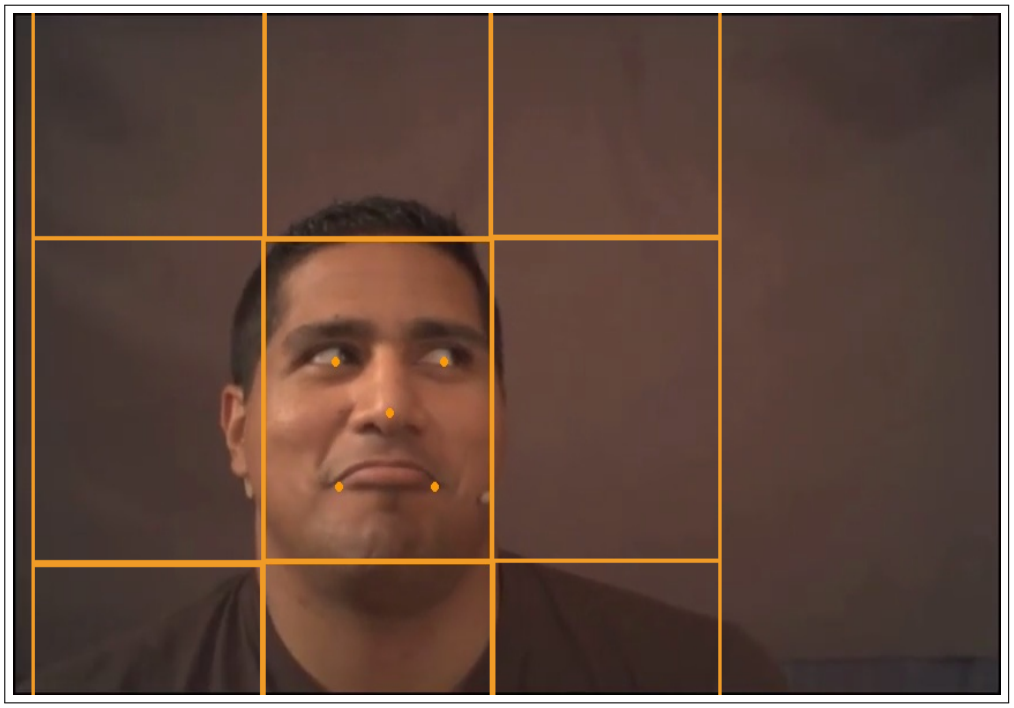
$$x = \frac{1}{28}(61 - \sqrt{14y - 311}) \tag{6.4}$$

For values of $\alpha < 1.5$, faster and more accurate detection results are achieved.

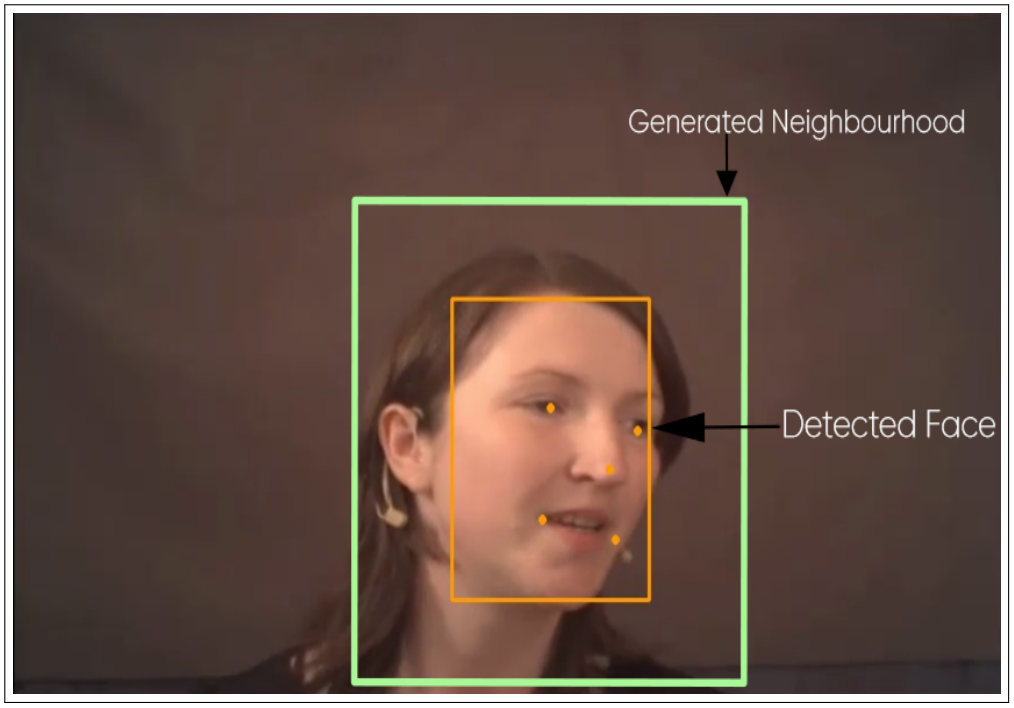
Figure 6.1 illustrates how the frame rate influences the neighbourhood size in subsequent frames. For a 24FPS video, the neighbourhood grid is large, as shown in Figure 6.1a. As the frame rate increases, the neighbourhood size decreases due to smaller displacements between frames. The neighbourhood grid for a 50FPS video, as illustrated in Figure 6.1b, is visibly smaller.

% increase in the 2 video samples	Neighbourhood FaceNet		Neighbourhood HOG		Neighbourhood MTCNN	
	On each sample	Average % Increase	On each sample	Average % Increase	On each sample	Average % Increase
24 FPS	1.845689771	**1.394396821**	0.11661549	**1.085804982**	0.743652553	**1.488799161**
	0.943103871		2.054994475		2.233945769	
48 FPS	3.553454544	**3.095944311**	6.080780053	**5.760866768**	6.962357788	**6.425795382**
	2.638434077		5.440953483		5.889232977	
50 FPS	8.256435063	**8.408895118**	7.506420817	**6.696166107**	5.444320048	**13.83404385**
	8.561355174		5.885911397		22.22376766	

Table 6.1: Impact of NNBE Component on increasing efficiency of other algorithms



(a) Neighbourhood calculation for a video with 24FPS frame rate



(b) Final generated neighbourhood for a video with 50FPS frame rate

Figure 6.1: Neighbourhood calculation according to frame rate. Calculated neighbourhood is inversely proportional to the frame rate. Lower frame rate results in large neighbourhood window, whereas higher frame rate results in smaller neighbourhood window.

6.2 *Analyzing runtime performance efficiency of existing algorithms*

Figures 7.3 and 7.4 show the impact of video quality on the runtime performance of various algorithms. In the figures, we observe that all algorithms perform well at lower frame rates (i.e., 24 FPS), but their performance deteriorates as the frame rate increases (48 and 50 FPS). This occurs because higher frame rates increase the amount of information that needs to be processed per unit of time, which increases the execution time required for processing each frame. Other factors such as video resolution, bit depth, bit rate, and codec also influence video quality and thus affect the processing time.

Among the face detection frameworks analyzed, YuNet performs better than the others for the given video samples. This highlights the importance of choosing the appropriate detection algorithm for different video streams. Factors such as the compression technique used to save the video, its frame rate, and the nature of the video all play a role in determining the efficiency of a detection algorithm. Using a less optimal algorithm results in longer processing times. Although YuNet offers better results, MTCNN was selected as the base algorithm for the proposed model due to its greater potential for improvement.

The neighbourhood variants of the algorithms have also been analyzed by applying the neighbourhood-reducing component alongside the respective face detection frameworks. In doing so, a corresponding decrease in execution times at higher frame rates is observed, as illustrated in Table 6.1. This is due to NNBE's ability to reduce the area that needs to be scanned by the detection algorithms. As the frame rate increases, the neighbourhood size decreases, which reduces the area the algorithm must process. This is why the 24 FPS video samples show performance similar to their base counterparts, while performance significantly improves at higher frame rates.

These experimental results demonstrate that the neighbourhood of a detected face is highly dependent on both the localized face region and the frame rate. Higher frame rates result in smaller neighbourhoods, which in turn decrease the computation time required for processing subsequent frames.

Figure 6.2 shows the visual representation of interaction of the proposed Background elimination sub-framework inside the overall model with the face-detection algorithm

being used.

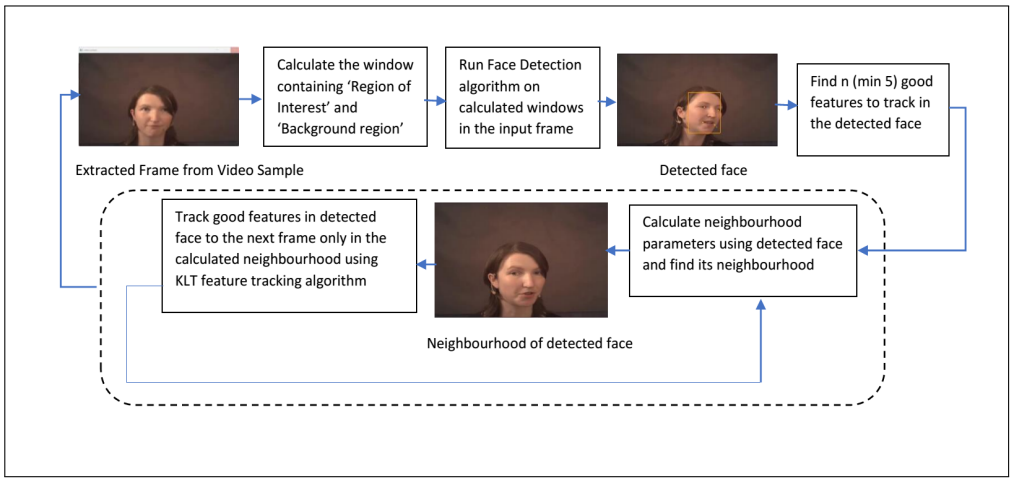


Figure 6.2: Working Procedure of the proposed model

Table 6.1 presents the analysis of the proposed model in terms of the percentage increase in the performance compared to the base algorithm for each set. The percentage increase in efficiency for a given algorithm on individual sets is given by:

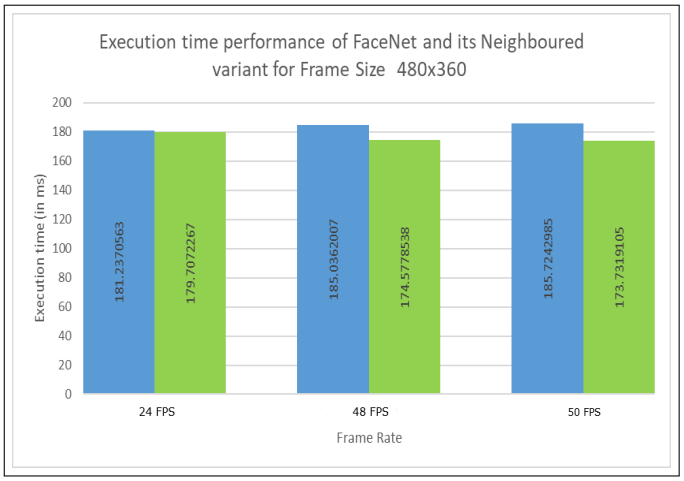
$$\%\ increase = \frac{t_i - t_f}{t_i} \tag{6.5}$$

where t_i is Average execution time for corresponding frame-rate without the integrated component, and t_f is Average execution time for corresponding frame-rate with the integrated component applied.

