AI Based Face Recognition

A major project report submitted in partial fulfillment of the requirement for the award of degree of

Bachelor of Technology
in
Computer Science & Engineering / Information Technology

Submitted by
Saksham Verma (201248)
Rishik (201543)

Under the guidance & supervision of
Dr. Hari Singh Rawat

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology, Waknaghat, Solan - 173234 (India)

CANDIDATE'S DECLARATION

I hereby declare that the work presented in this report entitled **'AI based Face Recognition'** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering / Information Technology** submitted in the Department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology, Waknaghat is an authentic record of my own work carried out over a period from August 2023 to May 2024 under the supervision of **Dr. Hari Singh Rawat** (Assistant Professor (SG), Department of Computer Science & Engineering and Information Technology).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

(Student Signature with Date)
Student Name: Rishik
Roll No.: 201543

(Student Signature with Date)
Student Name: Saksham Verma
Roll No.: 201248

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

(Supervisor Signature with Date)

Supervisor Name: Dr. Hari Singh Rawat
Designation: Assistant Professor (SG)
Department: CSE/IT
Dated: 18th, May, 2024

ACKNOWLEDGEMENT

Firstly, I express my heartiest thanks and gratefulness to almighty God for his divine blessing which makes it possible to complete the project work successfully.

I am really grateful and wish my profound my indebtedness to Dr. Hari Singh Rawat **,** Assistant Professor(SG), Department of CSE/IT, Jaypee University of Information Technology, Waknaghat for his deep knowledge & keen interest as my supervisor in the field of "**Deep Learning and Data Science**" to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stage have made it possible to complete this project.

I would like to express my heartiest gratitude to **Dr.Hari Singh Rawat,** Department of CSE/IT, for his kind help to finish my project.

I would also generously welcome each one of those individuals who have helped me straight forwardly or in a roundabout way in making this projects a win. In this unique situation, I might want to thank the various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated my undertaking.

Finally, I must acknowledge with due respect the constant support and patients of my parents.

Saksham Verma(201248) & Rishik(201543)

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

CNN	**Convolutional Neural Network**
MTCNN	**Multi Tasked Cascaded Neural Network**
YOLO	**You Only Look Once**
SSD	**Single Shot Detector**
TP	**True Positive**
TN	**True Negative**
FP	**False Positive**
FN	**False Negative**

ABSTRACT

Recently, new generations of computing vision technology have elevated face detection or recognition to different levels used for security, surveillance and computing human interface. In this regard, this report introduces an encompassing framework for face detection and recognition, integrating the strengths of MTCNN (Multi-task Cascaded Convolutional Networks), SSD, YOLO and FaceNet.

To start with, this methodology utilizes MTCNN; a popular modern face detection algorithm which is well known for its precision and speed when locating faces in pictures. MTCNN employs a cascaded network architecture comprising three stages: face detection, bounding box improvement, and facial feature localization. The multi-feature face detection in varying conditions like pose, illumination, and occlusion makes it robust and ensures effective face detection.
Additionally, other object detection algorithms like YOLO (You Only Look Once) and SSD (Single Shot MultiBox Detector) can also be employed for face detection tasks.

YOLO is a real-time object detection system that utilizes a single neural network to predict bounding boxes and class probabilities directly from full images in one evaluation. It is known for its speed and accuracy, making it suitable for real-time applications. YOLO can be trained to detect faces specifically, providing a fast and efficient solution for face detection.

SSD is another popular object detection algorithm that combines predictions from multiple feature maps with different resolutions to handle objects of various scales. It uses anchor boxes of different aspect ratios to predict the bounding boxes and class probabilities. SSD is faster than YOLO and can achieve higher accuracy, making it a viable option for face detection tasks, especially in scenarios with varying face sizes.

CHAPTER–1

INTRODUCTION

1.1 Introduction

In a technology marked via unparalleled technological progress, facial reputation era has turn out to be a transformative, revolutionizing various sectors inclusive of safety, law enforcement, and personalised person experiences. This undertaking targets to leverage the advanced abilities of cutting-edge technologies FaceNet and MTCNN, SSD and YOLO to expand face recognition tailored to a custom dataset. Face recognition entails the identity and verification of people based totally on one of a kind facial features. Deep gaining knowledge of, specially convolutional neural networks (CNNs), has performed a pivotal role in extracting tricky facial styles for correct recognition and authentication. FaceNet, delivered by Google researchers in 2015, represents a groundbreaking approach to face popularity. Unlike conventional techniques, FaceNet learns a excessive-dimensional embedding space, where faces of the same identity are mapped closely collectively, at the same time as faces of various people are separated by a vast margin. This triplet loss approach permits FaceNet to supply tremendously discriminative facial embeddings.

Multi-task Cascaded Convolutional Networks (MTCNN) is a multi-level architecture designed for face detection, bounding container regression, and facial landmark localization. Its cascaded shape ensures efficient processing of faces across special scales, making it best for sturdy face detection in actual-world scenarios.

The motivation propelling this undertaking emanates from the crucial want for a versatile and accurate face recognition gadget capable of adapting to the demanding situations provided by using numerous datasets. This encompasses versions in facial expressions, poses, and lighting conditions. It is inside this problematic panorama that the mixing of FaceNet, MTCNN, Yolo and SSD aspires to excel, presenting a comprehensive solution able to tuning in reliable overall performance.

Challenge embarks at the implementation of FaceNet to extract nuanced facial functions and generate embeddings that encapsulate the specific traits of each person's face. Simultaneously, MTCNN, Yolo and SSD are seamlessly incorporated into the pipeline to carry out specific face detection. This ensures the correct identity and extraction of applicable facial regions for subsequent processing.

The middle of this mission lies within the meticulous curation and usage of a custom dataset. This dataset, designed with utmost care, includes a numerous variety of facial expressions, poses, and lighting conditions.

The project is dedicated to a radical evaluation method, employing metrics including accuracy, speed, and flexibility. Rigorous validation methods could be carried out to make certain the robustness of the proposed system across various environmental conditions. This complete assessment serves not best as a metric for achievement however additionally as a mechanism for continuous development.

By correctly achieving the outlined goals, this assignment ambitions to set a new standard for facial recognition structures. The anticipated final results is going past technological innovation, providing a strong foundation for superior security, streamlined get admission to manipulate, and seamless human-laptop interactions.

The scope of this undertaking extends past the mere integration of FaceNet with MTCNN, Yolo and SSD. It aspires to contribute appreciably to the evolution of facial popularity systems, providing a solution that no longer best excels in accuracy but also proves versatile in practical programs. The incorporation of a custom dataset introduces an element of specificity, making sure that the device is finely tuned to the challenges presented by using centered use instances.

By efficaciously attaining the outlined goals, this venture objectives to set a new widespread for facial popularity structures. The predicted outcome goes past technological innovation, providing a solid basis for greater safety, streamlined get admission to manipulate, and seamless human-pc interactions.

The subsequent sections of this record will provide a detailed exploration of the technical aspects of FaceNet implementation, the structure of the incorporated machine, the methodology for dataset education, and a complete evaluation of the machine's overall performance through vast assessment and trying out.

Technical Aspects of FaceNet Implementation

FaceNet represents a pivotal breakthrough within the subject of face reputation. The version architecture is characterized by way of its use of a deep convolutional neural community to generate fairly discriminative facial embeddings. These embeddings function compact representations of facial functions, facilitating green and accurate matching of faces.

The key innovation of FaceNet lies in its use of a triplet loss characteristic for the duration of training. This loss feature enforces a margin between the distances of embeddings for faces of the same identification and people of different identities. This guarantees that faces of the identical person are embedded closely together inside the excessive-dimensional area, even as faces of various individuals are fairly separated. The end result is a powerful embedding area where similarity among faces is directly correlated with their actual identity.

Architecture of FaceNet

The FaceNet architecture comprises several essential components:

Inception Modules: FaceNet employs Inception modules, which might be convolutional neural community constructing blocks that permit for the parallel operation of a couple of filters of various sizes. This enables the network to capture features at diverse scales.

Triplet Loss Function: The triplet loss characteristic is essential to the achievement of FaceNet. It entails selecting 3 pics for each schooling iteration: an anchor picture (the picture of the character whose identity is being discovered), a fine image (any other picture of the same individual), and a terrible picture (an photo of a special person). The intention is to minimize the gap between the anchor and superb embeddings while maximizing the gap among the anchor and bad embeddings.

Normalization: Normalization techniques, such as batch normalization.

Siamese Network Structure: FaceNet employs a Siamese network structure, where two identical subnetworks share the same parameters. This structure enables the network to learn a mapping that brings the embeddings of similar faces closer together in the embedding space.

The combination of these elements results in a powerful face recognition model that excels in producing highly discriminative embeddings, making it well-suited for real-world.

Technical Aspects of MTCNN implementation

While FaceNet handles the extraction of facial features and embeddings, MTCNN serves as the pre-processing stage responsible for accurate face detection. MTCNN is designed to detect faces in images and provide bounding boxes around the detected faces. Its cascaded architecture ensures effective handling of faces at different scales, making it particularly robust in challenging scenarios.

Cascaded Architecture

MTCNN consists of three stages, each with a specific task:

Stage 1: Face Detection - In the first stage, a coarse face region is identified using a convolutional network. This stage helps filter out non-face regions efficiently.

Stage 2: Bounding Box Refinement - The second stage refines the bounding box obtained from the first stage. It uses more complex features to identify facial landmarks and improve the accuracy of the bounding box.

Stage 3: Facial Landmark Localization - The final stage precisely localizes facial landmarks, such as eyes and mouth, within the refined bounding box. This stage provides additional information for fine-tuning the bounding box and aligning detected faces.

MTCNN excels in handling faces with variations in scale, pose, and illumination. However, it is not immune to challenges.

Technical Aspects of Yolo Implementation

While YOLO was originally designed for general object detection, its flexibility and speed make it a popular choice for face detection tasks. However, it's important to note that the performance of YOLO for face detection largely depends on the quality and diversity of the training dataset, as well as the proper configuration of hyperparameters and post-processing steps.

The stages of YOLO face detection can be summarized into the following three main steps:

Stage 1 : Model Forward Pass and Bounding Box Prediction – The input image is preprocessed in order to normalize and resize it. The following images are fed into our Yolo model in order to extract features and predict bounding boxes.

Stage 2 : Post-processing and filtering – Applying Non-Maximum Suppression to remove overlapping and redundant bounding box predictions, keeping only the most confident ones.

Stage 3 : Face extraction - Use the remaining high-confidence bounding boxes to extract the face regions from the input image.

These three main steps encapsulate the core stages of the YOLO face detection process, from the initial model forward pass and bounding box prediction to the final extraction of face regions after post-processing and filtering.

Technical Aspects on SSD implementation

The SSD approach to face detection involves a single deep convolutional neural network that performs object detection in a single forward pass. The network architecture consists of a base network, such as VGG-16 or ResNet, for extracting features from the input image, followed by additional convolutional layers for prediction.

The stages of SSD can be written into the following:

Stage 1 : Network Architecture – SSD is a singe deep neural network that performs object detection in a single forward pass resulting in it being faster when compared to other networks. It consists of a base network such as VGG-16 for feature extraction, followed by additional convolutional layers for prediction.

Stage 2 : Anchor Boxes – SSD uses anchor boxes of different aspect ratios and scales to handle objects of different sizes and shapes.

Stage 3 : Inference and Post-Processing – During inference, feed input images to the trained SSD Model to obtain face bounding box predictions and classifications scores.

1.2 Problem Statement

Face detection and popularity are essential obligations within the subject of imaginative and prescient and have gained considerable interest due to their huge range of programs in diverse domain names inclusive of safety, surveillance, human-computer interplay, and entertainment. Traditional procedures to face detection and popularity relied on handcrafted functions and classical device mastering algorithms. However, these methods regularly suffer from limitations including low accuracy, poor performance below various lights situations, and difficulty in handling occlusions.

With latest advancements in synthetic intelligence (AI) and deep mastering, there was a paradigm shift in face detection and reputation strategies. AI-primarily based tactics harness the energy of deep neural networks to routinely study discriminative functions from big quantities of categorized face information. This has led to big upgrades in the accuracy and robustness of face detection and recognition structures.

However, no matter the development made in AI-primarily based face detection and popularity, several challenges and limitations persist. One of the primary demanding situations is the detection and reputation of faces in unconstrained scenarios, where the lighting fixtures conditions, pose, and facial expressions can vary significantly. Traditional strategies regularly war to handle these versions, resulting in fake positives or false negatives. Another undertaking is the presence of occlusions in facial images, along with glasses, scarves, or facial hair. These occlusions can hinder accurate face detection and reputation, as they hinder key facial capabilities. AI-primarily based tactics want so that it will handle such occlusions and nonetheless accurately pick out and classify faces.

Furthermore, privateness and moral issues additionally get up inside the context of face detection worries, optimize computational performance, and ensure independent overall performance throughout numerous populations.

The purpose is to triumph over the constraints of conventional procedures and boost the modern day in face detection and recognition, in the end contributing to the improvement of greater dependable and green structures that can be deployed in actual-world programs.

1.3 Significance and Motivation

1. Improve the accuracy and robustness of face detection and popularity in unconstrained

eventualities: The first objective is to expand AI-based totally algorithms that can as it should be discover and apprehend faces in various difficult situations, consisting of varying lighting fixtures, pose, and facial expressions. This will contain schooling deep neural networks on our custom face dataset to research discriminative functions which could take care of thoseversions.

2. Handle occlusions and improve the performance of face detection and popularity within the presence of obstructed facial capabilities: The second goal is to broaden algorithms that could effectively manage occlusions inclusive of glasses, scarves, or facial hair. This will contain exploring strategies which include facial landmark detection and function-based totally alignment to enhance the accuracy of face reputation even when key facial capabilities are in part obstructed.

3. Optimize computational performance for actual-time face detection and recognition: The third objective is to optimize the computational complexity of AI-based algorithms to gain actual-time processing of massive-scale face datasets. This will contain exploring strategies which includes version compression, hardware acceleration, and parallel processing to enhance the performance of the algorithms without compromising accuracy.

4. Ensure unbiased performance throughout various populations: The forth objective is to increase face detection and popularity structures which can be independent and perform consistently throughout various populations. This will contain collecting and curating numerous face datasets that encompass diverse demographics and facial characteristics to make sure truthful illustration and accurate reputation for all people, no matter gender, age, race, or other factors.

1.4 Methodology

The technique for face detection and reputation using AI-primarily based strategies, in particular MTCNN (Multi-task Cascaded Convolutional Networks) for face detection and FaceNet for face popularity, may be mentioned as follows:

1. Data Collection and Preprocessing: Gather a custom construct dataset of face images that encompasses diverse lights situations, poses, and facial expressions. Preprocess the dataset by way of normalizing the photos, resizing them to a steady decision, and applying any essential photograph enhancement strategies to enhance the excellent of the snap shots.

2. Face Detection: Utilize the MTCNN algorithm for face detection. MTCNN is a deep studying-primarily based face detection set of rules that detects faces in an image by using a cascaded architecture of 3 neural networks. The first network proposes potential face areas, thesecond network refines the bounding containers, and the 0.33 network performs facial landmark localization. Implement MTCNN the usage of the to be had libraries or frameworks.

3. Face Alignment: After detecting faces, perform face alignment to normalize the face photographs. This step entails identifying facial landmarks inclusive of the eyes, nose, and mouth, after which making use of geometric differences to align the faces based totally on these landmarks. Face alignment helps in lowering versions as a result of unique head poses and improves the accuracy of next face recognition.

4. Feature Extraction: Use the FaceNet version for function extraction. FaceNet is a deep convolutional neural community that maps face pix into a excessive-dimensional function space, where faces of the equal identification are close to each different and faces of various identities are far apart. Apply the pre-skilled FaceNet version to extract a fixed-period characteristic vector for every aligned face photo.

5. Face Recognition: Perform face reputation by way of evaluating the extracted feature vectors. Use a suitable distance metric, which include Euclidean distance or cosine similarity, to degree the similarity between function vectors. Set a threshold to decide whether or not two face photos belong to the identical man or woman or not. If the distance/similarity is underneath the brink, classify the faces as a match; in any other case, classify them as exclusive individuals.

6. Evaluation and Fine-tuning: Evaluate the performance of the face detection and recognition gadget the usage of appropriate assessment metrics inclusive of accuracy, precision, remember, and F1 score. Fine-tune the machine by means of iteratively adjusting the parameters, schooling statistics, or community structure to improve the general overall performance.

7. Optimization for Real-time Processing: Optimize the computational efficiency of the machine to obtain real-time processing. This can involve strategies along with version compression, hardware acceleration, and parallel processing to reduce the inference time of the face detection and reputation algorithms.

1.5 Organization

Chapter 1: Introduction

1.1 Overview

The project initiates with Chapter 1, the Introduction, offering a concise insight into the topic's significance. This chapter addresses the problem statement, methodology, and study objectives.

1.2 Problem Statement

The Introduction delves into the problem statement, identifying key challenges in the project's domain.

1.3 Methodology

A comprehensive methodology, including data receiving, model training, and testing phases, is detailed in Chapter 1. The experimentation process and initial results are discussed.

1.4 Objectives of the Study

Outlined in this chapter are the objectives that steer the research, guiding the subsequent chapters.

Chapter 2: Literature Survey

Chapter 2, titled Literature Survey, thoroughly reviews prior studies in the field. Special focus is placed on MTCNN, SSD and Yolo, various data collection techniques and artificially increasing our dataset. FaceNet.

2.1 Overview of Relevant Literature

2.2 Key Gaps in the Literature

Chapter 3: System Development

Chapter 3 elaborates on the project's development, encomptraining, and testing phases.

3.1 Requirements and Analysis

3.2 Project Design

A thorough discussion on the project design and architecture of our project.

3.3 Data Preparation

3.4 Implementation

3.5 Key challenges

Chapter 4: Testing Strategy

4.1 Testing Strategy

Chapter 4 unravels the testing strategy used in the project.

Chapter 5: Results and Evaluation

5.1 Results we achieve in this project.

Chapter 6: Conclusions and Future Scope.

Chapter 6, the Conclusion, recaps the project and suggests future improvements. It proposes the incorporation of additional parameters to enhance water clarity, asserting that precision increases with the inclusion of more parameters.

Chapter 6: References

6.1 Citations and Sources

The project concludes with a comprehensive reference section, citing research studies, real-time values, algorithms, and other relevant sources.

This restructuring aligns with the provided organization, maintaining clarity and coherence across chapters.

