

FACENET: UNVEILING THE MECHANICS OF ADVANCED FACE RECOGNITION THROUGH COMPARATIVE ANALYSIS

*A project report submitted in partial fulfilment
of the requirements for the award of the degree of*

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING (Internet of Things)

Submitted by

I. SAI PAVAN

(20BQ1A4918)

B. HARICHARAN REDDY

(20BQ1A4910)

K. NAVEEN CHAND

(21BQ5A4901)

under the esteemed guidance of

Mrs. M. RAJYA LAKSHMI

Associate Professor



[Program: Computer Science and Engineering (Internet of Things) – CSO]

VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

(Autonomous)

Approved by AICTE, Permanently Affiliated to JNTUK, NAAC Accredited with ‘A’

Grade, ISO 9001:2015 Certified

Nambur (V), Pedakakani (M), Guntur (Dt.), Andhra Pradesh – 522 508

APRIL - 2024

[Program: Computer Science and Engineering (Internet of Things) – CSO]

VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

(Autonomous)

**Approved by AICTE, Permanently Affiliated to JNTUK, NAAC Accredited with ‘A’
Grade, ISO 9001:2015 Certified**

Nambur (V), Pedakakani (M), Guntur (Dt.), Andhra Pradesh – 522 508



CERTIFICATE

This is to certify that the project report entitled “**FACENET: UNVEILING THE MECHANICS OF ADVANCED FACE RECOGNITION THROUGH COMPARATIVE ANALYSIS**” is being submitted by **I. Sai Pavan** (Regd.No: **20BQ1A4918**), **B. Haricharan Reddy** (Regd.No: **20BQ1A4910**), **K. Naveen Chand** (Regd.No: **21BQ5A4901**) in partial fulfillment of the requirement for the award of the degree of the **Bachelor of Technology** in **Computer Science and Engineering (Internet of Things)** to the Vasireddy Venkatadri Institute of Technology is a record of bonafide work carried out by them under my guidance and supervision.

The results embodied in this project have not been submitted to any other university or institute for the award of any degree or diploma.

Signature of the Supervisor

Mrs. M. Rajya Lakshmi
Associate Professor,
Department of CSO, VVIT.

Head of the Department

Dr. Chintalapudi V Suresh
Professor & HoD,
Department of CSO, VVIT.

DECLARATION

We hereby declare that the work embodied in this project entitled **“FACENET: UNVEILING THE MECHANICS OF ADVANCED FACE RECOGNITION THROUGH COMPARATIVE ANALYSIS”**, which is being submitted by us in requirement for the B. Tech Degree in **Computer Science and Engineering (Internet of Things)** from Vasireddy Venkatadri Institute of Technology, is the result of investigations carried out by us under the supervision of Mrs. M. Rajya Lakshmi, Associate Professor.

The work is original and the results in this thesis have not been submitted elsewhere for the award of any degree or diploma.

Signature of the Candidates

I. Sai Pavan (Regd.No: **20BQ1A4918**)

B. Haricharan Reddy (Regd.No: **20BQ1A4910**)

K. Naveen Chand (Regd.No: **21BQ5A4901**)

ACKNOWLEDGEMENT

We take this opportunity to express deepest gratitude and appreciation to all those people who made this project work easier with words of encouragement, motivation, discipline, and faith by offering different places to look to expand my ideas and help me towards the successful completion of this project work.

First and foremost, we express deep gratitude to **Sri. Vasireddy Vidya Sagar**, Chairman, Vasireddy Venkatadri Institute of Technology for providing necessary facilities throughout the B.Tech programme.

We express sincere thanks to **Dr. Y. Mallikarjuna Reddy**, Principal, Vasireddy Venkatadri Institute of Technology for his constant support and cooperation throughout the B.Tech programme.

We express sincere gratitude to **Dr. Ch. Venkata Suresh**, Professor & HOD, Computer Science & Engineering (Internet of Things), Vasireddy Venkatadri Institute of Technology for his constant encouragement, motivation and faith by offering different places to look to expand my ideas.

We would like to express sincere gratitude to our Guide **Mrs. M. Rajya Lakshmi**, Associate Professor, CSO for her insightful advice, motivating suggestions, invaluable guidance, help and support in successful completion of this project.

We would like to take this opportunity to express thanks to the **Teaching and NonTeaching** Staff in the Department of Computer Science & Engineering, VVIT for their invaluable help and support.

Names of Candidates

I. Sai Pavan (20BQ1A4918)

B. Haricharan Reddy (20BQ1A4910)

K. Naveen Chand (21BQ5A4901)

Department Vision

To accomplish the aspirations of emerging engineers to attain global intelligence by obtaining computing and design abilities through communication that elevate them to meet the needs of industry, economy, society, environmental and global.

Department Mission

- To mould the fresh minds into highly competent IoT application developers by enhancing their knowledge and skills in diverse hardware and software design aspects for covering technologies and multi-disciplinary engineering practices.
- To provide the state-of-the-art facilities to forge the students in industry-ready in IoT system development.
- To nurture the sense of creativity and innovation to adopt the socio-economic related activities.
- To promote collaboration with the institutes of national and international repute with a view to have best careers.
- To enable graduates to emerge as independent entrepreneurs and future leaders.

Program Educational Objectives (PEOs)

PEO-1: To formulate the engineering practitioners to solve industry's technological problems

PEO-2: To engage the engineering professionals in technology development, deployment and engineering system implementation

PEO-3: To instill professional ethics, values, social awareness and responsibility to emerging technology leaders

PEO-4: To facilitate interaction between students and peers in other disciplines of industry and society that contribute to the economic growth.

PEO-5: To provide the technocrats the amicable environment for the successful pursuing of engineering and management.

PEO-6: To create right path to pursue their careers in teaching, research and innovation.

Program Outcomes (POs)

PO1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues, and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: Individual and teamwork: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: Life-long learning: Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes (PSOs)

PSO-1: Proficient and innovative with a strong cognizance in the arenas of sensors, IoT, data science, controllers, and signal processing through the application of acquired knowledge and skills.

PSO-2: Apply cutting-edge techniques and tools of sensing and computation to solve multi-disciplinary challenges in industry and society.

PSO-3: Exhibit independent and collaborative research with strategic planning while demonstrating professional and ethical responsibilities of the engineering profession.

Project Outcomes

Students who complete a minor project will:

- PW-01.** Use the design principles and develop concept for the project.
- PW-02.** Estimate the time frame and cost for the project execution and completion.
- PW-03.** Analyze the project progress with remedial measures individual in a team.
- PW-04.** Examine the environmental impact of the project.
- PW-05.** Demonstrate the project functionality along with report and presentation.
- PW-06.** Apply the Engineering knowledge in design and economically manufacturing of components to support the society need.
- PW-07.** Assess health, safety and legal relevant to professional engineering practices.
- PW-08.** Comply the environmental needs and sustainable development.
- PW-09.** Justify ethical principles in engineering practices.
- PW-010.** Perform multi-disciplinary task as an individual and / or team member to manage the project/task.
- PW-011.** Comprehend the Engineering activities with effective presentation and report.
- PW-012.** Interpret the findings with appropriate technological / research citation.

MAPPING OF PROJECT OUTCOMES TO POs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
PW-01	3	2	2	2								
PW-02	3	2	2								3	
PW-03	3	3		2	3					3		
PW-04	3					3	3	3				3
PW-05	3	2									3	
PW-06	3	2	2	2	3							
PW-07						3						
PW-08							3					
PW-09								3				
PW-10									3		3	
PW-11										3		
PW-12												3
PW-PO	3	2	2	2	3	3	3	3	3	3	3	3

MAPPING OF PROJECT OUTCOMES TO PSOs

	PSO1	PSO2	PSO3
PW-01	2	2	2
PW-02			2
PW-03			
PW-04	3	3	3
PW-05			
PW-06	2	2	2
PW-07	2	2	2
PW-08	1	1	1
PW-09	2	2	2
PW-10	2	2	2
PW-11	2	2	2
PW-12	1	1	1
PW-PSO	2	2	2

Note: Strong – 3, Moderate – 2, Low – 1

CONTENTS

	Page No
ABSTRACT	i
LIST OF TABLES	ii
LIST OF FIGURES	iii
CHAPTER-1 INTRODUCTION	01
1.1	Triplet Loss and Selection
	03
1.2	Triplet Selection
	04
1.3	Deep Learning Basics
	05
1.3.1	Stochastic Gradient Descent
	05
1.3.2	AdaGrad
	06
1.3.3	ReLU
	07
1.4	CNN Architectures
	08
1.4.1	Zeiler & Fergus architecture
	08
1.4.2	Inception Model
	09
1.5	Evaluation
	10
CHAPTER-2 EXISTING MODELS	12
CHAPTER-3 SYSTEM DESIGN	19
3.1	Preprocessing
	21
3.2	Training
	21
CHAPTER-4 RESULTS AND ANAKYSIS	32
4.1	Comparision with previous methods
	35

4.2	Effect of CNN Model	36
4.3	Sensitivity to Image Quality	37
4.4	Embedding Dimensionality	38
4.5	Amount of Training Data	38
4.6	Performance on LFW	39
4.7	Performance on YouTube Faces DB	41
4.8	Face Clustering	41
CHAPTER-5 CONCLUSIONS AND FUTURE SCOPE		43
5.1	Conclusion	44
5.2.	Future Scope	45
REFERENCES		47

ABSTRACT

Facial recognition technology, an integral element in identifying individuals based on distinct patterns, is gaining widespread traction for diverse applications, especially within security systems. Various methodologies for facial recognition have been proposed to elevate precision, and FaceNet has emerged as an innovative approach rooted in deep convolutional networks and triplet loss training. Nevertheless, the intricate nature and time requirements of the training procedure prompted the incorporation of TensorFlow machine learning and pre-established models, resulting in a substantial reduction in training duration. This approach performs surveys, assesses performance, and compares accuracy between FaceNet and assorted previously developed facial recognition techniques. The investigation delves into FaceNet's effectiveness in facial recognition, considering its distinct training methodology and efficiency in representation, accomplishing cutting-edge performance with a mere 128 bytes allocated per face. The manuscript contributes to unraveling the complexities of FaceNet, presenting an exhaustive comparative analysis with prevailing models and deliberating on its potential influence on the trajectory of facial recognition technology. Ethical considerations, practical applications, and encountered challenges are also scrutinized in this all-encompassing exploration of FaceNet's facets.

LIST OF TABLES

Table. No	Description	Page No
3.1	Face Image Database Used	20
3.2	Pre-Trained Models	21
3.3	Face Recognition Results using FaceNet	34
3.4	Comparative Analysis	36
3.5	Image Quality	37
3.6	Embedding Dimensionality	37
3.7	Training Data Size	38

LIST OF FIGURES

Figure.No	Description	Page No
1.1	Triplet Loss	02
1.2	Triplet Loss and Selection	03
1.3	Formula of Triplet Loss	04
1.4	Terms of Triplet Loss	04
1.5	Equations of Hard positives and hard negatives	04
1.6	SGD	05
1.7	AdaGrad	06
1.8	ReLU Activation Function	08
1.9	Zeiler & Fergus architecture	09
1.10	Inception Network	10
2.1	High Level Modal Structure	14
2.2	Triplet loss example Anchor Positive, Negative	16
3.1	An instance of cropping the image during the preprocessing stage using the MTCNN method	21
3.2	Training Process	22
3.3	Architecture of the proposed attendance system	23
3.4	Flow chart for facial recognition	24
3.5	Face registration	25
3.6	68 facial landmarks	27
3.7	Feature Extraction	28
3.8	Taking Attendance	29
3.9	Attendance Tracker Sheet	31
4.1	LFW errors	40
4.2	Face Clustering	42

5.1	Comparative Analysis of Algorithms on different datasets	44
-----	--	----

CHAPTER-1

INTRODUCTION

FaceNet offers a consolidated representation for face recognition, authentication, and grouping assignments. It transforms every facial image into a Euclidean realm where the disparities reflect facial resemblance. For instance, a depiction of individual A will be positioned nearer to all other depictions of individual A, contrasting with depictions of any other individuals within the dataset.

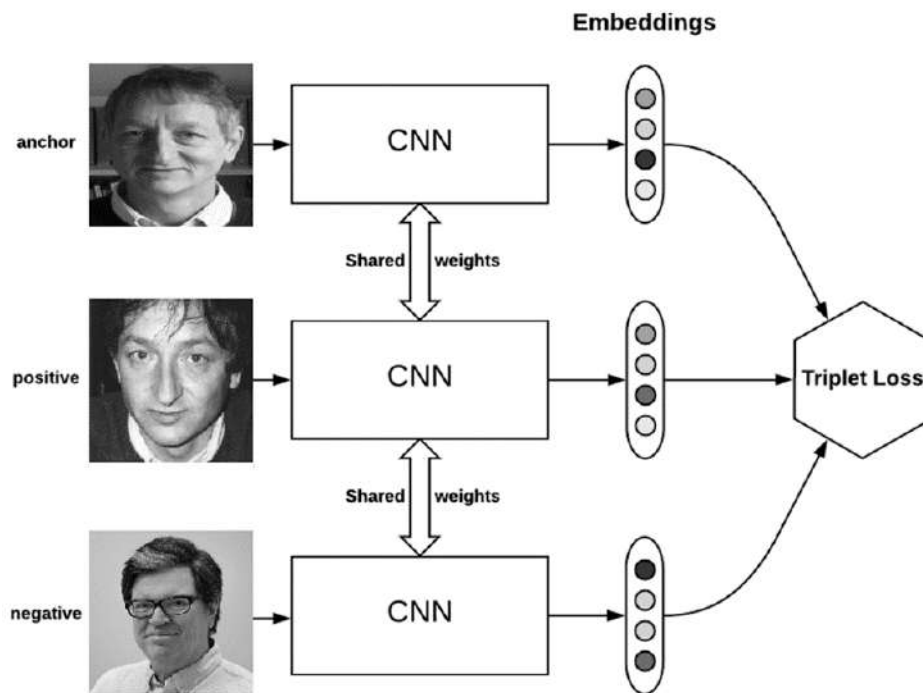


Fig 1.1: Triplet Loss

A key contrast between FaceNet and other methods lies in its approach: rather than relying on bottleneck layers for recognition or verification tasks, it learns the mapping directly from images, generating embeddings. Once these embeddings are generated, various tasks such as verification and recognition can be executed using conventional techniques pertinent to that domain, leveraging these freshly created embeddings as the feature vector. For instance, facial recognition can employ k-NN with embeddings serving as the feature vector, while clustering faces can utilize any clustering technique.

