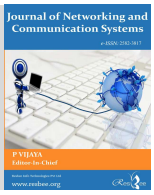
Vol.8 No.1 Jan 2025

Journal of Networking and Communication Systems

Received 4 December, Revised 20 December, Accepted 4 January



Face Detection with MTCNN Using Densenet for Enhance Security

Mehedi Hassan
Institute of Information Technology, Jahangirnagar University, Dhaka-1342, Bangladesh

Arnob Biswas
Institute of Information Technology, Jahangirnagar University, Dhaka-1342, Bangladesh

Nayem Al Hakim
Institute of Information Technology, Jahangirnagar University, Dhaka-1342, Bangladesh

Maisha Mumtaj
Department of Statistics and Data Science, Jahangirnagar University, Dhaka-1342, Bangladesh

SK Al Nahia Samin
Department of Mechanical Engineering, Ahsanullah University of Science and Technology, Dhaka-1208, Bangladesh

Md Sazzad Hossen
Department of Computer Science and Engineering, City University, Dhaka-1215, Bangladesh

Mahir Labib Hossain
Department of Electrical and Electronics Engineering, Ahsanullah University of Science and Technology, Dhaka-1208, Bangladesh

Sajal Panday Soumo
Veterinary Animal and Biomedical Sciences, Sylhet Agricultural University

Auchto Dey
Department of Civil Engineering, Stamford University Bangladesh

Abstract: The research tackles a major security system vulnerability: the ability of current face detection systems to identify changes in circumstances and spoofing attacks. The objective is to create a system that is resistant to security threats and can detect faces reliably under a range of circumstances and scales. The study suggests integrating Dense Convolutional Networks (Dense Net) with Multi-Task Cascaded Convolutional Networks (MTCNN) to do this. Dense Net is a deep learning model that takes advantage of its densely linked layers to improve on MTCNN, a widely used face detection technique. The proposed system improves the performance of face detection, making it more robust and secure. The effectiveness of the combined model is validated through rigorous experimentation and evaluation of standard datasets. The results demonstrate the potential of Dense Net in improving the robustness and security of face detection systems, paving the way for more secure applications in areas such as surveillance, biometric authentication, and social media.

Keywords: Face Detection, MTCNN, DenseNet, Enhanced Security, Spoofing Attacks.

Nomenclature

Abbreviation	Expansion
IIT	Institute of Information Technology
JU	Jahangirnagar University
CNN	Convolutional Neural Network
SVM	Support Vector Machine
MTCNN	Multi-Task Cascaded Convolutional Neural Network
LBP	Local Binary Patterns
R-CNN	Region-based Convolutional Neural Network
YOLO	You Only Look Once
PCA	Principal Component Analysis
RPN	Region Proposal Network
SSD	Single Shot Detector
LFW	Labeled Faces in the Wild
HCI	Human Computer Interaction
AI	Artificial Intelligence
ML	Machine Learning
DL	Deep Learning
Dense Net	Dense Convolutional Networks
RFE	Receptive Field Enhancement
IoU	Intersection over Union
FCN	Fully Convolutional Network
NMS	Non-Maximum Suppression
TPR	True Positive Rate
TN	True Negative
FP	False Positive
FN	False Negative
TNR	True Negative Rate

1. Introduction

Computer intelligence is growing due to the availability of cutting-edge sensing, analysis, and rendering tools and technologies, as well as the quickening pace at which processing power is increasing. Numerous studies and commercial products have shown that computers can interact with people in a natural way by using cameras to see people, microphones to hear people, and an understanding of these inputs to respond to people kindly. Face detection is one of the core methods that makes this kind of organic HCI possible. Face detection is the foundation for all facial analysis algorithms, which include head pose tracking, gender and age recognition, face alignment, face modeling, face relighting, face recognition, face verification/authentication, and many more. The main objective of face detection, given a digital image, is to ascertain whether the image contains faces. For humans, this activity seems simple, but it presents a significant challenge to computers, making it one of the most researched academic areas in recent decades. Numerous differences in scale, position, orientation (in-plane rotation), pose (out-of-plane rotation), facial expression, lighting conditions, occlusions, etc. can be blamed for the challenges faced by face detection.

As a branch of AI, ML focuses on creating statistical models and algorithms that let computers learn from their experiences and become more intelligent without the need for explicit programming. With the use of data, algorithms may identify patterns and independently generate predictions and judgments, all without the need for explicit programming. Through the process of training a model on a labeled dataset, new, unknown data can be predicted by the model through learning and pattern recognition. The intention is to free the computer from human involvement so that it can continue to grow on its own.

DL leverages neural networks to find patterns and relationships in massive datasets, allowing the system to automatically learn from experience and get better at it. It is an essential part of consumer voice control devices, autonomous vehicles, and many other applications that make feasible outcomes that were not before imaginable. DL is used in research related to graphical modeling, FR, audio recognition, computer vision, signal processing, pattern recognition, speech recognition, and audio recognition. Using multiple-layer neural network designs and extensive labeled datasets for training, these models can achieve state-of-the-art precision, sometimes even surpassing human capabilities.

1.1 Convolutional Neural Network

CNNs are a kind of neural network that has completely changed pattern recognition, image processing, and computer vision. Deep neural networks, such as CNNs, are modeled after the composition and operations of the human brain. We shall define CNNs, describe their operation, and discuss some of their uses in this paragraph.

CNNs are made to handle data that has a grid-like layout, like pictures. They are made up of multiple layers, including fully linked, max-pooling, and convolutional layers. Features like edges, lines, forms, and textures are extracted from the input image by the convolutional layers. The computation is more efficient thanks to the pooling layers, which decrease the dimension of feature maps. Ultimately, the task of regression or classification is carried out by the completely connected layers.

The convolutional layer's main concept is to process the image by applying several filters, or kernels. Every filter is a weight matrix, a small matrix that moves across the image and uses the local pixels to do a dot product. A new feature map that emphasizes the existence of patterns in the input image is the end product. One kernel may detect vertical edges, whereas another kernel may detect horizontal edges. Multiple filter applications enable CNNs to recognize increasingly intricate patterns and features.

To reduce the size of the feature map, the pooling layer is usually added after the convlayer. While average pooling calculates the average value, max pooling chooses the highest value in each local area of the feature map. The root mean square of the values in each local region is calculated via L2 pooling. By lessening the network's sensitivity to minute changes in the input image, the pooling operation increases the network's resistance to noise and distortions.

The last classifying or regression task is handled by the fully connected layer. To compute the output, it takes the output from the earlier layers and applies a set of weights and biases. The fully connected layer may generate a probability distribution for each class in a classification problem. It may result in a single output value in a regression task.

When it comes to image processing and computer vision, CNNs are superior to conventional ML algorithms in several ways. Initially, they don't require manual engineering because they can extract high-level features from pixel data. Second, because they are distributed and parallel, they can manage massive datasets containing millions of photos. Thirdly, all layers can be optimized to minimize a loss function if they are trained end-to-end. Because of this, the network may acquire intricate representations of the statistical relationships between the input and output.