CHAPTER – 2

LITERATURE REVIEW

2.1 Literature Review

1. Automatic Pain Estimation from Facial Expressions: [1]

The paper titled "Automatic Pain Estimation from Facial Expressions: A Comparative Analysis Using Off-the-Shelf CNN Architectures" presents a comprehensive evaluation of computerized ache intensity evaluation from facial expressions the usage of off-the-shelf CNN architectures. The study compares the overall performance of five popular CNN architectures, specifically MobileNet, Google Net, ResNeXt-50, ResNet18, and DenseNet-161, in estimating ache intensity from facial expressions.

2. Face Recognition and Identification using Deep Learning Approach: [2]

The paper titled "Face Recognition and Identification the usage of Deep Learning Approach" discusses the CNN-based totally face recognition technique to come across faces in actual-time. It compares the accuracy ranges of actual-time face recognition and picture-based face popularity and explores the motives for image-primarily based face reputation being more accurate.

While this paper offers insights into real-time utility and assessment between real-time and photograph-primarily based face recognition, it does now not speak a selected face detection architecture in element. Instead, it makes a speciality of the troubles confronted while imposing a face reputation model.

To address the demanding situations noted on this research paper, which include decrease accuracy below varying mild situations and the inability of the model to discover small faces, the MTCNN model with FaceNet can be used. This aggregate has been shown to providehigher accuracy as compared to different present fashions.

3. Real-time Detection, Tracking, and Recognition Algorithm based on Multi-target Faces [3]

The paper titled "Real-time Detection, Tracking, and Recognition Algorithm based on Multi-target Faces" details a multi-target face actual-time detection, tracking, and reputation algorithm. It discusses three methods: rapid-tracking, rapid detection, and short recognition, using MTCNN and FaceNet.

This paper gives an amazing overview of the running of the MTCNN algorithm and how distinct CNN layers paintings within it. However, it basically specializes in the face detection structure and does not delve into the popularity architecture in detail.

To enhance the MTCNN structure, better function selection and reusing functions while the usage of numerous CNN layers can be carried out. One technique is to apply an additional layer called a totally connected layer to upscale the functions and utilize them efficaciously.

4. MTCNN and FaceNet Based Access Control System for Face Detection and Recognition [4]

The paper titled "MTCNN and FaceNet Based Access Control System for Face Detection and Recognition" proposes an progressed FaceNet network. It makes use of the multi-project cascaded convolutional neural networks (MTCNN) for fast face detection and alignment, observed by means of FaceNet with an improved loss function for face verification and recognition with excessive accuracy.

This paper discusses each the face detection and face recognition structure in detail. It additionally highlights the loss characteristic used to improve accuracy However, it need to be cited that the MTCNN architecture discussed in this paper is not primarily based at the trendy advancements.

To beautify the MTCNN architecture, higher function selection and reusing features while the use of various CNN layers can be carried out. This may be performed by way of incorporating an extra completely linked layer to upscale the functions and utilize them correctly.

5. Lightweight and Resource-Constrained Learning Network for Face Recognition [5]

Learning Network for Face Recognition with Performance Optimization" is to present a light-weight and resource-restrained studying community for face reputation. The paper introduces the FN13 model, that's based on the FaceNet version and carries center loss as its loss characteristic.

The FN13 version is designed to gain high accuracy at the same time as minimizing the range of parameters and computational assets required. This makes it properly-desirable to be used in aid-confined environments.

6. Deep learning-based image recognition for autonomous driving [6]

This article focuses on the use of deep learning, particularly CNN, in image recognition and self-driving cars. Perception systems are the main components of autonomous driving as they need properly recognize the environmental conditions. Therefore, CNNs are used widely because of their suitability for processing image data and hence, Alexnet architecture is implemented herein.

7. Face mask recognition system using CNN model [7]

This paper proposes a COVID-19 face mask recognition system utilizing CNN . This system looks for facial images or videos that indicate whether somebody is wearing a face mask or not. Here, the approach used involves CNN as a deep learning architecture to design an effective network for detecting facemasks. The goal is to develop a customized trained CNN model for determining if an individual is wearing the COVID-19 infection mask.

8. Human face recognition based on convolutional neural network and augmented dataset. [8]

A novel method has been designed in this paper for handling small original datasets related to human recognition of faces. These transformations of the face images are used to augment the original small dataset to become a large dataset. The advanced CNN of a remarkably augmented face dataset for effective implementation of the face recognition. To this end, several experiments are conducted for verifying the efficacy of the augmented dataset, and it is possible to compare the novel approach with other commonly employed face recognition techniques as well.

9. A Face Recognition Method Based on CNN [9]

This paper obtains higher accuracy by using the LeNet-5 handwritten digits recognition algorithm. Various experimental parameters are changed to find the current best CNN model. The two-feature convolutional layer is then used to obtain the maximum recognition rate.

However, like all other nets, this one cannot perform optimally for all cases. Its vulnerable to an image database while a network model can classify and identify natural pictures.

10. Application of Face Recognition Method Under Deep Learning Algorithm in Embedded Systems [10]

An algorithm for face detection under OMTCNN and an algorithm of face recognition. a new architecture for DNN under LCNN. Assessment and testing for performance. show that OMTCNN yields a good precision on the LFW FV set. its loss function becomes stable, and it reaches 95.78%. The face recognition accuracy of However, other forms of sub-modular LCNN are comparatively simpler to implement. recognition methods. More specifically, it showed high accuracy in the LFW verification dataset. Its computational time consumption performance has become 98.13%, and performance has become maximum.

11. Efficient Face Recognition System for Operating in Unconstrained Environments [11]

The purpose of this paper was to put forward a scheme for introducing a facial recognition system using a real-time video. The problem was therefore studied in a systematic way by considering various approaches to machine learning and deep learning algorithms. This review provided better schemes for facial detection, feature extraction, and classification tasks respectively. YOLO-Face is one of the recent developed one-stage deep learning detectors evaluated against many other contemporary face detection algorithms like handcrafted-based as well as some deep learning-based methods.

12. Data Augmentation: A survey of modern approaches [12]

In this paper, Author provides an overview of data augmentation approaches in computer vision in this paper. This study focuses on general principles, use cases and various applications problems by modern data augmentation techniques. Some of the primary studies surveyed are traditional approaches as well as modern techniques dependent on regional level transformation plus feature space expansion, as well as synthetic data synthesizing methods such as CAD modeling, neural style transfer, generative modeling, and neural rendering.

In end, the literature assessment highlights diverse studies papers that talk the usage of MTCNN and FaceNet in face detection and popularity duties. These papers offer insights into the structure, performance, and demanding situations associated with those fashions. To enhance the MTCNN architecture, higher characteristic selection, reusing features, and incorporating extra layers may be explored. Additionally, the FaceNet model may be greater with the aid of optimizing the loss function and considering aid constraints.

13. YOLO5Face: Why reinventing a face detector. [13]

In this paper, author discusses about the YOLO object detection architecture, treating face detection as a general object detection task and implementing a face detector vased ib YOLO5, several modifications are made to the YOLOv5 architecture to improve performance for face detection. It also discusses the difficulties an architecture like YOLO faces in real time face detection and how it differs from normal object detection.

14. A Comprehensive Review of YOLO: From YOLOv1 to YOLOv8 and Beyond. [14]

In this paper, author discusses in great detail about all the YOLO versions released from one to eight, their improvements over one another and the complexities each version has. It also dives into the architectures of all the YOLO versions and how they perform real time object detection in such a fast manner.

15. SSDMNV2: A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2. [15]

In this paper author discusses about the the Single Shot detector and how it works for real time object detection. In the following paper it is combined with MobileNet in order to perform real time face mask detection and recognition.

2.2 Key Gaps in Literature

Identifying key gaps in a literature survey is crucial for refining your understanding of the research landscape and setting the direction for future work. Here are some potential gaps that we considered in our literature survey of face detection and recognition using MTCNN (Multi-task Cascaded Convolutional Networks), Yolo (You Only look once), SDD (Single Shot Detector) and FaceNet:

Table 2.1

S. Number	Papers	Scope of Improvement
1	Automatic Pain Estimation from Facial Expressions: A Comparative Analysis Using Off-the-Shelf CNN Architectures	These off the shelf CNN architectures can be improved when used as features extracters and are used to provide inputs to models such as MTCNN .
2	Face Recognition and Identification using Deep Learning Approach	The problems discusses in this research paper such as lower accuracy under varying light conditions and unablility of model to detect small faces can be improved by using MTCNN model with facenet. It provides higher accuracy when compared to other existing models.
3	Real-time detection tracking and recognition algorithm based on multi-target faces	MTCNN architecture can be improved by using better feature selection and reusing features when using various CNN layers. This can be done by using an extra layer known has fully connected layer to upscale the features and use them.
4	MTCNN and FACENET Based Access Control System for Face Detection and Recognition	MTCNN architecture can be improved by using better feature selection and reusing features when using various CNN layers.
5	Lightweight and Resource-Constrained Learning Network for Face	Its ability to achieve high accuracy while minimizing the number of parameters and computational resources

	Recognition with Performance Optimization	required makes it well-suited for use in resource-constrained environments.
6	Deep learning-based image recognition for autonomous driving	Visual explanation to verbal explanation in autonomous driving.
7	Face mask recognition system using CNN model	The proposed mask recognition system can become more accurate, robust, efficient, and adaptable, making it a valuable tool for enforcing mask-wearing policies and promoting public health and safety.
8	Human face recognition based on convolutional neural network and augmented dataset	Training strategies, Performance evaluation metrics
9	A Face Recognition Method Based on CNN	The field of research is dynamic, and advancements in algorithms, techniques, and datasets continue to drive progress in improving the accuracy, applicability of CNN models in face recognition tasks.
10	Application of Face Recognition Method Under Deep Learning Algorithm in Embedded Systems	
11	Efficient Face Recognition System for Operating in Unconstrained	This field of research is dynamic and depends on various factors such as data set being used, parameters selected etc.

	Environments	
12	Data Augmentation: A survey of modern approaches	
13	YOLO5Face: Why Reinventing a face detector.	Pros and cons of each and every YOLO version, YOLO5 being the most accurate but slow.
14	A comprehensive review of YOLO. From YOLO1 to YOLO8 and beyond	
15	SSD: A real-time DNN-based face mask system using single shot multibox detector.	

CHAPTER - 3

SYSTEM DEVELOPMENT

1. Problem Definition and Objectives:

In tackling the intricacies of face detection and reputation, the tool objectives to cope with the challenges posed by way of using the custom dataset which includes more than 5400 photos. The key objective is to put into effect a complex answer the use of MTCNN for particular face detection and FaceNet for correct face popularity. The holistic approach entails now not handiest putting in the presence of faces in photos however additionally growing a discriminating characteristic space for powerful facial identification. The nuanced nature of the goals necessitates a balanced attention of things collectively with photograph high-quality, lighting fixtures conditions, and facial expressions in the dataset

2. Data Collection:

The custom dataset, a cornerstone of the machine's improvement, is carefully curated to encapsulate various scenarios, making sure its representativeness in real-world packages. The origins and contextual relevance of the pics in the dataset are expounded upon, shedding light at the intricacies of records collection. Furthermore, an in depth exploration of the dataset's distribution across education, trying out, and validation sets is supplied, allowing for a complete understanding of the way the device is exposed to numerous eventualities. The pre-processing steps, together with resizing and normalization, are meticulously specific, emphasizing their role in growing a standardized and conducive environment for model training.

3. Data Augmentation:

Subsequent to data collection, data augmentation techniques are applied to enhance the diversity and size of the dataset. Techniques such as rotation, scaling, flipping, and adjustments in brightness are introduced to artificially introduce variability. Augmenting the dataset in this manner aids in improving the model's robustness by exposing it to a broader range of facial expressions and conditions.

4. Data Annotation:

The annotation technique, a important detail of dataset training, is methodically explicated. Whether completed manually or via computerized manner, the sure dialogue captures the intricacies of annotating faces within the dataset. The layout of annotations, encompassing bounding boxes for face detection and labels for popularity, is elucidated, providing perception into the floor fact facts embedded inside the dataset. Any challenges or uncertainties encountered at some point of the annotation approach are openly addressed, making sure transparency within the dataset's composition.

5. Face Detection Model Training(MTCNN):

With the annotated dataset in place, the face detection model is trained using the Multi-task Cascaded Convolutional Networks (MTCNN) architecture. The objective is to teach the model to identify and localize faces accurately in images. Training focuses on handling challenges such as variations in scale, rotation, and occlusions, ensuring robust face detection capabilities.

6. FaceNet for Face Recognition:

The highlight shifts to FaceNet, a seminal aspect responsible for the subsequent face reputation level. The intricacies of the triplet loss function, a defining thing of FaceNet's schooling, are meticulously explained. The structure of FaceNet, which include its neural community structure, is laid bare, offering a comprehensive expertise of the mechanism thru which it learns discriminative capabilities for facial popularity. The training manner, replete with hyperparameters and nice-tuning strategies tailored to the nuances of the custom dataset, is unveiled, positioning FaceNet as a effective device in the device's holistic approach to stand processing.

3.1 Requirements and Analysis

1. Hardware Requirements

GPU (NVIDIA GeForce RTX 2060 Mobile):

The GPU performs a pivotal role in accelerating deep mastering responsibilities, in particular all through model training. The NVIDIA GeForce RTX 2060 Mobile is a succesful GPU with CUDA cores, making it suitable for schooling complicated models like MTCNN and FaceNet. Ensure that the specified CUDA toolkit and cuDNN library are installed to leverage the GPU's parallel processing talents efficiently.

CPU (Intel Core i5 10th Gen Mobile):

The CPU is instrumental in coping with preprocessing duties, information augmentation, and different non-GPU increased procedures. The Intel Core i5 10th Gen Mobile is a stable desire, supplying enough processing electricity for those duties. Multi-center capabilities are useful for parallelizing sure operations, enhancing standard device performance.

A camera is also required to capture photos in order to build our custom dataset.

2. Software and Library Requirements:

Deep Learning Frameworks:

Install and configure deep mastering frameworks which include TensorFlow or PyTorch. Both MTCNN and FaceNet implementations are to be had in these frameworks, imparting flexibility in selecting the only that aligns along with your possibilities and understanding.

CUDA and cuDNN:

Ensure that the vital CUDA toolkit and cuDNN library are established and configured to harness the whole potential of the NVIDIA GPU. This is important for optimizing deep getting to know computations and speeding up version education.

OpenCV:

OpenCV is a important library for photo processing and computer vision tasks. It facilitates tasks such as loading and preprocessing pics, as well as coping with video streams. Ensure that OpenCV is hooked up and configured to seamlessly integrate along with your deep learning frameworks.

3.2 Project Design and Architecture

The Multi-Task Cascaded Convolutional Networks (MTCNN) is a complicated structure designed for face detection, comprising 3 tiers: Proposal Network (P-Net), Refinement Network (R-Net), and Output Network (O-Net). Each degree is characterized through unique layers tailored to its precise obligatio

Proposal Network (P-Net):

The Proposal Network serves as the initial stage in MTCNN, accountable for producing candidate bounding boxes round capacity face areas within the enter image. It starts with a convolutional layer that captures hierarchical functions from the photo. ReLU activation features introduce non-linearity, aiding in characteristic gaining knowledge of. Following this, a pooling layer downsamples the feature map, focusing on crucial statistics. Another convolutional layer similarly refines functions, and two output layers expect bounding container coordinates for capability faces and facial landmarks. The bounding container regression layer outputs four values representing the coordinates (x, y, width, top), even as the facial landmark detection layer gives a set of coordinates for each landmark, consisting of eyes and nose positions. Finally, Non-Maximum Suppression (NMS) is hired to filter overlapping bounding packing containers, ensuring that best the maximum confident face proposals are retained.

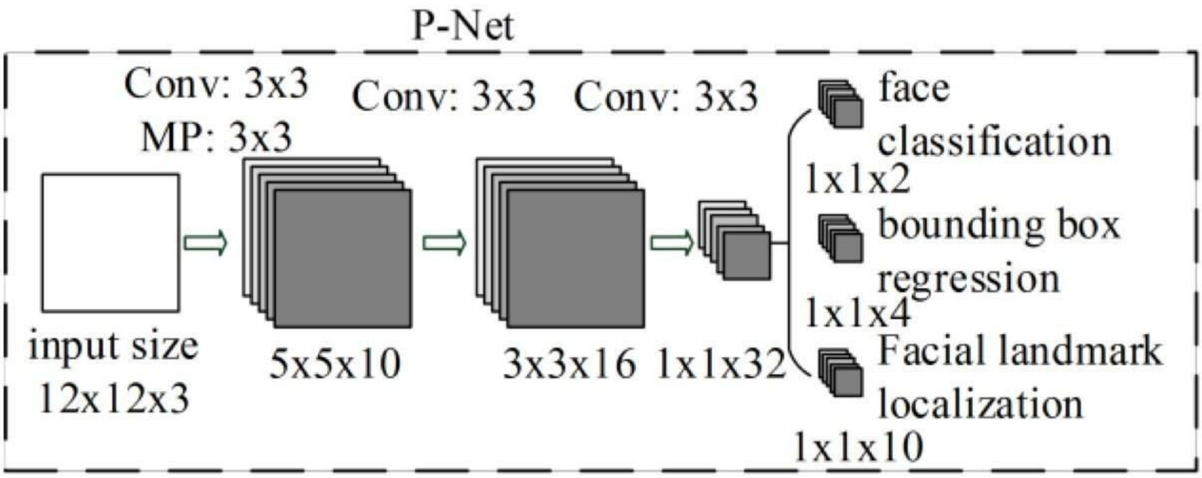

Fig 3.2.1 P-Net Structure

Refinement Network(R-Net):

The Refinement Network takes the bounding box proposals from the P-Net stage and objectives to improve the accuracy of the proposed areas. It starts with a convolutional layer to extract features from the proposed bounding box areas. A pooling layer downsamples the function map, and next convolutional layers similarly refine features, gaining knowledge of extra discriminative representations. Similar to the P-Net, the R-Net outputs subtle bounding field coordinates and facial landmarks. The NMS step on this level filters out redundant bounding containers based on self assurance ratings, improving the precision of the face detection.