The crucial point to highlight here is that FaceNet doesn't introduce any novel algorithms for the tasks mentioned; instead, it focuses on creating embeddings directly usable for face recognition, verification, and clustering.

Utilizing a deep convolutional neural network (CNN), FaceNet is structured to train embeddings where the squared L2 distance reflects face similarity. During training, images undergo scaling, transformation, and precise cropping around the facial region.

Another noteworthy facet of FaceNet pertains to its loss function. It adopts the triplet loss function (as depicted in Fig 1.1). This involves the requirement of three images for computation: anchor, positive, and negative.

1.1 TRIPLET LOSS AND SELECTION

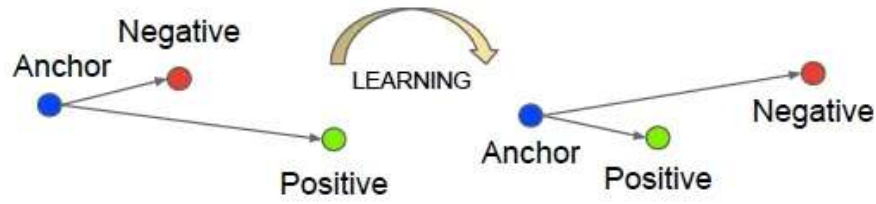


Fig 1.2: Triplet Loss and Selection

The concept driving the triplet loss function is to ensure that the embedding of an anchor image (depicting a particular person, say A) is closer to the embeddings of positive images (representing all images of person A) compared to the embeddings of negative images (depicting all other individuals).

Put differently, the aim is for the distance between the embedding of the anchor image and the embeddings of positive images to be lesser than the distance between the embedding of the anchor image and the embeddings of negative images. Formally, the triplet loss function can be defined as follows:

$$\sum_i^N \left[\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha \right]_+$$

Fig 1.3: Formula of Triplet Loss

- x_i -- It represents an image
- $f(x_i)$ -- It represents the embedding of an image
- α -- It represents the margin between positive and negative pairs

Fig 1.4: Terms of Triplet Loss

In this context, the superscripts "a," "p," and "n" represent the anchor, positive, and negative images, respectively.

The parameter alpha serves as the margin between positive and negative pairs. It acts as a threshold value that governs the disparity between image pairs. For instance, if alpha is set to 0.5, it implies that the difference between anchor-positive and anchor-negative image pairs should be at least 0.5.

1.2 TRIPLET SELECTION

Ensuring the selection of appropriate image pairs is paramount since numerous pairs may meet the condition, thereby hindering significant learning and slowing convergence. To expedite convergence, it is vital to choose triplets that defy the triplet constraint. Basically, the following elements are required for selection:

$$\text{Argmax } || f(x_i^a) - f(x_i^p) ||_2^2 \quad \text{-- Eq(1)}$$

$$\text{Argmin } || f(x_i^a) - f(x_i^n) ||_2^2 \quad \text{-- Eq(2)}$$

Fig 1.5: Equations of Hard positives and hard negatives

Equation (1) suggests that provided an anchor image depicting person A, the objective is to identify a positive image of A where the distance between them is maximized.

Equation (2) implies that when presented with an anchor image portraying person A, the aim is to pinpoint a negative image for which the distance between them is minimized.

In essence, the focus lies on selecting the hard positives and the hard negatives. This methodology expedites convergence as the model acquires meaningful representations. However, a challenge arises with this approach: computing the hard positives and the hard negative across the entire dataset proves computationally infeasible.

An alternative approach involves computing hard positives and negatives within the mini batch. In this scenario, approximately 1000–2000 samples (with a typical batch size of around 1800) are chosen in most experiments.

To ensure meaningful representations of anchor-positive distances, it's imperative to restrict the number of samples per identity within each mini-batch. The objective is to ensure approximately 40 faces per identity within each mini batch, along with the inclusion of randomly selected negative faces.

1.3 DEEP LEARNING BASICS

FaceNet employs Stochastic Gradient Descent (SGD) in combination with standard backpropagation and AdaGrad optimization techniques for training Convolutional Neural Networks (CNNs). The starting learning rate is set at 0.05, while alpha is configured to 0.2. ReLU is selected as the activation function.

1.3.1 STOCHASTIC GRADIENT DESCENT

It serves as an optimization technique employed to optimize the loss function.

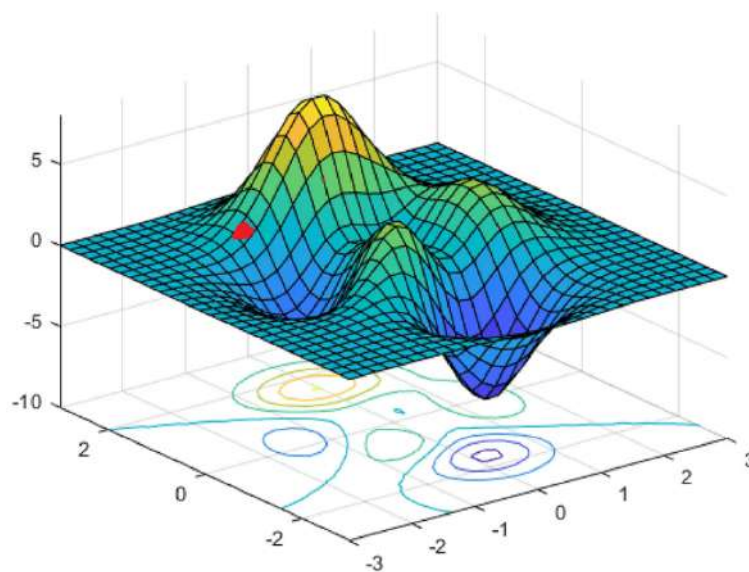


Fig 1.6: SGD

The 2 axes (x and y) depict weights, while the 3rd axis (z) signifies loss concerning those 2 weights.

Designating the red point as A, this journey commences here. The essence of SGD lies in navigating this hill-like structure to attain the global minimum (the lowest point of hill). At this moment, the "Descent" aspect of SGD should be clear. Let's now delve into "Gradient" part.

Gradient merely indicates the direction of the steepest ascent in an n-dimensional plane (akin to the derivative, which indicates slope of a line). It's crucial to emphasize that it provides the direction of the steepest ascent, not the descent. Therefore, negative of the value provided by the gradient is taken to move downhill.

1.3.2 AdaGrad

AdaGrad is employed to produce adaptive learning rates since constant learning rates are ineffective in deep learning. In the case of CNNs, where each layer is tasked with detecting various features (such as patterns, edges etc.), a constant learning rate proves inadequate. This is because different layers in the network necessitate distinct learning rates for optimal performance. To gain a clearer understanding of AdaGrad, let's examine a few equations.

$$W_t = W'_{t-1} - \eta * g_t \quad \text{-- Eq (1)}$$

$$W_t = W'_{t-1} - \eta'_t * g_t \quad \text{-- Eq (2)}$$

$$\eta'_t = \eta / \text{sqrt}(\alpha_{t-1} + \epsilon) \quad \text{-- Eq (3)}$$

$$\alpha_{t-1} = \sum_{i=1}^{t-1} g_i^2 \quad \text{-- Eq (4)}$$

$$g_1^2 + g_2^2 + g_3^2 + \dots + g_{t-1}^2$$

(Sum of squares of gradients upto t-1)

$$g_i = (\partial L / \partial W)$$

Fig 1.7: AdaGrad

Equation (1) — This represents the standard weight update equation of SGD, utilizing a fixed learning rate (η).

Equation (2) — Here, the weight update equation of AdaGrad is encountered, employing an adaptive learning rate ($\eta't$).

Equation (3) — This equation outlines the formula for computing the adaptive learning rate.

Equation (4) — It illustrates the formula for calculating G_{t-1} .

The value G_{t-1} represents the sum of squares of gradients up to iteration $t-1$. Here, 't' denotes the number of iterations. Hence, at each step, calculate the gradient and accumulate their squares to generate G_{t-1} . Given that G_{t-1} alters with every iteration, the learning rate also undergoes changes accordingly.

1.3.3 ReLU

ReLU serves as the non-linear activation function implemented in this framework. Before delving into the specifics of ReLU, let's grasp the necessity of non-linear activation functions. Their requirement arises because employing solely a linear activation function results in the output being merely a linear combination of the input, regardless of the network's layer count.

Moreover, lacking non-linear activation functions prevents the creation of neural networks capable of tackling complex problems. Without them, the decision boundary remains linear. Hopefully, the indispensability of non-linear activation functions is now recognized. Let's explore the fundamentals of ReLU.

ReLU represents an advancement over sigmoid and tanh activation functions. The primary drawback of both sigmoid and tanh lies in the issue of vanishing gradients. As they output values between 0 and 1, calculating gradients via backpropagation involves multiplying various values within this range. After a few iterations, these values may become minuscule, leading to stagnant weight updates. Additionally, both sigmoid and tanh incur high computational costs involving computationally intensive functions like exponent and tan.

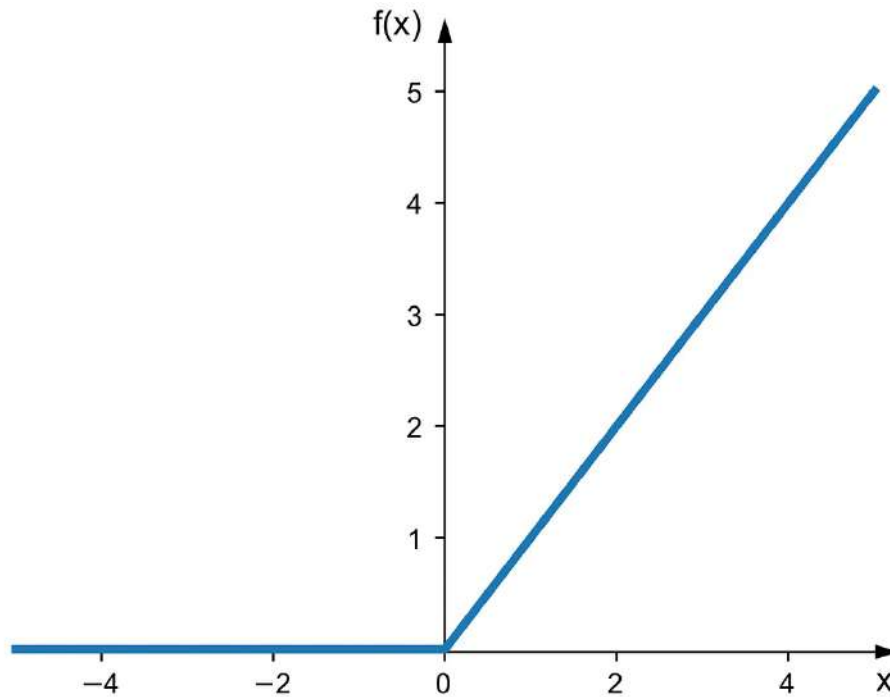


Fig 1.8: ReLU Activation Function

The ReLU formula is $f(x) = \max(0, x)$

In this context, it's evident that the value doesn't reside between 0 and 1, and there's no requirement to calculate any expensive function. Hence, ReLU resolves both problems.

1.4 CNN ARCHITECTURES

FaceNet employs two types of CNNs: the Zeiler & Fergus architecture and the GoogLeNet-style Inception model.

1.4.1 ZEILER & FERGUS ARCHITECTURE

The Zeiler & Fergus architecture serves the purpose of visualizing the training process of a CNN. It aids in comprehending the internal mechanisms of a CNN by introducing a novel visualization technique. This technique provides insights into the function of intermediate layers and the operation of classifiers.

Below is the Zeiler & Fergus architecture utilized in the FaceNet research paper.

layer	size-in	size-out	kernel	param	FLPS
conv1	220×220×3	110×110×64	7×7×3, 2	9K	115M
pool1	110×110×64	55×55×64	3×3×64, 2	0	
rnorm1	55×55×64	55×55×64		0	
conv2a	55×55×64	55×55×64	1×1×64, 1	4K	13M
conv2	55×55×64	55×55×192	3×3×64, 1	111K	335M
rnorm2	55×55×192	55×55×192		0	
pool2	55×55×192	28×28×192	3×3×192, 2	0	
conv3a	28×28×192	28×28×192	1×1×192, 1	37K	29M
conv3	28×28×192	28×28×384	3×3×192, 1	664K	521M
pool3	28×28×384	14×14×384	3×3×384, 2	0	
conv4a	14×14×384	14×14×384	1×1×384, 1	148K	29M
conv4	14×14×384	14×14×256	3×3×384, 1	885K	173M
conv5a	14×14×256	14×14×256	1×1×256, 1	66K	13M
conv5	14×14×256	14×14×256	3×3×256, 1	590K	116M
conv6a	14×14×256	14×14×256	1×1×256, 1	66K	13M
conv6	14×14×256	14×14×256	3×3×256, 1	590K	116M
pool4	14×14×256	7×7×256	3×3×256, 2	0	
concat	7×7×256	7×7×256		0	
fc1	7×7×256	1×32×128	maxout p=2	103M	103M
fc2	1×32×128	1×32×128	maxout p=2	34M	34M
fc7128	1×32×128	1×1×128		524K	0.5M
L2	1×1×128	1×1×128		0	
total				140M	1.6B

Fig 1.9: Zeiler & Fergus architecture

The model boasts 140 million parameters and performs 1.6 billion FLOPS (Floating Point Operations Per Second) per image.