CNNs are widely used in computer vision and image processing applications, including segmentation, object detection, face recognition, and captioning of images. CNNs are capable of identifying things in an

image, locating them, and even categorizing them using object detection. CNNs can extract discriminative features from facial photographs and match them to a database of recognized faces in the face recognition domain. CNNs are capable of segmenting images into distinct areas and labeling each one according to its content. Given an image's visual content, CNNs can produce a natural language description of the image for image captioning.

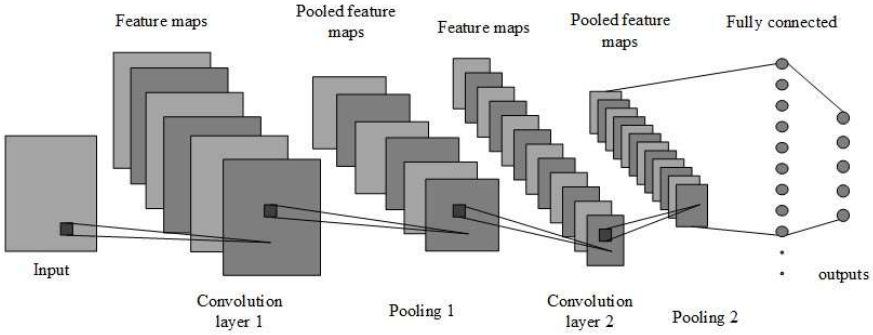


Fig 1. *A CNN with multiple convolutional layers*

CNNs (Fig 1) have also been applied to other domains, like natural language processing, speech, and reinforcement learning. In natural language processing, CNNs can learn to classify text documents into different categories or to generate new sentences that match a given input.

1.2 Convolution Operation Process

In a lot of signal and image processing applications, such as computer vision and DL, convolution operation is essential. That is represented in Fig.2. This mathematical function can be thought of as a filtered version of the original signal because it combines two functions to create a new one. The first step in the convolution process is to choose a pixel from the input image and apply the filter on top of it. The filter is then advanced to the subsequent pixel, and the matching input pixel values and the filter's dot are calculated. Every pixel in the input image goes through this process again, creating an output image or feature map that shows the filtered version of the original image.

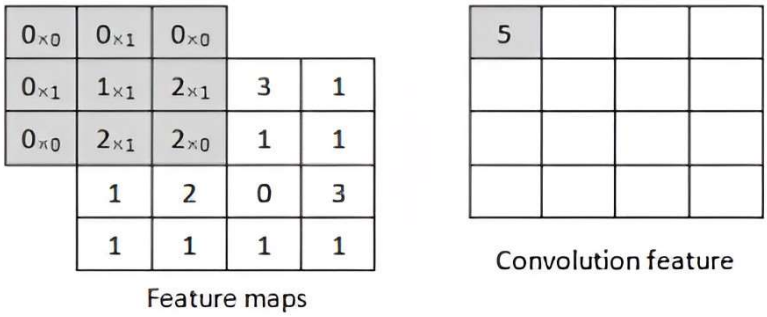


Fig. 2. *Convolution Process*

The size of the resultant image depends on the input image's size, the filter size, and the stride of the convolution operation. The stride determines the step size at which the filter is moved over the input image. A larger stride results in a smaller output image.

1.3 Pooling Layer

This layer, which is usually added after each convolutional layer, is an essential part of CNNs. The layer's goal is to minimize the height and width of the provided feature map while maintaining all its important characteristics. Although there are various kinds of pooling operations, max pooling is the most widely used kind. The input feature map is divided into non-overlapping rectangular using max pooling, which then chooses the maximum value in each sector. A new feature map with fewer parameters and smaller spatial dimensions is the end product.

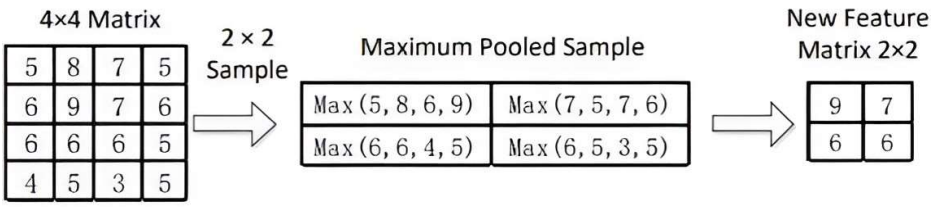


Fig. 3. *Pooling Layer*

Max pooling has various advantages. It first lessens the CNN's sensitivity to minute translations in the input image. Secondly, it lowers the CNN's parameter count, which lessens overfitting and enhances generalization. Thirdly, it reduces the size of the feature maps, which boosts CNN efficiency.

1.4 SVM

It is a potent ML algorithm that is commonly tried for regression analysis and categorization. To do this, a high-dimensional space is searched for the best hyperplane to use to divide two data points with the greatest margin. The distance between the nearest data points from each class and the hyperplane is known as this margin.

When dealing with complicated classification issues with many characteristics and a small number of data points, SVM is especially helpful. SVM uses a kernel approach to transform the input data into a higher space to handle non-linear data. This makes it simple to identify the hyperplane that will accurately separate the data.

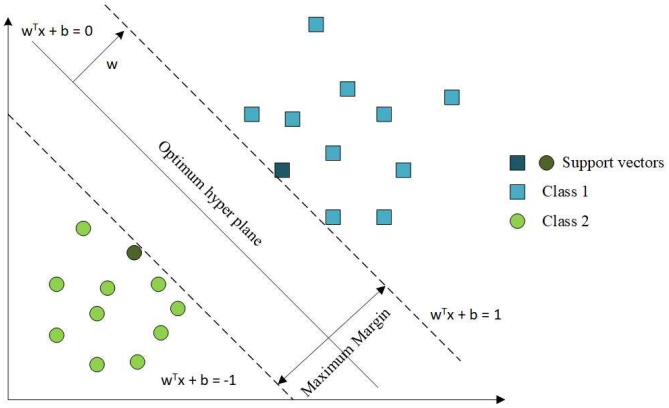


Fig. 4. *SVM with an example*

One of the main advantages of SVM is its robustness to overfitting. It achieves this by maximizing the margin, which reduces the risk of misclassifying new data points.

1.5 Motivation

DL techniques, particularly deep CNN, have shown impressive results in a variety of computer vision tasks in recent years. These tasks include semantic segmentation, object detection, image categorization, and more. DL techniques, as opposed to conventional computer vision methods, have won numerous renowned benchmark evaluations, including the ImageNet Large Scale Visual Recognition Challenge [1], and do not require a hand-crafted design process. Research interest in using DL to solve face identification [2] tasks has increased along with the field's popularity in computer vision.