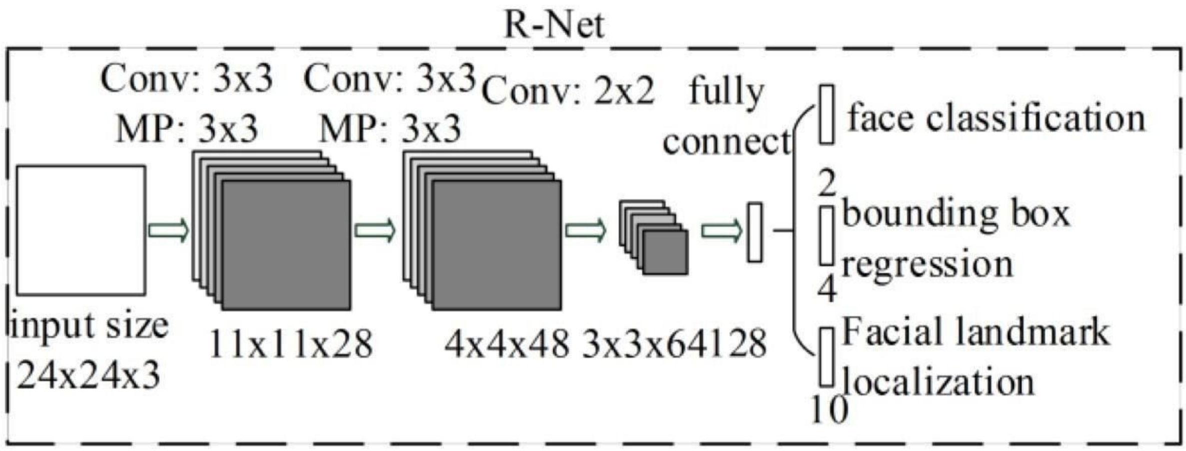

Fig 3.2.2 R-Net Structure

Output Network (O-Net):

The Output Network is the final stage in MTCNN, refining the bounding container proposals from the R-Net. It follows a comparable shape, inclusive of convolutional layers to beautify features and a pooling layer to downsample the function map. The community predicts the very last bounding field coordinates with improved precision and refines facial landmarks. The NMS step is yet again carried out, serving as a final clear out to discard duplicate and low-self assurance bounding containers. This degree completes the cascaded approach of MTCNN, offering extraordinarily accurate predictions for face detection and localization.

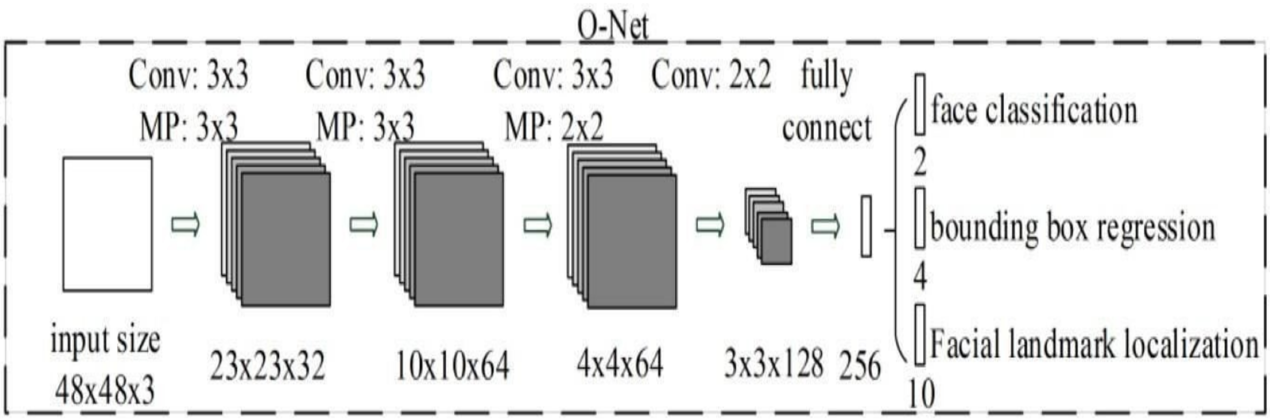

Fig 3.2.3 O-Net Structure

In short, the MTCNN architecture progresses thru these 3 tiers, regularly refining face proposals with every step. This cascaded layout allows for efficient narrowing down of capability face regions, ensuring robust and correct face detection in diverse eventualities.

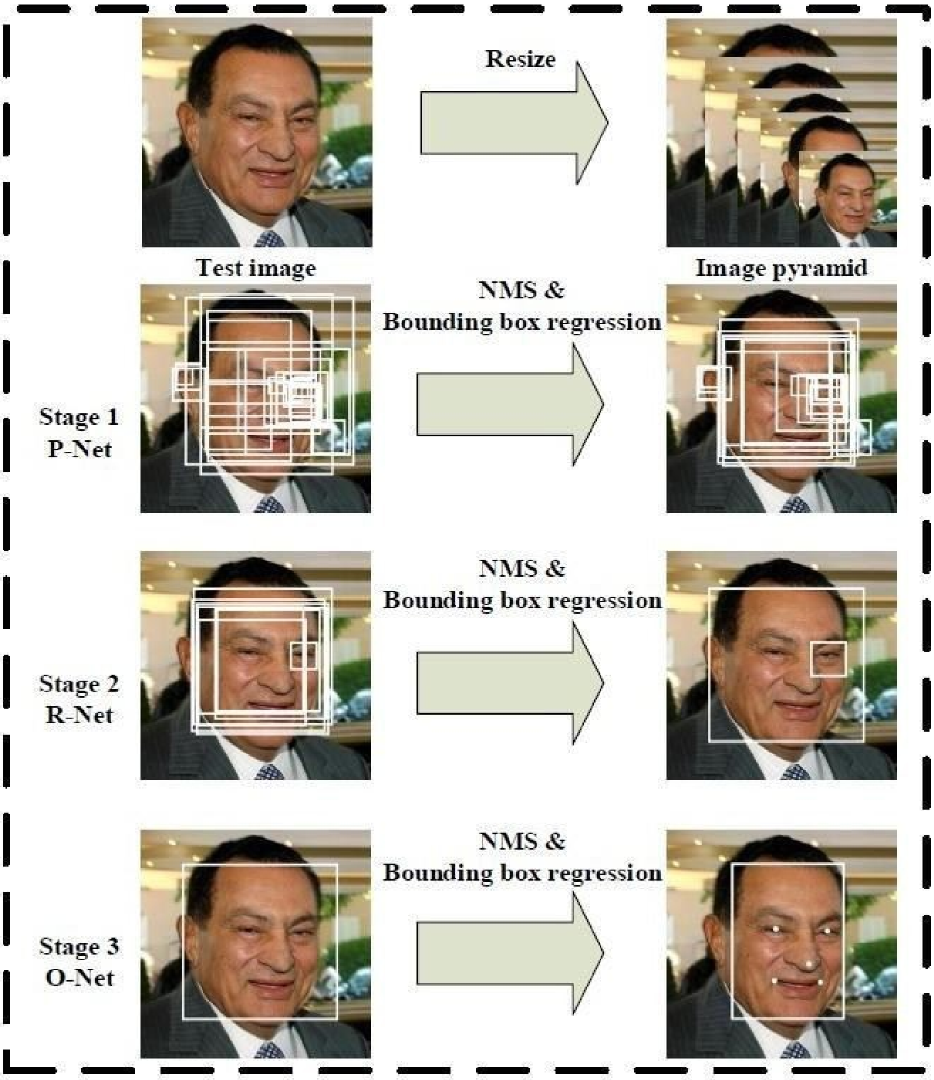

Fig 3.2.4 Overall Architecture of MTCNN

SSD(Single Shot Detector):

The Single Shot Detector (SSD) is a popular object detection algorithm that aims to detect objects in real-time with high accuracy. It follows a single-stage approach, which means that it performs object localization and classification simultaneously, making it faster and more efficient compared to traditional two-stage object detection methods.

The working principle of SSD revolves around a deep convolutional neural network that generates a collection of bounding boxes and scores for the presence of object classes in those boxes. The network architecture is based on a base network, such as VGG-16 or ResNet.

These feature maps are then fed into additional convolutional layers, called auxiliary convolution layers, which are responsible for predicting bounding boxes and classifying the objects contained within them. The auxiliary convolution layers are added to multiple feature maps from different stages of the base network, allowing the SSD to detect objects of various sizes.

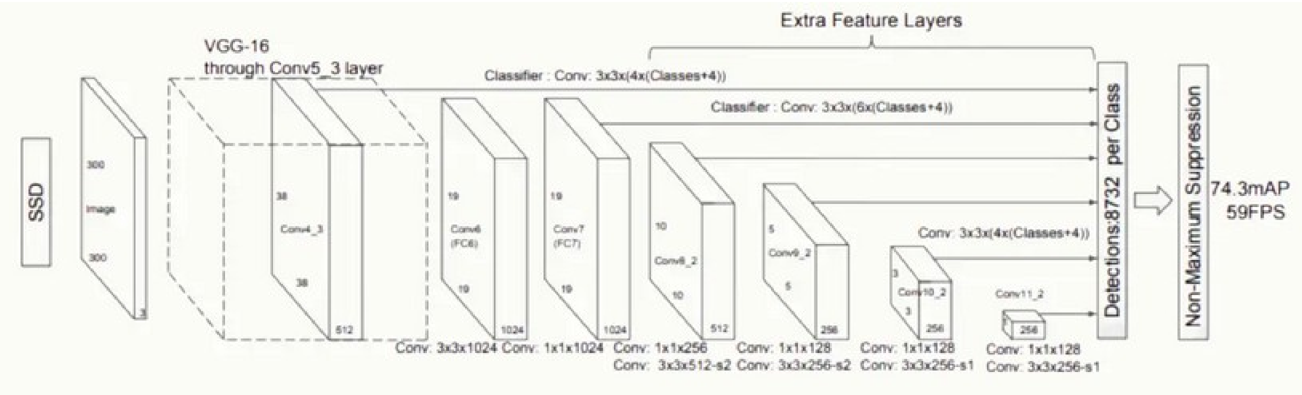

Fig 3.2.5 SSD architecture

YOLO (You only look once)

YOLO (You Only Look Once) v8 is the latest version of the renowned real-time object detection system developed by Ultralytics. It is a state-of-the-art deep learning model that has been extensively trained on a vast dataset of images, enabling it to accurately detect and classify objects in real-time.

YOLO v8 employs a novel architecture called YOLOv8, which is a combination of various advanced techniques such as Anchor Boxes, Feature Pyramid Networks (FPN), and Batch Normalization. Anchor Boxes are pre-defined boxes of different aspect ratios that help the model detect objects of varying sizes and shapes more accurately. FPN is a technique that combines feature maps from different layers of the network, allowing the model to detect objects at multiple scales. Batch Normalization is a technique that normalizes the input to each layer, improving the training process and enhancing the model's performance.

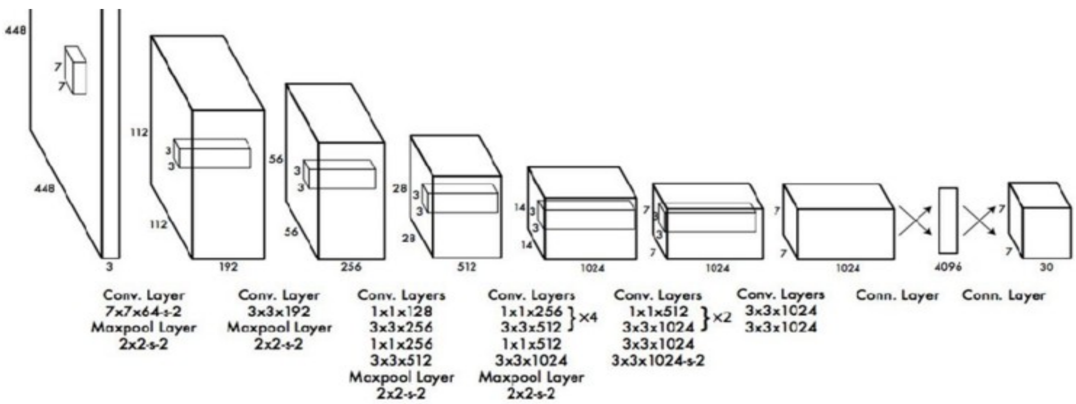

Fig 3.2.6 YOLO architecture

Facenet:

After the a success detection and localization of faces the usage of MTCNN, FaceNet is deployed to address the face reputation assignment. The bounding containers generated by MTCNN across the detected faces are used to extract facial regions from the original picture. These facial areas undergo preprocessing, ensuring compatibility with FaceNet's input necessities. FaceNet, a deep studying model, then generates high-dimensional embeddings for every face, representing particular facial functions in a vector space. During the training section, those embeddings are stored in a database along corresponding identification labels. In the popularity segment, the embeddings of detected faces are compared with the saved embeddings using a distance metric together with Euclidean distance or cosine similarity. The device then determines the identification of people based on the similarity. A predefined threshold is employed to differentiate among fits and non-matches, with smaller distances indicating better similarity. The output consists of the recognized identity of people in the photograph, permitting applications along with get entry to manipulate and customized person reviews. FaceNet's training entails getting to know discriminative functions to minimize the gap among embeddings of the same character and maximize the distance between embeddings of different individuals. For actual-time programs, issues such as threshold tuning and model optimization may be important for green and correct face popularity.

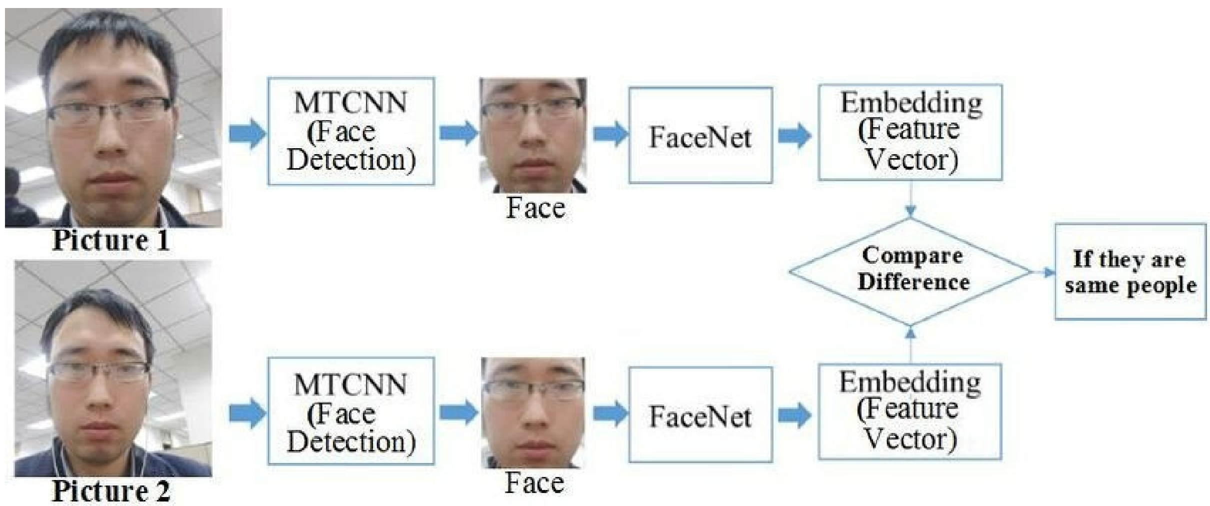

Fig 3.2.7 FaceNet Working

FaceNet converts facial pix into 128-dimensional vectors thru a deep neural network. The number one architecture of FaceNet is primarily based on a siamese network shape with triplet loss. Here's a detailed explanation of how FaceNet achieves this conversion:

1. Siamese Network Architecture:

FaceNet employs a siamese community, a type of neural network that consists of two same subnetworks (twins) with shared weights. These twin networks take the equal enter (anchor photo) and bring embeddings (feature vectors) as output.Triplets and Triplet Loss:

During education, FaceNet uses triplets of pix: an anchor picture, a positive image (identical character because the anchor), and a negative photograph (different character from the

anchor). The objective is to minimize the gap between the anchor and positive embeddings whilst maximizing the distance between the anchor and poor embeddings.

2. Embedding Layer:

The very last layer of the siamese network is the embedding layer. This layer produces a set-length vector (embedding) for each enter face. In the case of FaceNet, the embedding is ready to be 128-dimensional.

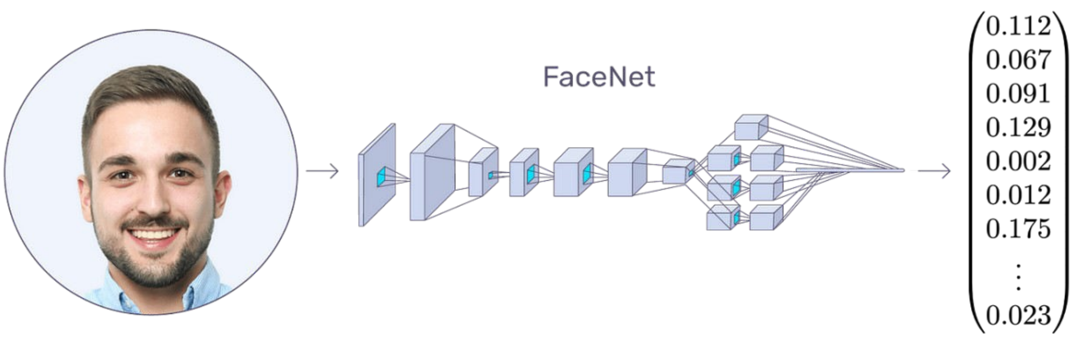

Fig 3.2.8 FaceNet working

3. Euclidean Distance in Embedding Space:

The Euclidean distance between embeddings is used as a degree of similarity. Smaller distances imply better similarity, at the same time as larger distances indicate dissimilarity.

4. Triplet Loss Function:

The goal function, or loss feature, used in training FaceNet is the triplet loss. This loss feature encourages embeddings of the identical character to be near within the embedding space and embeddings of different humans to be a long way apart. The mathematical components entails minimizing the space among the anchor and tremendous embeddings even as simultaneously maximizing the space between the anchor and negative embeddings.

5. Normalization of Embeddings:

To make the embeddings greater strong and similar across one-of-a-kind faces, FaceNet

normalizes the embedding vectors to have unit duration. This step is vital for making sure that the gap among embeddings is invariant to scale.

6. Output:

During inference, when a today's face is provided, the model generates a 128-dimensional vector (embedding) for that face the use of the discovered weights from the schooling phase. This embedding may be used for face reputation responsibilities, wherein similarity with special embeddings is measured for identity or verification.