1.4.2. INCEPTION MODEL

The central concept behind the Inception network architecture involves employing multiple filters of different sizes concurrently. Unlike traditional network architectures where a single filter size, such as 3x3 or 5x5, is typically chosen, the Inception architecture utilizes several filters at the same time and concatenates their outcomes.

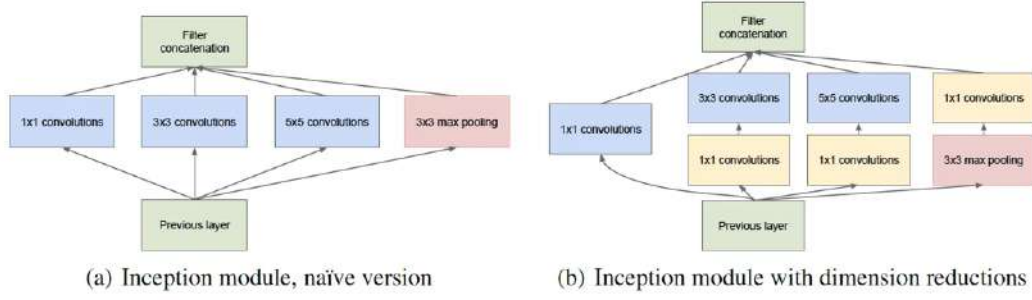


Fig 1.10: Inception Network

In Fig 1.10 (a), the approach involves utilizing multiple filters of sizes 1x1, 3x3, and 5x5 in conjunction with a max-pooling layer, followed by concatenation of results. This constitutes the primary intuition behind the Inception network architecture. However, this method poses a computational challenge due to its high complexity. To mitigate this issue, 1x1 convolutions are employed for dimensionality reduction.

In Fig 1.10(b), a 1x1 filter is incorporated with every other convolution to reduce dimensionality and render the architecture computationally feasible.

The Inception model architecture featured in FaceNet research paper comprises 6.6M to 7.5M parameters and approximately 500M to 1.6B FLOPS. Different versions of the Inception model are utilized in the FaceNet, some of which are optimized for mobile phone deployment, thus possessing relatively fewer parameters and filters.

1.5 EVALUATION

The calculation for true accepts (TA) is performed as follows:

$$TA(d) = \{(i, j) \in \mathcal{P}_{\text{same}}, \text{ with } D(x_i, x_j) \leq d\} .$$

True accepts refer to the face pairs correctly classified as identical at threshold 'd'. The definition of false accepts (FA) is given by:

$$FA(d) = \{(i, j) \in \mathcal{P}_{\text{diff}}, \text{ with } D(x_i, x_j) \leq d\}$$

False accepts denote the face pairs inaccurately classified as identical. Where:

P same – It represents pairs of identical identities.

P diff – It represents pairs of different identities.

$D(x_i, x_j)$ - Denotes the squared L2 distance between the image pair.

d – signifies distance threshold

The validation rate (VAL) and false accept rate (FAR) for a specified face distance 'd' are defined as:

$$\text{VAL}(d) = \frac{|\text{TA}(d)|}{|\mathcal{P}_{\text{same}}|}, \quad \text{FAR}(d) = \frac{|\text{FA}(d)|}{|\mathcal{P}_{\text{diff}}|}.$$

CHAPTER-2

EXISTING MODELS

Similar to recent works leveraging deep networks [7, 9], this approach is entirely data-driven, learning its depiction directly from facial pixels. Instead of relying on engineered features, this approach utilizes a substantial dataset of labeled faces to acquire necessary invariances to pose, illumination, and the other variable conditions.

This report delves into the exploration of two distinct deep network architectures that have demonstrated significant victory in the computer vision community. Both architectures are characterized as deep convolutional networks [2, 4]. The initial architecture draws inspiration from the Zeiler & Fergus [12] model, incorporating several alternating layers of convolutions, local response normalizations, max pooling layers and non-linear activations. Additionally, the integration of many $1 \times 1 \times d$ convolution layers, influenced by the work of [3]. The next architecture is built upon the Inception model by Szegedy et al., notably recognized as the conquering approach for the ImageNet 2014 [8]. These networks employ mixed layers that simultaneously run various convolutional and pooling layers, concatenating their responses. The findings indicate that these models have the capability to decrease the number of parameters by as much as 20 times and can diminish the number of Floating-Point Operations Per Second (FLOPS) needed for comparable performance.

A vast body of literature exists on the subject of face verification and recognition, but conducting an exhaustive review is beyond the scope of this paper. Therefore, this paper will provide a brief discussion of the most pertinent recent work.

Several notable works, including those by [7, 9, 13], have adopted a sophisticated system comprising multiple stages. These systems combine the output of a deep convolutional network with Principal Component Analysis (PCA) for dimensionality reduction and employ Support Vector Machines (SVMs) for classification.

Zhenyao et al. [13] implement a deep network to "warp" faces into a canonical frontal view. Subsequently, they train a Convolutional Neural Network (CNN) to classify each face based on its known identity. For verification purposes, PCA is applied to the network output, coupled with an ensemble of SVMs.

Taigman et al. [9] present a multi-stage approach focusing on aligning faces with a common 3D shape model. They train a multi-class network for face recognition, covering over 4000

identities. The authors also explore a Siamese network, optimizing the L1-distance between the two face features. Achieving their top performance on Labeled Faces in the Wild (LFW) at 97.35%, they use an ensemble of 3 networks with different color channels and alignments. Predicted distances from these networks, computed through the non-linear SVM predictions based on the χ^2 kernel, are further coupled using another non-linear SVM.

Sun et al. [6, 7] suggested an economically feasible network in terms of computational cost. They employ an ensemble of twenty-five such networks, each working on a distinct face patch. Their last performance on LFW reaches 99.47% [7], achieved by combining 50 responses, including regular and flipped ones. The authors incorporate both Principal Component Analysis (PCA) and Joint Bayesian model [1], equivalent to the linear transformation in the embedding space. Notably, their method operates without explicit 2D/3D alignment requirements. The training process involves the union of classification and verification loss, resembling the triplet loss used in [5, 11]. While the verification loss reduces the L2-distance between faces of the very same identity and imposes a margin between faces of distinct identities, it differs from triplet loss by focusing on comparing pairs of images rather than encouraging a relative distance constraint. A very similar loss was found in Wang et al. [10] for ranking pictures based on semantic and visual resemblance.



Fig 2.1: High Level Modal Structure

The network comprises a batch input layer and a deep CNN, succeeded by L2 normalization, yielding the face embedding. Subsequently, during training, the triplet loss is applied.

The Triplet Loss (refer to Fig 1.2) aims to minimize the distance between an anchor and a positive example, both belonging to the same identity, while simultaneously maximizing the distance between the anchor and a negative example from a different identity.

FaceNet is constructed using the building blocks mentioned above, and now each of these components will be delved into sequentially. The deep neural network depicted in the figure is derived from the GoogLeNet architecture. While the FaceNet paper doesn't extensively

delve into the internal intricacies of the GoogleNet architecture, treating the deep neural network as a black box, let's see the crucial concepts to understand how it is utilized and for what specific purpose within the context of FaceNet.

A. Deep Network — GoogleNet

GoogleNet emerged as the victor in the ImageNet 2014 challenge, introducing several ground breaking features and advancements over conventional Convolutional Neural Networks (CNNs). Some notable characteristics of GoogleNet include:

- **Depth:** GoogleNet is a 22-layers deep network, a substantial increase compared to the 8-layered AlexNet.
- **Efficiency:** It demonstrates computational efficiency, performing computations at an accelerated pace. The computational cost of GoogleNet is reported to be approximately two times less than that of AlexNet.
- **Accuracy:** GoogleNet exhibits significantly higher accuracy in comparison to AlexNet, showcasing advancements in image classification tasks.
- **Resource Usage:** It boasts low memory usage, making efficient use of system resources, and is designed with low power consumption in mind.

The inception of the GoogleNet architecture was primarily inspired by these features, leading to the development of something known as the 'inception module' or 'network-in-network.' This architectural innovation played a pivotal role in shaping the efficiency, speed, and accuracy of GoogleNet.

B. Inception-v2 and Inception-v3

This release incorporates Factorization, aimed at reducing parameters and mitigating the overfitting issue. The introduction of Batch Normalization addresses the need for stabilizing and accelerating the training process. Additionally, label smoothing is implemented to prevent a specific logit from becoming disproportionately large compared to others, thereby applying regularization at the classifier layer.

C. Inception-v4 and Inception-ResNet-v1

This release streamlines the stem of the network, which serves as the preamble connecting to the initial inception module. The inception blocks maintain their structure, now denoted as A, B, C. In the ResNet Version, a significant enhancement is the introduction of residual

connections, substituting pooling in the inception module. David Sandberg's FaceNet implementation adopts the 'Inception-ResNet-v1' version. During the training of FaceNet, the deep network extracts and assimilates diverse facial features. These facial features are subsequently transformed directly into 128D embeddings, where similar faces should exhibit proximity, while distinct faces should have considerable separation in the embedding space (which is essentially the feature space). This concept is translated into implementation through a loss function known as Triplet Loss.

D. Cost Function

FaceNet's distinctive attribute lies in its loss function. While the primary loss function for face recognition is the Triplet Loss, David's FaceNet implementation incorporates 2 loss functions: 'Triplet loss' and the 'Softmax activation with cross-entropy loss'. The structure of the Triplet cost function is as follows:

$$\text{Cost Function} = \sum_i^N \text{Triplet Loss Function} + \text{L2 Regularization}$$

E. Triplet Loss

Consider a function $f(x)$ that generates embeddings in a n -dimensional space for an image x . Here are examples of images:

- **Anchor:** An image of Dhoni that is intended for comparison.
- **Positive:** Another image of Dhoni, serving as a positive example.
- **Negative:** An image of Kohli, representing a negative example.

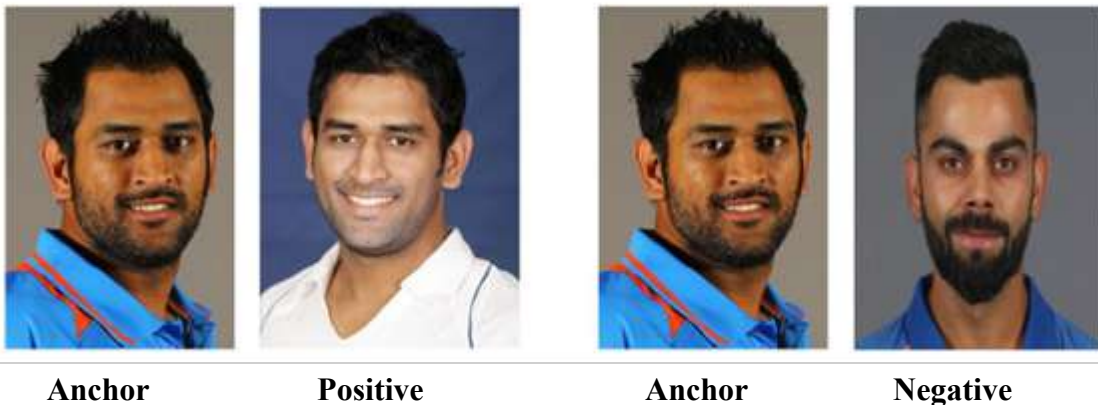


Fig 2.2: Triplet loss example Anchor Positive, Negative

Theoretically, the anchor image should be closer to the positive image and farther from the negative image in Euclidean space. This relationship can be calculated as:

$$\text{distance (Anchor, Positive)} \quad \text{distance (Anchor, Negative)}$$

$$\|f(\text{Anchor}) - f(\text{Positive})\|^2 + \alpha \leq \|f(\text{Anchor}) - f(\text{Negative})\|^2$$

similarly,

$$\|f(\text{Anchor}) - f(\text{Positive})\|^2 + \alpha - \|f(\text{Anchor}) - f(\text{Negative})\|^2 \leq 0$$

Here,

$\|f(\text{Anchor}) - f(\text{Positive})\|^2$ is the distance between anchor image and positive image,

$\|f(\text{Anchor}) - f(\text{Negative})\|^2$ is the distance between anchor image and negative image.

To ensure a greater separation between the positive set and negative set, a margin α is introduced to the positive set, pushing it further away. The loss function can be zero and the equation will look like the following (as values below zero are not needed):

$$L(\text{Anchor}, \text{Positive}, \text{Negative}) = \max(\|f(\text{Anchor}) - f(\text{Positive})\|^2 + \alpha - \|f(\text{Anchor}) - f(\text{Negative})\|^2, 0)$$

F. Triplet Selection

A clear inquiry arises regarding the selection process for $f(\text{Anchor}, \text{Positive})$ and $f(\text{Anchor}, \text{Negative})$ pairs. If chosen randomly, the aforementioned equation might be satisfied effortlessly, but the network would not effectively learn. Furthermore, random selection could potentially lead to encountering local minima, leading to incorrect weight convergence during gradient descent. The paper proposes that the utilization of very challenging examples might lead to convergence problems at an early stage and could potentially yield a flawed model. Instead, semi-hard examples are the recommended choice. This can be accomplished by employing a reasonable mini-batch size; in the paper, the author utilized 40 faces in a mini-batch. Therefore, it is crucial to pair 'semi-hard' examples and present them to the network, ensuring:

$$\text{distance (Anchor, Positive)} \approx \text{distance (Anchor, Negative)}$$

The α margin consistently ensures their separation, even when they are in proximity to one another.