Much research uses deep CNNs for face detection, motivated by the notable achievements of deep DL techniques in computer vision applications.

1.6 Problem Statement

Though they work well in many situations, current face detection systems frequently have trouble in varied situations with things like lighting, scale, and stance. These systems are also susceptible to spoofing attacks, which include the use of false data to fool the system. Face detection systems' security and robustness are compromised by these issues, which reduces their dependability in crucial applications like biometric authentication and surveillance. The creation of a more reliable and secure face detection system that can manage a variety of situations and fend off spoofing attempts is the issue this thesis attempts to

solve. The suggested method combines the advantages of both models—MTCNN and DenseNet to improve face detection performance.

1.7 Objective

Objectives of this study are-

1. To develop a robust and secure face detection system by combining the strengths of MTCNN and DenseNet.
2. To enhance face detection performance across various conditions such as brightness, occlusion, face alignment, etc., and achieve better accuracy and reliability.
3. To improve security, prevent unauthorized access and ensure system integrity.

1.8 Proposed Method with Key Result

To extract detailed information, the suggested solution combines DenseNet's densely connected convolutional layers with MTCNN's capacity to recognize, classify, and align facial landmarks. We have used SVM to check the accuracy of the classification. Our system provides better detection accuracy and fewer false positives by integrating these designs. Experimental findings demonstrate that our approach outperforms current benchmarks with a whopping detection accuracy of 96.7%. It is also appropriate for real-time deployment because the model processes images at a faster average pace. These results demonstrate our method's potential to improve facial detection technologies for more robust security systems.

The rest of the paper is structured as follows. Chapter II discusses a Literature review describing the existing works. Chapter III discusses the methodology and the performance of the system framework and finally, a conclusion is drawn in Chapter IV.

2. Literature Review

Over the years, numerous worthwhile research initiatives about our subject have been completed. Still, most of the effort yields insufficient outcomes. It takes a lot of research to produce current systems and diverse datasets are also needed. This study proposes the integration of two methods for face detection: MTCNN and DenseNet. Numerous researchers have also used alternative approaches to create their models. This section discusses a few of them.

2.1 Rule-Based Technique-Related Networks

The usage of the Haar Cascade algorithm for human face detection from images with simple and complex backgrounds has been studied by Shahad Salh Ali et al. [3] The AdaBoost cascade classifier, Integral Image, and Haar-like Filter are the three major components of this algorithm. The Haar Cascade algorithm is not without flaws, though: it can mistakenly identify non-face objects as faces due to fundamental characteristics (false positives); it may fail to recognize faces that are partially obscured or oriented unusually (false negatives); it is vulnerable to changes in illumination that affect accuracy; and although it is quick, accuracy is lost in comparison to more recent models.

Yang Tao et al [4] enhance the classical LBP algorithm for face recognition. Their proposed “double circle” LBP addresses LBP’s deficiency by significantly improving rotation invariance. Notably, it performs well on small pose face images. However, it’s essential to recognize its limitations related to spatial support and discriminative capability.

2.2 Feature Based Technique Related Networks

The utilization of the Viola-Jones algorithm for human face detection from images with simple and complex backgrounds is investigated by Mayukh Ghosh et al. [5] The AdaBoost cascade classifier, Integral Image, and Haar-like Filter are the three major components of this algorithm. In addition to cropping and extracting individual faces from photographs with several faces, the study proposes a method for detecting human faces in both vibrant and grayscale photographs. This algorithm, although efficient, has certain drawbacks: it operates successfully when encountered with welllit, frontal views; it struggles while encountered with non-frontal views; it is difficult to deal with partially occluded faces or complex backgrounds; and its performance varies depending on the resolution of the image; high-resolution images can fill the detection window, while low-resolution images lack unique characteristics.

2.3 Template Matching Related Networks

Bnu Utomo Wahyu Mulyono et al. [6] explore the use of the Eigenface approach for human face recognition, an algorithm within PCA used to reduce dimensionality and distribute facial images optimally in facial space. Widely implemented in various research, PCA not only detects human faces under normal conditions but also recognizes images in different expressions and challenges like post-plastic surgery. However, the Eigenface approach, while effective, has limitations: it performs best with well-centered face images, with deviations impacting accuracy; it is sensitive to lighting variations and shadows, affecting recognition reliability; and its effectiveness varies significantly with image scale and resolution, influencing its performance across different image types.

2.4 Deep Learning Related Networks

By training it on the vast WIDER face dataset, Huaizu Jiang et al. [7] investigate the application of Faster R-CNN, which is renowned for its outstanding performance in object detection standards, for face detection. They present innovative results on the WIDER test set, the recently released IJB-A, and two other well-known face detection benchmarks, FDDB [7] Faster R-CNN has limits in face detection, despite its efficacy in object detection. This is because it mainly relies on RPNs to produce candidate regions, which adds an enormous handling burden. The quality of these suggestions for regions is critical to face detection accuracy, and imprecise or inaccurate proposals might negatively affect subsequent years detection stages. Furthermore, Faster R-CNN struggles with non-frontal views, and severe poses, effectively handling blocked faces or challenging backgrounds, whereas it performs well with frontal faces.

Stronger R-CNN-based deep identification of faces is implemented by Hao Wang et al. [8], who highlight that Faster R-CNN is one of the most promising and representative approaches to object detection and is getting more and more well-liked in a range of applications. It's essential to be mindful of its limits, though, as the technique depends on RPNs to generate candidate regions, which can be computationally prohibitive. The precision of these region suggestions has an enormous effect on face detection accuracy, and imprecisions or inaccuracies could adversely affect afterward detection stages. Although Faster R-CNN demonstrates strong results on frontal faces, it continues to have problems with non-frontal views, severe poses, and efficiently handling occluded faces or complicated backdrops, which are still problems in the face detection area.

To identify plausible face regions within images, Qihang Wang et al. [9] offer a face identification technique based on Fast R-CNN that utilizes the use of the CNN, Haar-Adaboost, and candidate search algorithms. Following that, a trained CNN analyzes these regions, conducting pooling and convolution operations to extract final convolution features ROI based on the Fast R-CNN network structure. It's vital to acknowledge its limitations, though: To create candidate regions, Fast R-CNN requires RPNs, which can be quite computationally expensive. The quality of these region suggestions has an important effect on face detection accuracy, and imprecisions or inaccuracies could adversely affect later detection stages. Fast R-CNN does well when dealing with frontal faces, but it struggles when dealing with non-frontal views, extreme poses, obstructed faces, or complicated backdrops. These issues emphasize the continued difficulties in face detection.