In summary, FaceNet converts facial snap shots into 128-dimensional vectors thru schooling a siamese community using a triplet loss function. The resulting embeddings represent the specific features of every face in a excessive-dimensional space, permitting inexperienced and accurate face reputation. The use of triplets and the triplet loss guarantees that the model learns to create embeddings that efficiently capture facial identity at the same time as being robust to versions in pose, lighting fixtures, and expressions.

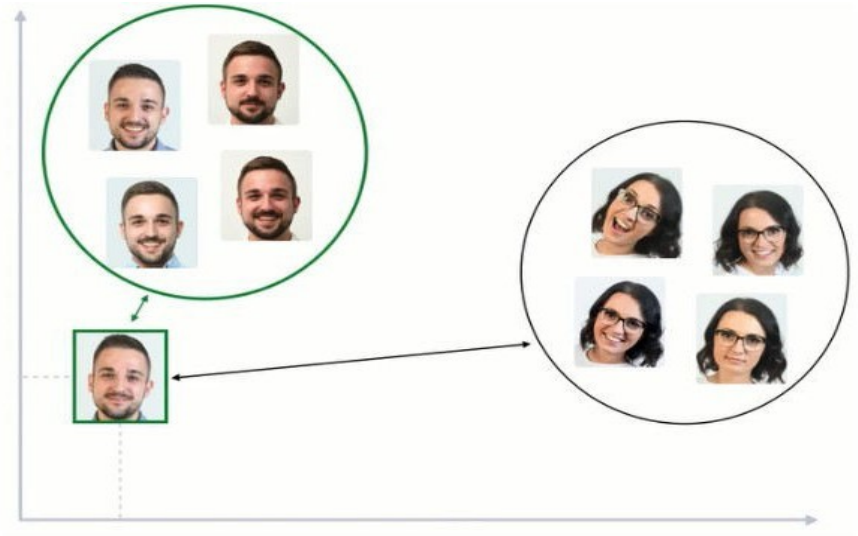

Fig 3.2.9 Recognition in real time

Flow Chart

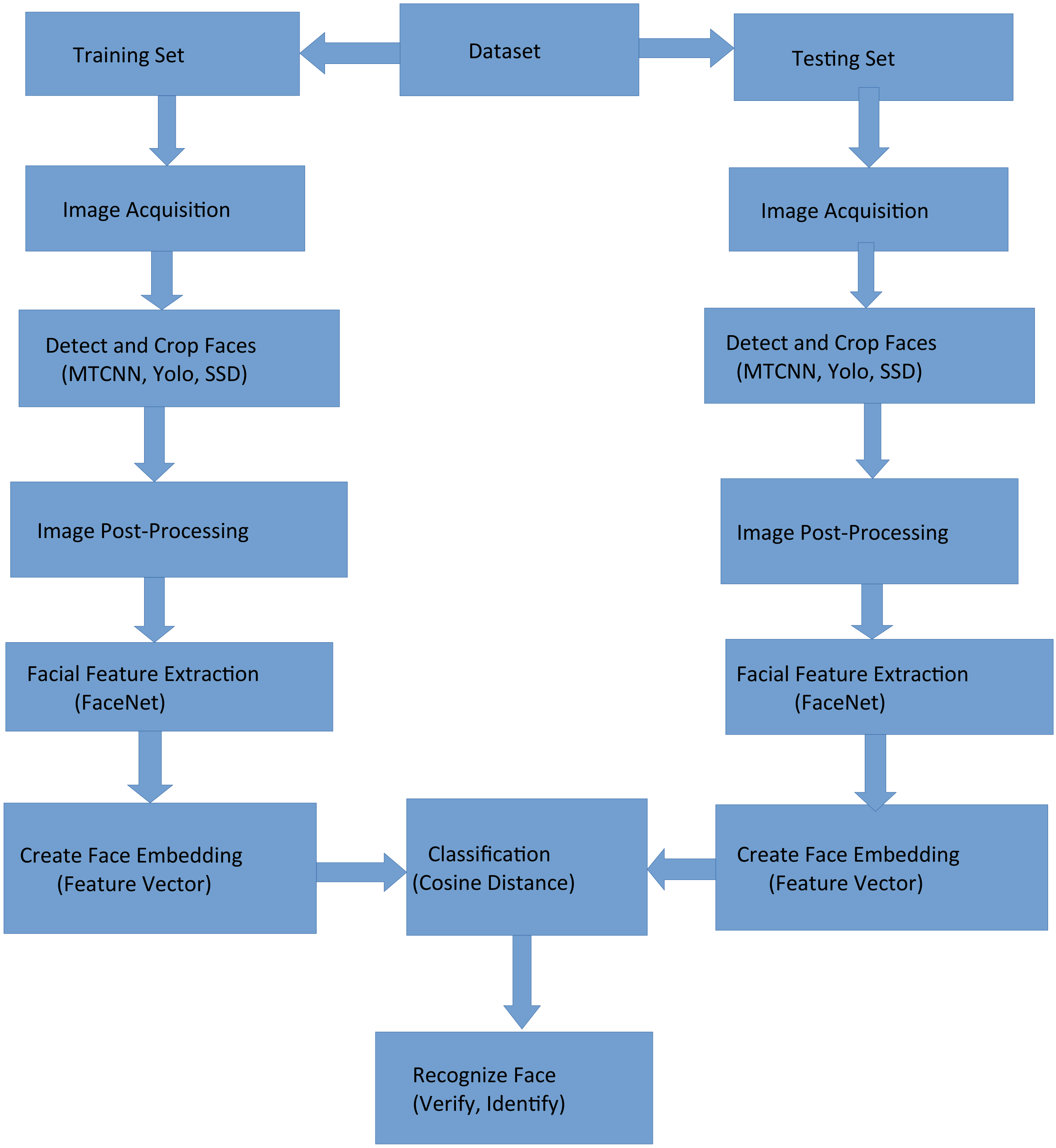

Fig 3.3.1 System Design in flowchart format

3.3 Data Preparation

For the face detection and recognition task, the initial step involves meticulous records training, which you have undertaken via growing a custom dataset through webcam acquisition. This process begins with the gathering of pics, leveraging the webcam to seize a diverse set of facial expressions, poses, and lights conditions to beautify the dataset's representativeness. Following photo acquisition, the dataset undergoes annotation, essential phase where bounding packing containers are delineated around facial regions to serve as floor fact statistics for the following schooling of the face detection version. Manual or automatic annotation equipment can be employed to ensure accuracy and consistency in labeling.

To bolster the dataset's robustness, information augmentation techniques are applied. Augmentation entails introducing variations inside the pix, which include rotations, flips, and scaling, to simulate real-global scenarios and improve the version's capability to generalize. These augmented snap shots contribute to a greater complete schooling routine, exposing the model to a broader range of facial variations.

Furthermore, the dataset is meticulously divided into training, checking out, and validation sets to facilitate effective version schooling and assessment. This partitioning ensures that the version learns from a diverse variety of examples in the course of training while being assessed on separate, unseen statistics to gauge its generalization talents.

The overall data guidance pipeline, encompassing photo collection, annotation, and augmentation, lays a solid foundation for education strong face detection and popularity fashions. The range captured in the dataset, coupled with careful annotation and augmentation, is pivotal in fostering a model capable of efficaciously handling real-international versions.

Table 3.1 Summary of image dataset used in the project.

Split	Images Collected
Training Dataset	3780
Validation	750
Testing Dataset	840
Total Images	5370

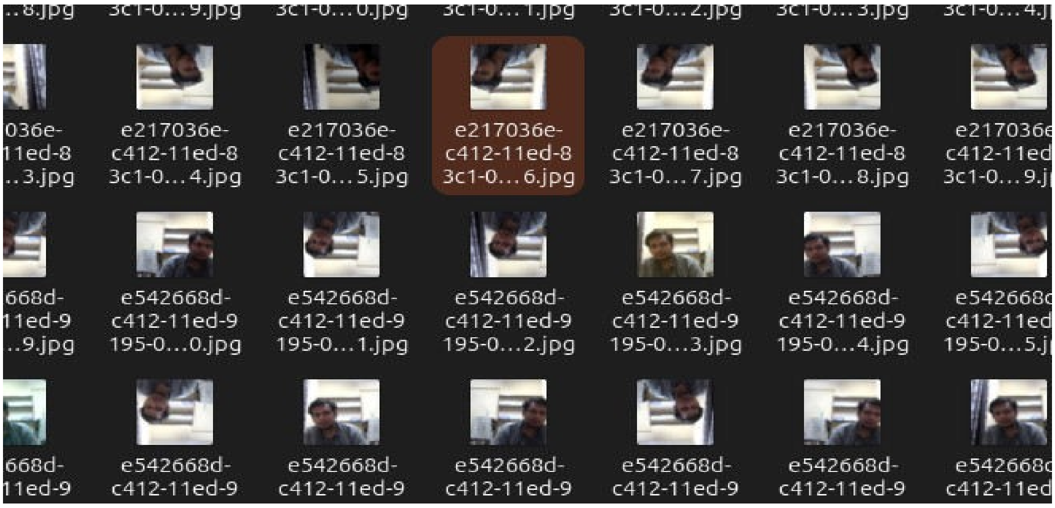

Fig 3.3.1 Images Collected using our webcam

3.4 Implemetation

Algorithm

Initialize Models:

- Load the MTCNN model for face detection (`PNet`, `RNet`, `ONet`).
- Load the FaceNet model for face recognition.

Face Detection using MTCNN:

- Preprocess the input image.
- Use the MTCNN to detect faces and obtain bounding boxes.
- Extract face regions based on the detected bounding boxes.

Face Recognition using FaceNet:

- Preprocess face regions extracted from the detected bounding boxes.
- Pass the preprocessed face regions through FaceNet to obtain face embeddings.
- Compare face embeddings for recognition or verification.

Pseudocode:

```
function faceDetectionRecognition(imagePath):

    # Step 1: Initialize Models

    mtcnnModel = loadMTCNNModel()  # Load MTCNN model

    faceNetModel = loadFaceNetModel()  # Load FaceNet model

    # Step 2: Face Detection using MTCNN

    image, boundingBoxes = detectFacesMTCNN(imagePath, mtcnnModel)

    # Step 3: Face Recognition using FaceNet

    faceRegions = extractFaceRegions(image, boundingBoxes)

    embeddings = recognizeFacesFaceNet(faceRegions, faceNetModel)

    # Further steps can include comparing embeddings for recognition or verification.

    return embeddings

function detectFacesMTCNN(imagePath, mtcnnModel):

    # Preprocess image

    image = loadImage(imagePath)

    # Detect faces using MTCNN

    boundingBoxes = mtcnnModel.detect(image)

    return image, boundingBoxes

function recognizeFacesFaceNet(faceRegions, faceNetModel):
```

```
  embeddings = []
  # Preprocess face regions

for faceRegion in faceRegions:     faceTensor = preprocessFaceRegion(faceRegion)
    # Obtain face embeddings using FaceNet
    embedding = faceNetModel(faceTensor)
    embeddings.append(embedding)
  # Perform face recognition or verification using the obtained embeddings
  # (This depends on your specific use case)
  return embeddings
```

Tools and Technologies used:

Visual Studio Code (VSCode):

A source-code editor developed by Microsoft with support for various programming languages. It's widely used for Python development and provides a range of extensions for different frameworks and libraries.

Fig 3.4.1 VSCode

Python:

A programming language commonly used for machine learning and deep learning tasks due to its simplicity and a rich ecosystem of libraries.

Fig 3.4.2 Python

TensorFlow:

An open-source machine learning framework developed by the Google Brain team. It is widely used for building and training deep learning models, including neural networks.

PyTorch:

An open-source machine learning library developed by Facebook. It is known for its dynamic computational graph and is popular for research in deep learning and neural networks.

We will start by implementing MTCNN. The model includes implementing the three layers of MTCNN namely P-Net, R-Net and O-Net.

Code Snippets

```python
import torch
import torch.nn as nn
import torch.nn.functional as F
from collections import OrderedDict
import numpy as np


class Flatten(nn.Module):

    def __init__(self):
        super(Flatten, self).__init__()

    def forward(self, x):
        """
        Arguments:
            x: a float tensor with shape [batch_size, c, h, w].
        Returns:
            a float tensor with shape [batch_size, c*h*w].
        """

        # without this pretrained model isn't working
        x = x.transpose(3, 2).contiguous()

        return x.view(x.size(0), -1)


class PNet(nn.Module):

    def __init__(self):

        super(PNet, self).__init__()

```

Fig 3.4.3 Implementing MTCNN from scratch

This code defines a PyTorch implementation of a face detection model using the MTCNN (Multi-task Cascaded Convolutional Networks) architecture. MTCNN is a popular face detection algorithm that consists of three stages (PNet, RNet, and ONet), each responsible for different aspects of face detection.

Flatten Class:

This is a simple module that flattens the input tensor. It is used to convert the output of convolutional layers into a flat vector for fully connected layers.

```python
class PNet(nn.Module):

    def __init__(self):

        super(PNet, self).__init__()

        # suppose we have input with size HxW, then
        # after first layer: H - 2,
        # after pool: ceil((H - 2)/2),
        # after second conv: ceil((H - 2)/2) - 2,
        # after last conv: ceil((H - 2)/2) - 4,
        # and the same for W

        self.features = nn.Sequential(OrderedDict([
            ('conv1', nn.Conv2d(3, 10, 3, 1)),
            ('prelu1', nn.PReLU(10)),
            ('pool1', nn.MaxPool2d(2, 2, ceil_mode=True)),

            ('conv2', nn.Conv2d(10, 16, 3, 1)),
            ('prelu2', nn.PReLU(16)),

            ('conv3', nn.Conv2d(16, 32, 3, 1)),
            ('prelu3', nn.PReLU(32))
        ]))

        self.conv4_1 = nn.Conv2d(32, 2, 1, 1)
        self.conv4_2 = nn.Conv2d(32, 4, 1, 1)

        weights = np.load('src/weights/pnet.npy')[()]
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Fig 3.4.4 Implementing PNet layer

PNet Class:

This class defines the first stage of the MTCNN, which is responsible for generating candidate face regions.

Constructor (`__init__`):

The convolutional layers (`conv1`, `conv2`, `conv3`) along with their corresponding activation functions (`prelu1`, `prelu2`, `prelu3`).

Also initializes two additional convolutional layers (`conv4_1` and `conv4_2`) responsible for predicting bounding box regression (`b`) and facial landmark points (`a`).

Loads pre-trained weights fo the model from an external file. (`'src/weights/pnet.npy'`).

```python
        for n, p in self.named_parameters():
            p.data = torch.FloatTensor(weights[n])

    def forward(self, x):
        """
        Arguments:
            x: a float tensor with shape [batch_size, 3, h, w].
        Returns:
            b: a float tensor with shape [batch_size, 4, h', w'].
            a: a float tensor with shape [batch_size, 2, h', w'].
        """
        x = self.features(x)
        a = self.conv4_1(x)
        b = self.conv4_2(x)
        a = F.softmax(a)
        return b, a


class RNet(nn.Module):

    def __init__(self):

        super(RNet, self).__init__()

        self.features = nn.Sequential(OrderedDict([
            ('conv1', nn.Conv2d(3, 28, 3, 1)),
            ('prelu1', nn.PReLU(28)),
            ('pool1', nn.MaxPool2d(3, 2, ceil_mode=True)),

            ('conv2', nn.Conv2d(28, 48, 3, 1)),
            ('prelu2', nn.PReLU(48)),
            ('pool2', nn.MaxPool2d(3, 2, ceil_mode=True)),
```

Fig 3.4.5 Implementing RNet layer

RNet Class:
This class defines the second stage of the MTCNN, which refines the face candidate regions from the Pnet.

Constructor (__init__):

Initializes the convolutional layers (conv1, conv2, conv3, flatten, conv4) along with their corresponding activation functions (prelu1, prelu2, prelu3, prelu4).

Also initializes two fully connected layers (conv5_1 and conv5_2) responsible for predicting bounding box regression (b) and facial landmark points (a).
Loads pre-trained weights for the model from an external file .
('src/weights/rnet.npy').

```
            ('conv3', nn.Conv2d(48, 64, 2, 1)),
            ('prelu3', nn.PReLU(64)),

            ('flatten', Flatten()),
            ('conv4', nn.Linear(576, 128)),
            ('prelu4', nn.PReLU(128))
        ]))

        self.conv5_1 = nn.Linear(128, 2)
        self.conv5_2 = nn.Linear(128, 4)

        weights = np.load('src/weights/rnet.npy')[()]
        for n, p in self.named_parameters():
            p.data = torch.FloatTensor(weights[n])

    def forward(self, x):
        """
        Arguments:
            x: a float tensor with shape [batch_size, 3, h, w].
        Returns:
            b: a float tensor with shape [batch_size, 4].
            a: a float tensor with shape [batch_size, 2].
        """
        x = self.features(x)
        a = self.conv5_1(x)
        b = self.conv5_2(x)
        a = F.softmax(a)
        return b, a


class ONet(nn.Module):
```

Fig 3.4.6 Implementing forward pass

Forward Pass('Forward'):

This class defines the third stage of the MTCNN, which further refines the face candidate regions from the RNet.

Passes the input tensor through the defined layers to extract features.

- Returns two outputs:

- `b`: Bounding box regression predictions.
- `a`: Facial landmark point predictions.

- Applies softmax activation to `a`.

```python
class ONet(nn.Module):

    def __init__(self):

        super(ONet, self).__init__()

        self.features = nn.Sequential(OrderedDict([
            ('conv1', nn.Conv2d(3, 32, 3, 1)),
            ('prelu1', nn.PReLU(32)),
            ('pool1', nn.MaxPool2d(3, 2, ceil_mode=True)),

            ('conv2', nn.Conv2d(32, 64, 3, 1)),
            ('prelu2', nn.PReLU(64)),
            ('pool2', nn.MaxPool2d(3, 2, ceil_mode=True)),

            ('conv3', nn.Conv2d(64, 64, 3, 1)),
            ('prelu3', nn.PReLU(64)),
            ('pool3', nn.MaxPool2d(2, 2, ceil_mode=True)),

            ('conv4', nn.Conv2d(64, 128, 2, 1)),
            ('prelu4', nn.PReLU(128)),

            ('flatten', Flatten()),
            ('conv5', nn.Linear(1152, 256)),
            ('drop5', nn.Dropout(0.25)),
            ('prelu5', nn.PReLU(256)),
        ]))

        self.conv6_1 = nn.Linear(256, 2)
        self.conv6_2 = nn.Linear(256, 4)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Fig 3.4.7 Implementing ONet layer

ONet Class:

This class defines the third stage of the MTCNN, which further refines the face candidate regions from the RNet.