G. SVM Training — Inference







These embeddings are then utilized to determine the Euclidean distance for photo matching or validation. SVM, an optimal “machine learning algorithm for classification, is trained on these” produced “embeddings and can” subsequently applied for the inference on tested data.

CHAPTER-3 SYSTEM DESIGN

During testing, the FaceNet model in this project leveraged various public datasets, including YALE, JAFFE, AT&T, Georgia Tech, and Essex, to assess the accuracy of the face recognition approach.

The dataset of facial images is employed due to its extensive utilization in numerous face recognition studies, rendering it a standard dataset in the field. This choice facilitates seamless comparison with other methods proposed in various prior studies. Table I presents a sample of facial images alongside the details of each dataset.

Table 3.1: Face Image Database Used

Dataset	The number of persons	Total number of images in the databases	Sample images
Yale Database	15	164	
JAFFE	10	213	
AT&T	40	400	
Georgia Tech	50	750	
Essex_faces94	153	3078	
Essex_faces95	72	1440	

Essex_faces96	152	3016	
Essex_grimace	18	360	

Face recognition using FaceNet involves the following steps:

3.1 PREPROCESSING

During the preprocessing stage, detection of facial images is conducted on each image using MTCNN. Upon successful detection of face, the original image from the dataset, with dimensions x pixels * y pixels, is cropped based on the detected facial area, resulting in a size of 182 pixels * 182 pixels. Fig. 3.1 illustrates this pre-processing procedure employing the MTCNN method.

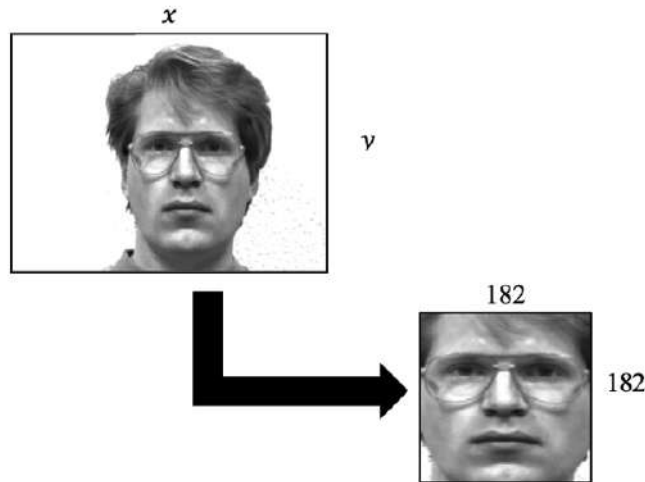


Fig. 3.1 An instance of cropping the image during the preprocessing stage using the MTCNN method

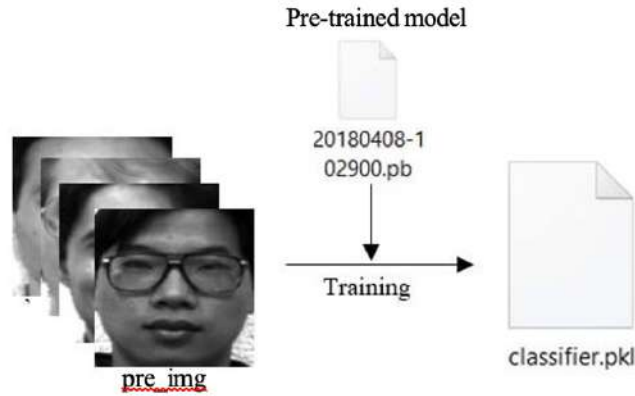
3.2 TRAINING

Upon completion of the pre-processing stage on all datasets, the subsequent step involves training the model. In this project, pre-existing model pre-training data is utilized to streamline the process. It's worth noting that the dataset within the pre-trained model significantly impacts the accuracy of face recognition. Therefore, in this project, 2 types of pre-trained models are obtained from CASIA-WebFace and VGGFace2 for comparison, to ascertain if they indeed affect the quality of the results. Table II provides more detailed descriptions of the pre-trained models utilized.

Table 3.2: Pre-Trained Models

Model Name	LFW Accuracy	Training dataset	Architecture
20180408-102900	0.9905	CASIA-WebFace	Inception ResNet v1
20180402-114759	0.9965	VGGFace2	Inception ResNet v1

Following the pre-processing of images, the dataset is trained, with the pre-processed images organized into a folder named "pre_img," labeled according to their respective names. TensorFlow machine learning assistance is employed for training. The outcome of this process yields files in the .pkl format. For a visual representation, please refer to Fig. 3.2.

**Fig. 3.2. Training process**

The proposed attendance system comprises four main components: web camera face capture, student image database, face recognition, and attendance record update, as depicted in Fig. 3.3. The system is implemented on a laptop computer equipped with an integrated web camera.

Let's consider the task of checking attendance for a class. To facilitate the attendance check, all students in the class are required to capture pictures (approximately 10 images in this project) using the integrated web camera. These images are then utilized to generate the student face database, acting as a reference for real-time face recognition.

To determine the attendance of a student in the class, the computer captures facial images of the student via the live video stream. It then employs deep learning neural networks to predict whether the student matches anyone in the database and, if a match is identified,

further identifies the student's name. The outcome of this face recognition process is then used to update the attendance record in the database.

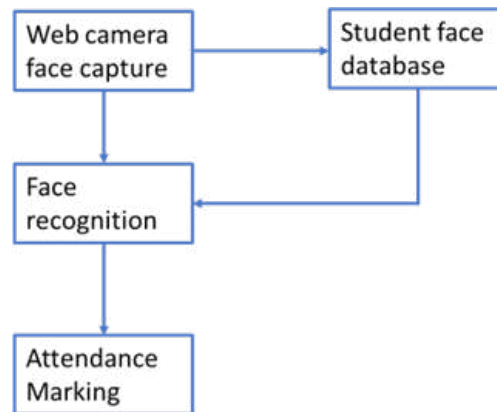


Fig 3.3: Architecture of the proposed attendance system

At the heart of the algorithm resides the deep neural network referred to as FaceNet, which explicitly generates a mapping from facial images to a condensed Euclidean space, where the distances act as a direct indication of facial similarity. Essentially, FaceNet gives a 128-dimensional representation—a vector with 128 components—from an image, ensuring that:

1. The representations of 2 images depicting the same individual are near each other.
2. The representations of 2 images featuring different individuals are distinctly distant from each other.

Euclidian's distance between the encodings acts as a metric to decide whether 2 face images belong to the same individual. Considering the resource-intensive process of training FaceNet, which demands extensive amounts of data and computational power, the choice is made to utilize a pre-trained model named inception_blocks_v2. This results in a model with 3,743,280 parameters.

Within the database, each person is represented by 128 - dimensional encoding. To the minimize variation, 10 - face images are captured for each of student, the corresponding 10 - encodings are derived. The average of the encodings is then stored as the representative encoding for that individual in the database. The database stores, for each student, their name alongside the corresponding encoding.

To improve facial recognition accuracy, the program takes multiple real-time face images for a person, independently recognizes each picture, and consolidates results based on these individual recognitions. For example, this implementation acquires 10 face images, subsequently conducting independent face recognitions.

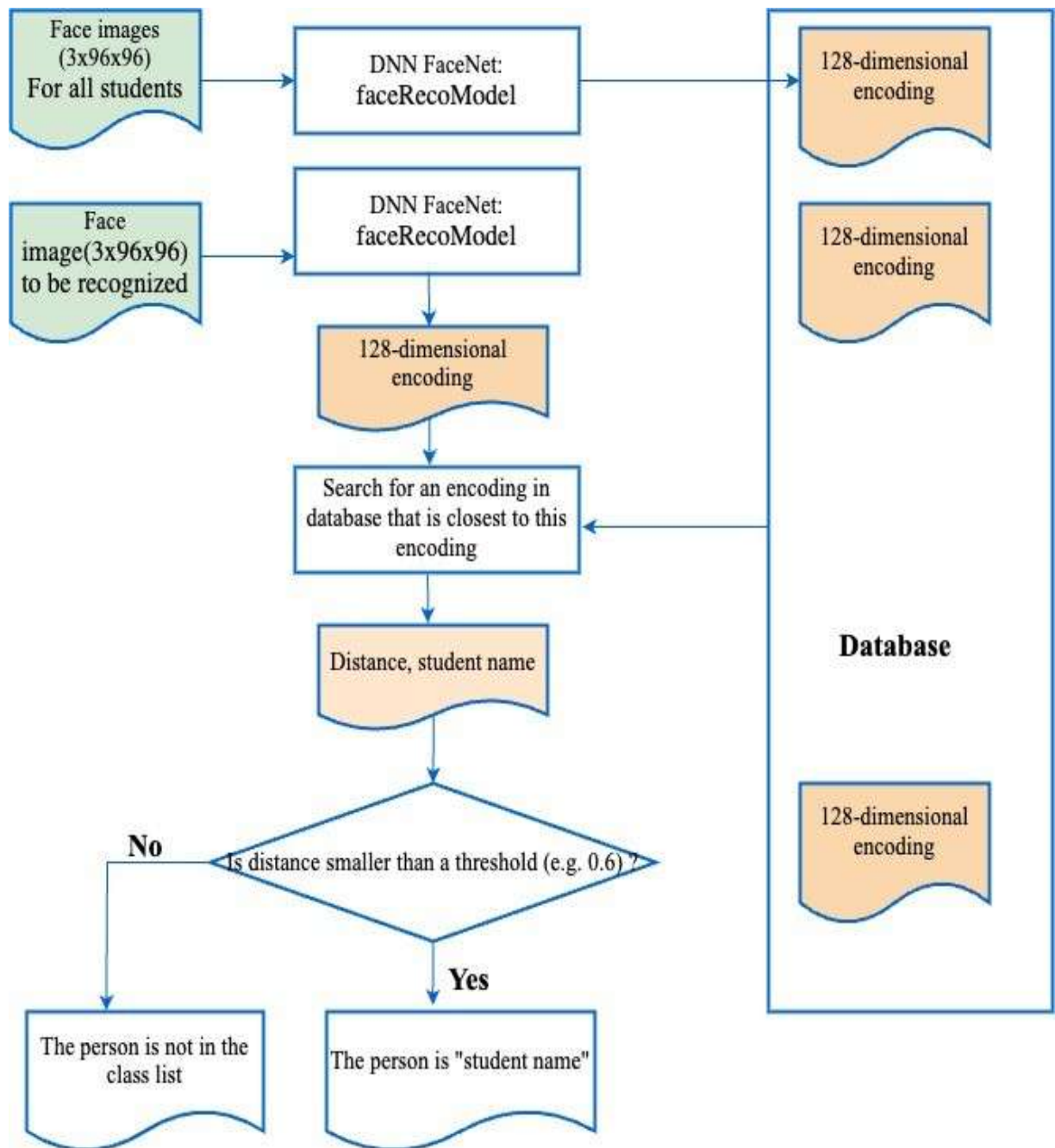


Fig 3.4: Flow chart for facial recognition

Step 1: Face registration

The system allows users to register their faces by capturing a sequence of images from a webcam. Users can provide a name for the registered face, which is used to organize the captured images.

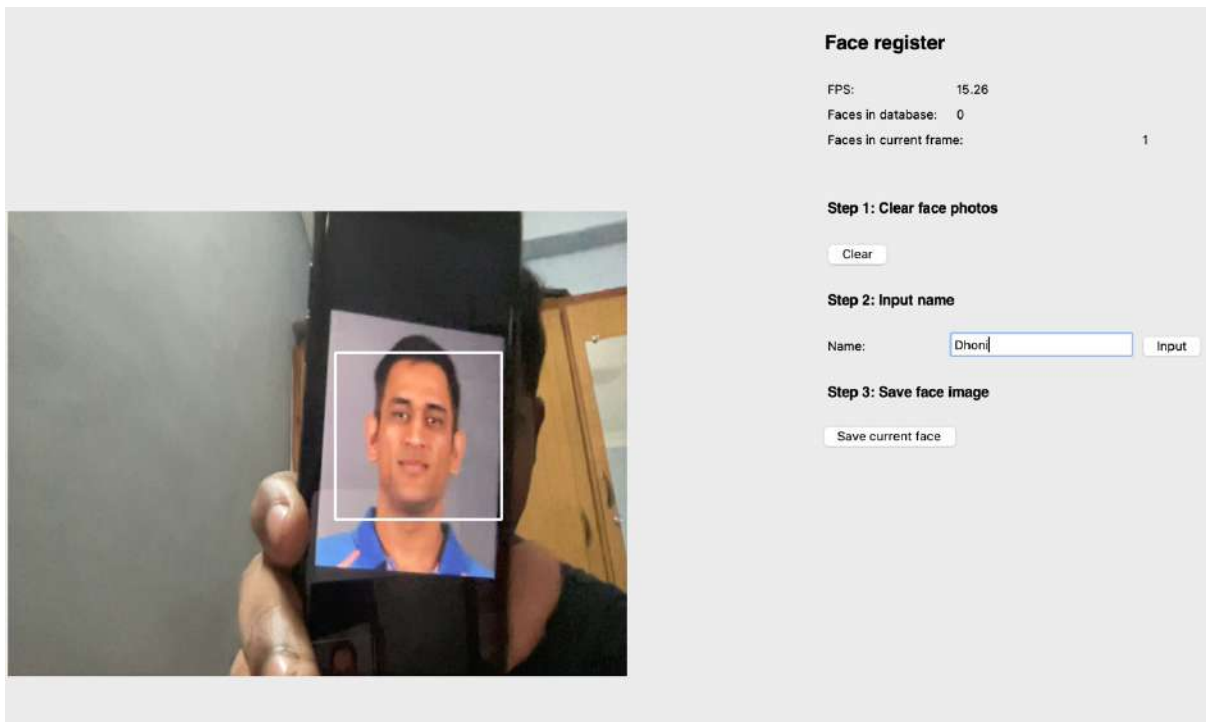


Fig 3.5: Face registration

- **Initialization:**

The class constructor initializes various member variables for storing data like frame counters, GUI elements, file paths, and face detection results.