Delong Qi et al. [10] present YOLO5Face, a face detector based on the YOLOv5 object detector. Treating face detection as a generic object detection task, the authors make significant modifications to optimize YOLOv5 for face detection, including adding a five-point landmark regression head, using a stem block at the input of the backbone, employing smaller-size kernels in the SPP, and integrating a P6 output in the PAN block. Their study demonstrates that these enhancements enable their face detector to achieve state-of-the-art performance on the WiderFace dataset. However, it's important to acknowledge its limitations: YOLO5Face inherits YOLO's challenge in detecting small objects, which can impact its performance on small or occluded faces. While these modifications improve accuracy, they may also increase computational complexity. Similar to YOLO, YOLO5Face may struggle with non-frontal views and extreme poses, highlighting ongoing challenges in effectively managing diverse facial orientations.

Dweepna Garg et al. [11] focus on improving the accuracy of face detection using DL models, employing YOLO, a popular DL library, for their implementation. The paper compares the efficiency and accuracy of their approach against traditional methods in face detection. Dweepna Garg and colleagues used YOLO5Face, a face detector based on YOLOv5, achieving state-of-the-art performance. However, it's essential to recognize its limitations: YOLO5Face inherits YOLO's challenge in detecting small objects, impacting its performance on small or occluded faces. While modifications enhance face detection accuracy, they may also increase computational complexity. Similar to YOLO, YOLO5Face may struggle with non-frontal views and extreme poses, presenting ongoing challenges in effectively managing diverse facial orientations.

Ziping Yu et al. [12] implemented YOLO-FaceV23, a real-time face detector based on the one-stage detector YOLOv5. They introduce the RFE module, designed to enhance the receptive field for small faces and utilize NWD Loss to mitigate the sensitivity of IoU to the location deviation of tiny objects. Their approach aims to improve detection performance by addressing these challenges. However, it's important to acknowledge its limitations: YOLO-FaceV23 inherits YOLO's difficulty in detecting small objects, potentially affecting its performance on tiny faces. While modifications enhance face detection accuracy, they may also increase computational complexity. Like YOLO, YOLOFaceV23 may face challenges with non-frontal views and extreme poses, highlighting ongoing difficulties in effectively managing diverse facial orientations.

Ning Zhang et al. [13] deeply analyze MTCNN and provide a detailed implementation principle. They compare the characteristics of Two-stage and One-stage detection models in face detection tasks, demonstrating MTCNN's effectiveness through experimental verification. The study evaluates MTCNN's performance using the Wider Face dataset and compares it with the results of the YOLOv3 model. Ning Zhang and colleagues delve into the details of MTCNN for face detection, emphasizing its effectiveness while acknowledging its limitations. MTCNN's multi-stage approach—comprising detection, alignment, and feature extraction—can impact real-time performance, posing challenges in balancing accuracy with computational efficiency. Moreover, MTCNN's reliance on predefined facial landmarks for alignment underscores the importance of landmark quality for accurate detection. Like other methods, MTCNN faces difficulties with non-frontal views and extreme poses, indicating ongoing efforts are needed to improve its capability to handle diverse facial orientations effectively.

Kaipeng Zhang et al. [14] mentioned a deep cascaded multi-task framework exploiting the inherent correlation between face detection and alignment to enhance performance. The framework adopts a cascaded structure with three stages of deep convolutional networks designed to predict face and landmark locations in a coarse-to-fine manner. They introduced a new online hard sample mining strategy during learning to automatically improve performance without manual sample selection. Kaipeng Zhang and colleagues' deep cascaded multi-task framework for face detection and alignment aims to improve accuracy but also faces limitations. The cascaded structure enhances accuracy but potentially increases computational complexity, relying heavily on a large and diverse training dataset for effectiveness. Coarse-to-fine prediction stages within the framework may accumulate errors throughout the cascade, necessitating careful balancing of accuracy and efficiency across stages. While their online hard sample mining strategy enhances performance by targeting challenging cases, it may not fully address issues such as extreme poses, occlusions, and lighting variations, underscoring ongoing challenges in achieving robust face detection and alignment.

Changing Cao et al. [15] discuss an improved version of Faster R-CNN designed specifically for detecting small objects, reflecting advancements in training data and machine performance that have solidified CNNs as the dominant algorithm in object detection. Changing Cao and colleagues propose enhancements to Faster R-CNN aimed at improving performance in detecting small objects. However, it's crucial to acknowledge its limitations: Faster R-CNN, like YOLO, faces challenges in accurately detecting small objects, which can impact its effectiveness with tiny or occluded targets. While modifications aimed at small object detection enhance accuracy, they may also introduce higher computational complexity. Similar to the original Faster R-CNN, effectively managing diverse orientations and extreme poses remains a significant challenge, underscoring the ongoing need for improvements in robustness across various object detection scenarios.

In their influential paper, Shaoqing Ren et al. [16], used a RPN that innovatively shares full-image convolutional features with the detection network, enabling nearly cost-free region proposals. This RPN is meticulously trained end-to-end to produce high-quality region proposals, subsequently utilized by Fast R-CNN for detection purposes. However, despite its advancements, several limitations must be acknowledged. The CNN layers within the RPN are not trained in an end-to-end manner with feedback from the regressor and SVM, impacting the overall efficiency of training. Moreover, each region proposal necessitates a complete pass through the convolutional network, leading to significant memory usage during computations. Furthermore, while the integration of RPN in Faster R-CNN enhances object detection accuracy, its detection speed may not meet the requirements for real-time applications. Nevertheless, the RPN substantially improves region proposal efficiency and contributes to achieving state-of-the-art object detection across diverse datasets. To overcome these challenges, future research should focus on enhancing real-time performance and memory efficiency, while also carefully considering the societal implications of deploying such systems in safety-critical applications.

This study introduces R-CNN, a method developed by Ross Girshick et al. [17] for object detection in images, leveraging a single deep neural network. R-CNN integrates region proposals with CNNs to identify objects within images effectively. However, R-CNN has several drawbacks. Firstly, its multi-stage pipeline—comprising region proposal generation, CNN feature extraction, and SVM classification—results

in significant computational costs, making it slower compared to newer, streamlined methods like Fast R-CNN and Faster R-CNN. Secondly, the training process of R-CNN involves training SVMs individually for each object category, which is resource-intensive and time-consuming. Consequently, its inference speed is comparatively slower due to the multiple sequential steps involved. Lastly, while R-CNN excels in object detection, its localization accuracy may suffer due to potential mismatches between proposed regions and actual objects in images, arising because region proposals are generated independently of the CNN feature extraction process.

LiDAR R-CNN, a second-stage detector that enhances any current 3D detector, is presented by Zhi Chao Li et al. [18]. It shows how R-CNN may be modified for point cloud 3D object identification, which is crucial for autonomous driving. This strategy does have several drawbacks, though. First off, R-CNN's adaption to 3D detection raises computing complexity, which may affect autonomous vehicles' critical real-time performance. Second, in situations where there is sparse or noisy data, performance may suffer due to the dependence on high-quality point cloud data. Finally, it can be difficult to implement LiDAR R-CNN integration with current 3D detectors in situations with limited resources since it necessitates a large amount of computational power and optimization.