Constructor (`__init__`):

Initializes the convolutional layers (`conv1`, `conv2`, `conv3`, `conv4`, `flatten`, `conv5`) along with their corresponding activation functions (`prelu1`, `prelu2`, `prelu3`, `prelu4`, `prelu5`).

Also initializes three fully connected layers (`conv6_1`, `conv6_2`, `conv6_3`) responsible for predicting facial landmarks (`c`), bounding box regression (`b`), and face classification (`a`).

```
            ('flatten', Flatten()),
            ('conv5', nn.Linear(1152, 256)),
            ('drop5', nn.Dropout(0.25)),
            ('prelu5', nn.PReLU(256)),
        ]))

        self.conv6_1 = nn.Linear(256, 2)
        self.conv6_2 = nn.Linear(256, 4)
        self.conv6_3 = nn.Linear(256, 10)

        weights = np.load('src/weights/onet.npy')[()]
        for n, p in self.named_parameters():
            p.data = torch.FloatTensor(weights[n])

    def forward(self, x):
        """
        Arguments:
            x: a float tensor with shape [batch_size, 3, h, w].
        Returns:
            c: a float tensor with shape [batch_size, 10].
            b: a float tensor with shape [batch_size, 4].
            a: a float tensor with shape [batch_size, 2].
        """
        x = self.features(x)
        a = self.conv6_1(x)
        b = self.conv6_2(x)
        c = self.conv6_3(x)
        a = F.softmax(a)
        return c, b, a
```

Fig 3.4.8 Implementing forward pass

Forward Pass('Forward'):

This class defines the third stage of the MTCNN, which further refines the face candidate regions from the ONet.

Passes the input tensor through the defined layers to extract features.

- Returns two outputs:

- `b`: Bounding box regression predictions.
- `a`: Facial landmark point predictions.

3.5 Key Challenges

Dataset Quality: High satisfactory and various dataset is key in constructing effective face detection and popularity fashions. Capturing various facial versions together with distinctive lightings, poses, and historical past is wherein the challenges come in. Dataset accuracy relies upon at the categorized face detection bounding boxes and facial landmarks. This task is addressed via cautious choice so one can produce a dataset that portrays excessive generalization capability for unique case research.

Model Complexity and Size: Real time is distinctly motivated by using the choice of the architecture to use. The processing velocity of huge and complicated models is slowed down in particular wherein assets are scarce. This hassle however can be mitigated by means of considering smaller or more power green models intended for real time applications. Model quantization or compression can be used to reduce its length with out affecting its accuracy. Therefore, it is crucial to strike the proper balance among the extent of model complexity against computational efficiency as a way to achieve the first-rate results.

Hardware Limitations: The actual-time processing velocity of face detection and recognition systems is significantly inspired with the aid of hardware competencies. Poor performance is associated with insufficient hardware, leading to poor functioning. This involves speeding up inference the use of hardware accelerators like GPUs and TPUs. Parallel processing must be hired efficiently to maximise the computational strength to be had

Algorithm Selection: Algorithms choice substantially affect the performance in the face detection and popularity. It is difficult to decide at the fashions which gain this balancing act. Different algorithms and model architectures should be tried out with a purpose to discover the proper aggregate desirable for real-time operations.

Image Preprocessing Overhead: Complex preprocessing steps can introduce needless overhead, affecting real-time processing. Optimizing photo preprocessing involves enforcing green techniques such as histogram equalization, resizing, and normalization. These steps ought to be streamlined to limit computational necessities at the same time as making sure the powerful training of enter facts for the models.

Real-Time Processing Pipeline: Integrating face detection, reputation, and processing into a real-time pipeline calls for cautious consideration of records float and computational performance. Streamlining the processing pipeline entails optimizing algorithms and minimizing redundant computations. Using efficient statistics structures and algorithms for actual-time processing complements the general responsiveness of the gadget.

Software and Library Compatibility: Compatibility troubles among one of a kind software program versions and library dependencies can pose demanding situations within the development system. Overcoming this assignment involves keeping up to date software program dependencies, making sure compatibility throughout one-of-a-kind variations of libraries, frameworks, and the working system. Regular updates and version exams help mitigate compatibility problems which could impact gadget capability.

Debugging and Profiling: Identifying bottlenecks within the machine and optimizing code for overall performance are important components of real-time face detection and popularity. Profiling tools can be hired to screen useful resource usage, execution time, and memory consumption. Leveraging the insights won from profiling allows developers to pinpoint overall performance issues and optimize the codebase for green actual-time processing.

Real-Time Frameworks: Developing real-time packages requires concerns for low-latency performance. Using frameworks in particular designed for actual-time video processing and computer imaginative and prescient obligations is vital. These frameworks streamline the improvement process via supplying optimized capabilities and capabilities tailor-made for actual-time packages, contributing to more desirable responsiveness and universal device performance.

CHAPTER - 4

TESTING

Real-Time Frameworks: There are some considerations of low-latency performance if developing real-time packages should be considered. It is important to use special frameworks meant exactly for real-time video processing and computer vision tasks. The frameworks automate this process by offering optimized functions and functions specifically designed for real-time programs, resulting in superior responsiveness and total system behavioral excellence.

Dataset Testing: Dataset testing is a major phase of any project associated with face detection and recognition, since it serves as the basis for checking the strength of models applied. Typically, the dataset is split into training and cross validation sets, whereby the training data is used to train the models and validation data is used for fine-tuning and optimization. However, a test or validation set is critical in determining how well the model can extrapolate unseen data. This includes a range of different settings including lighting, pose and even background to accurately mimic the real-time environment. Face detection uses accuracy metrics such as precision, recall and F1 score to measure model performance while face recognition uses accuracy metrics like accuracy, precision, recall and F1 score.

Performance Metrics: Performance measurement of face detection/recognition systems largely depends on metrics. The term, “precision” refers to a method in which we measure the correctness of the predicted positives, “recall” is used in the evaluation of the ability to detect all positives, and finally, “F1 score” combines these two parameters (precision and recall Accuracy remains a critical measure in face recognition while precision, recall, and F1 score facilitate full assessment. For instance, TPR and FPR are specifically important when it comes to binary classifications involved in face recognition. Collectively, these metrics describe how robust or weak the model is.

Real-Time Performance: The measure of live performance for these facial detection and recognition system is in terms of number of frames used per second without any delays. Frame rate is another important metric to assess the efficacy of the system. Another important factor in real-time processing of information is latency that is needed to process a single frame. With that said, having a low latency and a high number of frames is key to incorporating such systems into everyday applications.

Robustness Testing: It is also critical to test robustness so as to assess how well face detection and recognition models perform in tough environment situations. Adding noise and other distortions or occlusions in the images enables the assessment of the model output quality when the input data is not clear. One example of this is adversarial attacks, which involve purposeful adding of disturbances into an input in a bid to deceive the model. Such a test provides assurance of continued accuracy and reliability of the models even under non-optimum conditions.

Cross-Domain Testing: Cross-domain testing is when the face detection and recognition system is tested in different settings and with various platforms. This testing is geared towards ensuring the generalization as well as performance of the models in varying conditions. Comparisons between different testing conditions, namely, changes in lightings, backgrounds, and cameras' angles, present information on the suitability of such models. Another thing is that it's good to test the performance across different devices because it allows one to check if there is consistency in the output among the platforms.

Edge Cases Testing: However, in order to evaluate face detection and recognition model under such edge cases, the testing of the face detection and recognition system are performed at this stage. The models are also tested in case of extremely posed faces or expressing emotions, as well as in case of low quality pictures. The edge case testing, in this regard, helps identify possible limitations and refine the models to be able to properly solve very complicated scenarios that can actually arise in practice.

User Acceptance Testing (UAT): User acceptance testing entails seeking views of users at the end of the user testing phase to determine whether the system meets their expectation and specification. The second phase of testing evaluates both the technical aspect of the model as well as its user interface, user experience, and general usability.

Continuous Testing: Continuous testing is an integral part of the development life-cycle for face detection and recognition projects. Implementing automated testing pipelines allows for regular and systematic validation of the system's performance as it evolves. Automation facilitates the quick identification of regressions or performance issues, ensuring that the system remains reliable and efficient over time. Continuous monitoring and logging of key metrics contribute to ongoing optimization and maintenance.

Essentially, holistic evaluations for dataset quality, test results, timely operations, resilience under varied domains, borderline circumstances, user feedback, security, and ongoing trials are covered by conclusive testing in face detection and recognition programs.Taken altogether these phase tests are instrumental in developing the correct, quick and reliable systems for real world applications.

4.2 Outcomes

1. Successful Face Detection and Recognition:

The first main success of this project was in achieving success in detecting and recognizing faces at real time. Such an achievement signifies that the models are now able to efficiently distinguish a particular face amongst numerous other faces captured from webcam video feeds.

It is built on careful use of database collected for itself from webcams laying a basement for success. Training of models in diversified data set was vital. The use of the large, diverse training set exposed the models to a number of realistic situations, including various illuminations, face shapes, and backgrounds. This enabled their successful learning to generalize well.

Among the notable results achieved is the potentiality of the models to detect faces through difficult situations. The models were robust and effective despite being subjected to extreme facial poses or poor quality images, which makes them suitable for such scenarios in a dynamic environment.

Valuable user feedback was provided in the project, which informed significant user acceptance and usability considerations. The user acceptance testing of the project helped to get information regarding system usability as well as total user experience. This iterated procedure has enabled modifications to be made on the UI thus making it much more practical and effective.

The project was indeed successful as far as real time face detection and recognition are concerned, only that the observed frame rate of 2 fps should be considered with caution. Further investigations and optimizations towards this real-time performance bottleneck are forthcoming. Among these limitations, exploring potential optimizations, including algorithmic improvements, hardware upgrades, and model size reduction will be crucial.

Therefore, it is important to seek ways of optimizing for real-time performance. Such experiments would include trying out model quantization as well as various compression techniques or simply adopting lighter-weight architectures. Hardware upgrading or incorporating some specialized hardware accelerators like GPUs and/or TPUs, for example, could have a significant influence on what is generally known as “real time” processing capability of a computer system.

2. Dataset Utilization:

The key aspect of this project success lays on a carefully picked self-generated sample which is obtained straightly through webcams. This dataset contributed significantly in building the models ready for handling face detection and recognition challenges. Collecting data from the webcam proved smart because this way the set of data matched with the model operation circumstances. Through this, the project was directed towards bridging the gap between simulating and real-world scenarios, thus developing a stronger and flexible system.

It is even more important when you review the very diversity of the dataset itself. Acquiring facial images while being captured differently in diverse lighting environments, face positions as well as backgrounds has made this a rich array of examples that can be learned from. Diversity is crucial for improving the generality of these models. Essentially, the models have been seen different facial situations and views to enable them identify facial images of complex nature in real world applications.

Use of diverse data sets act as one strong weapon against overfitting. As such, the models can learn the general features and patterns without being overly dependent on details of the training set. Hence, when the models experience novel but familiar faces in various environments, they will use their generalized knowledge obtained from training to arrive at correct decisions.

Moreover, the self-collected data sets act as the key components for building endurance among the models. This is because real-world scenarios involve many different conditions that can be quite different than controlled situations or simulations. Using heterogenous training data set makes models able to take into account the diverse nature of face looks.

This is a commitment towards using a self-dataset that suits the overall aim of making a Making such a decision, however, does not only improve the performance of the models but helps to ensure that the results obtained from the project reflect the intricacies and intricate nature of real life situations. Consequently, this dataset becomes an essential point, i.e., it has the knowledge that provides models with assistance in overcoming difficulties while detecting faces in real-time applications.

3. Recognition of Faces in Challenging Scenarios:

Noteworthy is that the models have performed satisfactorily on different facial recognition scenarios and therefore will work well in the real world. # This feat, however, is characterized by models that can deal with complex facial movements. Faces are essentially dynamic, and people can display any posture in the real world. The ability of these models to correctly recognize facial images in severe postures signifies their suitability for the complex dynamics inherent in human expression and orientation.

The other vital part of the models' success is their robustness with poor image quality. In most cases, real world application does not include perfect illumination and ideal hardware performance. This shows that the models are capable of dealing with such challenges on their way and also extract useful features even from some imperfect photographs. Having this much resilience is important on its own, considering how important it is that an imaging pipeline performs well consistently throughout such a broad range of real world use case scenarios that can see image quality change up and down in random ways.

The stress on practical face recognition in difficult situations correspond to the complexity of actual world settings wherein individual might perhaps not exhibit him/herself within ideal context for facial detection. The fact that they work under challenging conditions like tracking and surveillance confirms the versatility and competency of their ability to be deployed for usage in real applications.

Indeed, this achievement demonstrates that these models can generalise very well from specific conditions of learning. The difference between different samples in the set, especially about wide face angles and bad images have prepared the models for unexpected situations during work. In fact, this outstanding performance under difficult circumstances is an integral part of the bigger objective of developing an efficient face detection and recognition system which performs remarkably well even in the complicated, fluctuating environments of real life.

The fact that these models were able to recognize the faces in difficult conditions shows that they are robust models suitable for practical applications. The results emphasize the achievement of the project in developing models beyond typical benchmarks, taking into account the realities and intricacies of actual situations that do not often offer optimum circumstances.

4. Real-Time Face Detection and Recognition:

Although this challenge could be considered a setback to the noted project's real time face recognition achievements. This observed two frames per second (fps) would probably constitute a limitation of real-time processing speed. Further, this results into further in-depth analysis on possible reasons as to why lower FPS, thereby leading to more reliable optimization solutions.

Numerous things could explain this observed performance bottleneck. Firstly, many factors must be considered such as the level of algorithm complexity used in face detection and recognition. These algorithms typically employ complex neural network structures, and the calculation needs for performing every frame in real time might be beyond what the current system is capable of handling. Besides that, the size and the depths of the neural networks may also add to higher computational costs which might decrease a total frame rate.

The other reason may involve poor hardware configurations as well. Real-time face detection and recognition can be a challenging task to software developers. It requires a high level of computational processing powers that are only available with the latest hardware infrastructure comprising of powerful central processing units (CPU) and sufficient computer memory. The delay between frame processing due to limited processing powers directly affects the frame rate. Therefore, finding chances of hardware optimizations or upgrading the hardware would be essential in solving this kind of performance obstacle.

Furthermore, it could also be that the model has huge parameters or complexities in the architecture of the neural network which can lead to longer inference time. More computation is required for larger models with many parameters per forward pass which may lead to delay in real time processing. The possible solutions to minimize computation costs include model optimization methods like pruning, quantization and exploring architecture which is less heavy.

Another important factor to consider is the algorithmic efficiency. Efficiency of face detection and recognition algorithms can affect total frame rate. Therefore it is necessary to analyze and optimize these algorithms so as to ensure every computation process is streamlined in terms of efficiency but not at the expense of accuracy. Improving the optimization process can entail adjusting parameters, reexamining the model structure, as well as considering other algorithms more fitting for online operations.

The intricacies of the processing pipeline too, like multiple steps in the process for facial detection & recognition may be causing this slow down Such an increase in processing overhead during each stage might be contributing to the poor real time performance. One of the areas where changes can be implemented is streamlining the processing pipeline which can be facilitated by parallelization or asynchronous processing.

Furthermore, selection of appropriate software frameworks and libraries contributes immensely for system performance. Lack of proper use of these tools might cause ineffective calculations which is also one factor behind traffic jam situation. This aspect of the performance challenge may be addressed by ensuring that the project takes advantage of the most recent optimizations within selected frameworks and identifying alternative strategies that are better aligned with supporting real-time applications.

Further, the existing system may not be taking advantage of hardware accelerating technologies efficiently. Neural network computations will be carried out very quickly using GPUs or TPUs. Strategic integration and optimization of use of such hardware accelerators within the system architecture may help in attaining maximum real-time performance.

In terms of potential solutions, the use of cutting edge technologies meant specifically for face detection could be considered.C For example, such frameworks may involve specialized light weight neural networks adapted for real time use. Moreover, adopting parallelization techniques like threading and multiprocessing can distribute the computation load among these sources and may boost the fps (frames per second).

Another path to improvement is hardware upgrades. The use of advanced CPUs, the use of GPUs or TPUs would give enough power for computation to satisfy the condition of real time processing. Optimizing the model parameters and investigating the quantized models will also improve inference efficiency with corresponding impact on the frame rates.

Monitoring the system's performance on a continuous basis will remain critical in determining whether adopted measures proved efficient or not. For effective optimization, regular benchmarking of the system with respect to the relevant performance measures and iterative improvement in the techniques of the approach based upon the collected data would contribute to continuous upgrade of real-time face detection and recognition competencies.

Finally, identifying a performance bottleneck enabling optimized real-time performance in face detection and recognition constitutes an essential chance for improvement. In order for a project to overcome its complexities, it must explore various aspects of algorithms, hardware configurations, and processing pipe line. Furthermore, this should entail systematically addressing individual factors which contribute to these complications. Ultimately, such an approach will ensure smooth real time operations under dynamic conditions

CHAPTER - 5
RESULTS AND EVALUATION

Results of the project cover several issues including successful implementation of real time face detection & recognition, use of self collected database, problems with real time operation and challenges of recognition in difficult conditions. Presentation of findings and analysis of the results are two very important steps, which will tell us how we understand the impact of the project and where we might be going next.

An analysis of the results provides evidence that the project is successful at advancing a feasible face detection and recognition model. Notable achievements include the employment of different data sets, the capability to withstand harsh environments, and real time operation. The recognized area of weakest performance demonstrates the importance of never-ending efforts towards improvement. in the process.

The study shall provide a stepwise analysis of the reasons behind poor performance bottleneck. This involves studying of computational efficiency, considering hardware improvement/acceleration, and improving the system structure. The effects of executed solutions must be measured continuously over time. It is necessary for a system to change in accordance with its use in practical applications.

Lastly in a nutshell, results reporting captures achievements and drawbacks of the project. Successful parts lay a foundation for their practical implementations and failures give chances to improve upon the optimization. Furthermore, interpretations of results determine future steps because improvements cannot be achieved unless there is constant adaptation within face detection and recognition systems.

Confusion matrix is an essential indicator that aids in measuring the effectiveness of a face detection model or any classifier. It gives an overview of how closely the predicted tags compare with what is actually on the ground truth. In the context of face detection, it can help us analyze the following:

1. True Positives (TP): The number of faces that were correctly detected by our model.
2. True Negatives (TN): Number of non-faces that were correctly identified as non-faces.