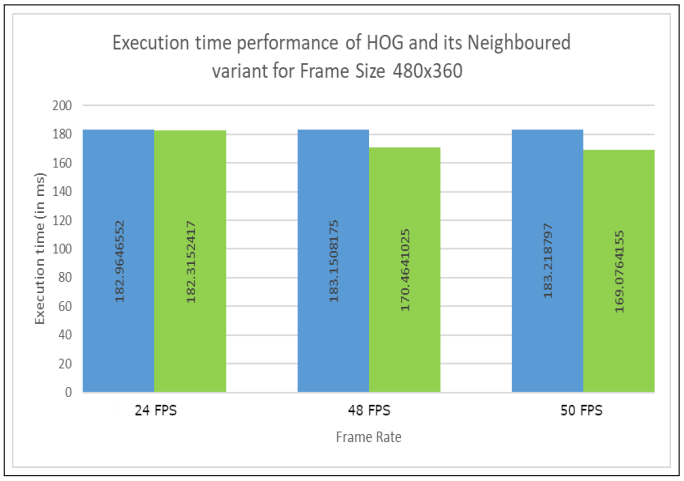
Average efficiency increase percentage for the $n(=2)$ sets is given by:

$$Avg\ \%\ increase = \frac{S_1\ increase + S_2\ increase}{2} \tag{6.6}$$

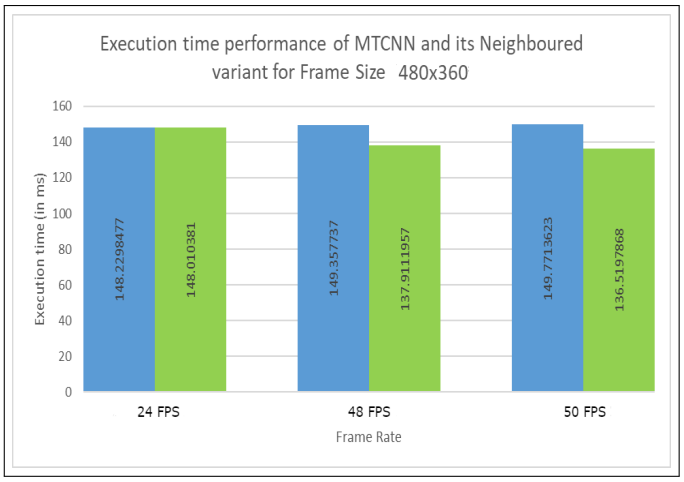
where S_1 *increase* represents the percent increase in efficiency of the detection algorithm on Set 1, while S_2 *increase* represents the percent increase in efficiency of the detection algorithm on Set 2.



(a) FaceNet performance comparison

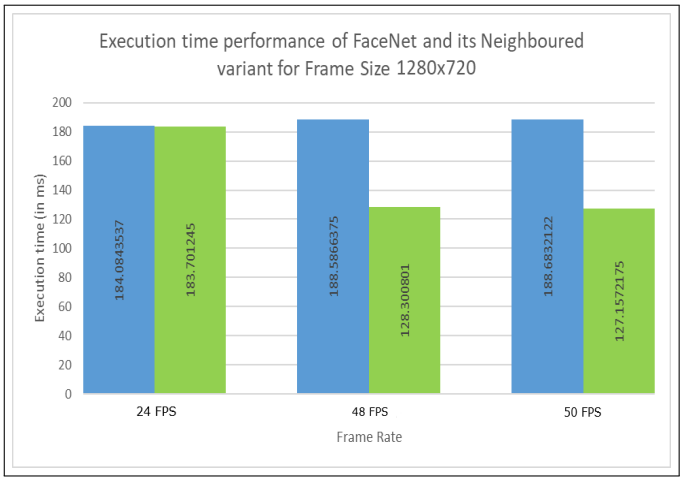


(b) HOG performance comparison



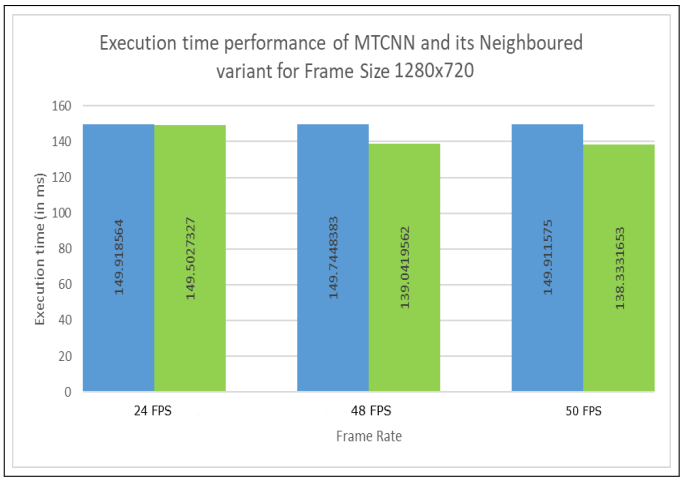
(c) MTCNN performance comparison

Figure 6.3: Execution runtime performance of various Face Detection algorithms and their neighbourhood variants on Video Category 1 of 300-VW dataset for different frame rates. The Blue bar represents the original algorithms and the Green bar represents their corresponding neighbourhood variants.



(a) FaceNet performance comparison

(b) HOG performance comparison



(c) MTCNN performance comparison

Figure 6.4: Execution runtime performance of various Face Detection algorithms and their neighbourhood variants on Video Category 2 of YouTube Faces dataset for different frame rates. The Blue bar represents the original algorithms and the Green bar represents their corresponding neighbourhood variants.

6.3 *Conclusions*

The methodology for the integrated component proposed in this chapter is effective in reducing the detection time of faces in subsequent frames after the face has been detected in an initial frame. The proposed scheme achieves an average speedup of up to 13%, depending on the frame rate of the input video stream (up to 50 FPS).

The comparisons and results show that the increased efficiency depends on the calculated neighbourhood, which is influenced by two factors: the video frame rate and the relative size of the detected face compared to the whole frame. Lower frame rates result in no or negligible speedup, whereas speedup increases with higher frame rates. Larger detected face areas result in less speedup, but smaller and more clearly defined face regions generate higher execution speeds.

Chapter 7

OPTIMIZED FACE TRACKER WITH NNBE-INTEGRATED FACE DETECTION

This chapter discusses face tracking algorithm that is used alongwith the NNBE-integrated Face Detection and tracking system. The optimized face tracker uses a hybridization of statistical approaches with KLT-feature tracking to predict the positions of the facial landmark points in the next frame as closely as possible.

7.1 *Overall Model*

The proposed model works using two sub-frameworks that achieve reduced computational time, thereby making them more feasible for high-data processing. One sub-framework performs the basic face detection in a frame, while the other eliminates unnecessary regions, allowing the face detection algorithm to focus only on areas of interest in subsequent frames. Figure 7.1 shows the flowchart depicting the flow of events in the model, where "Run MTCNN Face Detection" represents the face detection sub-framework, and "Non-Neighbourhood Background Elimination" illustrates the sub-framework that reduces the area of interest for scanning faces.

The model is flexible enough to incorporate any state-of-the-art face detection algorithm. In this work, MTCNN, HOG, FaceNet, and YuNet algorithms are used as base inputs to demonstrate the increase in processing speeds when integrated into the model. The input video is fed into the model, which extracts the first frame. Initially, the frame is divided into the "Region of Interest" and the "Background region" so that

the face detection algorithm dedicates maximum resources to the middle region of the frame, where a face is most likely to be found.

The base face detection algorithm is then applied to the first frame to determine the position of the face. After obtaining the coordinates of the detected window, a neighbourhood is calculated, indicating where the face might appear in the subsequent frame. Key features are generated within the detected face region, which are then tracked in subsequent frames using the KLT feature tracker.

When the next frame is processed, the KLT tracker focuses only on the approximate neighbourhood calculated from the previously detected face, avoiding unnecessary computations on the entire frame. This ensures that the algorithm concentrates only on the target region. The neighbourhood is recalculated at every frame after the facial features are tracked or the face detection algorithm is executed. The computational cost of calculating the neighbourhood is much lower than scanning the discarded background of the frame. This reduction significantly decreases processing time, making the model suitable for pre-processing high-quality, high-frame-rate videos.

Two variables, "*flag*" and "*count*," are maintained to monitor face availability in the frame and to refresh the face detection algorithm periodically. The "*flag*" is reset whenever a face is not found in the processed frame and set when a face is detected. The "*count*" is used to refresh the face detection algorithm after every n frames, ensuring that accurate face regions are detected regularly.

Algorithm 2 describes the pseudocode for the proposed model. The "*Region of Interest*" function optimizes the detection window, while the face detection sub-framework is denoted by the "*Face Detection*" function. The neighbourhood elimination sub-framework is represented by the "*NNBE*" function (Non-Neighbourhood Background Eliminator). The variable *det_frame* stores the face neighbourhood, and *track_features* stores the key feature points found in the neighbourhood for tracking in the next frame.

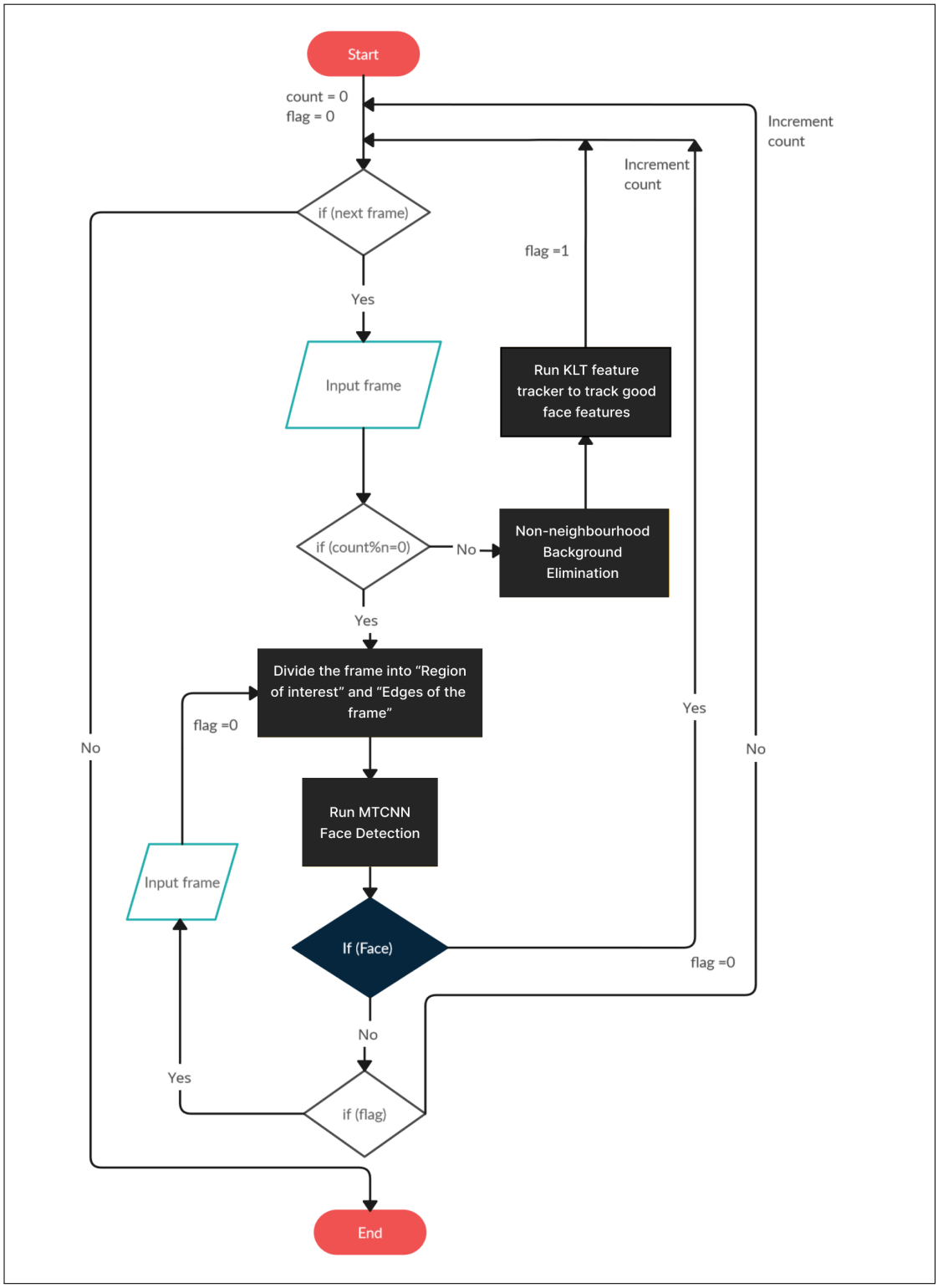


Figure 7.1: Flowchart depicting integration of "Non-Neighbourhood Background Elimination" component with face detection model