Faster R-CNN, an improved R-CNN variant that uses an RPN to produce object placement hypotheses was introduced by Shaoqing Ren et al. [19]. Faster R-CNN can now recognize objects in real-time thanks to this enhancement. Still, there are a few drawbacks. First off, adding RPN to a model makes it faster, but it also makes it more complex, requiring more processing power. To get optimal performance, it can be difficult to train the RPN in conjunction with the detection network, and requires careful parameter tweaking. Finally, even though Faster R-CNN increases the speed of detection, it might still have trouble performing in real-time in high-resolution images or video streams because processing time is still a bottleneck.

Along with another well-liked object identification technique, YOLO, Rohith Gandhi [20] offers a thorough explanation of R-CNN and its variations. Even with a thorough comparison, the topic has some limitations. First off, the overview could be too shallow to fully convey the subtleties and practical difficulties that arise when putting these algorithms into practice—a vital aspect of their real-world applications. Even though the research compares R-CNN with YOLO, it may not adequately address the scenarios in which one algorithm performs better than the other, which could result in an inadequate comprehension of the advantages and disadvantages of each. Finally, a comprehensive picture of the state of the field today may not be provided by the overview, as it may not completely address the most recent developments and enhancements in object identification algorithms outside of

R-CNN and YOLO.

Edgar Kaziakhmedov et al. [21] examine the security of the popular MTCNN face detection system, a cascade CNN, and present a reliable and easily replicable attack technique. The document suggests applying various facial characteristics directly to the face or a medical face mask, printed on a standard white and black printer. This strategy does have several drawbacks, though. First off, the accuracy and quality of the printed attributes may have a substantial impact on the attack's effectiveness, which could limit its consistency. Second, the study's results might not apply to other face identification systems because it mainly focuses on one particular kind of face detection system (MTCNN). Finally, the suggested attack approach may be less applicable if the viability of utilizing these printed properties in dynamic and unpredictable real-world circumstances is not fully investigated.

A face detection method based on Multi-task Cascaded CNN (MTCNN) optimization is put forth by Lijuan Zhang et al. [22]. The research talks about improving face detection performance by fine-tuning the MTCNN model. There are a few drawbacks to this strategy, though. The optimization process can be computationally demanding, involving a substantial amount of time and computing power, which may not be possible for all applications. Second, while enhancing detection accuracy is the paper's main goal, it might not fully solve the trade-offs between accuracy and speed, which are crucial for real-time applications. Last but not least, the optimization strategies covered may be specially designed for particular datasets or circumstances, which could restrict their applicability and efficacy in a variety of real-world settings.

2.5 Review Table

Table 1. *Summary Table of Research*

Type	Algorithm/Technique
Rule Based	Haar Cascades Local Binary Patterns
Feature-Based	Viola Jones Object Detection Framework
Template Matching	Correlation Based Matching Eigenfaces
Landmark-Based	Active Shape Model (ASM), Constrained Local Models (CLM)
DL Based	SSD, R-CNN, YOLO, MTCNN, Faster R-CNN

2.6 Summary of the Research Work

***Table 2**. Review of Existing Literature*

Authors	Methodology	Advantages	Disadvantages
Shahad Salh Ali et al. [3]	Makes use of Haar-like filters, Integral Images, and the AdaBoost cascade classifier to detect faces.	1) Effective for real-time applications and frontal face identification.	1) Less accurate than more recent techniques, 2) Sensitive to illumination 3) Prone to false positives and negatives.
Yang Tao et al. [4]	To increase rotation invariance and performance for small pose face images, a "double circle" LBP was introduced.	1) Include increased adaptation to small pose images. 2) Improved rotation invariance.	1) Has trouble with spatial support. 2) Not very good at discriminating.
Mayukh Ghosh et al. [5]	Uses AdaBoost, Integral Image, and Haar-like Filters to identify faces in both colorful and grayscale photos.	1) In typical circumstances, it is robust for frontal face detection.	1) Has trouble with occlusions, non-frontal perspectives, and different resolutions.
Bnu Utomo Wahyu Mulyono et al. [6]	Under regulated circumstances, PCA is used for facial recognition.	1) Works well for a range of expressions in regulated settings.	1) Sensitive to changes in resolution, image scale, and lighting.
Hao Wang et al. [8]	1) By improving detection phases, enhanced R-CNN is used for deep face identification.	1) Increased precision in difficult circumstances.	1) Constant problems with complicated backdrops and occlusions.
Qihang Wang et al. [9]	For accurate face detection, candidate regions are analyzed.	1) Moderate difficulty. 2) Effective for frontal faces.	1) Problems with extreme poses and occlusions.
Ziping Yu et al. [12]	Expands on YOLOv5 by using NWD Loss and Receptive Field Enhancement to detect small faces.	1) Better recognition of small faces.	1) Occlusions and non-frontal perspectives continue to provide difficulties.
Ning Zhang et al. [13]	Face alignment and detection are done using a multi-stage framework.	1) Excellent frontal face alignment and detection accuracy.	1) The computational inefficiency. 2) Problems with extreme poses.
Kaipeng Zhang et al. [14]	Incorporates cascaded tasks to detect and align joints.	1) Improved face detection precision and adaptability.	1) Increased computing complexity

2.6 Research Gap

Research gaps provide chances for additional development and innovation by highlighting the shortcomings and unsolved issues in current face detection techniques. The gaps found in the existing literature are discussed below:

- **Sensitivity to Changes in Lighting and Pose:** The performance of several conventional techniques (such as Haar Cascade, Viola-Jones, and Eigenface) is constrained in real-world situations by their sensitivity to changes in lighting, position, and picture quality.
- **Unable to Manage Occlusions:** Several methods, such as YOLO-based models and Faster R-CNN, have trouble identifying faces when they are obscured by objects or other faces, which reduces detection accuracy in complicated or crowded scenarios.
- **Problems with Non-Frontal Faces:** Conventional algorithms like Viola-Jones and Haar Cascade are not very good at identifying faces with extreme positions or those that are not frontal.
- **Inefficiency and Complexity of Computation:** Current DL models, such as YOLO5Face, MTCNN, and Faster R-CNN, frequently require large amounts of processing power, which makes them inappropriate for real-time applications on hardware-constrained devices.
- **Poor Performance in Identifying Small Faces:** Several algorithms, such as the R-CNN variation and the regular YOLO, show shortcomings in identifying small or distant faces in images, which lessens their usefulness in high-resolution or wide-field situations.
- **Lack of Robustness to Diverse contexts**: Techniques such as Eigenface and LBP are designed for particular, controlled situations and do not generalize to a variety of dynamic contexts.
- **Restricted Multi-Task Capabilities:** Although models such as MTCNN combine detection and alignment, multitasking frameworks can still be improved to address other problems including facial expressions, gender recognition, and emotion detection.
- **High False Positives and Negatives:** Reliability is decreased by algorithms like Haar Cascade's propensity for false positives and negatives, especially in complicated or cluttered backdrops.