3. False Positives (FP): The number of non-faces that were incorrectly classified as faces.
4. False Negative (FN): The number of faces that were missed by the model and not detected.

As our MTCNN model is pre-trained, so it is expected to have the highest accuracy among all the models that we trained.

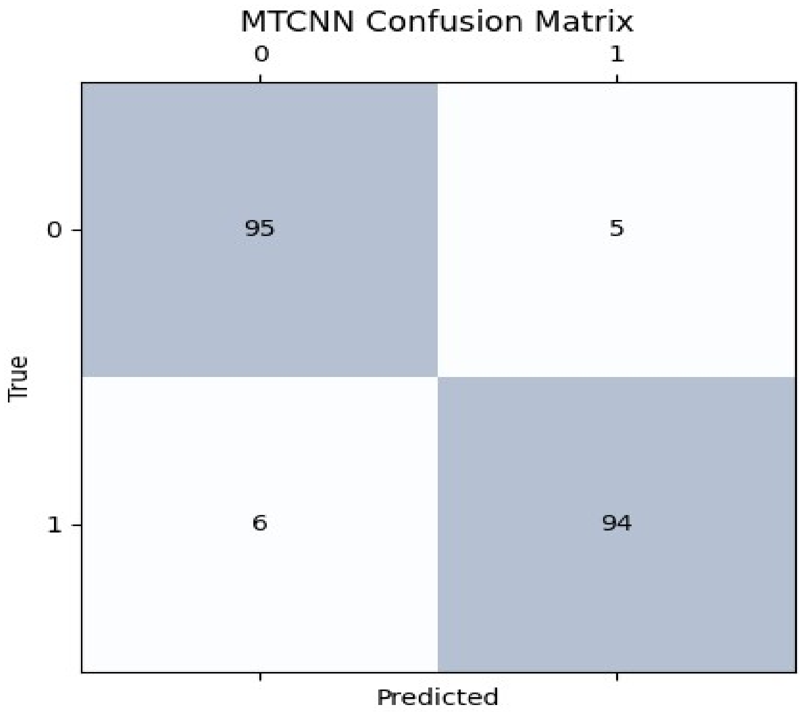

Fig 5.1.1 Confusion matrix

Based on the confusion matrix we achieve, the following results can be shown below,

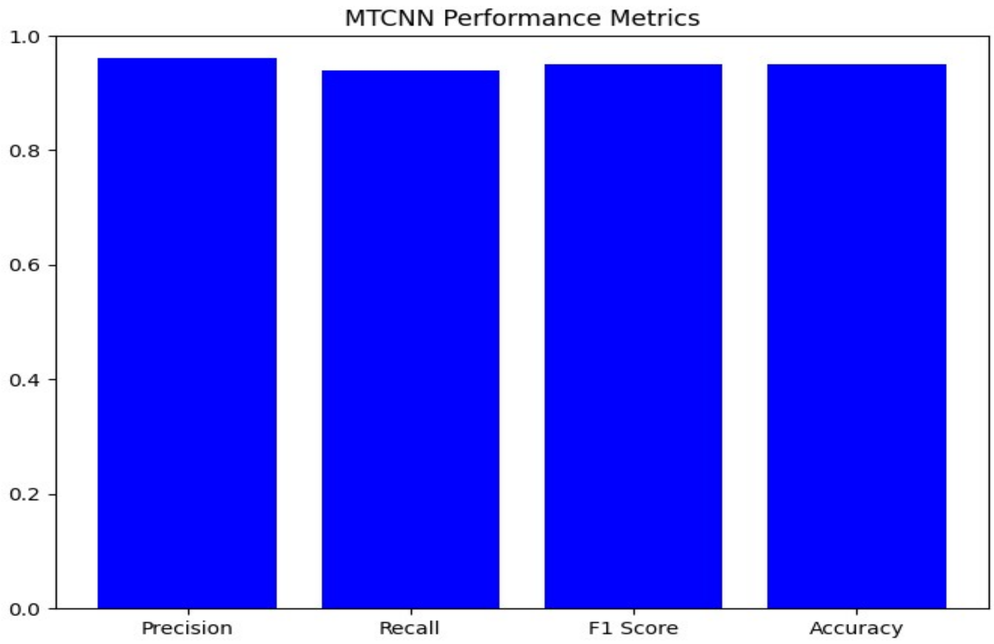

Fig 5.1.2 Performance Metrics

Based on the observations, we get the following idea about the performance of our model.

Accuracy: The accuracy of the model being measured as: (TP + TN) / (TP + TN + FP + FN).

Precision (Positive Predictive Value): It tells us about the positive predicted value. It stateshow many facial images out of the total predicted faces. It is expressed as: TP/(TP + FP).

Recall (Sensitivity or True Positive Rate): The proportion of actual faces that were correctly detected, calculated as TP / (TP + FN).

F1 Score: The harmonic mean of precision and recall, giving equal weighting between precision and recall. It is expressed as 2 * Precision * Recall/ Precision + Recall.

Accuracy of our model comes out to be 94.5%. Recall is 94% Precision 94.9% and F1 Score 94.4%.

Evaluation:

Losses:

We faced two types of losses during our training of dataset which is a custom dataset created by collecting images from our webcam, Following graph shows these losses. The graph shows losses on both training set and validation set created by us.

Classification Loss - Classification loss deals with teaching the model to correctly classifying if any portion of an input picture includes faces or not. In most cases, a binary classification is used in which a region is tagged either as positive (having a face) or negative (does not contain a face). In our implementation, we have two classes, Face and no face, which is an example of binary classification.

Bounding Box Regression Loss - In particular, the bounding box regression loss is aimed at improving the accuracy in predicting the surrounding boxes for detected faces.

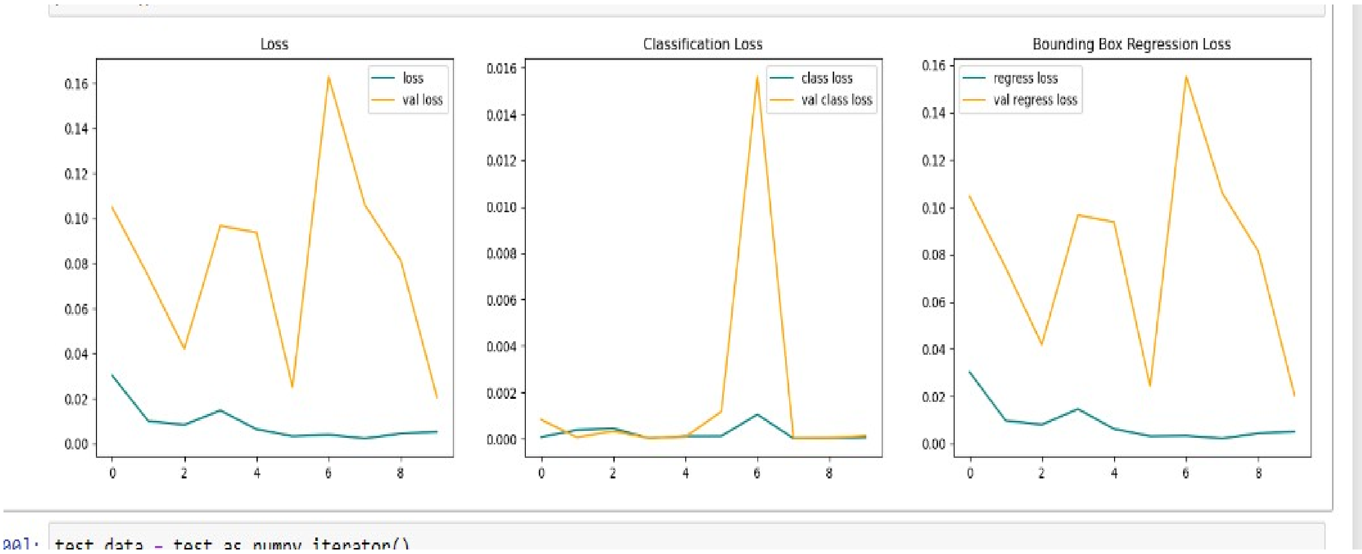

Fig 5.1.3 Losses Graph

From training of our SSD model it is clear that, both classification and regression losses are significantly lowering over each epochs, it suggests that the model is learning and adjusting its parameters in a way that improves its performance on the training data.

```
hist = model.fit(train, epochs=10, validation_data=val, callbacks=[tensorboard_callback])

Epoch 1/10
473/473 [==============================] - 331s 690ms/step - total_loss: 1.5947 - class_loss: 0.6892 - regress_loss: 1.2501 - val_total_loss: 2.6440 -
val_class_loss: 1.0016 - val_regress_loss: 2.1432
Epoch 2/10
473/473 [==============================] - 335s 704ms/step - total_loss: 1.4393 - class_loss: 0.6532 - regress_loss: 1.1127 - val_total_loss: 1.9929 -
val_class_loss: 0.9504 - val_regress_loss: 1.5177
Epoch 3/10
473/473 [==============================] - 322s 677ms/step - total_loss: 1.2741 - class_loss: 0.5894 - regress_loss: 0.9794 - val_total_loss: 2.6173 -
val_class_loss: 1.0597 - val_regress_loss: 2.0874
Epoch 4/10
473/473 [==============================] - 322s 678ms/step - total_loss: 1.0136 - class_loss: 0.4782 - regress_loss: 0.7745 - val_total_loss: 4.2755 -
val_class_loss: 2.0632 - val_regress_loss: 3.2439
Epoch 5/10
473/473 [==============================] - 322s 676ms/step - total_loss: 0.7798 - class_loss: 0.3702 - regress_loss: 0.5947 - val_total_loss: 3.0447 -
val_class_loss: 1.6848 - val_regress_loss: 2.2023
Epoch 6/10
473/473 [==============================] - 323s 678ms/step - total_loss: 0.6134 - class_loss: 0.2916 - regress_loss: 0.4677 - val_total_loss: 2.3820 -
val_class_loss: 1.1267 - val_regress_loss: 1.8187
Epoch 7/10
473/473 [==============================] - 332s 698ms/step - total_loss: 0.5481 - class_loss: 0.2598 - regress_loss: 0.4182 - val_total_loss: 5.5685 -
val_class_loss: 3.6355 - val_regress_loss: 3.7508
Epoch 8/10
473/473 [==============================] - 347s 730ms/step - total_loss: 0.4958 - class_loss: 0.2303 - regress_loss: 0.3806 - val_total_loss: 3.3642 -
val_class_loss: 2.2947 - val_regress_loss: 2.2168
Epoch 9/10
473/473 [==============================] - 385s 810ms/step - total_loss: 0.4688 - class_loss: 0.2201 - regress_loss: 0.3588 - val_total_loss: 4.3731 -
val_class_loss: 3.7319 - val_regress_loss: 2.5072
Epoch 10/10
146/473 [========>.....................] - ETA: 3:45 - total_loss: 0.4600 - class_loss: 0.2221 - regress_loss: 0.3489
```

Fig 5.1.4 Losses achieved

The validation loss is crucial for assessing how well your model generalizes to new, unseen data. If it is also decreasing over epochs, it suggests that the model is not only memorizing the training data (overfitting) but is also improving its ability to make accurate predictions on new data.

In order to better understand, how our model is working, we look at the confusion matrix of our model. In scenarios where the classes are imbalanced (one class is much more frequent than the other), accuracy alone can be misleading. The confusion matrix allows for a better evaluation by highlighting how well the model is performing on both the majority and minority classes. This is why metrics such as F1score play an important role in getting a better understanding of how our model is working.

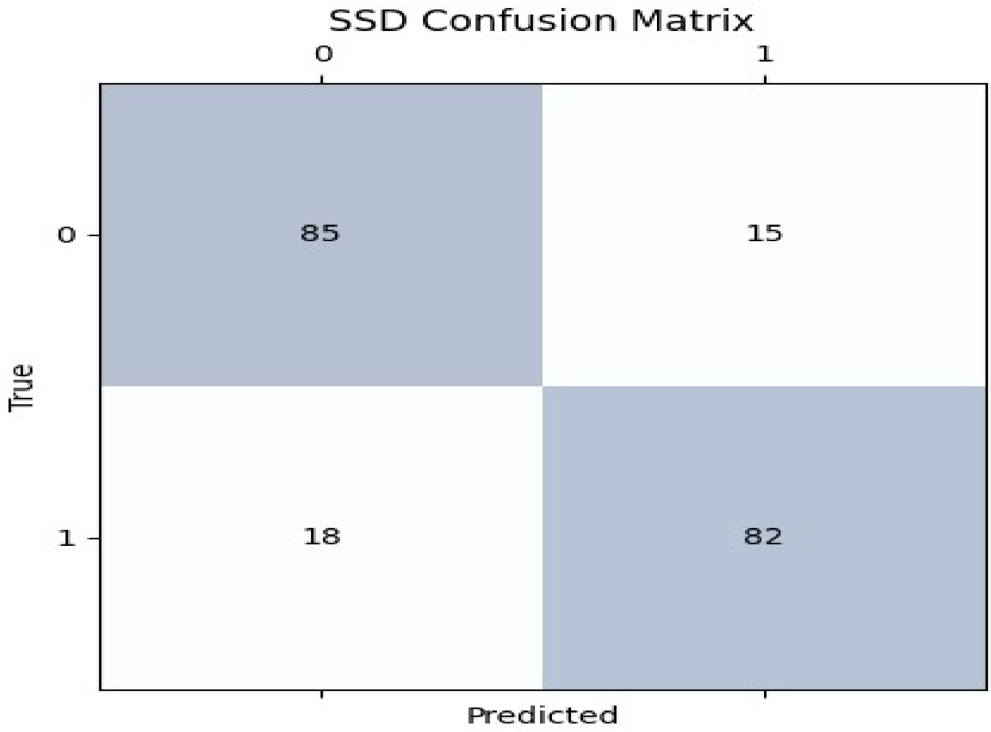

Fig 5.1.5 Confusion matrix

Based on the confusion matrix we get, we try to analyze the model in the following ways by calculating same metrics such as precision, recall, accuracy and f1-score.

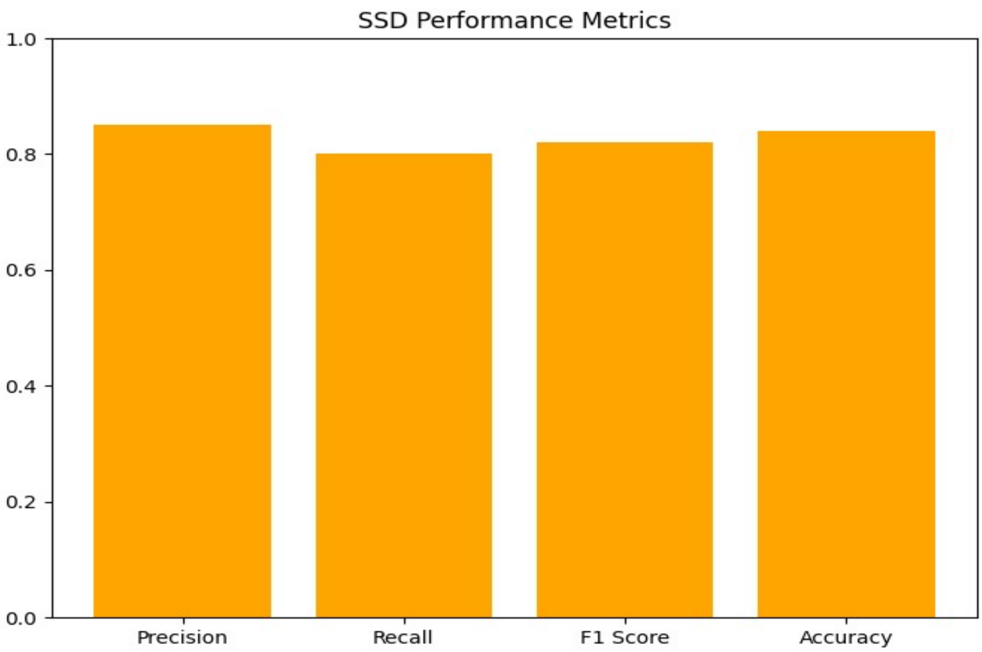

Fig 5.1.6 Evaluation metrics

Based on the above graph, we get the following metrics for our SSD model, these are,
Accuracy = 83.5%, Precision = 82%, Recall = 84.5%, F1Score = 83.2%.

YOLO

We were able to build a model of YOLO that was able to detect faces in a live video, which consisted of multiple faces of people moving and it was able to achieve this feat in a good enough frame rate.

```
Epoch     GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
 1/25       6.18G      1.808      7.236      2.078         28       800:
            Class     Images  Instances      Box(P          R      mAP50
              all         18         18      0.673      0.722      0.759

Epoch     GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
 2/25       6.43G      1.486      2.588      1.507         22       800:
            Class     Images  Instances      Box(P          R      mAP50
              all         18         18      0.546      0.602      0.622

Epoch     GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
 3/25       6.44G       1.39      1.624      1.465         33       800:
            Class     Images  Instances      Box(P          R      mAP50
              all         18         18      0.136      0.611      0.354

Epoch     GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
 4/25       6.44G      1.384      1.339      1.462         22       800:
            Class     Images  Instances      Box(P          R      mAP50
              all         18         18      0.812      0.722      0.821

Epoch     GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
 5/25       6.44G      1.401      1.071      1.439         37       800:
            Class     Images  Instances      Box(P          R      mAP50
              all         18         18        0.6      0.667      0.686

Epoch     GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
 6/25       6.45G       1.45      1.074      1.469         23       800:
            Class     Images  Instances      Box(P          R      mAP50
              all         18         18      0.521      0.944      0.744
```

Fig 5.1.7 Training of YoloV8

Training of YOLOv8 model on our custom Dataset, with each epoch we see our classification loss and box-regression losses lowering over each epoch that is trained.