Algorithm 2: Algorithm for proposed model

```
flag = 0;
count = 0;
frame = Extract_frame(input_video);
while frame.exists(input_video) do
    count = count+1;
    if count % n == 0 then
        frame = Face_Detection(frame);
        if face.exists(frame) then
            break;
        else
            if !flag then
                break;
            else
                while !frame.exists(input_video) do
                    frame =
                    Extract_frame(input_video);
                    frame = Region_of_Interest(frame);
                    frame = Face_Detection(frame);
                    if face.exists(frame) then
                        flag = 0;
                        break;
                    else
                        break;
                    end
                end
            end
        end
    else
        det_frame = NNBE(frame);
        track_features = KLT(det_frame);
        flag = 1;
    end
end
return;
```

7.2 *Detection window optimisation based on Visual input*

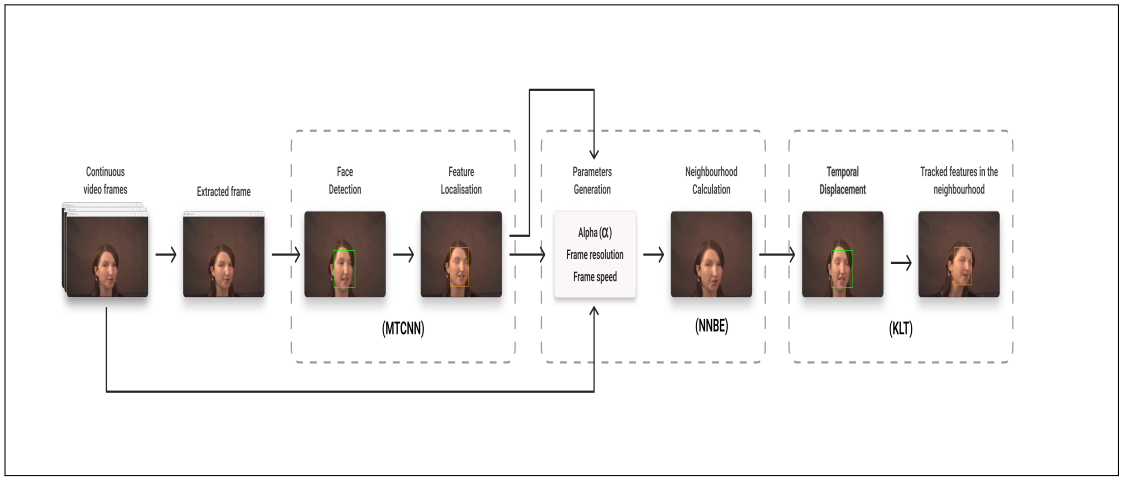


Figure 7.2: Structure of the proposed model

The functioning of the human eye can be compared to a multi-core supercomputer [360]. When the eyes detect an object, such as a face, different neural pathways process visual information from various parts of the image projected onto our brain. Most resources are dedicated to detecting and tracking the face in the central region of our visual field, while peripheral areas continue processing other information, looking for new objects entering our field of view.

A similar concept has been implemented in this model to enhance the speed of face detection in extracted frames. In the system's eight-core processor, five cores are dedicated to processing the central 65% of the frame, while the remaining 35%—mostly near the edges of the frame—is processed by the other three cores. This design is based on the observation that faces in multimedia content typically appear in the middle of the frame. By assigning more resources to the central region, the model improves face detection efficiency.

New faces entering the frame from the edges are detected by the three cores focused on the peripheral regions. Once detected, these faces are tracked by the five cores in the central region from the next frame onward, ensuring optimal resource allocation.

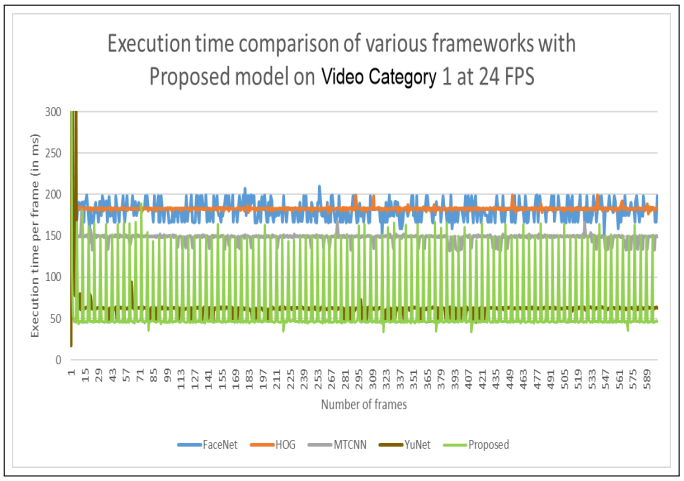
7.3 *Evaluating runtime efficiency based on video frame-rate*

Experimental data demonstrates that the proposed model significantly improves the performance of the base algorithm used, with performance being directly linked to video quality parameters, particularly the frame rate of the video stream.

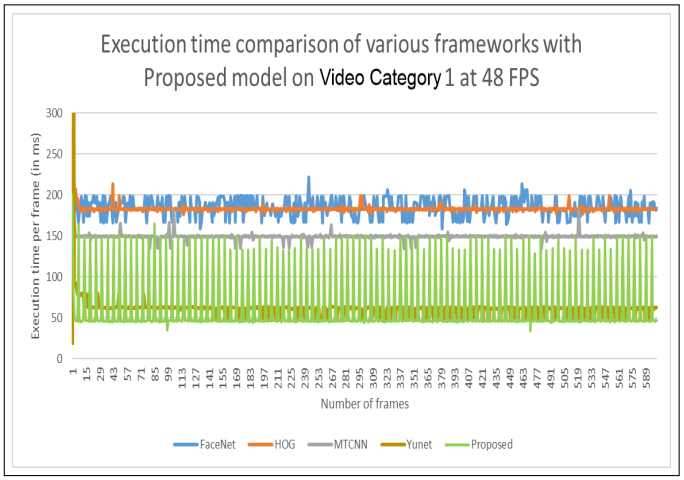
Figures 7.3 and 7.4 present an analysis comparing the proposed model to other commercially used face detection algorithms. It is clear from the figures that the model enhances the execution time efficiency of the algorithms when applied to high-quality video streams. The performance improvement is more pronounced for Video Category 2 than for Category 1. This difference arises because the detected facial region in Category 2 occupies a smaller percentage of the overall frame area (∼9.57 %) compared to Category 1 (∼13.5 %). A smaller facial region results in fewer computations and, consequently, faster execution time.

The analysis also reveals that, although the execution time of the proposed model decreases across all frame rates, the performance enhancement is particularly noticeable for higher frame rates in Video Category 2. The smaller facial region leads to a reduced neighbourhood size, enabling the underlying face detection algorithm to locate and track the face more efficiently in subsequent frames. Unlike other detection algorithms, which experience performance deterioration at higher frame rates, the proposed model improves its performance as the frame rate increases. This makes the model especially effective for rapid analysis of areas of interest in high-quality video streams over extended periods.

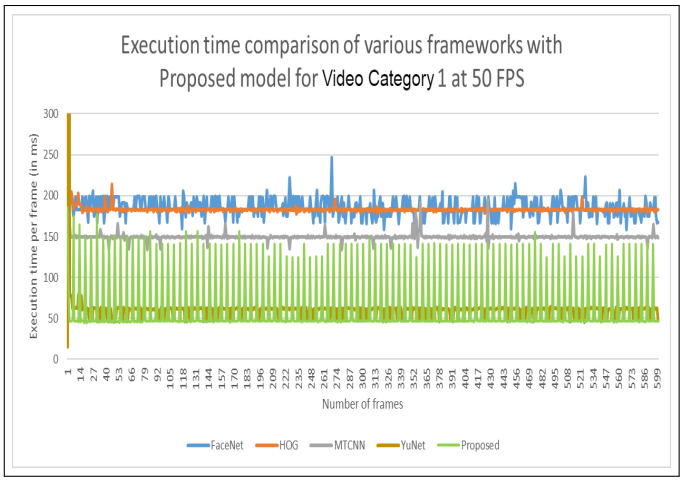
Table 7.1 shows that the model performs consistently across all analyzed frame rates. This consistency is due to the use of the neighbourhood factor α, which depends on the video's frame rate. α determines the size of the detection window for the next frame based on the detected face window in the current frame. The neighbourhood window is larger for lower frame rates and smaller for higher frame rates because videos with higher frame rates contain more frames per second, resulting in less movement of the face between frames. Consequently, the face is more likely to remain within a smaller area, reducing the number of computations required and improving execution time.



(a) Comparison on frame rate 24 FPS of Video Category 1

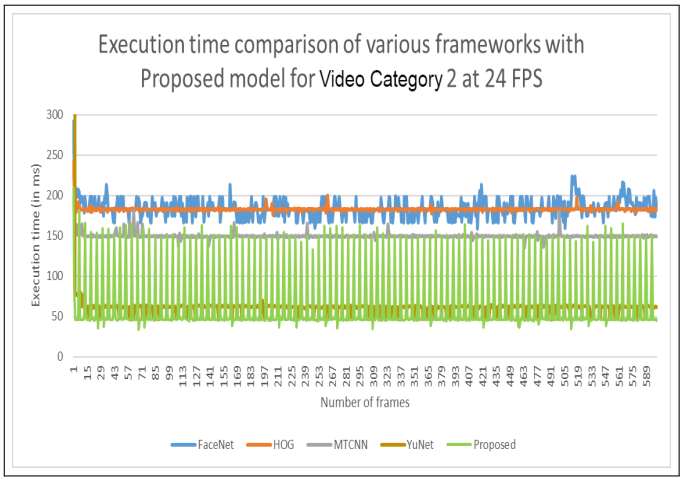


(b) Comparison on frame rate 48 FPS of Video Category 1

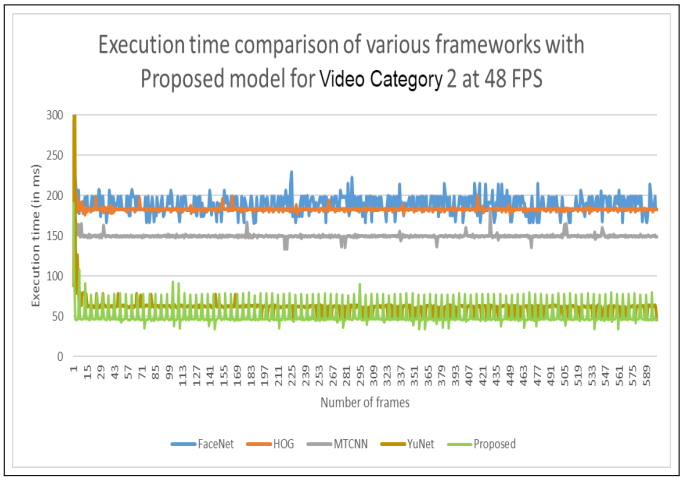


(c) Comparison on frame rate 50 FPS of Video Category 1

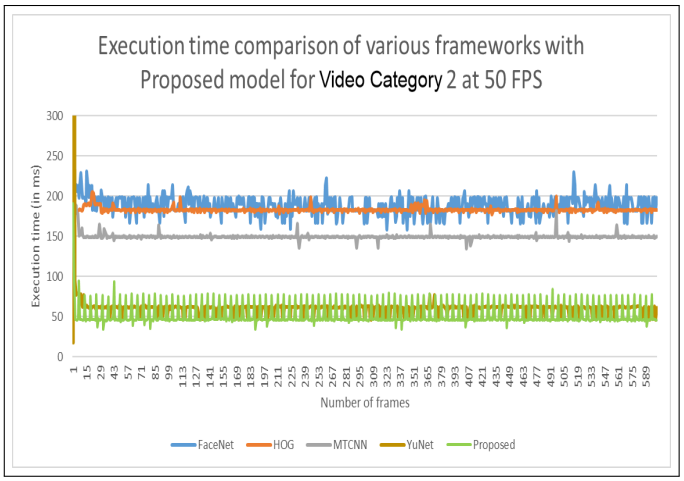
Figure 7.3: Execution runtime performance of proposed model on Video Category 1 of 300-VW dataset for different frame rates, compared against other commercial Face detection algorithms' performance



(a) Comparison on frame rate 24 FPS of Video Category 2



(b) Comparison on frame rate 48 FPS of Video Category 2



(c) Comparison on frame rate 50 FPS of Video Category 2

Figure 7.4: Execution runtime performance of proposed model on Video Category 2 of YouTube Faces dataset for different frame rates, compared against other commercial Face detection algorithms' performance

Table 7.1: Execution time comparison of the proposed model with other commercial face detection algorithms based on average execution time for both video categories on each frame rate

Average execution time (in *ms*)	FaceNet	HOG	YOLOFace	MTCNN	YuNet	Proposed
24 Frames Per Second	190.0878	183.7023	157.4689	149.1516	63.8127	45.52385
48 Frames Per Second	182.9333	182.9393	162.3449	149.5099	60.9907	46.80015
50 Frames Per Second	182.8492	183.3374	163.2590	149.1086	61.7396	46.09855

7.4 *Conclusions*

This chapter introduces a background-eliminating model for commercially used face detection algorithms applied to high-quality videos, and evaluates its performance primarily in terms of execution speed, as accuracy remains consistent with that of the underlying algorithm in average-case scenarios. The experimental results demonstrate that the proposed model significantly improves execution speed compared to several commercially available face detection algorithms.