- **Limited Scalability to 3D Data:** R-CNN 3D adaptations, such as point clouds in driverless cars, are computationally costly and have not yet been refined for use in more general face detection applications.
- **Suboptimal Handling of severe postures:** The inadequacy of current approaches (such as Yolo5Face and Faster R-CNN) in managing severe facial postures impacts their applicability in unrestricted environments.
- **Trade-offs Between Accuracy and Speed:** While speedier models like YOLO can compromise detection precision, high-performance models like speedier R-CNN offer accuracy at the expense of speed, leaving a gap for balanced solutions.
- **Inadequate Training for Seldom Occurring Scenarios:** A lot of models perform biasedly because they have not been thoroughly trained on datasets that contain uncommon situations such as partial occlusions, intense lighting, or varied cultural characteristics.
- **Dataset Restrictions:** A lot of approaches use small or out-of-date datasets that don't accurately capture the complexity and variances of face identification jobs in the real world.

These research gaps point to areas where our suggested model can be improved, including creating lightweight, effective models that are better at generalizing in a variety of settings and have improved detection skills for small, obstructed, or non-frontal faces.

3. System Model

The human face is an amazing and complex structure that is essential to our existence. It is essential to social interaction, emotional signaling, identity, recognition, communication, expression, cultural relevance, and evolution. The human face serves as a means of communication and understanding in addition to being a physical characteristic. Finding and recognizing faces in an image or video is known as face detection, and it's a basic computer vision task. Face detection has a significant influence because faces are so important. Security, monitoring, biometrics, authentication, HCI, sentiment analysis, personalization, automated photo tagging, medical imaging, and other applications rely on it. Using MTCNN, DenseNet201, and SVM techniques, this research project detects faces from several datasets and classifies them according to their label extracting some more features. The study aims to enhance the security measures in surveillance, and authentication where faces need to be detected. As this study has used an ensemble technique that has never been used before, it does have a lot of scope for more development.

3.2 Basic Workflow

The working process of the system is depicted in Fig- 5.

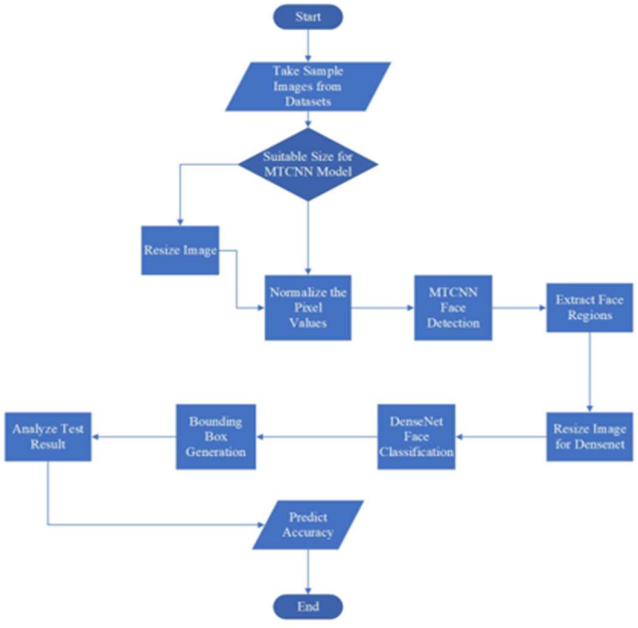


Fig. 5. *Workflow Diagram*

3.3 System Model

3.3.1 MTCNN

Stage 1 - Proposal Network (P-Net): This is the first stage and it is a FCN. The P-Net is used to obtain candidate windows and their bounding box regression vectors. Bounding box regression is a technique to predict the localization of boxes when the goal is detecting an object of some pre-defined class, in this case, faces. Once the bounding box vectors are obtained, overlapping regions are combined through some refining. After refining to reduce the number of candidates, the stage's final output is all candidate windows.

Stage 2 - Refine Network (R-Net): All candidates from the P-Net are fed into the Refine Network. Because of the dense layer at the very end of the network architecture, this network is a CNN rather than an FCN like the previous one. The R-Net utilizes bounding box regression for calibration, NMS for merging overlapping candidates, and further reduction of the number of candidates.

Stage 3 - Output Network (O-Net): This is the final stage where a more complex CNN is used to further refine the result and output facial landmark positions. The MTCNN system quickly and accurately detects, aligns, and extracts facial features from digital photos using a cascading succession of neural networks.

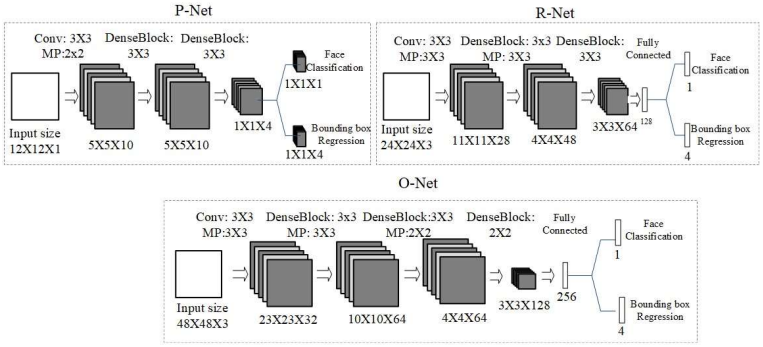


Fig. 6. *MTCNN Face Detection Technique*

3.3.2 Densenet 201

The output of the MTCNN model is fetched to the densenet201 model as input to extract features. It refines the output of MTCNN more and classifies the images into different labels. DenseNet-201 is a deep CNN architecture that stands out for its densely connected structure. Unlike traditional CNNs, where layers are stacked sequentially, DenseNet introduces direct connections between all layers within a dense block. Each layer receives feature maps from all preceding layers, leading to a highly efficient and expressive network. These dense connections facilitate feature reuse, gradient flow, and better parameter efficiency. DenseNet-201, specifically, has 201 layers and has been pre-trained on a large dataset (e.g., ImageNet) for image classification tasks. It's a powerful choice for tasks like object recognition, segmentation, and feature extraction.