It also sets up a video capture object to access the webcam feed.

- **GUI Management:**

The class provides functions for interacting with the GUI elements. This includes functions to clear previously registered data, retrieve user input for the registered face name, and create and manage the layout of the GUI for displaying instructions, frame rate, face counts, and system logs.

- **Data Management:**

The code includes functionalities to handle data organization for captured faces. It creates folders to store captured images for each registered face and a separate file to store information about registered faces.

- **Face Detection and Processing:**

- The system utilizes Dlib for face detection within each video frame captured from the webcam.
- It calculates the region of interest (ROI) for detected faces and verifies if the face is within the frame boundaries.
- If a face is detected within the frame, the code draws a rectangle around the face for visual feedback and saves the extracted face ROI as an image file within the user-specified folder.

- **Main Loop:**

The core functionality resides in the process method. This method continuously captures video frames, detects faces, updates the frame rate display, and refreshes the GUI with the processed frame.

- **Running the Application:**

The main function serves as the entry point for the application. It sets up logging functionalities and creates an instance of the Face Register class. Finally, it calls the run method of the class to initiate the face registration application.

Step 2: Feature extraction

Dlib is equipped with a pre-trained facial landmark detector capable of identifying 68 distinct facial landmarks. The outputs from this facial landmark detection process serve as input to a Convolutional Neural Network (CNN), which subsequently generates 128 values, encapsulating a representative encoding of the face.

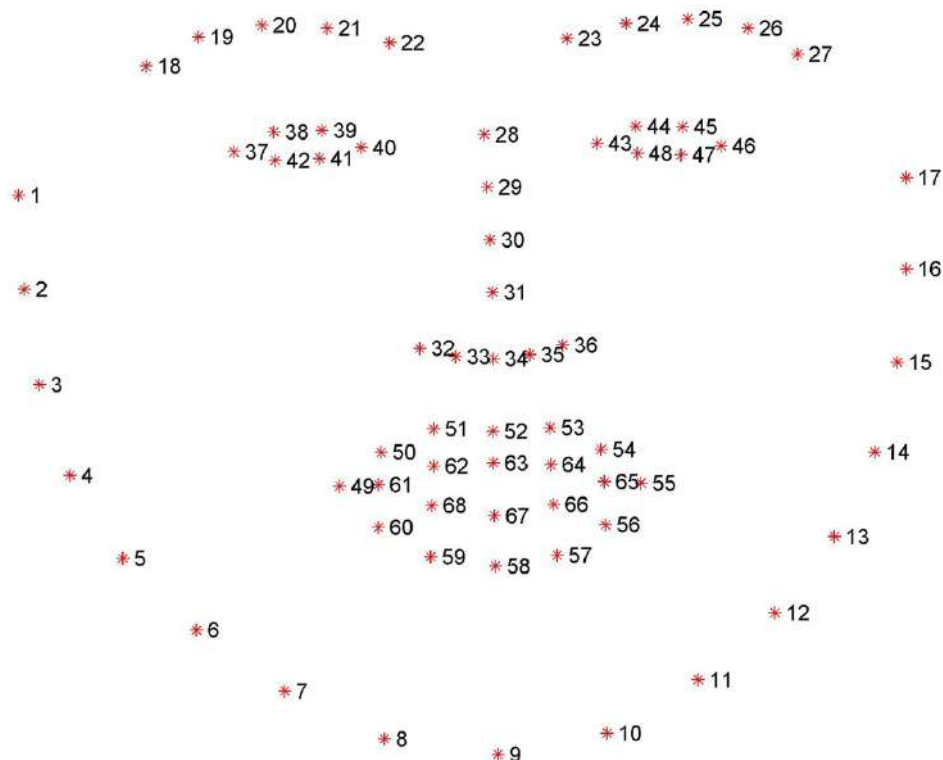


Fig 3.6: 68 facial landmarks

The following functions generate a 128-dimensional representation for each distinct registered face:

□ **return_128d_features(path_img) :**

- This function takes an image path (path_img) as input.
- It reads the image, detects faces using the Dlib detector.
- If a face is detected, it extracts facial landmarks and computes a 128-dimensional feature vector using the Dlib face recognition model.
- The function logs informative messages and returns the 128D feature vector if a face is detected, or 0 otherwise.

□ **return_features_mean_personX(path_face_personX) :**

- This function takes a directory path (path_face_personX) as input, representing a folder containing images of a single person.
- It iterates through all images in the directory.
- For each image, it calls return_128d_features to extract the 128D feature vector.
- The function calculates the average (mean) of all extracted feature vectors for the person and returns it as a NumPy array.
- If no faces are detected in any image, it returns a zero-filled array of size 128.

Main Function:

□ **main():**

- This function sets up logging to display informational messages.
- It retrieves a list of directory names within path_images_from_camera, assuming these represent folders containing images of different people.
- It opens the output CSV file named "features_all.csv" in write mode.
- The function iterates through each directory (person):
- It calculates the average feature vector for the person using return_features_mean_personX.
- It extracts the person's name from the directory name format ("person_name" or "person_number_name").
- It creates a new list by prepending the person's name to the 128D feature vector (resulting in a 129D list).
- It writes the person's name and features as a row to the CSV file.
- Finally, the function logs a message indicating successful completion and the location of the output CSV file.

Dhoni	-0.12097756564617200	0.10286712894837100	0.09513374914725620	-0.01330598040173450	-0.10541309416294100	-0.05315342545
-------	----------------------	---------------------	---------------------	----------------------	----------------------	----------------

Fig 3.7: Feature Extraction

Step 3: Taking Attendance

The "Face_Recognizer" class utilizes Dlib for face detection, facial landmark prediction, and face recognition. SQLite is employed for managing the attendance records, and a table is created in the database for the current date to store attendance information. The attendance is marked with the person's name, the timestamp, and the date, ensuring a comprehensive and organized record.

The attendance method within the class plays a crucial role in updating attendance records. It checks whether an entry already exists for the person on the current date. If the person is already marked present, it notifies that the attendance has already been recorded; otherwise, it inserts a new record with the person's name, current time, and date.

The process method in the class handles the real-time processing of video frames, performing facial recognition and attendance tracking. It retrieves known faces from the "features_all.csv" file and then continuously captures frames from the video stream. The program detects faces within each frame, computes facial features using a pre-trained Dlib model, and compares them with the features of known faces in the database.

The program distinguishes between scenarios where the face count remains the same or changes in consecutive frames. In cases where the face count changes, it dynamically updates the list of recognized faces and performs face recognition, linking faces across frames using centroid tracking. It also updates attendance records based on the recognized person's name.

The graphical user interface displays real-time information, including the frame number, frames per second (FPS), the count of detected faces, and specific instructions. The attendance-taking process occurs seamlessly within the frame processing loop, providing continuous monitoring and feedback through the graphical interface.

Finally, the program can be terminated by pressing the 'q' key. The run method initializes the video capture and invokes the process method, effectively initiating the attendance-taking process. The entire script combines the power of facial recognition with practical attendance tracking, making it suitable for various applications where automated attendance is required.

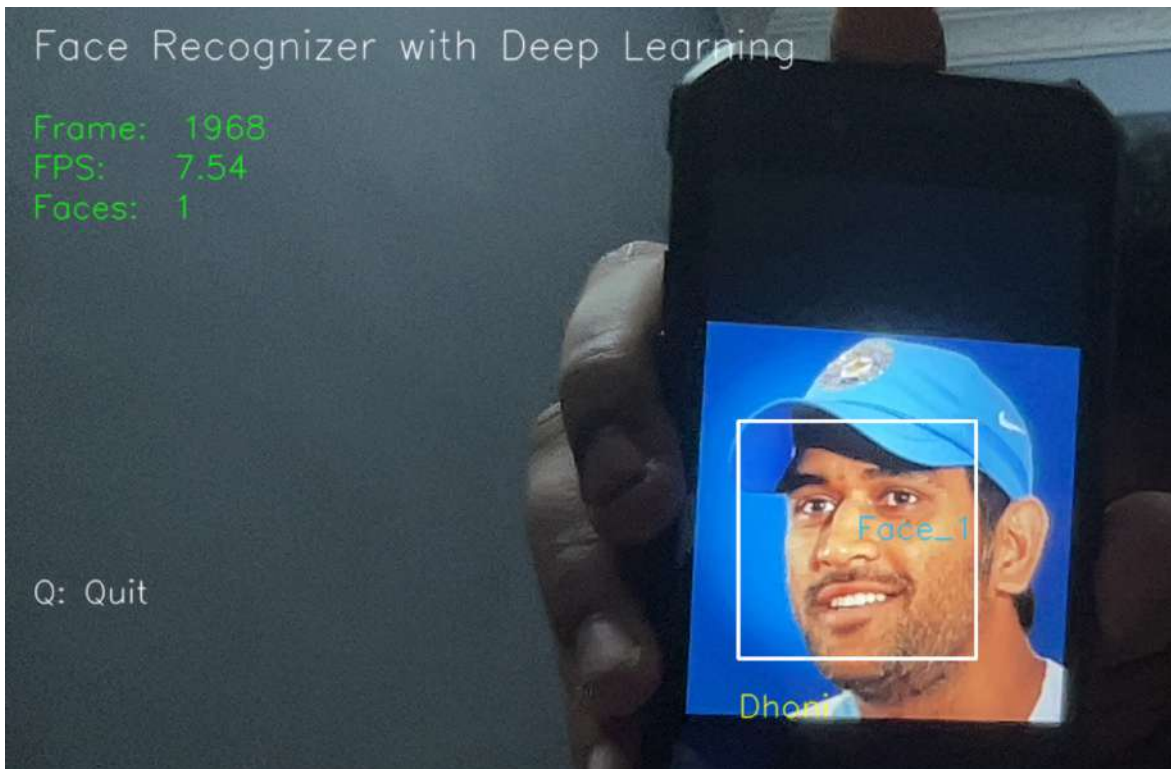


Fig 3.8: Taking attendance

Step 4: Flask web application to track the attendance

In this step, a Flask web application is implemented to provide a user interface for accessing and visualizing the attendance data recorded in the SQLite database. The web application consists of two routes defined by the `@app.route` decorator. The first route, `('/')`, corresponds to the home page. Upon accessing this route, the `'index'` function is called, rendering the `'index.html'` template. This template initially displays a date selection form and informs the user if there is no attendance data for the chosen date.

The second route, `('/attendance')`, is activated when the date selection form is submitted (using the POST method). The `'attendance'` function is then triggered, retrieving the selected date from the form submission. It converts the selected date to the required format and establishes a connection to the SQLite database. Subsequently, it queries the database for attendance records on the specified date, fetching data such as names and timestamps.

The retrieved attendance data is passed to the `'index.html'` template for rendering. If there is no attendance data for the selected date, the template is rendered with the `'no_data'` variable set to True, indicating that there is no data to display. Otherwise, the attendance data is displayed in a tabular format on the web page.

The Flask application is initiated by the `if __name__ == '__main__':` block, and the `app.run(debug=True)` command starts the development server, allowing the web application to be accessed through a web browser. The web server runs in debug mode (`debug=True`), which facilitates real-time code changes without the need to restart the server.

In summary, this Flask application provides a user-friendly interface to view attendance records based on selected dates, enhancing the usability and accessibility of the attendance system. Users can interact with the web application through their browser, making it convenient for checking attendance details.

Attendance Tracker Sheet

Select Date:

27/12/2023

Show attendance

Attendance Data Table

Name	Time
Charan	12:39:13
Akhil	13:01:27
Dhoni	20:48:08

Fig 3.9: Attendance Tracker Sheet

CHAPTER-4

RESULTS AND ANALYSIS

At this stage, accuracy testing of facial recognition is conducted across various face databases. Eight widely-used public face databases are selected for this purpose, enabling comparisons with other methodologies. The results of the accuracy measurements on each face dataset are presented in Table 3. The measurement approach involves employing Euclidean space, where each face image undergoes comparison with the index label assigned to it. The `numpy.mean` function is applied to compare the best class indices, resulting from Euclidean space calculations, with the original labels of each face.

The findings from the FaceNet testing, as detailed in Table 3, reveal its high accuracy in facial recognition tasks. However, it is observed that the accuracy of FaceNet is not optimal in the `Essex_faces96` dataset. This discrepancy might be attributed to the presence of multiple face labels, each containing distinct faces. Consequently, significant variations in facial conditions can have a notable impact on the accuracy achieved using FaceNet.