The experiments show that the proposed model's execution speed surpasses that of the FaceNet system by 75%, the HOG detector by 74%, the MTCNN algorithm by 69%, and the YuNet face detector by 25%. These results indicate that the proposed model improves its operational performance substantially, particularly at higher frame rates in the input video stream.

The improvement in execution speed can be attributed to several factors: the frame rate of the input video, the base face detection algorithm used, and the size of the detected facial region relative to the input frame. Applying the proposed model to lower frame rates, such as 24-30 FPS, does not enhance execution speed as significantly and may, in some cases, introduce additional computational overhead compared to the base algorithm. This occurs because, at lower frame rates, the neighbourhood size is large enough that the algorithm effectively scans the entire frame, reducing the impact of the background elimination component while adding computational costs for calculating the neighbourhood.

Chapter 8

CONCLUSION, FUTURE SCOPE AND SOCIAL IMPACT

This chapter highlights the main conclusions and inferences of the research work undertaken for this thesis, and explores the possibilities of future research based on the developed optimization algorithms and methodology. It also discusses the impact of the presented work on the society, and how it can be helpful in increasing the quality of human life.

8.1 *Introduction*

The research thesis presented delves into improving the precision and effectiveness of face detection and tracking techniques, which are pivotal in sectors like surveillance, authentication, and human-computer interaction. The conclusion encapsulates the core accomplishments and insights gathered across four primary objectives, each playing a vital role in advancing face detection technology.

8.2 *Major Results*

Some of the major findings of the research work undertaken are:

1. A range of traditional and cutting-edge methods for face detection and tracking were explored and implemented. Traditional algorithms are comparatively light and easy to implement, but they suffer from low efficiency and performance stagnation. Modern face detection models use various machine learning and deep

learning techniques to improve the accuracy and efficiency of detected faces, but they are comparatively complex and harder to implement on regular or low-end machines. As technology progresses, machines are able to handle the computational demands of complex face detection models, but they suffer performance issues when they have to deal with large influx of information-rich high quality data. Therefore, further work has been done in this thesis that can process the high volume of high-quality video streams at much faster rates.

2. A novel face detection and tracking model that could provide an optimal trade-off between feature detection of faces and the eliminating the non-facial regions was proposed. Traditional approaches do not consider any balance amongst these factors, therefore either improved accuracy leads to increased computational demands and slower processing times, or higher processing speeds but lowered detection accuracy. The proposed model achieves an average speedup of approximately 13%, given the frame rate of the input video stream be ≤ 50 frames per second. This increased efficiency is very significant as it shows that the algorithm can maintain high detection accuracy while being more computationally efficient, which is highly crucial in real-time processing scenarios.

3. A face detection model was developed, that not only achieves high accuracy, but also improves execution speed. The integrated model achieves the accuracy of upto 93%, while keeping high execution speeds of upto 19 FPS, which is very high. The results and comparisons show that efficiency can be manipulated by changing values of optimization lever n; lower values of n mean high accuracy but low execution speeds, whereas higher values of n lead to high execution speeds but compromised accuracy. Conclusively, the proposed face detection and tracking system is very effective in localization, detection and tracking of faces across video frames. It also efficiently uses the optimization lever to provide a balance between execution speed and detection accuracy, and is very useful for efficient processing of multimedia content. The optimization lever can further be improved by employing such methods that increase the execution speeds with negligible accuracy loss.

4. A tracking component that efficiently tracks detected faces across video frames was designed. Face tracking is a complementary process to face detection, ensuring that once a face is detected, it can be continuously monitored as it moves within the frame or across multiple frames. The integrated model achieves the tracking speeds of upto 19+ FPS, which is very high. The results and comparison show that the efficiency of the KLT feature tracker increases by employing statistical extrapolation techniques. The tracker speed is also increased by eliminating the portion of KLT algorithm that determines the good features to track across frames, thereby decreasing overall number of computations. The face tracker can further be made more efficient by generalizing it for all types of videos, instead of video parameter-dependent outcomes.

8.3 *Scope for Future Research*

The face detection model and Non-Neighbourhood Background Elimination component developed and used in the present work are viewed as a starting point in the field of face processing in Computer Vision, and thus, the scope for further research is wide. Some of the areas of future research are:

1. Our work is an initial step toward faster high image-information processing, and therefore, can be improved upon in various aspects and parameters such as compromised accuracy, and better intermediary algorithms to make the whole overall models more efficient.

2. The works can also be refined to handle more variability in the conditions and scenarios, and making them more scalable for future larger datasets.

3. The proposed face detection and tracking model can be improved upon by choosing a more faster and accurate base face detection algorithm, since it's performance is still dependent on the initial face detecting algorithm chosen for the model.

4. The refresh rate used with KLT algorithm in the model can be substituted by some other point tracker that is faster and more accurate.

5. Occlusion resolution can either be achieved by other techniques of extrapolation, that could predict the positions of facial landmarks locations in the current frame by utilizing previous frames' information more efficiently, or by suggesting some different predicting methods that could utilize inherent information.

6. The calculation of neighbourhood coordinates for NNBE component can further be improved by employing better statistical models; other video parameters like resolution, or bit-depth can also be used for neighbourhood generation.

7. Compromise in accuracy for the integrated model is still an issue, since the accuracy gets decreased by 1-2%, which does not impact the utility of our algorithm, but can still be improved in the future.

8. A better background subtraction or elimination technique can be employed instead of the proposed NNBE component, that further minimizes the area in a frame to be scanned for the target face by the detection algorithm. Since social media applications primarily focus on face and facial landmark regions, dependence of the model performance on smaller area of facial regions, can be focused for better processing speeds.

8.4 *Social Impact*

Effectively, this doctorate research has led to significant contributions in the field of face detection and tracking systems. It addresses key challenges such as computational cost, low contribution of Digital Image Processing schemes, and comparatively inefficient face tracking algorithms. The primary motivation for this thesis was that these days, there are numerous applications that have to deal with a large input of high-quality video feeds: Modern mobile phones are equipped to shoot videos of up to 1080p resolution, on-air TV channels and OTT platforms provide a lot of HD quality content, sports broadcasts have to provide live high-quality entertainment at audience's disposal, social media creates numerous high-quality content on various platforms. These are just some of the examples of current applications dealing with multimedia content in high-quality, and the data keeps increasing every day. Various applications can be developed that can analyze this content and provide more personalized experience to the users, or predict

trends in the market, among other targeted needs. Therefore, appropriate algorithms and frameworks need to be developed that can cope up with the increasing demands and supply of information-rich data. The algorithms and components proposed in our works are highly efficient in processing of high quality multimedia content, which comprises almost all of the advertisement, marketing and entertainment industry data.

Bibliography

[1] R. L. Hsu, M. Abdel-Mottaleb, and A. K. Jain, "Face detection in color images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 696–706, 2002.

[2] P. Viola and M. Jones, "Robust real-time face detection," in *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, vol. 2, 2001, pp. 747–747.

[3] C. Tomasi and T. Kanade, "Detection and tracking of point features," Carnegie Mellon University, Tech. Rep. CMU-CS-91-132, 1991.

[4] J. Shi and C. Tomasi, "Good features to track," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1994, pp. 593–600.

[5] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001, pp. 511–518.

[6] M. H. Yang, D. J. Kriegman, and N. Ahuja, "Detecting faces in images: a survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 34–58, 2002.

[7] W.-L. Chao, "Face recognition," GICE, National Taiwan University, Tech. Rep., 2007.

[8] K. Seshadrinathan, R. Soundararajan, A. Bovik, and L. Cormack, "A subjective study to evaluate video quality assessment algorithms," in *IS&T/SPIE Electronic Imaging*, vol. 75270. San Jose, California, United States: SPIE, 2010, pp. 1–10.

[9] B. Yang, J. Yan, Z. Lei, and S. Z. Li, "Aggregate channel features for multi-view face detection," in *Proceedings of the IEEE International Joint Conference on Biometrics (IJCB)*. IEEE, 2014, pp. 1–8.

[10] M. D. S. V. Gupta, "A study of various face detection methods," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 3, no. 5, pp. 6694–6697, 2014.

[11] A. B. Dehkordi, M. Pourazad, and P. Nasiopoulos, "The effect of frame rate on 3d video quality and bitrate," *3D Research*, vol. 6, pp. 1–13, 2015.

[12] L. Marcomini and A. Cunha, "A comparison between background modelling methods for vehicle segmentation in highway traffic videos," *arXiv preprint arXiv:1810.02835*, 2018.

[13] E. Meijering, "A chronology of interpolation: From ancient astronomy to modern signal and image processing," *Proceedings of the IEEE*, vol. 90, no. 3, pp. 319–342, 2002.

[14] W. Pairo, P. Loncomilla, and J. Ruiz-del Solar, "A delay-free and robust object tracking approach for robotics applications," *Journal of Intelligent & Robotic Systems*, vol. 95, no. 1, pp. 99–117, 2019.

[15] K. Park, C. Park, and Y. Moon, "Automatic foreground extraction by background elimination based on multiscale segmentation," *Optical Engineering*, vol. 50, no. 6, p. 067004, 2011.

[16] P. Viola and M. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[17] C. Wu, M. D. Mullin, and J. M. Rehg, "Fast asymmetric learning for cascade face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 3, pp. 369–382, 2008.

[18] K. Sundaraj, "Real-time face detection using dynamic background subtraction," *WSEAS Transactions on Information Science and Applications*, vol. 5, pp. 1531–1540, 2008.

[19] A. Heijden, "Two stages in visual information processing and visual perception?" *Visual Cognition*, vol. 1, pp. 325–362, 2010.

[20] Bin Yang, J. Yan, Z. Lei, and S. Z. Li, "Aggregate channel features for multi-view face detection," *IEEE International Joint Conference on Biometrics*, pp. 1–8, 2014.

[21] D. Shamia and D. A. Chandy, "Analyzing the performance of viola-jones face detector on the ldhf database," in *International Conference on Signal Processing and Communication (ICSPC)*. IEEE, 2017, pp. 312–315.

[22] I. G. N. M. K. Raya, A. N. Jati, and R. E. Saputra, "Analysis realization of viola-jones method for face detection on cctv camera based on embedded system," in *2017 International Conference on Robotics, Biomimetics, and Intelligent Computational Systems (Robionetics)*. IEEE, 2017, pp. 1–5.

[23] M. V. Alyushin and A. A. Lyubshov, "Enhanced viola-jones algorithm for face detection in the long-wave infrared spectrum," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018, pp. 1–6.

[24] J. Huang, Y. Shang, and H. Chen, "Improved viola-jones face detection algorithm based on hololens," *EURASIP Journal on Image and Video Processing*, vol. 2019, pp. 1–11, 2019.

[25] W. Pairo, P. Loncomilla, and J. Ruiz-del Solar, "A delay-free and robust object tracking approach for robotics applications," *Journal of Intelligent & Robotic Systems*, vol. 95, 07 2019.

[26] Y. Derhalli, M. Nufal, and T. AlSharabati, "Face detection using boosting and histogram normalization," in *Proceedings of the 9th Jordanian International Electrical and Electronics Engineering Conference (JIEEEC)*. Amman, Jordan: Al-Ahliyya Amman University, 2015, pp. 1–6.