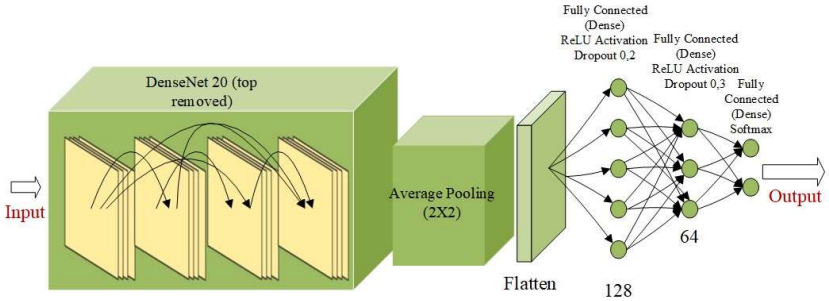


Fig. *7. Densenet*

3.3.3 SVM

After finding out the features and labels, we train an SVM model with those features and labels and test if the MTCNN has detected faces properly or not. An SVM is a supervised ML algorithm used for classification tasks. Given labeled training data for each category, SVM finds an optimal hyperplane (or line) that maximizes the distance between classes in an N-dimensional space. It's particularly suitable for text classification, offering advantages like speed and good performance with limited samples. The goal is

to separate data points into different classes based on their features, making it useful for tasks like natural language processing and image analysis.

3.3.4 Dataset Collection

Labeled Faces in the Wild: The LFW dataset comprises 13,233 face images collected from the web. It features 5,749 unique identities, with 1,680 individuals having two or more images. The dataset serves as a benchmark for unconstrained face recognition, allowing researchers to explore face verification in real-world scenarios [23].

Fig. 8. *LFW Image*

Selfies: The Selfies dataset comprises 5591 sets, which include 2 photos of a person from his documents and 13 selfies. 571 sets of Hispanics and 3512 sets of
Caucasians [24].

Fig. 9*. Selfie Image*

Nepaliceleb: The dataset contains images of 50 famous individuals from Nepal, including actors, actresses, singers, and social figures.

The individuals in the dataset vary in age, gender, and profession. The images were taken from different angles and forms, including front-facing, side-facing, and profile views, to capture each individual's facial features [25].

Fig. 10. *Neplaiceleb Image*

Human Faces: A diverse compilation of human facial images encompassing various races, age groups, and profiles, to create an unbiased dataset that includes coordinates of facial regions suitable for training object detection models [26].

Fig. 11*. Extracted face Image*

Images: A database of face photographs designed for the creation of face detection and recognition models. This dataset has been derived from the LFW Dataset [27].

3.4 Performance Metrics

Evaluation metrics are crucial for figuring out how accurate the classification is. The evaluation metrics that are used here are: Accuracy, LogLoss, ROC curve, Matthew Correlation Coefficient, Balanced Accuracy, and Confusion Matrix for each class and overall. For building the confusion matrix, a few things need to be looked up:

1. True Positive (TP): The number of correctly predicted positive instances (i.e., instances that fall into the positive class) is known as the TP.
2. True Negative (TN): The number of correctly predicted negative instances (i.e., instances that fall into the negative class) is known as the TN.
3. False Positive (FP): Also referred to as Type I error, it is the number of cases that fall into the negative class but were mistakenly predicted as positive.
4. False Positive (FP): Also referred to as Type II error, it is the number of cases that fall into the positive class but were mistakenly predicted as negative.

3.4.1 Accuracy

Accuracy measures the proportion of correctly predicted instances out of the total.

A higher value indicates a better result. The formula to find accuracy is:

$$Accuracy = \frac{Correct\ Prediction}{Total\ Instances}$$

3.4.2 Log Loss (Logarithmic Loss)

Log Loss quantifies the difference between predicted probabilities and actual class labels. Lower values indicate better alignment. The formula is:

$$LogLoss = -\frac{1}{N}\sum_{i=1}^{n}(y_i \log(p_i) + (1 - y_i)\log(1 - p_i)$$

3.4.3 ROC Curve (Receiver Operating Characteristics)

ROC curve is a graphical representation of a classifier's performance across different thresholds. It plots the TPR against the FPR.

The model works better when the area under the ROC curve is higher.

3.4.4 Matthew Correlation Coefficient

Matthew Correlation Coefficient Measures the quality of binary classification predictions, considering true positives, true negatives, false positives, and false negatives. The higher the value of the Mathew Correlation Coefficient the higher accuracy the model has. The formula is:

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

3.4.5 Balanced Accuracy

Balanced accuracy is an adjusted accuracy that accounts for class imbalance. It's the average of sensitivity (TPR) and specificity (TNR). A higher value of balanced accuracy also means higher usability of that model. The formula is:

$$Balanced\ Accuracy = \frac{TPR + TNR}{2}$$

3.5 Experiments and Results

Face detection is a fundamental computer vision task that involves locating and identifying faces within an image or video. After fetching 6 datasets, we found the faces detected from the given datasets and stored them in a different directory with a bounding box and key points- eyes, nose, sides of mouth with features extracted from the images. Then SVM model was trained with these features and labels. After that, different evaluation metrics were evaluated and it came away with really satisfying results.

Fig. 12. *Detected face*

Accuracy Score: Accuracy score = 0.9666496945010183. This Accuracy score notifies us that this technique is very efficient.

Log Loss: Log Loss = 0.13793843960173627. This score is close to 0 means the model is working very well.

Matthew Correlation Coefficient: Matthews Correlation Coefficient: 0.9017407610825772 This score is close to 1 which means the predicted label and the actual has almost a linear relation. That means the model is working well.

Balanced Accuracy: Balanced Accuracy: 0.7953597858710163. This value is close to the accuracy score which means the model is reliable.

Table 3. *Performance Summary*

Accuracy Score	**0.967**
LogLoss	0.138
MCC	0.902
Balanced Accuracy	0.795

ROC Curve: The area under the ROC curve means the efficiency, accuracy, and integrity of a model. The ROC curve for our model is:

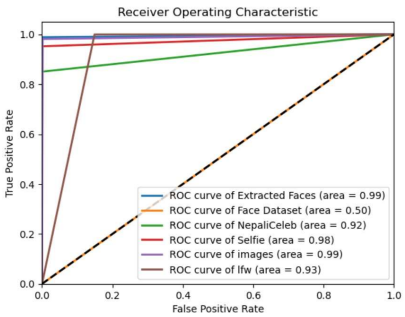


Fig. 13. *ROC Curve*

Here, we can see that the areas are very big. That means the model works fine.

Confusion Matrix: The confusion matrix for our model is given below:-

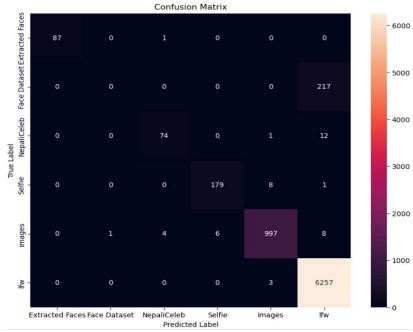


Fig. 14. Confusion Matrix

4. Summary and Conclusion

Our meticulously crafted face detection system, which synergistically combines MTCNN for face localization and key points extraction with DenseNet201 for feature representation, has yielded remarkable results. The high accuracy score of 0.967, coupled with a log loss of 0.138 and a Matthews correlation coefficient (MCC) of 0.902, underscores its robustness. Moreover, the balanced accuracy of 0.795 indicates consistent performance across different classes. The bounding box key points provided by MTCNN and the rich feature vectors extracted by DenseNet201 contribute synergistically to the system's efficacy. In summary, your system detects faces accurately and provides essential facial features for downstream tasks. The combination of MTCNN and DenseNet201 showcases the power of integrating specialized components to achieve superior performance.