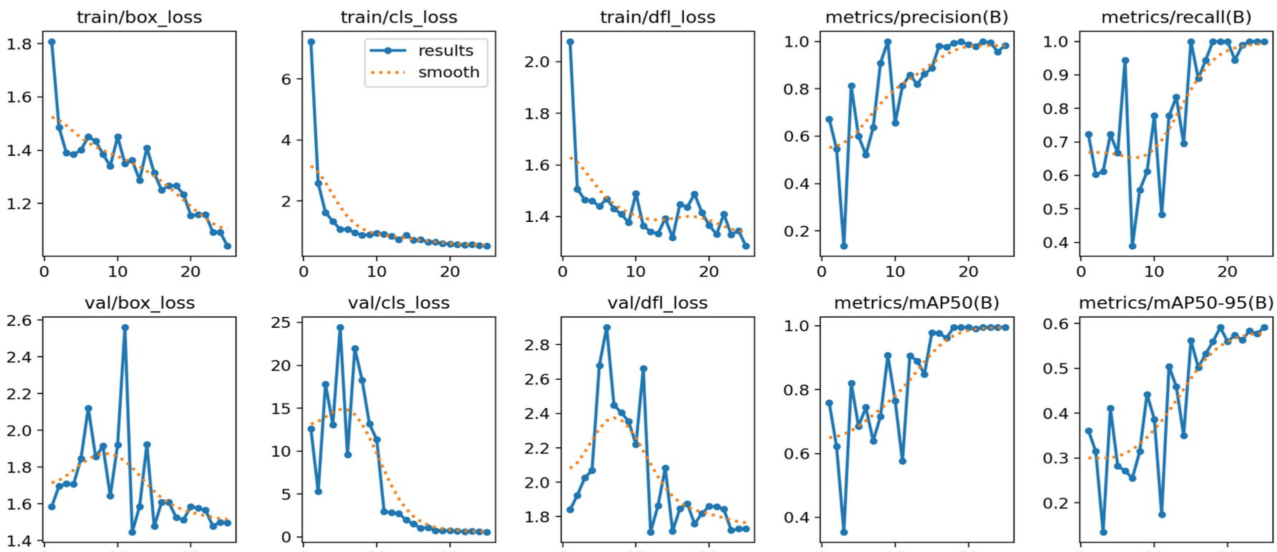

Fig 5.1.8 Losses over training

As we can see from the given graph of losses that both classification loss and regression losses decrease as we keep on training our data on increasing epochs.

Confusion matrix for our YOLO model.

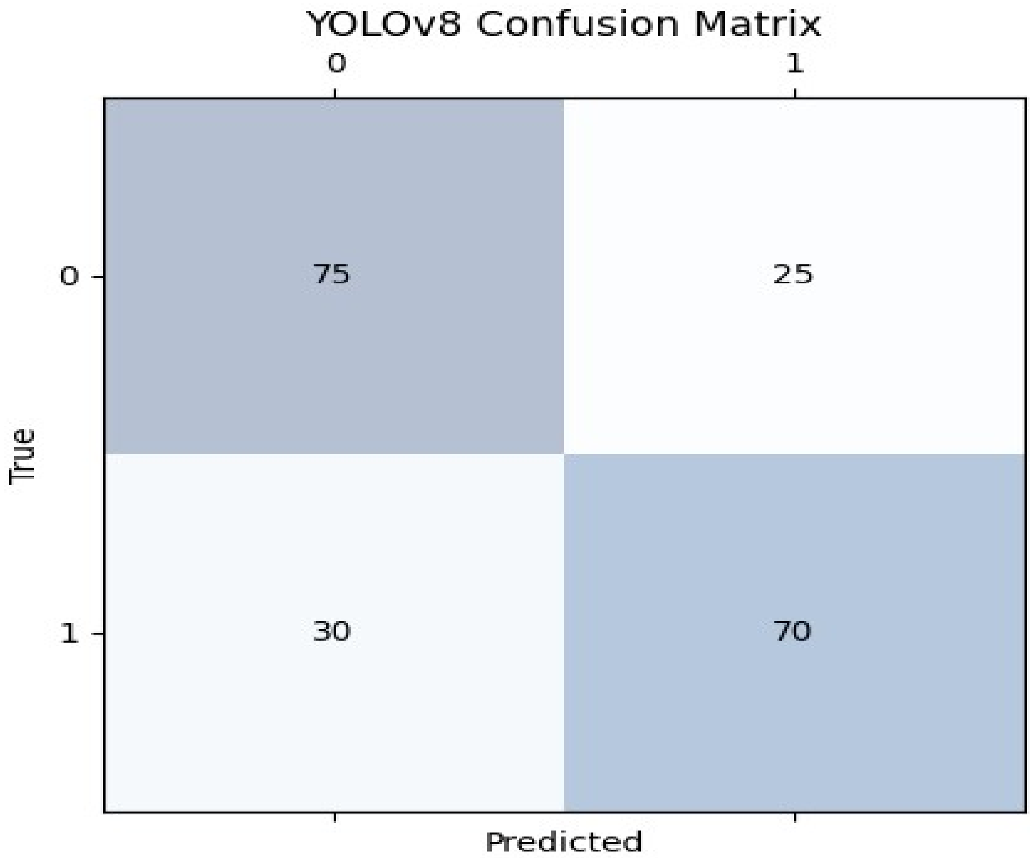

Fig 5.1.9 Confusion matrix for our model

From the given confusion we can calculate the metrics such as accuracy, precision, F1-score and recall in order to get a better understanding of how our model is working.

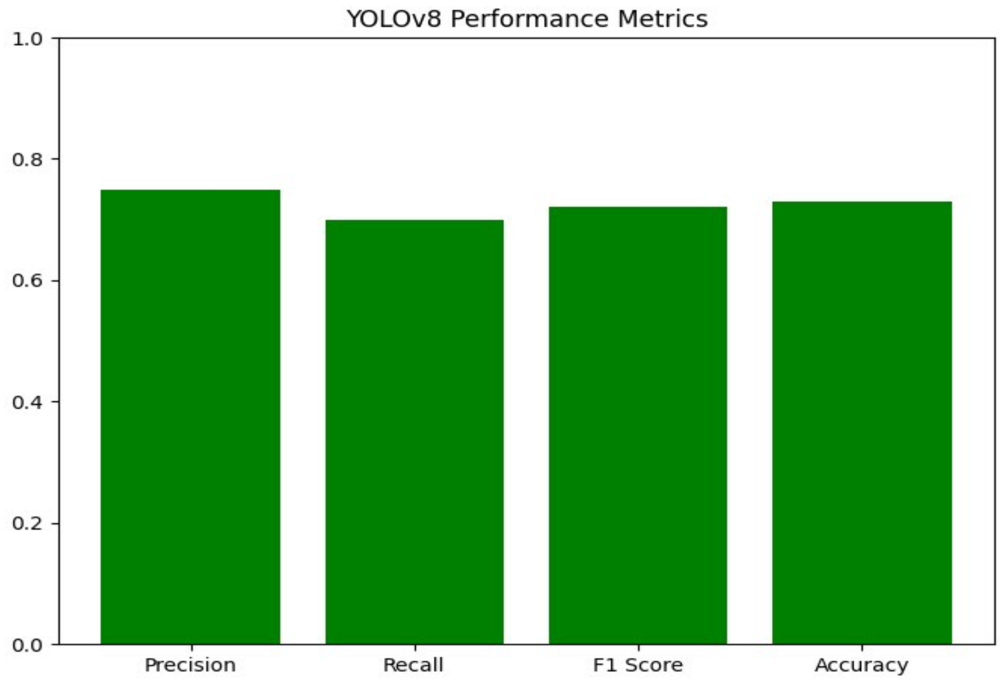

Fig 5.2.0 Evaluation Metrics

Based on the above graph, we get the following metrics for our YOLO model, these are the results we achieve.

Accuracy = 72.5%, Precision = 73.7%, Recall = 70%, F1Score = 71.8%.

Comparing all the metrics of MTCNN, SSD and YOLO, we can see a clear picture on how these models stack up against each other in terms of performance. It is also to be noted that due to our models being working on real time, there is a huge discrepancy on how they perform, also a lot of things depend on what dataset they are trained on and results may vary if the dataset is increased.

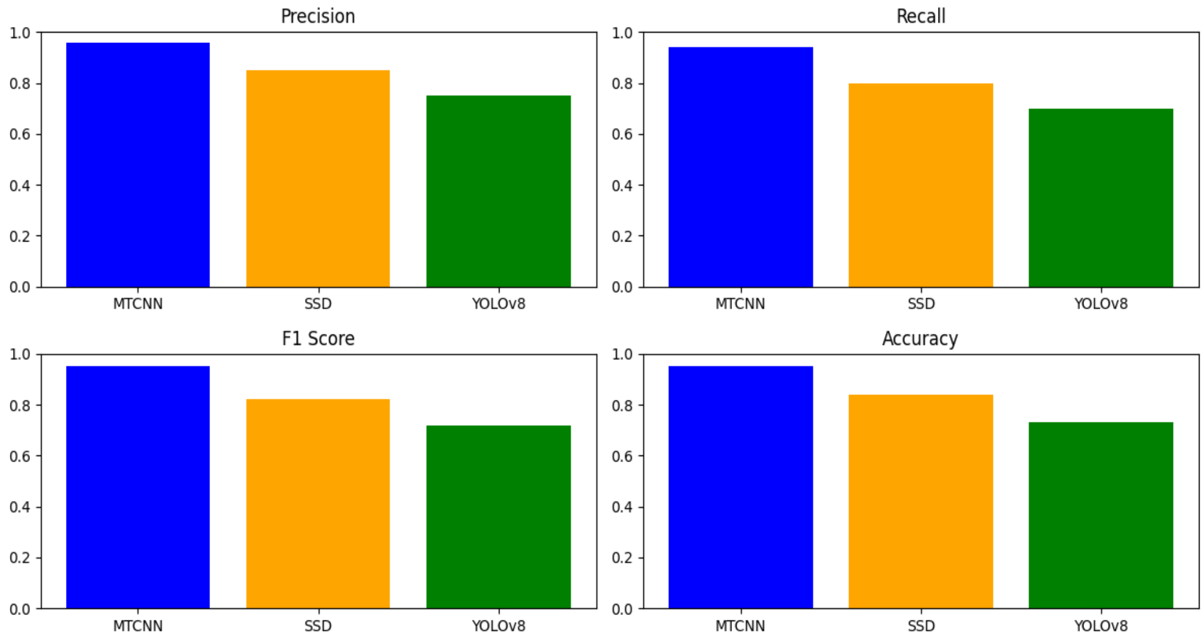

Fig 5.2.1 Comparison of all the metrics

5.1 Results

Multiple face real time detection is an important functionality that can be used in different scenarios. The rapid, accurate recognition of multiple faces at scenes like video surveillance, man-machine intercommunication, or security systems is very important in these situations to let the system respond quickly and be useful. This successful utilization of this feature implies that probably the utilized face detection algorithms such as MTCNN (Multi-task Cascaded Convolutional Networks) and many others can efficiently handle real time processing requirements.

In this case, “Real-Time” means such a face detection system has to process frames at high speed in order to follow the input stream and it has to perform at least the same rate as the resolution of the video. For achieving real-time performance of multiple faces, it’s important to consider algorithm efficiency, parallelization approaches and maybe even hardware acceleration.

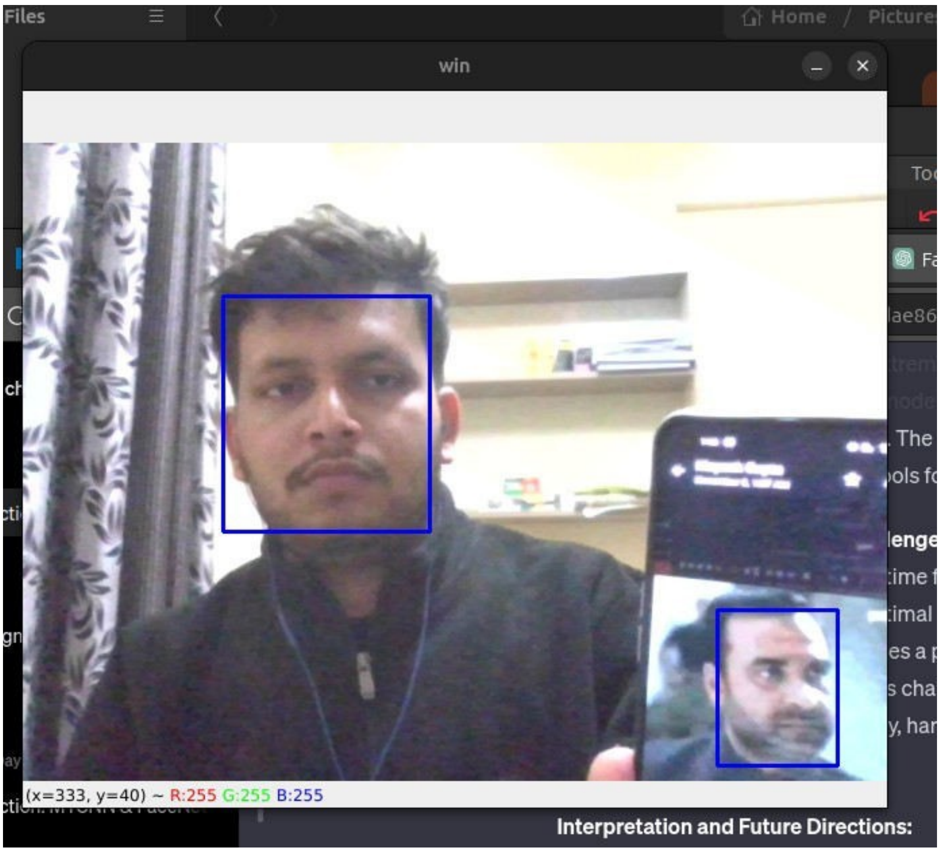

Fig 5.2.2 Face detection in real time

Face Recognition in real-time.

This marks a huge victory for the project because it signifies that the system is able to spot faces during streaming and recognize individuals within such short duration. In real-time face recognition one can rapidly process video frames or images, make decision in terms of identities of detected faces and assign them to one's previously known persons. This achievement points out a proper combination of high-powered face recognition models, like FaceNet, and RT processing technologies. Face detecting in real time gives birth to vast options such as secure access control, video surveillance, human computer interaction, etc. This success portrays that the system has met the requirements for computation needs during real-time face recognition. It is crucial in developing a fast and practicable face recognition.

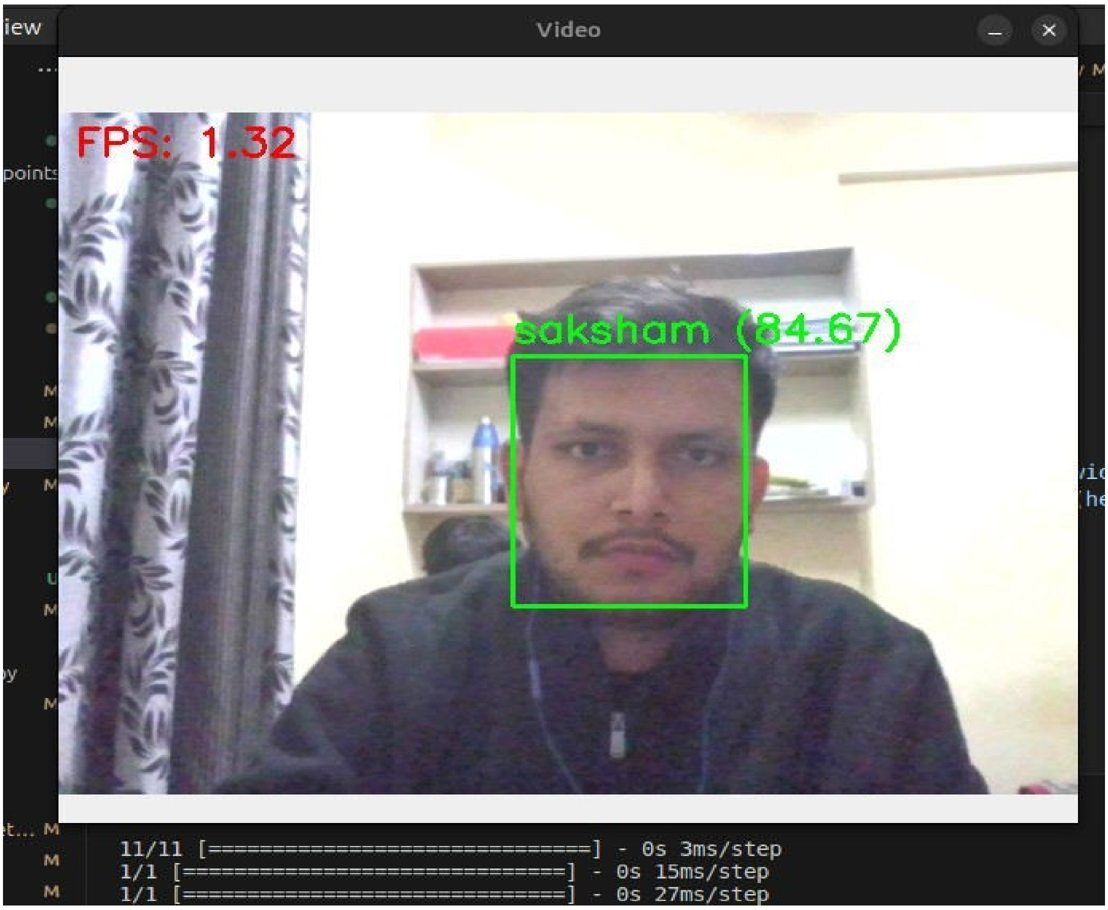

Fig 5.2.3 Face Recognition in real-time

Comparison of the three models based on FPS.

Model	FPS
YOLOV8	30
MTCNN	1.8 – 2.5
SSD	15

Table 5.1 Comparison on FPS

5.2 Limitations

Some weaknesses are present in the project which need to be taken into account for further improvement. Secondly, a small size of datasets consisting of over 5000 images may hinder model’s capability to generalize adequately for different situations. A bigger and heterogeneous dataset can increase the model’s resistance to lightings, poses, and facial expressions alterations so as to achieve better general performances.

The second limitation is that real-time detection was observed at about two frames per second (fps). It is important to attain higher frame rate since it helps in accurate, fast and effective face tracking in dynamic applications./ One can explore this limitation by investigating and optimizing the computational efficiency of the algorithms, exploring parallelization strategies.

As its limitation, this system does not have the ability to identify various faces at once during face recognition process. Ideally, in scenario with plurality of persons involved, the system should accurately and differently identify and read each face separately. For example, the face recognition algorithms can be refined to account for cases where several faces appear close together, or in other words where multi-face detection is required, as well as optimization.

These could be addressed by collecting a larger and wider variety of data through collection of high resolution images. Additionally, incorporating improved face recognition algorithms that are efficient computationally as well as implementing alternative parallel processing strategies would help. Moreover, the use of hardware accelerators can also enhance the speed at which frames are sent in real time detection or the upgrade of the computational infrastructure can assist in improving the same. Such features as refinement of algorithms on the face recognition or multiplex face recognition addition can improve an ability.