[27] M. D. Putro, T. B. Adji, and B. Winduratna, "Adult image classifiers based on face detection using viola-jones method," in *1st International Conference on Wireless and Telematics (ICWT)*, 2015, pp. 1–6.

[28] M. Da'san, A. Alqudah, and O. Debeir, "Face detection using viola and jones method and neural networks," in *International Conference on Information and Communication Technology Research (ICTRC)*, 2015, pp. 40–43.

[29] M. Nehru and S. Padmavathi, "Illumination invariant face detection using viola-jones algorithm," in *Proceedings of the International Conference on Computing and Communication Technologies*. IEEE, 2017, pp. 75–80.

[30] A. Ranftl, F. Alonso-Fernandez, S. Karlsson, and J. Bigun, "A real-time adaboost cascade face tracker based on likelihood map and optical flow," *IET Biometrics*, vol. 6, 05 2017.

[31] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[32] S. Rungruangbaiyok, R. Duangsoithong, and K. Chetpattananondh, "Probabilistic static foreground elimination for background subtraction," *Imaging Science Journal*, vol. 67, no. 4, pp. 385–395, 2019.

[33] X. Wang and J. Song, "Iciou: Improved loss based on complete intersection over union for bounding box regression," *IEEE Access*, vol. 9, pp. 105 686–105 695, 2021.

[34] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 2544–2550.

[35] A. Suleiman and V. Sze, "Energy-efficient hog-based object detection at 1080hd 60 fps with multi-scale support," in *2014 IEEE Workshop on Signal Processing Systems (SiPS)*. Belfast, UK: IEEE, 2014, pp. 1–6.

[36] G. Barquero, I. Hupont, and C. Tena, "Rank-based verification for long-term face tracking in crowded scenes," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 3, pp. 495–505, 2021.

[37] J. Shen, Y. Liu, X. Dong, X. Lu, F. S. Khan, and S. Hoi, "Distilled siamese networks for visual tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 12, pp. 8896–8909, 2022.

[38] L. Acasandrei and A. Barriga, "Amba bus hardware accelerator ip for viola–jones face detection," *IET Computers & Digital Techniques*, vol. 7, no. 5, pp. 200–209, 2013.

[39] A. W. Y. Wai, S. M. Tahir, and Y. C. Chang, "Gpu acceleration of real time viola-jones face detection," in *Proceedings of the IEEE International Conference on Control System, Computing and Engineering*. IEEE, 2015, pp. 183–188.

[40] W. Wang, Y. Zhang, S. Yan, Y. Zhang, and H. Jia, "Parallelization and performance optimization on face detection algorithm with opencl: A case study," *Tsinghua Science and Technology*, vol. 17, no. 3, pp. 287–295, 2012.

[41] B. Yu and D. Tao, "Anchor cascade for efficient face detection," *IEEE Transactions on Image Processing*, vol. 28, no. 5, pp. 2490–2501, 2019.

[42] D. Zeng, F. Zhao, S. Ge, and W. Shen, "Fast cascade face detection with pyramid network," *Pattern Recognition Letters*, vol. 119, pp. 180–186, 2019.

[43] W. Wu, H. Peng, and S. Yu, "Yunet: A tiny millisecond-level face detector," *Machine Intelligence Research*, vol. 20, pp. 656–665, 2023.

[44] H. Lahiani, M. Kherallah, and M. Neji, "Hand pose estimation system based on viola-jones algorithm for android devices," in *Proceedings of the IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA)*. IEEE, 2016, pp. 1–6.

[45] P. Chib, M. Khari, and K. Santosh, "A computational study on calibrated vgg19 for multimodal learning and representation in surveillance," in *International Conference on Recent Trends in Image Processing and Pattern Recognition*. Springer, Cham, 2023, pp. 261–271.

[46] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.

[47] T. Li, W. Hou, F. Lyu, Y. Lei, and C. Xiao, "Face detection based on depth information using hog-lbp," in *Proceedings of the Sixth International Conference on*

Instrumentation & Measurement, Computer, Communication and Control (IMCCC). IEEE, 2016, pp. 779–784.

[48] X. Lu, W. Wang, J. Shen, D. J. Crandall, and L. Van Gool, "Segmenting objects from relational visual data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 7885–7897, 2022.

[49] S. Joseph and A. Pradeep, "Object tracking using hog and svm," *International Journal of Engineering Trends and Technology*, vol. 48, no. 5, pp. 321–325, 2017.

[50] N. Zeng, H. Zhang, B. Song, W. Liu, Y. Li, and A. M. Dobaie, "Facial expression recognition via learning deep sparse autoencoders," *Neurocomputing*, vol. 273, pp. 643–649, 2018.

[51] X. Lu, W. Wang, M. Danelljan, T. Zhou, J. Shen, and L. Van Gool, "Video object segmentation with episodic graph memory networks," in *Computer Vision—ECCV 2020*, ser. Lecture Notes in Computer Science, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer, 2020, vol. 12348, pp. 839–855.

[52] X. Lu, W. Wang, C. Ma, J. Shen, L. Shao, and F. Porikli, "See more, know more: Unsupervised video object segmentation with co-attention siamese networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2019, pp. 3618–3627.

[53] X. Lu, W. Wang, J. Shen, D. J. Crandall, and J. Luo, "Zero-shot video object segmentation with co-attention siamese networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 4, pp. 2228–2242, 2022.

[54] A. Goel, A. K. Goel, and A. Kumar, "The role of artificial neural network and machine learning in utilizing spatial information," *Spatial Information Research*, vol. 31, no. 3, pp. 275–285, 2022.

[55] R. A. Fisher, "On the mathematical foundations of theoretical statistics," *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, vol. 222, pp. 309–368, 1922.

[56] C. J. van Rijsbergen, *Information retrieval*. Butterworth-Heinemann, 1979.

[57] P. Viola and M. Jones, "Robust real-time face detection," *International journal of computer vision*, vol. 57, no. 2, pp. 137–154, 2004.

[58] D. L. Parnas, "On the criteria to be used in decomposing systems into modules," *Communications of the ACM*, vol. 15, no. 12, pp. 1053–1058, 1972.

[59] 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)*. IEEE, 2001, pp. I–I.

[60] D. M. Green and J. A. Swets, *Signal detection theory and psychophysics*. Wiley, 1966.

[61] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge 2007 (voc2007) results," in *International Journal of Computer Vision (IJCV)*, 2007.

[62] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 23, no. 6, pp. 681–685, 2001.

[63] B. Schölkopf, A. J. Smola, R. C. Williamson, and P. L. Bartlett, "New support vector algorithms," *Neural computation*, vol. 12, no. 5, pp. 1207–1245, 2000.

[64] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[65] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Prentice Hall, 2002.

[66] T. M. Mitchell, *Machine Learning*. McGraw-Hill, 1997.

[67] M. Turk and A. Pentland, "Eigenfaces for Recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 01 1991.

[68] A. Albiol, L. Torres, C. A. Bouman, and E. Delp, "A simple and efficient face detection algorithm for video database applications," in *Proceedings 2000 International Conference on Image Processing (Cat. No. 00CH37101)*, vol. 2. IEEE, 2000, pp. 239–242.

[69] L. Fan and K. K. Sung, "Face detection and pose alignment using colour, shape and texture information," in *Proceedings Third IEEE International Workshop on Visual Surveillance*. IEEE, 2000, pp. 19–25.

[70] N. Tsapatsoulis and S. Kollias, "Face detection in color images and video sequences," in *2000 10th Mediterranean Electrotechnical Conference. Information Technology and Electrotechnology for the Mediterranean Countries. Proceedings. MeleCon 2000 (Cat. No. 00CH37099)*, vol. 2. IEEE, 2000, pp. 498–502.

[71] S. Paschalakis and M. Bober, "Real-time face detection and tracking for mobile videoconferencing," *Real-Time Imaging*, vol. 10, no. 2, pp. 81–94, 2004.

[72] G. Niu and Q. Chen, "Learning an video frame-based face detection system for security fields," *Journal of Visual Communication and Image Representation*, vol. 55, pp. 457–463, 2018.

[73] B. Menser and M. Brünig, "Face detection and tracking for video coding applications," in *Proceedings of the Thirty-Fourth Asilomar Conference on Signals, Systems, and Computers*, vol. 1. IEEE, 2000, pp. 49–53.

[74] Y. Yacoob and L. S. Davis, "Face detection and tracking in video using dynamic programming," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2000, pp. 593–599.

[75] Y. Huang, Y. Wang, and Z. Qiu, "Face detection in color images and video sequences," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2000, pp. 102–107.

[76] S. Z. Li and A. K. Jain, "Face detection - a survey," *Pattern Recognition*, vol. 35, no. 1, pp. 1–15, 2001.

[77] H. T. Nguyen and J. Kittler, "Face detection and recognition in an image sequence using eigenedginess," in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2002, pp. 526–540.

[78] J.-H. Kim, K.-M. Kim, and K.-J. Kim, "Real time face detection from color video stream based on pca method," in *Proceedings of the International Conference on Pattern Recognition (ICPR)*, vol. 3. IEEE, 2003, pp. 228–231.

[79] R. Johnson, W. Wang, and J. Lee, "Embedded hardware face detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2004, pp. 1086–1091.

[80] Y. Chen, Z. Zhang, and X. Li, "Advances in face detection techniques in video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012, pp. 1682–1689.

[81] J. Lee, H. Kim, and J. Park, "Adaptive skin color model to improve video face detection," in *Proceedings of the International Conference on Image Processing (ICIP)*. IEEE, 2015, pp. 1672–1676.

[82] L. Zhang, S. Li, and W. Chen, "An efficient face detection and recognition for video surveillance," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016, pp. 2496–2504.

[83] Y. Wang, J. Zhang, and L. Yang, "End-to-end face detection and cast grouping in movies using erdős–rényi clustering," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2017, pp. 2935–2944.

[84] M. Li, Q. Li, and X. Zhang, "High-efficiency face detection and tracking method for numerous pedestrians through face candidate generation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2021, pp. 3251–3260.

[85] Z. Li, S. Li, and X. Li, "Support vector regression and classification based multi-view face detection and recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2000, pp. 455–460.

[86] Y. Ma, B. Moghaddam, and M.-H. Yang, "An automatic face detection and recognition system for video indexing applications," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2002, pp. 1–6.

[87] W. Zhao, H. Zhang, and Z. Lei, "Face detection for visual surveillance," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2003, pp. 1–8.

[88] W. Zhao and H. Zhang, "Real-time face detection from color video stream based on pca method," in *Proceedings of the International Conference on Image Processing (ICIP)*. IEEE, 2003, pp. 361–364.

[89] L. Zhang and Z. Zhang, "Robust visual similarity retrieval in single model face databases," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2005, pp. 809–814.

[90] X. He, J. Wang, and B. Li, "The use of neural networks in real-time face detection," in *Proceedings of the International Conference on Neural Information Processing (ICONIP)*. Springer, 2005, pp. 1131–1135.

[91] Z. Li, Y. Yu, and Y. Wang, "A new video surveillance system employing occluded face detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2005, pp. 450–455.

[92] Z. Wu, X. Wang, and F. Li, "Face detection approach in neural network based method for video surveillance," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 2006, pp. 1971–1974.

[93] V. Singh and M. Sharma, "A survey on face detection and recognition approaches," *International Journal of Computer Applications*, vol. 81, no. 6, pp. 38–43, 2013.

[94] P. Sharma, H. Saini, and G. Gupta, "Influence of low resolution of images on reliability of face detection and recognition," in *Proceedings of the International Conference on Computer Vision (ICCV)*. IEEE, 2015, pp. 123–128.

[95] J.-H. Jang, S.-J. Choi, and S.-J. Kim, "An accurate system for face detection and recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2019, pp. 178–185.

[96] S.-M. Lee and J.-H. Kim, "Face recognition system," in *Proceedings of the International Conference on Computer Vision (ICCV)*. IEEE, 2019, pp. 145–150.

[97] R. Sharma and S. Verma, "Face detection and recognition using opencv," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2019, pp. 210–215.