Moreover, the accuracy testing utilizing Euclidean space measurement on eight widely used public face databases provides valuable insights into the performance of FaceNet in diverse scenarios. The chosen datasets facilitate a comprehensive comparison with other methods and offer a benchmark for evaluating FaceNet's effectiveness. The evaluation, as depicted in Table 3, employs the `numpy.mean` function to calculate accuracy by comparing the best class indices obtained through Euclidean space measurements with original labels of each face. The results underscore FaceNet's overall accuracy, positioning it as a highly reliable method for face recognition. However, the observed lower accuracy in the `Essex_faces96` dataset suggests potential challenges, possibly stemming from extreme variations in facial conditions within this specific dataset. Future optimizations may focus on addressing such variations to further enhance FaceNet's robustness across a broader range of scenarios and datasets.

Table 4.1: Face Recognition Results using FaceNet in Each Facial image Database

Dataset	Total images in the database	Total images successfully aligned	FaceNet rate (in %)	
			Casia-WebFace	VGGFace2
JAFFE	213	213	100	100
Yale	164	164	98.9	100
Georgia Tech	750	750	100	100
AT&T	400	400	97.5	100
Essex faces95	1440	1439	99.65	100
Essex faces94	3078	3059	99.37	99.37
Essex grimace	360	360	100	100
Essex faces96	3016	3016	76.86	77.67

The implementation of Face Recognition-Based Attendance System has been successfully completed. Face Registration module provides an intuitive user interface for capturing and registering faces. Users can input names and clear existing data, ensuring a clean and organized face database. The system leverages face detection to create dedicated face folders, allowing for the easy storage of face images. This step establishes a user-friendly foundation for further processing.

In the Face Feature Extraction step, the system utilizes Dlib's facial landmark predictor and a ResNet-based model to extract 128D face descriptors. These descriptors are crucial for accurately recognizing faces in subsequent steps. The system efficiently saves these features in a CSV file, providing a comprehensive database for face recognition.

The Face Recognition and Attendance module continuously processes video frames, detecting faces and comparing them with the stored face features. Real-time information, including recognized faces, frame count, and frames per second, is displayed. The

attendance records are updated in an SQLite database, ensuring accurate and efficient attendance tracking. The integration of a centroid tracker enhances face tracking capabilities. To enhance user accessibility, a web-based interface was implemented using Flask. Users can select a date, and the system retrieves attendance records from the SQLite database. The web interface provides a user-friendly means of checking attendance, displaying the selected date along with attendance data (if available). This step enhances the overall usability of the system.

The system seamlessly registers faces, extracts essential features, records attendance in real-time, and provides a convenient web interface for attendance checking. This solution combines the power of computer vision, machine learning, and web development to create an efficient and user-friendly automated attendance management system. The system is ready for deployment, offering a reliable solution for organizations seeking an advanced attendance tracking system.

4.1 COMPARISON WITH PREVIOUS METHODS

Following acquisition of the accuracy results on FaceNet test and subsequent phase in this investigation involves a comparative analysis of the FaceNet method against previously proposed face recognition approaches. The alternative methods under comparison encompass:

1. The LTDF_MLDN framework, based on local texture description, is enhanced with the Nearest Neighborhood Classifier (NNC), employing various distance metrics such as Euclidean, Manhattan, Minkowski, G-statistics, and chi-square distances, with the goal of identifying the most effective distance metric.
2. Eigenfaces methodology is employed, leveraging both Principal Component Analysis (PCA) and Kernel Principal Component Analysis (KPCA) techniques.
3. The sgFKNN algorithm utilizes a fuzzy approach to Grammar Fuzzy K-Nearest Strength for classification tasks.
4. A combined approach of PCA and Support Vector Machine (SVM) is utilized for classification tasks.

Examining comparative outcomes from all tests on each facial image dataset is depicted in the table below. The method presented for comparison demonstrates that the FaceNet approach achieves exceptionally high accuracy across various tested datasets, nearly reaching 100% accuracy for each assessment. This outstanding performance is credited to the meticulous comparison of each face against the TensorFlow pre-trained model,

reinforced by machine assistance. Notably, accuracy results on the JAFFE dataset closely align with those of other methods, underscoring that this dataset, primarily characterized by varied facial expressions, does not display significant differences in facial features. Despite diverse facial conditions in certain datasets, such as the use of accessories, the FaceNet method exhibits robust accuracy, unaffected by these variations. It's important to note that accuracy results may be affected if a face presents a highly distinct image observed in the Essex faces96 dataset. Additionally, the influence of pretrained models is evident in the LFW results, where VGGFace2 pre-trained accuracy measurements display a marginal difference of 0.006. While this disparity in LFW accuracy is subtle, it carries considerable significance in the context of facial image recognition testing.

Table 4.2: Comparative Analysis

Data set	LTDF_MLDN +NNC	Eigen Faces		sgFK NN	PCA+SVM	FaceNet	
		<i>PCA Algorithm</i>	<i>KPCA Algorithm</i>			<i>Casia-WebFace</i>	<i>VGGFace2</i>
JAFFE	100 (Manhattan & chi-square)	71.2	80	100	-	100	100
Yale Database	-	88.26	97.25	-	93	98.9	100
Georgia Tech	83.73 (chi-square)	-	-	79.57	-	100	100
AT&T	97.5 (chi-square)	-	-	99.25	98.75	97.5	100
Essex Faces94	-	70	70	-	-	99.37	99.37
Essex Faces95	99.09 (g-statistics)			-	-	99.65	100
Essex faces96	-			-	-	76.86	77.67
Essex grimace	100 (Manhattan & chi-square)			-	-	100	100

4.2 EFFECT OF CNN MODEL

Let's now explore a more in-depth analysis of the effectiveness of the four chosen models. On one side, there's the conventional architecture inspired by Zeiler & Fergus, incorporating 1x1 convolutions. On the other side, there are Inception-based models, recognized for their capacity to notably reduce model size. In terms of overall performance, the leading models from both architectures demonstrate comparable results. However, certain Inception-based

models, like NN3, achieve commendable performance while substantially reducing both FLOPS and model size.

Table 4.3: Image Quality

jpeg q	val-rate	#pixels	val-rate
10	67.3%	1,600	37.8%
20	81.4%	6,400	79.5%
30	83.9%	14,400	84.5%
50	85.5%	25,600	85.7%
70	86.1%	65,536	86.4%
90	86.5%		

The left table illustrates the influence of varying JPEG quality on the validation rate at $10E-3$ precision. Conversely, the right table shows how the image size in pixels affects the validation rate at $10E-3$ precision. This experiment was carried out using NN1 on the initial split of test hold-out dataset.

Table 4.4: Embedding Dimensionality

#dims	VAL
64	86.8% \pm 1.7
128	87.9% \pm 1.9
256	87.7% \pm 1.9
512	85.6% \pm 2.0

This table presents a comparison of the impact of embedding dimensionality on NN1 model. It includes the validation rate at $10E-3$, along with the standard error of the mean computed across five splits.

While the largest model shows a considerable improvement in accuracy compared to the smaller NNS2, the latter can still operate efficiently in 30ms per image on a mobile phone while maintaining sufficient accuracy for face clustering purposes. However, there is a noticeable decline in the ROC for $FAR < 10^{-4}$, indicating the presence of noisy labels in the ground truth of the test data. At extremely low false accept rates, even a single mislabelled image can significantly affect the curve.

4.3 SENSITIVITY TO IMAGE QUALITY

Table V illustrates the model's robustness across a wide range of image sizes. Remarkably, the network demonstrates resilience to JPEG compression, maintaining strong performance even at a JPEG quality of 20. The decrease in performance is minimal for face thumbnails as small as 120x120 pixels, and it remains satisfactory even at 80x80 pixels. This observation is significant, especially considering that the network was trained on 220x220 input images. Training with lower-resolution faces could potentially extend this range even further.

4.4 EMBEDDING DIMENSIONALITY

Exploration was carried out on different embedding dimensionalities, with 128 chosen for all experiments except for the comparison specified in Table VI. While one might expect larger embeddings to perform at least as well as smaller ones, it is conceivable that they require additional training to achieve comparable accuracy. Nonetheless, the differences in performance, as outlined in Table V, are statistically insignificant.

This table compares the performance after 700 hours of training for a smaller model using 96x96 pixel inputs. The model architecture resembles NN2 but does not include the 5x5 convolutions in the Inception modules. The differences in performance reported in Table V are not statistically significant.

Table 4.5: Training Data Size

#training images	VAL
2,600,000	76.3%
26,000,000	85.1%
52,000,000	85.1%
260,000,000	86.2%

It's crucial to emphasize that during training, a 128-dimensional floating-point vector is employed, but it can be quantized to 128 bytes without compromising accuracy. As a result, each face is succinctly represented by a 128-dimensional byte vector, which is ideal for large-scale clustering and recognition endeavors. Smaller embeddings are viable with a minor reduction in accuracy and can be implemented on mobile devices.

4.5 AMOUNT OF TRAINING DATA

Table VII illustrates the influence of extensive training data volumes. Due to time limitations, this assessment was performed on a smaller model; nevertheless, the impact might be even more pronounced on larger models. It's evident that employing tens of millions of samples results in a substantial enhancement in accuracy on the personal photo test set. Compared to utilizing only millions of images, the relative decrease in error is 60%. While employing another order of magnitude more images (hundreds of millions) still offers a slight improvement, the magnitude of enhancement begins to diminish.

4.6 PERFORMANCE ON LFW

The model undergoes evaluation on LFW using the established protocol for unrestricted, labelled external data. 9 training splits are utilized to establish the L2-distance threshold, following which classification (same or different) is carried out on the tenth test split. The chosen optimal threshold is 1.242 for all test splits, except for the eighth split, where it is 1.256.

Evaluation of the model occurs in two modes:

1. Utilizing a fixed center crop of the LFW provided thumbnail.
2. Applying a proprietary face detector to the provided LFW thumbnails. In cases where it fails to align the face (which occurs for two images), the LFW alignment is utilized.

A 3x6 grid of 18 portraits of men. The portraits include: Row 1: a man in a suit, a man in a white shirt, a man in a suit with a red background, an older man in a white shirt, a man in a suit with a blue tie, and a man in a suit with glasses. Row 2: a man in a military uniform, a man in a suit, a man in a suit with a red and white shield logo, a man in a blue tank top, a man in a green shirt, and a man in a tuxedo. Row 3: an older man in a suit, a man in a suit, a man in a patterned shirt, a man in a suit with a yellow tie, a man in a suit, and a man in a suit with a purple tie.

Fig 4.1. LFW errors.

This reveals all pairs of images that were inaccurately classified on LFW. Out of the 13 false rejections displayed here, only eight are genuine errors, while the remaining five are due to mislabeling in LFW.

Figure 4.1 offers an overview of all failure cases, displaying false accepts at the top and false rejects at the bottom. Utilizing the fixed center crop described in (1) achieves a classification accuracy of $98.87\% \pm 0.15$. Furthermore, incorporating additional face alignment (2) sets a new record with an accuracy of $99.63\% \pm 0.09$ standard error of the mean. This represents a substantial reduction in error compared to DeepFace [9] by more than a factor of 7 and a 30% improvement over the previously reported state-of-the-art for DeepId2+ [7]. These findings are based on the performance of model NN1, although even the much smaller NN3 achieves performance that is not statistically significantly different.

4.7 PERFORMANCE ON YOUTUBE FACES DB

The mean similarity of all pairs among the first one hundred frames identified by the face detector in each video is employed. This method produces a classification accuracy of $95.12\% \pm 0.39$. Extending the evaluation to the initial one thousand frames yields an accuracy of 95.18%. In comparison to [9], which evaluates one hundred frames per video and attains a 91.4% accuracy, this approach reduces the error rate by almost half. DeepId2+ [7] achieved 93.2%, and this method decreases this error by 30%, mirroring the improvement observed on LFW.

4.8 FACE CLUSTERING

The compact embedding employed is ideal for clustering a user's personal photos into groups based on individuals sharing the same identity. The specific constraints associated with clustering faces, as opposed to solely focusing on verification tasks, yield notably impressive results. Figure 4.2 showcases a cluster within a user's personal photo collection, created using agglomerative clustering. It serves as a vivid demonstration of the remarkable robustness to occlusion, pose variations, lighting changes, and age differences.



Fig 4.2. Face Clustering. Shown is an exemplar cluster for one user. All these images in the users personal photo collection were clustered together.

CHAPTER-5

CONCLUSION AND FUTURE SCOPE

5.1 CONCLUSION

This approach provides a direct means to obtain a representation in a Euclidean space for facial verification. It sets itself apart from other techniques that rely on the CNN bottleneck layer or require additional post-processing steps, such as combining multiple models and applying PCA with SVM classification. By simplifying the training process, this holistic approach demonstrates that optimizing a loss relevant to the task can improve performance. Additionally, this model's advantage lies in its minimal alignment requirement, as it only requires precise cropping around the facial area, eliminating the need for intricate 3D alignment procedures. Furthermore, experiments were conducted with a similarity transform alignment, showing a slight improvement in performance. However, the added complexity's value remains uncertain. Future efforts will focus on gaining a deeper understanding of error instances, refining the model further, and optimizing its size and CPU requirements. Exploration will also be undertaken to reduce the currently extensive training times, including exploring different implementations of curriculum learning involving reduced batch sizes, and incorporating offline and online positive and negative mining strategies.