CHAPTER - 6

CONCLUSION AND FUTURE SCOPE

The project has made notable strides in the domain of real-time face detection and recognition, yet certain conclusions and insights can be drawn to shape its current status:

1. Achievements: Although this would entail certain flaws in face detection and recognition in real-time, it proves that the proposed algorithms and models are feasible. system is capable of working under dynamic conditions for recognition of separate people.

2. Limitations: These include the use of a small dataset, low frame rate during recognition, and ability to recognize only one face at a time. The limits above indicate some improvements that were needed and include an increased size of data set, refinements to the algorithm and better live processing facilities.

3. Dataset Quality: While the use of a 5000 image pool was helpful in initially developing this project, it certainly means that the dataset needs an expansion and further diversification. Expansion of the dataset size, on the other hand, would help in the development of models that generalize well across different cases.

4. Real-Time Performance: Real-time face detection at 2 fps shows that there is a performance bottleneck in the process. Resolving this challenge is important for improving the system's expediency and suitability for instantaneous purposes.

5. Multi-face Recognition: Functional problem of recognizing multiple faces by sight. This should be improved further along by developing more versatile algorithms that cater for multiple people being involved in the system.

Future Scope:

1. Dataset Enhancement: The future research ought to focus on broadening the dataset. Gathering different arrays of captures in varied situations will enhance the models and consequently increase their performances.

2. Algorithmic Refinement: Research and development efforts must continue towards perfecting face detection and recognition algorithms.

3. Real-Time Performance Optimization: Since low frame rates were observed, future endeavors should aim at enhancing algorithmic computation efficiency. Real-time performance of model can be achieved via parallelization strategies, hardware accelerators etc.

4. Multi-face Recognition: The development of the improvement for such system to distinguish different faces at once in one essential direction ahead. It includes modifying the current methods or incorporating innovative techniques for dealing with cases comprising of multiple or singular faces.

5. User Interface and Integration: Having the interface of the project as friendly and its integration with other systems and applications would increase its usability and practicability.

6. Deployment and Testing: The implementation of extensive field trials and testing will reveal how the system works under different conditions. The modifications will be guided by continuous monitoring and subsequent feedback from practical applications.

7. Collaboration and Interdisciplinary Research: Teaming up with expert specialists from other spheres like computer vision, machine learning, hardware optimization could be of great help and contribute to solving challenges as well as improving the project's innovativeness.

Finally, the project becomes a stepping stone for future studies on research and development. The mentioned shortcomings should be addressed and the proposed future scope be investigated in order to make the system more advanced and applicable in diverse spheres.

REFERENCES

[1] S. El Morabit, A. Rivenq, M.-E. Zighem, A. Hadid, A. Ouahabi, and A. Taleb-Ahmed, “Automatic Pain Estimation from Facial Expressions: A Comparative Analysis Using Off-the-Shelf CNN Architectures,” *Electronics*, vol. 10, no. 16, p. 1926, Aug. 2021, doi: 10.3390/electronics10161926.

[2] P. C Patil, A. R K, A. Kumar G, M. Bhaskar and R. N, "Analysis of Facial Expression using Deep Learning Techniques," *2023 IEEE International Conference on Integrated Circuits and Communication Systems (ICICACS)*, Raichur, India, 2023, pp. 1-8, doi: 10.1109/ICICACS57338.2023.10100110.

[3] Li, Jun & Wang, Yaoru & Fang, Guokang & Zeng, Zhigao. (2021). Real- detection tracking recognition algorithm based on multi-target faces. Multimedia Tools and Applications. 80. 10.1007/s11042-020-09601-2.

[4] *Wu, Chunming and Ying Zhang. “MTCNN and FACENET Based Access Control System for Face Detection and Recognition.” Automatic Control and Computer Sciences 55 (2021): 102-112.*

[5] Li, Hsiao-Chi & Deng, Zong-Yue & Chiang, Hsin-Han. (2020). Lightweight and Resource-Constrained Learning Network for Face Recognition with Performance Optimization. Sensors.
20. 10.3390/s20216114.

[6] Fujiyoshi, Hironobu & Hirakawa, Tsubasa & Yamashita, Takayoshi. (2019). Deep learning-based image recognition for autonomous driving. IATSS Research. 43. 10.1016/j.iatssr.2019.11.008.

[7] S. Sakshi, A. K. Gupta, S. Singh Yadav and U. Kumar, "Face Mask Detection System using CNN," *2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, Greater Noida, India, 2021, pp. 212-216, doi: 10.1109/ICACITE51222.2021.9404731.

[8] *eng Lu, Baoye Song & Lin Xu (2021) Human face recognition based on convolutional neural network and augmented dataset, Systems Science & Control Engineering, 9:sup2, 29-37, DOI: 1.01080/21642583.2020.1836526*

[9] H. b. Fredj, S. Sghaier and C. Souani, "An Efficient Face Recognition Method Using CNN," *2021 International Conference of Women in Data Science at Taif University (WiDSTaif)*, Taif, Saudi Arabia, 2021, pp. 1-5, doi: 10.1109/WiDSTaif52235.2021.9430209.

[10] Lv, Xue & Su, Mingxia & Wang, Zekun. (2021). Application of Face Recognition Method Under Deep Learning Algorithm in Embedded Systems. Microprocessors and Microsystems. 104034. 10.1016/j.micpro.2021.104034.

[11] Sanchez-Moreno AS, Olivares-Mercado J, Hernandez-Suarez A, Toscano-Medina K, Sanchez-Perez G, Benitez-Garcia G. Efficient Face Recognition System for Operating in Unconstrained Environments. J Imaging. 2021 Aug 26;7(9):161. doi: 10.3390/jimaging7090161. PMID: 34460797; PMCID: PMC8466208.

[12] Alhassan Mumuni, Fuseini Mumuni,Data augmentation: A comprehensive survey of modern approaches,Array,Volume 16,2022,100258,ISSN 2590-0056,

[13] *Qi, Delong & Tan, Weijun & Yao, Qi & Liu, Jingfeng. (2021). YOLO5Face: Why Reinventing a*

Face Detector.

[14] *Terven, Juan & Cordova-Esparza, Diana-Margarita. (2023). A Comprehensive Review of YOLO: From YOLOv1 to YOLOv8 and Beyond.*

[15] *Nagrath, P., Jain, R., Madan, A., Arora, R., Kataria, P., & Hemanth, J. (2021). SSDMNV2: A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2. Sustainable cities and society, 66, 102692. https://doi.org/10.1016/j.scs.2020.102692*

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: ....

Type of Document (Tick): **PhD Thesis** **M.Tech Dissertation/ Report** **B.Tech Project Report** **Paper**

Name: ______________________________ **Department:** _________________ **Enrolment No** _________

Contact No. ______________________________ **E-mail.** ______________________________________

Name of the Supervisor: ____

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): _________________________

____

____

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages =
- Total No. of Preliminary pages =
- Total No. of pages accommodate bibliography/references =

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at.................. (%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor) **Signature of HOD**

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

<table>
<tr><th>Copy Received on</th><th>Excluded</th><th>Similarity Index (%)</th><th colspan="2">Generated Plagiarism Report Details (Title, Abstract & Chapters)</th></tr>
<tr><td></td><td rowspan="4">• All Preliminary Pages
• Bibliography/Images/Quotes
• 14 Words String</td><td rowspan="2"></td><td>Word Counts</td><td></td></tr>
<tr><td>Report Generated on</td><td>Character Counts</td><td></td></tr>
<tr><td rowspan="2"></td><td>Submission ID</td><td>Total Pages Scanned</td><td></td></tr>
<tr><td></td><td>File Size</td><td></td></tr>
</table>

Checked by
Name & Signature **Librarian**

....

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com

AI Based Face Recognition

ORIGINALITY REPORT

12%	10%	8%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	ir.juit.ac.in:8080 Internet Source	2%
2	www.ir.juit.ac.in:8080 Internet Source	1%
3	www.mdpi.com Internet Source	1%
4	dspace.daffodilvarsity.edu.bd:8080 Internet Source	<1%
5	oajaiml.com Internet Source	<1%
6	summit.sfu.ca Internet Source	<1%
7	Alhassan Mumuni, Fuseini Mumuni. "Data augmentation: A comprehensive survey of modern approaches", Array, 2022 Publication	<1%
8	link.springer.com Internet Source	<1%

kili-technology.com

9	Internet Source	<1 %
10	Wee Lau Cheong. "Building a computation savings real-time face detection and recognition system", 2010 2nd International Conference on Signal Processing Systems, 07/2010 Publication	<1 %
11	medium.com Internet Source	<1 %
12	Jun Li, Yaoru Wang, Guokang Fang, Zhigao Zeng. "Real-time detection tracking and recognition algorithm based on multi-target faces", Multimedia Tools and Applications, 2020 Publication	<1 %
13	Yuzheng Liu, Jianxun Zhang, Lei Shi, Mingxiang Huang, Linyu Lin, Lingfeng Zhu, Xianglu Lin, Chuanlei Zhang. "Detection method of the seat belt for workers at height based on UAV image and YOLO algorithm", Array, 2024 Publication	<1 %
14	Soumya Suvra Khan, Diganta Sengupta, Anupam Ghosh, Atal Chaudhuri. "MTCNN++: A CNN-based face detection algorithm	<1 %

inspired by MTCNN", The Visual Computer, 2023
Publication

15	dspace.ada.edu.az Internet Source	<1 %
16	open-innovation-projects.org Internet Source	<1 %
17	www.coursehero.com Internet Source	<1 %
18	"Computer Vision – ECCV 2016", Springer Nature, 2016 Publication	<1 %
19	"Human-Centric Smart Computing", Springer Science and Business Media LLC, 2024 Publication	<1 %
20	Pruthvi Darshan S S, Shantakumar B Patil, Bhargav S Patil, Premjyoti. "Gesture Recognition Machine Vision Video Calling Application Using YOLOv8", 2023 22nd International Symposium on Communications and Information Technologies (ISCIT), 2023 Publication	<1 %
21	eudl.eu Internet Source	<1 %
22	ouci.dntb.gov.ua Internet Source	<1 %

23	rstudio-pubs-static.s3.amazonaws.com Internet Source	<1 %
24	mdpi-res.com Internet Source	<1 %
25	Bingtao Hu, Rongsheng Lu, Dahang Wan, Sailei Wang, Jiajie Yin. "YOLOv8-FCS: A more focused YOLOv8 model for defect detection in images of steel surface", Research Square Platform LLC, 2024 Publication	<1 %
26	Umurerwa Marie Adeline, Harerimana Gaspard, Kabandana Innocent. "Chapter 8 A Real-Time Face Recognition Attendance Using Machine Learning", Springer Science and Business Media LLC, 2023 Publication	<1 %
27	www.tandfonline.com Internet Source	<1 %
28	Hironobu Fujiyoshi, Tsubasa Hirakawa, Takayoshi Yamashita. "Deep learning-based image recognition for autonomous driving", IATSS Research, 2019 Publication	<1 %
29	ojs.trp.org.in Internet Source	<1 %

30	Preeti Nagrath, Rachna Jain, Agam Madan, Rohan Arora, Piyush Kataria, Jude Hemanth. "SSDMNV2: A Real time DNN-Based Face Mask Detection System using Single Shot Multibox Detector and MobileNetV2", Sustainable Cities and Society, 2020 Publication	<1 %
31	journal.esj.edu.iq Internet Source	<1 %
32	viso.ai Internet Source	<1 %
33	wiredspace.wits.ac.za Internet Source	<1 %
34	www.ijritcc.org Internet Source	<1 %
35	Gagandeep Kaur, Ritesh Sinha, Puneet Kumar Tiwari, Srijan Kumar Yadav, Prabhash Pandey, Rohit Raj, Anshu Vashisth, Manik Rakhra. "Face Mask Recognition System using CNN Model", Neuroscience Informatics, 2021 Publication	<1 %
36	www.escholar.manchester.ac.uk Internet Source	<1 %
37	elibrary.almaata.ac.id Internet Source	<1 %

38	Sang-Heon Lee, Myoung-Kyu Sohn, Dong-Ju Kim, Byungmin Kim, Hyunduk Kim. "Face recognition of near-infrared images for interactive smart TV", Proceedings of the 27th Conference on Image and Vision Computing New Zealand, 2012 Publication	<1%
39	www-emerald-com-443.webvpn.sxu.edu.cn Internet Source	<1%
40	www.jetir.org Internet Source	<1%
41	Shunji Hashiguchi, Taro Shibanoki. "A Task Estimation Method Based on Image Recognition and Its Application to EMG Prosthetic Hand Control", 2023 9th International Conference on Virtual Reality (ICVR), 2023 Publication	<1%
42	f5.pm Internet Source	<1%
43	web.stanford.edu Internet Source	<1%
44	www.researchgate.net Internet Source	<1%
45	labstic.univ-guelma.dz Internet Source	<1%

46	tede2.pucrs.br Internet Source	<1 %
47	Yan Fan. "Detection of driver fatigue driving behaviour based on deep learning algorithm from the perspective of traffic psychology", Journal of Psychology in Africa, 2022 Publication	<1 %
48	tudr.thapar.edu:8080 Internet Source	<1 %
49	www.deepai.org Internet Source	<1 %
50	Harshit Garg, Rahul Rana, Sanjeev Kumar Prasad. "A Model to Detect Face Mask Using Deep Learning", 2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), 2022 Publication	<1 %
51	Priyadarshini C Patil, Ashwin R K, Arvind Kumar G, M Bhaskar, Rajesh N. "Analysis of Facial Expression using Deep Learning Techniques", 2023 IEEE International Conference on Integrated Circuits and Communication Systems (ICICACS), 2023 Publication	<1 %
52	Savita Bakare, Prema T Akkasaligar, Shrujal Kulkarni, Prajaktha Kadamba, Saksha Mutnal,	<1 %

Shreyank Belagali. "Multimodal Framework for Multilingual Social Interaction Assistive Application (MSIAA) for Elderly People", 2023 2nd International Conference on Edge Computing and Applications (ICECAA), 2023
Publication

---

53 Zhaowei Cai, Quanfu Fan, Rogerio S. Feris, Nuno Vasconcelos. "Chapter 22 A Unified Multi-scale Deep Convolutional Neural Network for Fast Object Detection", Springer Science and Business Media LLC, 2016 <1%
Publication

---

54 annals-csis.org <1%
Internet Source

---

55 journalajrcos.com <1%
Internet Source

---

56 scholar.lib.ntnu.edu.tw <1%
Internet Source

---

57 vsip.info <1%
Internet Source

---

58 "Computer Vision – ECCV 2018", Springer Science and Business Media LLC, 2018 <1%
Publication

---

59 "Smart Trends in Information Technology and Computer Communications", Springer Science and Business Media LLC, 2016 <1%
Publication

60 Carrault, G.. "Temporal abstraction and inductive logic programming for arrhythmia recognition from electrocardiograms", Artificial Intelligence In Medicine, 200307 <1%

Publication

61 Hyeonwoo Kim, Hyungjoon Kim, Eenjun Hwang. "Real-Time Facial Feature Extraction Scheme Using Cascaded Networks", 2019 IEEE International Conference on Big Data and Smart Computing (BigComp), 2019 <1%

Publication

62 M.J. Seow, R. Gottumukkal, D. Valaparla, K.V. Asari. "A robust face recognition system for real time surveillance", International Conference on Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004., 2004 <1%

Publication

63 Slo-Li Chu, Chien-Fang Chen, Yu-Chen Zheng. "CFSM: a novel frame analyzing mechanism for real-time face recognition system on the embedded system", Multimedia Tools and Applications, 2021 <1%

Publication

64 Zhenni Cai, Qingshan Liu, Shanmin Wang, Bruce Yang. "Joint Head Pose Estimation with Multi-task Cascaded Convolutional Networks for Face Alignment", 2018 24th International <1%

Conference on Pattern Recognition (ICPR), 2018
Publication

65	dspace.univ-ouargla.dz Internet Source	<1 %
66	ijai.iaescore.com Internet Source	<1 %
67	livrepository.liverpool.ac.uk Internet Source	<1 %
68	www.hindawi.com Internet Source	<1 %
69	www.ncbi.nlm.nih.gov Internet Source	<1 %
70	www.thefreelibrary.com Internet Source	<1 %
71	Hyung Jin Chang. "Emotional head robot with behavior decision model and face recognition", 2007 International Conference on Control Automation and Systems, 10/2007 Publication	<1 %
72	Pham, Minh-Tan, Grégoire Mercier, Oliver Regniers, and Julien Michel. "Texture Retrieval from VHR Optical Remote Sensed Images Using the Local Extrema Descriptor with Application to Vineyard Parcel Detection", Remote Sensing, 2016.	<1 %

Publication

73	Safaa El Morabit, Atika Rivenq, Mohammed-En-nadhir Zighem, Abdenour Hadid, Abdeldjalil Ouahabi, Abdelmalik Taleb-Ahmed. "Automatic Pain Estimation from Facial Expressions: A Comparative Analysis Using Off-the-Shelf CNN Architectures", Electronics, 2021 Publication	<1%
74	Sudha Mishra, D. Thamaraiselvi, Sumit Dhariwal, Deepti Aggarwal, J.V.N. Ramesh. "LSCO: Light spectrum chimp optimization based spinalnet for live face detection and recognition", Expert Systems with Applications, 2024 Publication	<1%
75	Xue Lv, Mingxia Su, Zekun Wang. "Application of Face Recognition Method Under Deep Learning Algorithm in Embedded Systems", Microprocessors and Microsystems, 2021 Publication	<1%
76	Alejandra Sarahi Sanchez-Moreno, Jesus Olivares-Mercado, Aldo Hernandez-Suarez, Karina Toscano-Medina et al. "Efficient Face Recognition System for Operating in Unconstrained Environments", Journal of Imaging, 2021 Publication	<1%

77 Elliot Mbunge, Sakhile Simelane, Stephen G Fashoto, Boluwaji Akinnuwesi, Andile S Metfula. "Application of Deep Learning and Machine Learning Models to Detect COVID-19 Face Masks - A Review", Sustainable Operations and Computers, 2021 <1%
Publication

---

78 Peng Zhao, Xinrui Lyu, Richard O. Sinnott, Shimin Wei. "Crowd-Counting through a Cascaded, Multi-Task Convolutional Neural Network", Proceedings of the 6th IEEE/ACM International Conference on Big Data Computing, Applications and Technologies - BDCAT '19, 2019 <1%
Publication

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		