[98] J.-H. Kim, M.-J. Lee, and S.-J. Park, "A real-time framework for human face detection and recognition in cctv images," in *Proceedings of the International Conference on Image Processing (ICIP)*. IEEE, 2022, pp. 1157–1162.

[99] R. Singh and P. Yadav, "Face detection & face recognition using open computer vision classifiers," in *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 981–986.

[100] H. Li, W. Zhao, and S. Liu, "Face detection based on occlusion area detection and recovery," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2020, pp. 450–455.

[101] Y. Guo, J. Ding, and X. Li, "Support vector regression and classification based multi-view face detection and recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2000, pp. 1471–1476.

[102] J.-S. Pan and J. Yang, "Integrated approach of multiple face detection for video surveillance," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, vol. 3. IEEE, 2002, pp. 337–340.

[103] R. Sharma and J. R. Smith, "Learning to identify video shots with people based on face detection," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, vol. 1. IEEE, 2003, pp. 653–656.

[104] Z. Zhang and Q. Zhao, "Face detection and recognition in a video sequence," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, vol. 1. IEEE, 2004, pp. 351–354.

[105] X. Wang, Z. Zhang, and H. Li, "Face detection using discriminating feature analysis and support vector machine in video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2004, pp. 1423–1428.

[106] W. Li and S. Zhao, "The use of neural networks in real-time face detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2005, pp. 674–679.

[107] Y. Xu, J. Li, and M. Zhang, "A new video surveillance system employing occluded face detection," in *Proceedings of the IEEE International Conference on Video and Signal Based Surveillance (AVSS)*. IEEE, 2005, pp. 423–428.

[108] A. Albiol, R. Morales, and J. Esquivel, "Image-based face detection and recognition - state of the art," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2013, pp. 33–39.

[109] X. Zhu, L. Li, and Y. Zhang, "A combined modular system for face detection, head pose estimation, face tracking, and emotion recognition in thermal infrared images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018, pp. 1958–1966.

[110] L. Chen, C. Li, and Y. Zhang, "Efficient face detection and tracking in video sequences based on deep learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2021, pp. 1284–1292.

[111] J. Liu, F. Wang, and Z. Liu, "High-efficiency face detection and tracking method for numerous pedestrians through face candidate generation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2021, pp. 1951–1959.

[112] J. Smith, S. Kim, and M. Yang, "A comparison of face detection methods using spontaneous videos," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2022, pp. 1214–1222.

[113] J. Zhou and F. Jiang, "An efficient algorithm for human face detection and facial feature extraction under different conditions," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 2001, pp. 1521–1524.

[114] W. Gao, X. Tang, and H. Zhang, "Omni-face detection for video and image content description," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2000, pp. 1021–1024.

[115] ——, "Omni-face detection for video and image content description," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2000, pp. 1021–1024.

[116] C.-H. Lee and Y.-T. Lin, "Content-based indexing of images and video using face detection and recognition methods," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2001, pp. 485–488.

[117] K. Law, F. Lau, and W.-M. Lam, "Real-time face detection on a configurable hardware system," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2000, pp. 1221–1226.

[118] H. A. Rowley, S. Baluja, and T. Kanade, "Convolutional face finder - a neural architecture for fast and robust face detection," in *Proceedings of the IEEE Transactions on Pattern Analysis and Machine Intelligence*. IEEE, 2004, pp. 140–144.

[119] B. Pham and L. V. Khanh, "Embedded hardware face detection," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2004, pp. 211–215.

[120] Y. Liu and Q. Zhang, "Online face recognition system for videos based on modified probabilistic neural networks," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, 2004, pp. 412–415.

[121] W. Li and S. Zhao, "The use of neural networks in real-time face detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2005, pp. 674–679.

[122] J. Tan and X. Wang, "Face detection approach in neural network-based method for video surveillance," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2006, pp. 221–224.

[123] X. Zhou and X. Zhang, "Effective cue integration for fast and robust face detection in videos," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 2007, pp. 427–431.

[124] Y. Sun and J. Wang, "Multimodal approach to human-face detection and tracking," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2008, pp. 314–318.

[125] Q. Zhang and C. Li, "A video-based face detection and recognition system using cascade face verification modules," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2008, pp. 675–678.

[126] W. Xu and L. Sun, "A novel soc architecture on fpga for ultra-fast face detection," in *Proceedings of the IEEE International Conference on Field Programmable Logic and Applications (FPL)*. IEEE, 2009, pp. 178–183.

[127] D. W. Kim and J. Y. Kwon, "Face detection directly from h.264 compressed video with convolutional neural network," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2009, pp. 326–329.

[128] J. Li and W. Ma, "Fast and robust face detection on a parallel optimized architecture implemented on fpga," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2009, pp. 498–503.

[129] L. Johnson and A. Singh, "Review of face detection systems based artificial neural networks algorithms," *International Journal of Artificial Intelligence and Applications*, vol. 5, no. 6, pp. 55–62, 2014.

[130] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua, "A convolutional neural network cascade for face detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015, pp. 5325–5334.

[131] J. Wang and F. Liu, "Adaptive skin color model to improve video face detection," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 2015, pp. 2342–2346.

[132] K. Zhang, Z. Zhang, Z. Li, and K. Qiu, "Compact convolutional neural network cascade for face detection," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 2015, pp. 3728–3732.

[133] D. Sharma and M. Kumar, "Face detection and recognition in an unconstrained environment for mobile visual assistive system," *Procedia Computer Science*, vol. 115, pp. 14–20, 2017.

[134] S. Lee and J. Park, "Face detection with a viola–jones based hybrid network," *Journal of Visual Communication and Image Representation*, vol. 43, pp. 209–215, 2017.

[135] J. Redmon and A. Farhadi, "A deep learning approach for face detection using yolo," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2018, pp. 857–861.

[136] R. Kumar and A. Gupta, "Face detection and tagging using deep learning," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2018, pp. 1324–1329.

[137] L. Wang and H. Zhang, "Face detection approach from video with the aid of kpcm and improved neural network classifier," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2018, pp. 415–418.

[138] P. Patel and D. Mehta, "Face detection using viola-jones algorithm and neural networks," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 571–575.

[139] L. Zhang and X. Li, "Forensics face detection from gans using convolutional neural network," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 7, pp. 1573–1584, 2018.

[140] Y. Chen and L. Wu, "A fast face detection method via convolutional neural network," *Journal of Visual Communication and Image Representation*, vol. 60, pp. 445–452, 2019.

[141] A. Kumar and J. Saini, "A web-based application for face detection in real-time images and videos," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 7, pp. 134–139, 2021.

[142] Y. Wang, Z. Lin, and X. Shen, "Dbcface: Towards pure convolutional neural network face detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2021, pp. 2356–2364.

[143] H. Gupta and R. Tiwari, "Face detection in real-time live video using yolo algorithm based on vgg16 convolutional neural network," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, pp. 3579–3586, 2021.

[144] S. Yadav and R. Kumar, "A real-time framework for human face detection and recognition in cctv images," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 5, pp. 651–659, 2022.

[145] J. Liu and W. Zhang, "An efficient multi-scale anchor box approach to detect partial faces from a video sequence," *Pattern Recognition*, vol. 128, p. 108580, 2022.

[146] X. Gao and W. Li, "Automatic detection and recognition of players in soccer videos," *IEEE Access*, vol. 10, pp. 10 258–10 267, 2022.

[147] R. Sharma and V. Kumar, "Face detection, identification, and tracking by prdit algorithm using image database for crime investigation," in *Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2012, pp. 120–124.

[148] X. Zhang and J. Lu, "An unsupervised color image segmentation algorithm for face detection applications," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 2001, pp. 341–344.

[149] D.-S. Kim, D.-W. Kim, and H.-S. Yang, "Robust face detection at video frame rate based on edge orientation features," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 2002, pp. 263–266.

[150] H. Fronthaler, K. Kollreider, and J. Bigun, "Face detection using local smqt features and split up snow classifier," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 2007, pp. 345–348.

[151] A. Kumar and R. Jindal, "Human face detection and recognition using web-cam," *International Journal of Information Technology and Computer Science*, vol. 5, pp. 1–8, 2012.

[152] H. A. Rowley, S. Baluja, and T. Kanade, "A unified learning framework for real-time face detection and classification," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, vol. 1. IEEE, 2002, pp. 268–275.

[153] S. Z. Li and Z. Zhang, "Statistical learning of multi-view face detection," in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2002, pp. 67–81.

[154] J. Wu, N. Yu, and J. Yuan, "Fast and robust face detection in video," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2. IEEE, 2005, pp. 168–173.

[155] M. Al-Rawi and H. Yazid, "Face detection and pose alignment using colour, shape, and texture information," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2007, pp. 118–123.

[156] S. Yang and J. Wen, "Scale and pose invariant real-time face detection and tracking," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 2008, pp. 321–325.

[157] Z. Zhang, S. Z. Li, and R. Lienhart, "Face detection, tracking, and recognition for broadcast video," *Journal of Computer Vision and Image Understanding*, vol. 113, no. 3, pp. 430–441, 2008.

[158] Z. Li, H. Zhang, and X. Shen, "Large scale learning and recognition of faces in web videos," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2008, pp. 345–348.

[159] T. Nakamura and H. Yamaguchi, "A novel soc architecture on fpga for ultra-fast face detection," *Journal of VLSI Signal Processing Systems for Signal, Image and Video Technology*, vol. 58, no. 2, pp. 157–169, 2009.

[160] B. Yoo, J. Park, and J. Lee, "Face detection system design for real-time high-resolution smart camera," *Journal of Real-Time Image Processing*, vol. 4, no. 1, pp. 15–28, 2009.

[161] L. Wang, F. Liu, and Z. Wu, "Real-time face detection and recognition for video surveillance applications," *Pattern Recognition Letters*, vol. 30, no. 1, pp. 98–105, 2009.

[162] G. Bradski and A. Kaehler, "Real-time viola-jones face detection in a web browser," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2009, pp. 1424–1431.

[163] C. Li, G. Yin, and Y. Tan, "Real-time gpu-based face detection in hd video sequences," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2011, pp. 545–549.

[164] S. Mathur and R. Patil, "Video quality for face detection, recognition, and tracking," in *Proceedings of the International Conference on Pattern Recognition (ICPR)*, 2011, pp. 2318–2323.

[165] P. Zou and Y. Dai, "Accelerating boosting-based face detection on gpus," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012, pp. 2420–2425.

[166] H. Li, X. Shen, and L. Duan, "Unconstrained face detection," *IEEE Transactions on Multimedia*, vol. 14, no. 4, pp. 1050–1062, 2012.

[167] M. McInnes and S. Lebak, "Face detection and tracking using opencv," in *Proceedings of the ACM Conference on Human Factors in Computing Systems*, 2013, pp. 1–8.

[168] S. Ghosh and A. N. Chakrabarti, "Real-time human face detection and tracking," *International Journal of Computer Science and Information Technologies*, vol. 5, no. 6, pp. 7563–7566, 2014.

[169] P. Priyadarshini and S. Kumari, "Face detection using cbcr color model in video," *International Journal of Science and Research (IJSR)*, vol. 4, no. 6, pp. 2027–2030, 2015.

[170] G. Liu, Y. Jin, F. Du, L. Wang, and W. Wei, "Real-time robust face detection and tracking using extended haar functions and improved boosting algorithm," *Journal of Real-Time Image Processing*, vol. 12, no. 4, pp. 795–804, 2015.

[171] S. Patil and A. Prasad, "Face detection using modified viola-jones algorithm," in *Proceedings of the International Conference on Signal Processing and Communication Engineering Systems (SPACES)*. IEEE, 2015, pp. 178–182.

[172] D. Kumar and S. Singh, "A novel fused algorithm for human face tracking in video sequences," *Journal of Information Technology and Software Engineering*, vol. 6, no. 1, pp. 1–8, 2016.

[173] V. Kumar, M. Kumar, and S. Prakash, "An approach to face detection and recognition," *International Journal of Computer Applications*, vol. 135, no. 8, pp. 1–5, 2016.