As we look for more development, we can consider exploring advanced neural network architectures such as Transformer-based models, capsule networks, or hybrid designs. These innovations promise further gains in accuracy and robustness. Enhance feature extraction using self-attention mechanisms, experimenting with different attention heads and layers for richer representations. Augment your dataset with diverse variations—lighting, pose, ethnicity, and occlusions—to improve generalization. Investigate domain adaptation techniques to handle real-world variations effectively. Address privacy concerns transparently, ensuring individuals are aware of data usage. Regularly audit for biases, implement fairness-aware training, explore federated learning, and fortify the system against adversarial attacks.

Our work impacts surveillance, security, law enforcement, social media, ads, healthcare, and retail. Address biases, educate the public, and collaborate with policymakers for responsible deployment. In summary, our work contributes to a safer, more efficient world embracing facial recognition responsibly, ensuring accuracy and ethical use.

Acknowledgements

We are grateful for the opportunity to all those who provided their valuable support and cooperation in the creation of this report.

This work is done under the supervision of Dr. Jesmin Akhter, Professor, Institute of Information Technology (IIT), Jahangirnagar University, Savar, Dhaka. She has provided us with several books, papers, and journals pertaining to current research during the course of the work. We would not be able to complete the project work efficiently and on schedule without her assistance, courteous support, and the

generous time she provided. We want to express our sincere gratitude to her for her leadership, insightful advice, encouragement, and amicable assistance.

We are incredibly grateful to the Director of IIT at Jahangirnagar University in Savar, Dhaka, for supporting us to deliver the project within the time frame. We also want to express our gratitude to the other IIT faculty members who, whether directly or indirectly, contributed to the success of this project by contributing their great guidance.

We want to thank all the other sources we got useful information. We owe a debt of gratitude to everyone who has assisted us in finishing this task.

Lastly, we would like to express our appreciation to all the staff of IIT, Jahangirnagar University, and friends and family who provided us with encouragement throughout the project.

Compliance with Ethical Standards

Conflicts of interest: Authors declared that they have no conflict of interest.

Human participants: The conducted research follows the ethical standards and the authors ensured that they have not conducted any studies with human participants or animals.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks", Advances in neural information processing systems, vol. 25, 2012.
[2] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification", Advances in neural information processing systems, vol. 27, 2014.
[3] S. S. Ali, J. H. Al'Ameri, and T. Abbas, "Face detection using haar cascade algorithm", in 2022 Fifth College of Science International Conference of Recent Trends in Information Technology (CSCTIT), pp. 198–201, 2022.
[4] Y. Tao and Y. He, "Face recognition based on lbp algorithm", in 2020 International Conference on Computer Network, Electronic and Automation (ICCNEA), pp. 21–25, 2020.
[5] M. Ghosh, T. Sarkar, D. Chokhani, and A. Dey, "Face detection and extraction using viola–jones algorithm", in Computational Advancement in Communication, Circuits and Systems: Proceedings of 3rd ICCACCS 2020, pp. 93–107, 2022.
[6] I. U. W. Mulyono, A. Susanto, E. H. Rachmawanto, A. Fahmi, et al., "Performance analysis of face recognition using eigenface approach", in 2019 International Seminar on Application for Technology of Information and Communication (iSemantic), pp. 1–5, 2019.
[7] H. Jiang and E. Learned-Miller, "Face detection with the faster r-cnn", in 2017 12th IEEE International Conference on Automatic face & gesture recognition, pp. 650–657, 2017.
[8] H. Wang, Z. Li, X. Ji, and Y. Wang, "Face r-cnn", arXiv preprint arXiv:1706.01061, 2017.
[9] Q. Wang and J. Zheng, "Research on face detection based on fast r-cnn", in Recent Developments in Intelligent Computing, Communication, and Devices: Proceedings of ICCD 2017, pp. 79–85, 2019.
[10] D. Qi, W. Tan, Q. Yao, and J. Liu, "Yolo5face: Why reinventing a face detector", in European Conference on Computer Vision, pp. 228–244, 2022.
[11] D. Garg, P. Goel, S. Pandya, A. Ganatra, and K. Kotecha, "A deep learning approach for face detection using yolo", in 2018 IEEE Punecon, pp. 1–4, 2018.
[12] Z. Yu, H. Huang, W. Chen, Y. Su, Y. Liu, and X. Wang, "Yolo-facev2: A scale and occlusion aware face detector", arXiv preprint arXiv:2208.02019, 2022.
[13] N. Zhang, J. Luo, and W. Gao, "Research on face detection technology based on MTCNN", in 2020 International Conference on computer network, electronic and automation (ICCNEA), pp. 154–158, 2020.
[14] 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.
[15] C. Cao, B. Wang, W. Zhang, X. Zeng, X. Yan, Z. Feng, Y. Liu, and Z. Wu, "An improved faster r-cnn for small object detection", Ieee Access, vol. 7, pp. 106838– 106846, 2019.
[16] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks", IEEE Transactions on pattern analysis and machine intelligence, vol. 39, no. 6, pp. 1137–1149, 2016.
[17] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation", in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587, 2014.
[18] Z. Li, F. Wang, and N. Wang, "Lidar RCNN: An efficient and universal 3d object detector", in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7546–7555, 2021.
[19] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks", Advances in neural information processing systems, vol. 28, 2015.
[20] R. Gandhi, "R-cnn, fast r-cnn, faster r-cnn, yolo—object detection algorithms. Medium", 2018.
[21] E. Kaziakhmedov, K. Kireev, G. Melnikov, M. Pautov, and A. Petiushko, "Realworld attack on MTCNN face detection system", in 2019 International MultiConference on Engineering, Computer and Information Sciences (SIBIRCON), pp. 0422–0427, 2019.

[22] L. Zhang, H. Wang, and Z. Chen, “A multi-task cascaded algorithm with optimized convolution neural network for face detection”, in 2021 Asia-Pacific Conference on Communications Technology and Computer Science (ACCTCS), pp. 242–245, 2021.
[23] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, “Labeled faces in the wild: A database for studying face recognition in unconstrained environments”, Tech. Rep. 07-49, University of Massachusetts, Amherst, 2007.
[24] Kaggle, “Selfies id images dataset, 95,000 files”, 01, 2023. 26, 2024.
[25] Kaggle, “Nepali celeb localized face dataset”, 2023, 2024.
[26] Kaggle, “Human faces (object detection)”, 2023, 2024.
[27] Kaggle, “Face recognition dataset - oneshot learning)”, 2021., 2024.