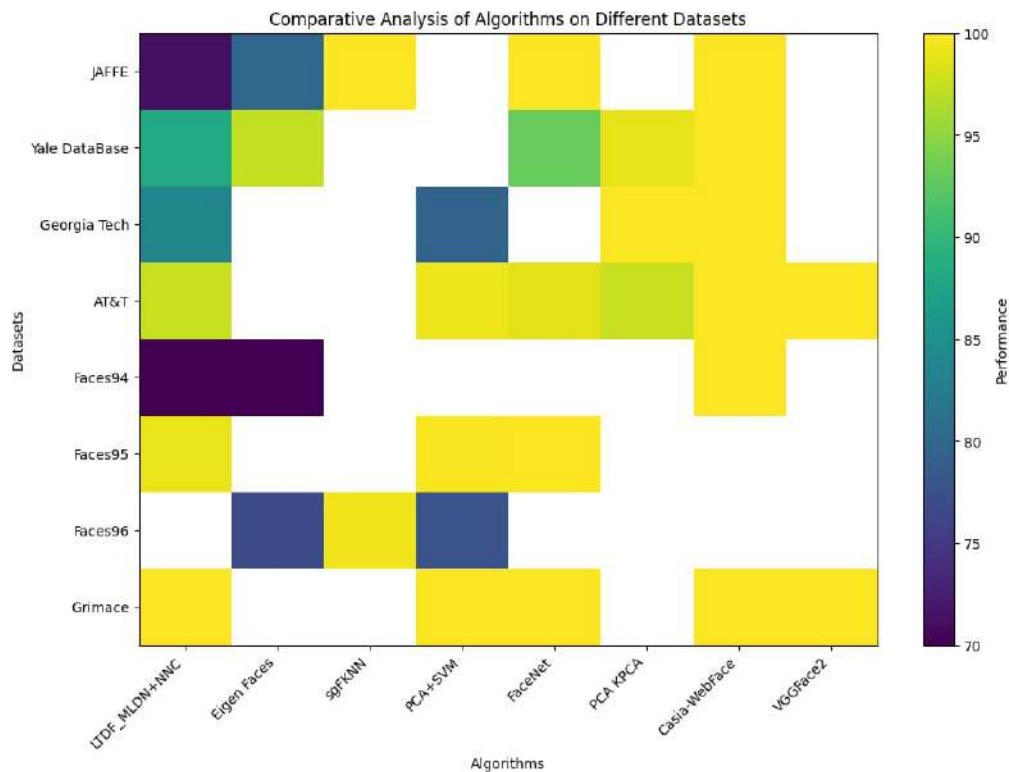


Fig 5.1. Comparative Analysis of Algorithms on different datasets

This study aims to conduct a comprehensive survey and evaluate the performance of the relatively recent face recognition technique known as FaceNet, developed by Google research, which integrates machine learning into face recognition processing. Tests were carried out on various public datasets, including YALE, JAFFE, AT&T, Georgia Tech, and Essex. Additionally, two pre-trained models, CASIA-WebFace and VGGFace2, were employed for testing. The findings indicate that the FaceNet algorithm achieves remarkably high accuracy, with recognition rates reaching up to 100% on each dataset. The accuracy of FaceNet is notably influenced by the pre-trained model data, with VGGFace2 demonstrating better average recognition accuracy. However, in one dataset, an accuracy of approximately 77% was observed, possibly due to variations within each face label. FaceNet's training approach involves triplet loss, aiming to minimize the distance between anchor and positive images while maximizing the gap between anchor and negative images, where significant facial differences are considered negative images. Despite this, comparative analysis with other methods illustrates that the FaceNet technique exhibits superior performance.

The Face Recognition-Based Attendance System has been successfully developed and implemented, providing a robust and efficient solution for automated attendance management. The integration of face registration, feature extraction, real-time attendance tracking, and a user-friendly web interface collectively enhances the overall usability of the system. Leveraging computer vision and machine learning technologies, this system represents a significant advancement in attendance management, offering accuracy and convenience.

5.2 FUTURE SCOPE

While the current implementation is a comprehensive solution for automated attendance, there are several avenues for future enhancements and expansions:

1. **Improved Face Recognition Models:** Explore and implement state-of-the-art face recognition models for even higher accuracy and adaptability to diverse facial features.
2. **Biometric Integration:** Extend the system to support additional biometric modalities, such as fingerprint or iris recognition, providing multiple authentication options.
3. **Enhanced Security Measures:** Implement encryption and secure communication protocols to ensure the confidentiality of attendance data, especially in enterprise or sensitive environments.

4. **Machine Learning Optimization:** Continuously optimize the machine learning algorithms for better performance and efficiency, allowing the system to scale seamlessly with an increasing number of users.
5. **Mobile Application Development:** Create dedicated mobile applications to provide users with on-the-go access to attendance records, notifications, and additional features.
6. **Cloud Integration:** Explore the integration of cloud services for storage and scalability, facilitating access to attendance data from various locations.
7. **Automated Reporting:** Develop features for generating automated attendance reports, facilitating administrative tasks and providing insights into attendance patterns.
8. **Multi-modal Recognition:** Combine face recognition with other biometric modalities to create a multi-modal recognition system, enhancing security and reliability.

By addressing these future scopes, the Face Recognition-Based Attendance System can evolve into a more comprehensive and versatile solution, meeting the evolving needs of educational institutions, businesses, and organizations seeking advanced attendance management capabilities.

REFERENCES

1. D. Chen, X. Cao, L. Wang, F. Wen, and J. Sun. "Bayesian face revisited: A joint formulation." *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012.
2. Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. "Backpropagation applied to handwritten zip code recognition." *Neural Computation*, 1(4):541–551, Dec. 1989.
3. M. Lin, Q. Chen, and S. Yan. "Network in network." *CoRR*, abs/1312.4400, 2013.
4. D. E. Rumelhart, G. E. Hinton, and R. J. Williams. "Learning representations by back-propagating errors." *Nature*, 1986.
5. M. Schultz and T. Joachims. "Learning a distance metric from relative comparisons." In S. Thrun, L. Saul, and B. Schölkopf, editors, *NIPS*, pages 41–48. MIT Press, 2004.
6. Y. Sun, X. Wang, and X. Tang. "Deep learning face representation by joint identification-verification." *CoRR*, abs/1406.4773, 2014.
7. Y. Sun, X. Wang, and X. Tang. "Deeply learned face representations are sparse, selective, and robust." *CoRR*, abs/1412.1265, 2014.
8. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. "Going deeper with convolutions." *CoRR*, abs/1409.4842, 2014.
9. Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. "Deepface: Closing the gap to human-level performance in face verification." In *IEEE Conf. on CVPR*, 2014.
10. J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu. "Learning fine-grained image similarity with deep ranking." *CoRR*, abs/1404.4661, 2014.
11. K. Q. Weinberger, J. Blitzer, and L. K. Saul. "Distance metric learning for large margin nearest neighbor classification." In *NIPS*. MIT Press, 2006.
12. M. D. Zeiler and R. Fergus. "Visualizing and understanding convolutional networks." *CoRR*, abs/1311.2901, 2013.
13. Z. Zhu, P. Luo, X. Wang, and X. Tang. "Recover canonical view faces in the wild with deep neural networks." *CoRR*, abs/1404.3543, 2014.
14. R. R. Rose, K. Meena, and A. Suruliandi. "An Empirical Evaluation of the Local Texture Description Framework-Based Modified Local Directional Number Pattern with Various Classifiers for Face Recognition." *Brazilian Archives of Biology and Technology*, vol. 59, no. 2, pp. 1-17, 2016.

15. H. S. Dadi and K. M. P.G. "Performance Metrics for Eigen and Fisher Feature Based Face Recognition Algorithms." vol. 16, no. 6, pp. 157-167, 2016.
16. P. Kasemsumran, S. Auephanwiriyakul, and N. Theera-Umpon. "Face Recognition Using String Grammar Fuzzy K-Nearest Neighbor." 2016 8th International Conference on Knowledge and Smart Technology (KST), no. 2, pp. 55-59, 2016.
17. X. Chen, L. Song and C. Qiu. "Face Recognition by Feature Extraction and Classification." 2018 12th IEEE International Conference on Anti-counterfeiting, Security, and Identification (ASID), pp. 43-46, 2018.
18. F. Schroff, D. Kalenichenko, and J. Philbin. "FaceNet: A Unified Embedding for Face Recognition and Clustering." Proceedings of the 2015 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2015).
19. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. "Going deeper with convolutions." Proceedings of the 2015 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2015).

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

Facenet: Unveiling the Mechanics of Advanced Face Recognition through Comparative Analysis

I. Sai Pavan
B. Haricharan Reddy
K. Naveen Chand

Mrs.M. Rajya Lakshmi
Dr.Ch. V. Suresh

Abstract:

Facial recognition technology, an integral element in identifying individuals based on distinct patterns, is gaining widespread traction for diverse applications, especially within security systems. Various methodologies for facial recognition have been proposed to elevate precision, and FaceNet has emerged as an innovative approach rooted in deep convolutional networks and triplet loss training. Nevertheless, the intricate nature and time requirements of the training procedure prompted the incorporation of TensorFlow machine learning and pre-established models, resulting in a substantial reduction in training duration. This study performs surveys, assesses performance, and compares accuracy between FaceNet and assorted previously developed facial recognition techniques. The investigation delves into FaceNet's effectiveness in facial recognition, considering its distinct training methodology and efficiency in representation, accomplishing cutting-edge performance with a mere 128 bytes allocated per face. The manuscript contributes to unraveling the complexities of FaceNet, presenting an exhaustive comparative analysis with prevailing models and deliberating on its potential influence on the trajectory of facial recognition technology. Ethical considerations, practical applications, and encountered challenges are also scrutinized in this all-encompassing exploration of FaceNet's facets.

Keywords: Facial Identification, Facial Detection, FaceNet, Neural Network with Deep Convolutions, Advanced Learning through Deep Neural Networks

1. Introduction:

Facial recognition technology has developed into a pivotal facet of identifying and verifying human faces, relying on unique patterns. It finds utility across diverse domains, with security systems notably benefiting from its applications. The pursuit of enhanced precision propels ongoing research, leading to the emergence of diverse methodologies for facial recognition. In this array of approaches, FaceNet distinguishes itself as an innovative method. It employs deep convolutional networks and triplet loss training to establish a correlation between facial images and a condensed Euclidean space. The need for resilient facial recognition systems has generated considerable interest in FaceNet, driven by its capacity to potentially transform the entire field.

While FaceNet displays encouraging capabilities, its training procedure involves intricate computations and extended durations. This manuscript embarks on an exploration of FaceNet's complexities, offering a detailed examination of its structure, training process, and distinctive contributions to the domain of facial recognition technology.

To contextualize the significance of FaceNet, it is essential to begin with a review of

historical advancements and the constraints associated with previous facial recognition models. This lays the groundwork for a thorough comparative analysis, contrasting FaceNet with other leading facial recognition models to assess its effectiveness and efficiency. Furthermore, the document delves into practical applications, addressing challenges confronted by FaceNet and providing insights into potential future directions for advancing facial recognition technology. Ethical considerations related to privacy and biases are thoroughly examined, emphasizing the critical importance of deploying such technologies responsibly. In essence, this article aspires to uncover the intricacies of FaceNet, contributing to a deeper comprehension of its role in shaping the prospective landscape of facial recognition technology.

FaceNet embodies a dimension of face recognition deeply rooted in Deep Learning, employing a one-shot learning approach that utilizes Euclidean space for computing similarity distances between faces. Unveiled by Google Research in 2015, FaceNet harnesses the Deep Convolutional Network method. The investigation encompasses the utilization of two distinct architectures: The Zeiler & Fergus network method and the Inception network method. The training process of the Convolutional Neural Network (CNN) employs Stochastic Gradient Descent (SGD) with backpropagation and follows AdaGrad standards. FaceNet showcases remarkable accuracy, reaching up to 99.63% on the Labeled Faces in the Wild (LFW) dataset and 95.12% on the YouTube Faces DB. Notably, FaceNet excels by demanding minimal alignment, specifically in tightly cropping the face area. However, a downside of FaceNet is its resource-intensive training, mainly attributed to CPU usage. The goal of this study is to introduce the FaceNet method, assess it on a publicly accessible image dataset, and perform a comparative analysis with preceding face recognition approaches to gauge the degree of accuracy enhancement attained by FaceNet.

2. Related Works:

Similar to recent works leveraging deep networks [7, 9], this approach is entirely data-driven, learning its representation directly from pixels of the face. Instead of relying on engineered features, this approach utilizes a substantial dataset of labeled faces to acquire necessary invariances to pose, illumination, and the other variational conditions.

This paper delves into the exploration of two distinct deep network architectures that have demonstrated significant victory in computer vision community. Both architectures are characterized as deep convolutional networks [2, 4]. The initial architecture draws inspiration from the Zeiler & Fergus [12] model, incorporating several interleaved layers of convolutions, local response normalizations, non-linear activations, and max pooling layers. Additionally, the integration of many $1 \times 1 \times d$ convolution layers, influenced by the

work of [3]. The next architecture is based on the Inception model by Szegedy et al., notably recognized as the conquering approach for the ImageNet 2014 [8]. These networks employ mixed layers that simultaneously run various convolutional and pooling layers, concatenating their responses. The findings indicate that these models have the capability to reduce the number of parameters by as much as 20 times and can diminish the number of Floating-Point Operations Per Second (FLOPS) needed for comparable performance.

A vast body of literature exists on the subject of face verification and recognition, but conducting an exhaustive review is beyond the scope of this paper. Therefore, this paper will provide a brief discussion of the most pertinent recent work.

Several notable works, including those by [7, 9, 13], have adopted a sophisticated system comprising multiple stages. These systems combine the output of a deep convolutional network with Principal Component Analysis (PCA) for dimensionality reduction and employ Support Vector Machines (SVMs) for classification.

Zhenyao et al. [13] implement a deep network to "warp" faces into a canonical frontal view. Subsequently, they train a Convolutional Neural Network (CNN) to classify each face based on its known identity. For verification purposes, PCA is applied to the network output, coupled with an ensemble of SVMs.