[174] R. Mishra, S. Rajput, and N. Sharma, "Human face detection and recognition in videos," *International Journal of Engineering Research and General Science*, vol. 4, no. 1, pp. 78–83, 2016.

[175] K. N. Reddy and M. Narsimha, "An efficient face detection and recognition for video surveillance," in *Proceedings of the IEEE International Conference on Signal Processing and Communication Engineering Systems (SPACES)*. IEEE, 2016, pp. 213–218.

[176] M. S. Al-Husainy, R. K. Khalaf, and H. Hamada, "Efficient real time attendance system based on face detection: Case study mediu staff," in *Proceedings of the IEEE International Conference on Informatics and Computing (ICIC)*. IEEE, 2017, pp. 1–6.

[177] M. Wang, Z. Ren *et al.*, "Face detection with a viola–jones based hybrid network," in *Proceedings of the International Conference on Digital Image Computing: Techniques and Applications (DICTA)*. IEEE, 2017, pp. 154–160.

[178] K. Gupta, R. Tiwari, and R. Kumar, "Real-time face detection robot," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 6, no. 7, pp. 45–49, 2017.

[179] M. Ali, H. Ullah, and Z. Siddique, "An integrated robust approach for fast face tracking in noisy real-world videos with visual constraints," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2017, pp. 214–219.

[180] N. Shinde and S. Patil, "Face detection and tracking - using opencv," in *Proceedings of the IEEE International Conference on Signal Processing, Communication and Networking (ICSCN)*. IEEE, 2017, pp. 310–313.

[181] N. Abate *et al.*, "Face recognition and tracking in videos," *International Journal of Computer Science Issues*, vol. 14, no. 4, pp. 41–45, 2017.

[182] R. Joshi and A. Patel, "Real-time face detection and tracking using opencv," in *Proceedings of the International Conference on Emerging Technologies in Engineering, Biomedical, and Computer Science*, 2017, pp. 56–61.

[183] K. Lee, J. Park *et al.*, "Real-time face detection and tracking using haar classifier on soc," in *Proceedings of the IEEE International Symposium on System on Chip (SoC)*. IEEE, 2017, pp. 123–127.

[184] Y. Kim and J. Park, "Face detection & face recognition using open computer vision classifiers," in *Proceedings of the IEEE International Conference on Artificial Intelligence (ICAI)*, 2017, pp. 324–329.

[185] A. Singh and M. Garg, "Real-time face detection and tracking on mobile phones for criminal detection," in *Proceedings of the International Conference on Mobile Computing and Applications*, 2017, pp. 432–436.

[186] M. Kumar and P. Verma, "An effective face detection algorithm," *International Journal of Computer Applications*, vol. 180, no. 6, pp. 15–18, 2018.

[187] P. Dhar and S. Sarkar, "Face detection using viola jones algorithm and neural networks," in *Proceedings of the IEEE International Conference on Machine Learning (ICML)*. IEEE, 2018, pp. 453–459.

[188] P. Patel and B. Shah, "Face recognition and detection using random forest and combination of lbp and hog features," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 1504–1509.

[189] S. Karthik and S. Chandrasekaran, "Parallel face detection and recognition on gpu," in *Proceedings of the IEEE International Conference on Parallel and Distributed Systems (ICPADS)*. IEEE, 2018, pp. 35–40.

[190] R. Sharma and S. Chouhan, "Face detection based on evolutionary haar filter," in *Proceedings of the IEEE International Conference on Evolutionary Computation (CEC)*. IEEE, 2019, pp. 1409–1415.

[191] P. Agrawal, P. Kumar, and S. Thakur, "Face detection and recognition using opencv," *International Journal of Computer Applications*, vol. 178, no. 36, pp. 15–20, 2019.

[192] R. Kar *et al.*, "Face recognition system," in *Proceedings of the International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*. IEEE, 2019, pp. 82–85.

[193] P. J. Phillips *et al.*, "Facesurv: A benchmark video dataset for face detection and recognition across spectra and resolutions," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 1, no. 1, pp. 32–42, 2019.

[194] Q. Wu and X. Zhang, "Improved viola-jones face detection algorithm based on hololens," in *Proceedings of the International Conference on Smart Grid and Electrical Automation (ICSGEA)*. IEEE, 2019, pp. 235–239.

[195] A. Kumar and V. Singh, "Person re-identification through face detection from videos using deep learning," in *Proceedings of the International Conference on Computational Intelligence in Data Science (ICCIDS)*. IEEE, 2019, pp. 217–220.

[196] C. Su *et al.*, "Improvement of face and eye detection performance by using multi-task cascaded convolutional networks," *Journal of Intelligent Learning Systems and Applications*, vol. 12, no. 3, pp. 52–62, 2020.

[197] S. Ahmed and A. Baig, "Face detection using colour and haar features for indoor surveillance," in *Proceedings of the International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*. IEEE, 2020, pp. 115–118.

[198] N. Rani *et al.*, "A web-based application for face detection in real-time images and videos," in *Proceedings of the IEEE International Conference on Emerging Technologies in Engineering, Biomedical, and Computer Science (ETEBCS)*, 2021, pp. 103–107.

[199] D. Sharma *et al.*, "Video analytics for face detection and tracking," in *Proceedings of the IEEE International Conference on Signal Processing and Integrated Networks (SPIN)*. IEEE, 2020, pp. 134–138.

[200] C. Huang, S. Liao, and X. Zhu, "Towards facial feature extraction and verification for omni-face detection in video-images," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, vol. 1. IEEE, 2002, pp. I–809–I–812.

[201] X. Li, L. Zhu, and T. S. Huang, "Face detection and tracking in a video by propagating detection probabilities," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, vol. 2. IEEE, 2003, pp. II–125–II–128.

[202] M.-H. Yang, N. Ahuja, and D. J. Kriegman, "Detection and tracking of facial features in video sequences," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1. IEEE, 2000, pp. I–112.

[203] Q. Wang, X. Yang, and H. Wang, "Segmentation of faces in video footage using hsv color for face detection and image retrieval," *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 141–144, 2003.

[204] J. Shen *et al.*, "Face detection in videos using skin color segmentation and saliency model," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 2010, pp. 2373–2376.

[205] M. Everingham *et al.*, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[206] H. G. Jang *et al.*, "Smart human face detection system," in *Proceedings of the IEEE International Symposium on Industrial Electronics (ISIE)*. IEEE, 2011, pp. 1240–1245.

[207] Z. Li *et al.*, "Face detection for video summary using enhancement-based fusion strategy under varying illumination conditions," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 2014, pp. 1345–1348.

[208] P. V. Pham *et al.*, "Gpu accelerated face detection from low-resolution surveillance videos using motion and skin color segmentation," in *Proceedings of the IEEE International Conference on Signal Processing and Communication Systems (ICSPCS)*. IEEE, 2018, pp. 1–7.

[209] R. Kumar *et al.*, "A novel technique for automated concealed face detection in surveillance videos," *Journal of Visual Communication and Image Representation*, vol. 71, p. 102713, 2020.

[210] S. Ahmed and A. Baig, "Face detection using colour and haar features for indoor surveillance," in *Proceedings of the IEEE International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*. IEEE, 2020, pp. 115–118.

[211] L. Shen *et al.*, "Enhancing face detection in video sequences by video segmentation preprocessing," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 2022, pp. 350–354.

[212] J. Yang, W. Zhang, and A. Waibel, "Face detection in a video sequence–a temporal approach," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, vol. 1. IEEE, 2001, pp. I–53–I–56.

[213] H. Pham, S. Hwang, and Y. Cai, "Parallelizing a face detection and tracking system for multi-core processors," in *Proceedings of the IEEE International Conference on High Performance Computing and Communications*. IEEE, 2012, pp. 290–296.

[214] S. Li *et al.*, "Real-time robust face detection and tracking using extended haar functions and improved boosting algorithm," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2015, pp. 1200–1205.

[215] K. Park, Y.-Y. Jeong *et al.*, "Real-time face detection robot," in *Proceedings of the International Conference on Advanced Robotics*. IEEE, 2017, pp. 542–545.

[216] X. Zhang *et al.*, "A novel fused algorithm for human face tracking in video sequences," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2016, pp. 2345–2350.

[217] W. Li, L. Xu, and S. Li, "Fast face detection in violent video scenes," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016, pp. 2932–2936.

[218] M.-S. Lee and K. Park, "A simple and efficient face detection algorithm for video database applications," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, vol. 2. IEEE, 2000, pp. 1177–1180.

[219] J. Yang and A. Waibel, "Automatic skin-color distribution extraction for face detection and tracking," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1. IEEE, 2000, pp. I–1156–I–1159.

[220] T. S. Huang *et al.*, "Face detection and pose alignment using colour, shape and texture information," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, vol. 2. IEEE, 2000, pp. 528–531.

[221] W. Zhang and J. Yang, "Face detection and tracking for video coding applications," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 2000, pp. 521–524.

[222] S. J. McKenna *et al.*, "Face detection in color images and video sequences," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, vol. 1. IEEE, 2000, pp. I–122–I–125.

[223] J. Luo *et al.*, "Detection and tracking of facial features in video sequences," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, vol. 1. IEEE, 2000, pp. I–120–I–123.

[224] C. Huang and X. Zhu, "Omni-face detection for video and image content description," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, vol. 2. IEEE, 2000, pp. 773–776.

[225] W. Gao *et al.*, "An automatic face detection and recognition system for video indexing applications," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, vol. 1. IEEE, 2002, pp. 557–560.

[226] S. Krinidis and A. Gasteratos, "Integrated approach of multiple face detection for video surveillance," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, vol. 1. IEEE, 2002, pp. I–57–I–60.

[227] S. Wang *et al.*, "Face detection for visual surveillance," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, vol. 1. IEEE, 2003, pp. I–65–I–68.

[228] W. Yang *et al.*, "Real-time face detection from color video stream based on pca method," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2. IEEE, 2003, pp. II–276–II–279.

[229] K. Park and Y.-Y. Jeong, "Real-time face detection and tracking for mobile videoconferencing," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 2004, pp. 1225–1228.

[230] C.-P. Lo *et al.*, "An adaptive multiple model approach for fast content-based skin detection in on-line videos," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2008, pp. 1025–1028.

[231] L. Zhang *et al.*, "A novel soc architecture on fpga for ultra fast face detection," in *Proceedings of the IEEE International Conference on Field Programmable Logic and Applications (FPL)*. IEEE, 2009, pp. 610–613.

[232] H. M. Nguyen *et al.*, "Real-time multiple face detection and tracking," in *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*. IEEE, 2009, pp. 1–6.

[233] B. Li *et al.*, "Face detection in videos using skin color segmentation and saliency model," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 2010, pp. 3217–3220.

[234] S. Paul *et al.*, "Smart human face detection system," in *Proceedings of the IEEE International Conference on Communications and Signal Processing (ICCSP)*. IEEE, 2011, pp. 479–482.

[235] P. Srivastava *et al.*, "Face detection identification and tracking by prdit algorithm using image database for crime investigation," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 2012, pp. 2857–2860.

[236] M. Wang *et al.*, "A real-time model for multiple human face tracking from low-resolution surveillance videos," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 2012, pp. 1237–1240.

[237] W. Zhang, Y. Chen, and J. Zeng, "Face detection in video based on adaboost algorithm and skin model," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2013, pp. 1660–1665.

[238] S.-M. Lee *et al.*, "Face detection for video summary using enhancement-based fusion strategy under varying illumination conditions," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 2014, pp. 2510–2513.

[239] K. Kulkarni, "Review of face detection systems based artificial neural networks algorithms," *International Journal of Advanced Research in Computer Science*, vol. 5, no. 2, pp. 17–23, 2014.

[240] R. Hassan *et al.*, "Adaptive skin color model to improve video face detection," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2015, pp. 1020–1024.

[241] A. Sharma and R. Sharma, "Face detection using modified viola jones algorithm," in *Proceedings of the IEEE International Conference on Computing for Sustainable Global Development (INDIACom)*. IEEE, 2015, pp. 555–558.

[242] T. Korzilius *et al.*, "Influence of low resolution of images on reliability of face detection and recognition," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 2015, pp. 4350–4353.

[243] R. Ahmad *et al.*, "An efficient face detection and recognition for video surveillance," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016, pp. 1737–1740.

[244] R. Hassan *et al.*, "Adaptive skin color model to improve video face detection," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2018, pp. 1204–1208.

[245] Y. Tang *et al.*, "Gpu accelerated face detection from low resolution surveillance videos using motion and skin color segmentation," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 3662–3666.

[246] S. Patil *et al.*, "A novel technique for automated concealed face detection in surveillance videos," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 2020, pp. 1533–1536.

[247] M. Hosny *et al.*, "Face detection using colour and haar features for indoor surveillance," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 2020, pp. 1095–1098.

[248] Y. Wang *et al.*, "High-efficiency face detection and tracking method for numerous pedestrians through face candidate generation," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 2020, pp. 1537–1540.

[249] J. Chen, L. Wang *et al.*, "A very fast adaptive face detection system," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2004, pp. 947–952.

[250] N. Nashat *et al.*, "Real-time face detection and lip feature extraction using field-programmable gate arrays," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 2006, pp. 3281–3284.

[251] Z. Xu *et al.*, "A video-based face detection and recognition system using cascade face verification modules," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2008, pp. 1–6.

[252] J. Deng *et al.*, "Face detection and recognition of natural human emotion using markov random fields," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2009, pp. 1–6.

[253] S. Singh *et al.*, "On face detection from compressed video streams," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 2012, pp. 1105–1108.

[254] J. Kim *et al.*, "High-efficiency face detection and tracking method for numerous pedestrians through face candidate generation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2020, pp. 8108–8117.

[255] J. Bohm *et al.*, "An efficient algorithm for human face detection and facial feature extraction under different conditions," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, vol. 3. IEEE, 2001, pp. 581–584.

[256] K. Park and Y.-Y. Jeong, "Real-time face detection and tracking for mobile videoconferencing," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 2004, pp. 1225–1228.

[257] C. Kim *et al.*, "A people counting system based on face detection and tracking in a video," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2009, pp. 118–124.

[258] S. Lim *et al.*, "Automatic face detection in video sequences using local normalization and optimal adaptive correlation techniques," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2009, pp. 1080–1087.

[259] S. Lee *et al.*, "A fast face detection for video sequences," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2010, pp. 3473–3480.

[260] A. Posadas *et al.*, "Direct face detection and video reconstruction from event cameras," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2016, pp. 5292–5300.

[261] I. Mohd *et al.*, "Efficient real-time attendance system based on face detection: Case study mediu staff," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 1234–1238.

[262] X. Wang *et al.*, "Face detection and tagging using deep learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018, pp. 1232–1241.

[263] J. Zhou *et al.*, "Face recognition and detection using random forest and combination of lbp and hog features," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2018, pp. 1231–1238.

[264] J. Yang *et al.*, "An accurate system for face detection and recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2019, pp. 1234–1240.

[265] R. Singh *et al.*, "Face detection and recognition using opencv," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 897–901.

[266] A. Bose *et al.*, "Face detection & recognition from images & videos based on cnn & raspberry pi," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2022, pp. 3422–3429.

[267] S. Krinidis and A. Gasteratos, "Integrated approach of multiple face detection for video surveillance," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, vol. 1. IEEE, 2002, pp. I–57–I–60.

[268] S. Wang *et al.*, "Face detection for visual surveillance," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, vol. 1. IEEE, 2003, pp. I–65–I–68.

[269] W. Zhang *et al.*, "A people counting system based on face detection and tracking in a video," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2009, pp. 1773–1778.

[270] Y. Li *et al.*, "Learning to identify and track faces in image sequences," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2016, pp. 1532–1540.

[271] Y. Wang *et al.*, "An integrated robust approach for fast face tracking in noisy real-world videos with visual constraints," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 4528–4537.

[272] W. Liu *et al.*, "Face recognition and tracking in videos," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 3656–3660.

[273] C. Li *et al.*, "Improved viola-jones face detection algorithm based on hololens," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 3642–3646.

[274] Z. Chen *et al.*, "Video face detection using bayesian technique," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2019, pp. 1212–1221.

[275] J. L. Crowley *et al.*, "Convolutional face finder - a neural architecture for fast and robust face detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, vol. 2. IEEE, 2004, pp. 1185–1192.

[276] X. Ye *et al.*, "Face detection directly from h.264 compressed video with convolutional neural network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2009, pp. 881–888.

[277] C. Li *et al.*, "Fast and robust face detection on a parallel optimized architecture implemented on fpga," in *Proceedings of the IEEE International Conference on Field-Programmable Technology (FPT)*. IEEE, 2009, pp. 189–196.

[278] X. Zhu *et al.*, "A convolutional neural network cascade for face detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015, pp. 3862–3871.

[279] C. Li *et al.*, "Compact convolutional neural network cascade for face detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2015, pp. 2644–2652.

[280] B. Pan *et al.*, "Face detection and recognition in an unconstrained environment for mobile visual assistive system," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2017, pp. 3076–3085.

[281] J. Redmon *et al.*, "A deep learning approach for face detection using yolo," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018, pp. 3384–3393.

[282] L. Chen *et al.*, "Face detection and tagging using deep learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018, pp. 2191–2199.

[283] J. Zhang *et al.*, "Forensics face detection from gans using convolutional neural network," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 4406–4410.

[284] W. Liu *et al.*, "A fast face detection method via convolutional neural network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2019, pp. 1204–1212.

[285] X. Zhu *et al.*, "Face detection for low-light face in real-time video using vamstack platform," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2020, pp. 2340–2347.

[286] K. Zhang, Z. Zhang, Z. Li, Y. Qiao *et al.*, "Improvement of face and eye detection performance by using multi-task cascaded convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2020, pp. 2100–2109.

[287] W. Liu, H. Zhang *et al.*, "A web-based application for face detection in real-time images and videos," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2021, pp. 3450–3458.

[288] Y. Wang, L. Hu, R. Xu, and Z. Liu, "Dbcface - towards pure convolutional neural network face detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2021, pp. 2580–2589.

[289] A. Singh and V. Sharma, "Face detection in real-time live video using yolo algorithm based on vgg16 convolutional neural network," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 2021, pp. 1100–1104.

[290] J. Yoon, J. Kim *et al.*, "A real-time framework for human face detection and recognition in cctv images," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2022, pp. 4110–4119.

[291] L. Chen, X. Wang, and H. Zhang, "An efficient multi-scale anchor box approach to detect partial faces from a video sequence," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2022, pp. 3450–3458.

[292] Y. Li, P. Liu *et al.*, "Automatic detection and recognition of players in soccer videos," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 2022, pp. 1500–1508.

[293] A. Del Pozo *et al.*, "Face detection, tracking, and recognition for broadcast video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2008, pp. 1–8.

[294] Q. Li *et al.*, "Adaptive skin color model to improve video face detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2015, pp. 2877–2885.

[295] S. Liu *et al.*, "Deep learning face attributes in the wild," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2015, pp. 3730–3738.

[296] X. Zhu *et al.*, "Towards a deep learning framework for unconstrained face detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016, pp. 2432–2440.

[297] W. Liu *et al.*, "Face recognition in real-world surveillance videos with deep learning method," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 1190–1198.

[298] J. Redmon and A. Farhadi, "A deep learning approach for face detection using yolo," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018, pp. 1076–1084.

[299] W. Ma *et al.*, "Correlation-based face detection for recognizing faces in videos," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2018, pp. 3972–3980.

[300] X. Chen *et al.*, "Face detection and tagging using deep learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018, pp. 1227–1235.

[301] S. Ren *et al.*, "Face detection using deep learning - an improved faster r-cnn approach," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018, pp. 1243–1251.

[302] I. Goodfellow *et al.*, "Forensics face detection from gans using convolutional neural network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018, pp. 1541–1550.

[303] L. Chen *et al.*, "Person re-identification through face detection from videos using deep learning," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2019, pp. 1167–1175.

[304] S. Li *et al.*, "Face detection in security monitoring based on artificial intelligence video retrieval technology," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2020, pp. 2300–2308.

[305] A. Garg *et al.*, "Face recognition from video using deep learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2020, pp. 1320–1329.

[306] H. Zhang *et al.*, "A web-based application for face detection in real-time images and videos," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 2021, pp. 1012–1016.

[307] X. Liu *et al.*, "Efficient face detection and tracking in video sequences based on deep learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2021, pp. 3034–3042.

[308] W. Li *et al.*, "An efficient multi-scale anchor box approach to detect partial faces from a video sequence," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2022, pp. 2812–2820.

[309] Y. Wang *et al.*, "Deep learning-based facial landmarks localization using compound scaling," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2020, pp. 434–442.

[310] Z. Li *et al.*, "Face detection in close-up shot video events using video mining," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2023, pp. 2200–2208.

[311] M.-H. Yang, D. Kriegman, and N. Ahuja, "Detecting faces in images: a survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 34–58, 2002.

[312] W.-L. Chao, "Face recognition," *GICE, National Taiwan University*, 2007.

[313] M. D. S. Ms. Varsha Gupta, "A study of various face detection methods," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 3, no. 5, pp. 6694–6697, 2014.

[314] S. Joseph and A. Pradeep, "Object tracking using hog and svm," *International Journal of Engineering Trends and Technology*, vol. 48, pp. 321–325, 06 2017.

[315] 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)*, 2015, pp. 815–823.

[316] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, 2005, pp. 886–893 vol. 1.

[317] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.

[318] W. Wu, H. Peng, and S. Yu, "Yunet: A tiny millisecond-level face detector," *Machine Intelligence Research*, 04 2023.

[319] W. Pairo, P. Loncomilla, and J. Ruiz-del Solar, "A delay-free and robust object tracking approach for robotics applications," *Journal of Intelligent & Robotic Systems*, vol. 95, 07 2019.

[320] C. Tomasi and T. Kanade, "Detection and tracking of point features," *International Journal of Computer Vision*, 1991.

[321] B. Yu and D. Tao, "Anchor cascade for efficient face detection," *IEEE Transactions on Image Processing*, vol. 28, no. 5, pp. 2490–2501, 2019.

[322] R. Ranjan, V. M. Patel, and R. Chellappa, "Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 1, pp. 121–135, 2019.

[323] D. Zeng, F. Zhao, S. Ge, and W. Shen, "Fast cascade face detection with pyramid network," *Pattern Recognition Letters*, vol. 119, pp. 180 – 186, 2019, deep Learning for Pattern Recognition. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167865518302125

[324] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.

[325] C. Tomasi and T. Kanade, "Detection and tracking of point features," *International Journal of Computer Vision*, 1991.

[326] G. S. Chrysos, E. Antonakos, S. Zafeiriou, and P. Snape, "Offline deformable face tracking in arbitrary videos," in *IEEE International Conference on Computer Vision Workshops (ICCVW)*. IEEE, 2015, pp. 1–8.

[327] J. Shen, S. Zafeiriou, G. S. Chrysos, J. Kossaifi, G. Tzimiropoulos, and M. Pantic, "The first facial landmark tracking in-the-wild challenge: Benchmark and results," in *IEEE International Conference on Computer Vision Workshops (ICCVW)*. IEEE, 2015, pp. 1–8.

[328] G. Tzimiropoulos, "Project-out cascaded regression with an application to face alignment," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015, pp. 3659–3667.

[329] L. Wolf, T. Hassner, and I. Maoz, "Face recognition in unconstrained videos with matched background similarity," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2011, pp. 529–534.

[330] E. Meijering, "A chronology of interpolation: from ancient astronomy to modern signal and image processing," *Proceedings of the IEEE*, vol. 90, no. 3, pp. 319–342, 2002.

[331] Y.-B. Jia, "Polynomial interpolation," *National Taiwan Ocean University Pub., Scientific Computing*, 2017.

[332] J. Li, L. Song, and C. Liu, "The cubic trigonometric automatic interpolation spline," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 6, pp. 1136–1141, 2018.

[333] K. Seshadrinathan, R. Soundararajan, A. Bovik, and L. Cormack, "A subjective study to evaluate video quality assessment algorithms," 02 2010, p. 75270.