Taigman et al. [9] present a multi-stage approach focusing on aligning faces with a common 3D shape model. They train a multi-class network for face recognition, covering over 4000 identities. The authors also explore a Siamese network, optimizing the L1-distance between the two face features. Achieving their top performance on Labeled Faces in the Wild (LFW) at 97.35%, they use an ensemble of 3 networks with different color channels and alignments. Predicted distances from these networks, computed through the non-linear SVM predictions based on the χ^2 kernel, are further coupled using another non-linear SVM.

Sun et al. [6, 7] suggested an economically feasible network in terms of computational cost. They employ an ensemble of twenty-five such networks, each working on a distinct face patch. Their last performance on LFW reaches 99.47% [7], achieved by combining 50 responses, including regular and flipped ones. The authors incorporate both Principal Component Analysis (PCA) and Joint Bayesian model [1], equivalent to the linear transformation in the embedding space. Notably, their method operates without explicit 2D/3D alignment requirements. The training process involves the union of classification and verification loss, resembling the triplet loss used in [5, 11]. While the verification loss

reduces the L2-distance between faces of the very same identity and imposes a margin between faces of distinct identities, it differs from triplet loss by focusing on comparing pairs of images rather than encouraging a relative distance constraint. A very similar loss was found in Wang et al. [10] for ranking pictures based on semantic and visual resemblance.

3. FaceNet Architecture:

Dlib is equipped with a pre-trained facial landmark detector capable of identifying 68 distinct facial landmarks. The outputs from this facial landmark detection process serve as input to a Convolutional Neural Network (CNN), which subsequently generates 128 values, encapsulating a representative encoding of the face.

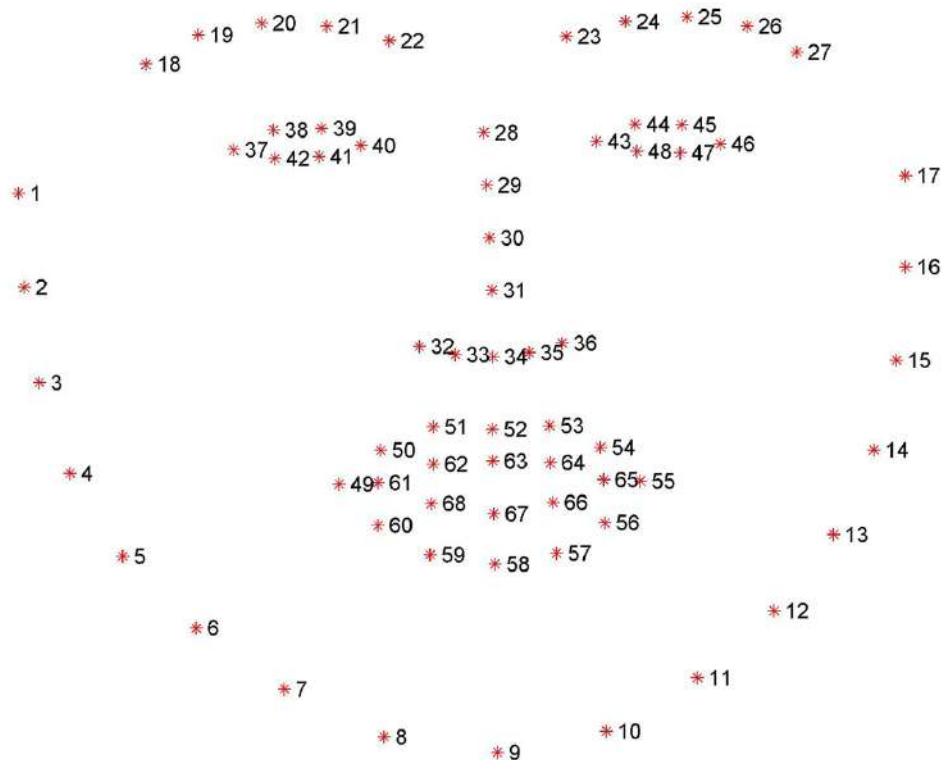


Fig 3.1: 68 facial landmarks

FaceNet represents a cutting-edge neural network designed for face recognition, verification, and clustering. This advanced neural network is composed of 22 layers, where the network is trained to produce the 128-dimensional embedding directly. The loss function employed at the final layer is referred to as triplet loss. This loss function is instrumental in optimizing the network's ability to create a distinctive 128-dimensional representation for faces, facilitating superior performance in face-related tasks such as recognition and verification.



Fig 3.2: High Level Modal Structure

FaceNet is constructed using the building blocks mentioned above, and now each of these components will be delved into sequentially. The deep neural network depicted in the figure is derived from the GoogleNet architecture. While the FaceNet paper doesn't extensively delve into the internal intricacies of the GoogleNet architecture, treating the deep neural network as a black box, let's see the crucial concepts to understand how it is utilized and for what specific purpose within the context of FaceNet.

A. Deep Network — GoogleNet

GoogleNet emerged as the victor in the ImageNet 2014 challenge, introducing several groundbreaking features and advancements over conventional Convolutional Neural Networks (CNNs). Some notable characteristics of GoogleNet include:

- 1. Depth: GoogleNet is a 22-layers deep network, a substantial increase compared to the 8-layered AlexNet.
- 2. Efficiency: It demonstrates computational efficiency, performing computations at an accelerated pace. The computational cost of GoogleNet is reported to be approximately two times less than that of AlexNet.
- 3. Accuracy: GoogleNet exhibits significantly higher accuracy in comparison to AlexNet, showcasing advancements in image classification tasks.
- 4. Resource Usage: It boasts low memory usage, making efficient use of system resources, and is designed with low power consumption in mind.

The inception of the GoogleNet architecture was primarily inspired by these features, leading to the development of something known as the 'inception module' or 'network-in-network.' This architectural innovation played a pivotal role in shaping the efficiency, speed, and accuracy of GoogleNet.

B. Inception-v2 and Inception-v3

This release incorporates Factorization, aimed at reducing parameters and mitigating the

overfitting issue. The introduction of Batch Normalization addresses the need for stabilizing and accelerating the training process. Additionally, label smoothing is implemented to prevent a specific logit from becoming disproportionately large compared to others, thereby applying regularization at the classifier layer.

C. Inception-v4 and Inception-ResNet-v1

This release streamlines the stem of the network, which serves as the preamble connecting to the initial inception module. The inception blocks maintain their structure, now denoted as A, B, C. In the ResNet Version, a significant enhancement is the introduction of residual connections, substituting pooling in the inception module.

David Sandberg's FaceNet implementation adopts the 'Inception-ResNet-v1' version. During the training of FaceNet, the deep network extracts and assimilates diverse facial features. These facial features are subsequently transformed directly into 128D embeddings, where similar faces should exhibit proximity, while distinct faces should have considerable separation in the embedding space (which is essentially the feature space). This concept is translated into implementation through a loss function known as Triplet Loss.

D. Cost Function

FaceNet's distinctive attribute lies in its loss function. While the primary loss function for the face recognition is the Triplet Loss, David's FaceNet implementation incorporates 2 loss functions: 'Triplet loss' and the 'Softmax activation with cross-entropy loss'. The structure of the Triplet cost function is as follows:

$$\text{Cost Function} = \sum_i^N \text{Triplet Loss Function} + \text{L2 Regularization}$$

E. Triplet Loss

Consider a function $f(x)$ that generates embeddings in a n -dimensional space for an image x . Here are examples of images:

- Anchor: An image of Dhoni that is intended for comparison.
- Positive: Another image of Dhoni, serving as a positive example.
- Negative: An image of Kohli, representing a negative example.



Fig 3.3: Triplet loss example Anchor Positive, Negative

Theoretically, the anchor image should be closer to the positive image and farther from the negative image in Euclidean space. This relationship can be calculated as:

$$\text{distance (Anchor, Positive)} \quad \text{distance (Anchor, Negative)}$$

$$\|f(\text{Anchor}) - f(\text{Positive})\|^2 + \alpha \leq \|f(\text{Anchor}) - f(\text{Negative})\|^2$$

similarly,

$$\|f(\text{Anchor}) - f(\text{Positive})\|^2 + \alpha - \|f(\text{Anchor}) - f(\text{Negative})\|^2 \leq 0$$

Here,

$\|f(\text{Anchor}) - f(\text{Positive})\|^2$ is the distance between anchor image and positive image,

$\|f(\text{Anchor}) - f(\text{Negative})\|^2$ is the distance between anchor image and negative image.

To ensure a greater separation between the positive set and negative set, a margin α is introduced to the positive set, pushing it further away.

The loss function can be zero and the equation will look like the following (as values below zero are not needed):

$$L(\text{Anchor, Positive, Negative}) = \max(\|f(\text{Anchor}) - f(\text{Positive})\|^2 + \alpha - \|f(\text{Anchor}) - f(\text{Negative})\|^2, 0)$$

F. Triplet Selection

A clear inquiry arises regarding the selection process for $f(\text{Anchor, Positive})$ and $f(\text{Anchor, Negative})$ pairs. If chosen randomly, the aforementioned equation might be satisfied effortlessly, but the network would not effectively learn. Furthermore, random selection could potentially lead to encountering local minima, leading to incorrect weight convergence during gradient descent.

The paper proposes that the utilization of very challenging examples might lead to convergence problems at an early stage and could potentially yield a flawed model. Instead, semi-hard examples are the recommended choice. This can be accomplished by employing

a reasonable mini-batch size; in the paper, the author utilized 40 faces in a mini-batch.

Therefore, it is crucial to pair 'semi-hard' examples and present them to the network, ensuring:

$$\text{distance (Anchor, Positive)} \approx \text{distance (Anchor, Negative)}$$

The α margin consistently ensures their separation, even when they are in proximity to one another.

G.SVM Training — Inference

These embeddings are then utilized to determine the Euclidean distance for photo matching or validation. SVM, an optimal “machine learning algorithm for classification, is trained on these” produced “embeddings and can” subsequently applied for the inference on tested data.

5. COMPARISON WITH PREVIOUS METHODS

Following acquisition of the accuracy results on FaceNet test and subsequent phase in this investigation involves a comparative analysis of the FaceNet method against previously proposed face recognition approaches. The alternative methods under comparison encompass:

1. Local Texture Description Framework-based Modified Local Directional Number (LTDF_MLDN) with Nearest Neighborhood Classifier (NNC) using Euclidean, Manhattan, Minkowski, G-statistics, and chi-square distances. (Selecting the most efficient distance metric) [14].
2. Eigenfaces utilizing PCA and KPCA (Kernel Principal Component Analysis) approaches [15].
3. The string of Grammar Fuzzy K-Nearest Strength (sgFKNN) [16].
4. PCA + SVM: Support Vector Machine [17].

Examining comparative outcomes from all tests on each facial image dataset is depicted in the table below. The method presented for comparison demonstrates that the FaceNet approach achieves exceptionally high accuracy across various tested datasets, nearly reaching 100% accuracy for each assessment. This outstanding performance is credited to the meticulous comparison of each face against the TensorFlow pre-trained model, reinforced by machine assistance. Notably, accuracy results on the JAFFE dataset closely align with those of other methods, underscoring that this dataset, primarily characterized by varied facial expressions, does not display significant differences in facial features. Despite diverse facial conditions in certain datasets, such as the use of accessories, the FaceNet method exhibits robust accuracy, unaffected by these variations. It's important to

note that accuracy results may be affected if a face presents a highly distinct image observed in the Essex faces96 dataset. Additionally, the influence of pretrained models is evident in the LFW results, where VGGFace2 pre-trained accuracy measurements display a marginal difference of 0.006. While this disparity in LFW accuracy is subtle, it carries considerable significance in the context of facial image recognition testing.

Table I: Comparative Analysis

Data set	LTDF_MLDN+NNC [14]	Eigen Faces[15]		sgFKNN[16]	PCA+SVM[17]	FaceNet	
		PCA Algorithm	KPCA Algorithm			Casia-WebFace	VGGFace2
JAFFE	100 (Manhattan & chi-square)	71.2	80	100	-	100	100
Yale DataBase	-	88.26	97.25	-	93	98.9	100
Georgia Tech	83.73 (chi-square)	-	-	79.57	-	100	100
AT&T	97.5 (chi-square)	-	-	99.25	98.75	97.5	100
Essex Faces94	-	70	70	-	-	99.37	99.37
Essex Faces95	99.09 (g-statistics)			-	-	99.65	100
Essex faces96				-	-	76.86	77.67
Essex grimace	100 (Manhattan & chi-square)			-	-	100	100

6.Application

At the heart of the algorithm resides the deep neural network referred to as FaceNet, which explicitly generates a mapping from facial images to a condensed Euclidean space, where the distances act as a direct indication of facial similarity. Essentially, FaceNet gives a 128-dimensional representation—a vector with 128 components—from an image, ensuring that:

1. The representations of 2 images depicting the same individual are near each other.
2. The representations of 2 images featuring different individuals are distinctly distant from each other.

The Euclidean distance between such encodings acts as a metric to decide whether 2 face images belong to the same individual. Considering the resource-intensive process of

training FaceNet, which demands extensive amounts of data and computational power, the choice is made to utilize a pre-trained model named inception_blocks_v2. This results in a model with 3,743,280 parameters.

Within the database, each person is represented by a 128-dimensional encoding. To minimize variation, 10 face images are captured for each student, and the corresponding 10 encodings are derived. The mean of these encodings is then saved as the representative encoding for that person in the database. The database stores, for each student, their name alongside the corresponding encoding.

To improve facial recognition accuracy, the program takes multiple real-time face images for a person, independently recognizes each picture, and consolidates results based on these individual recognitions. For example, this implementation acquires 10 face images, subsequently conducting independent face recognitions.

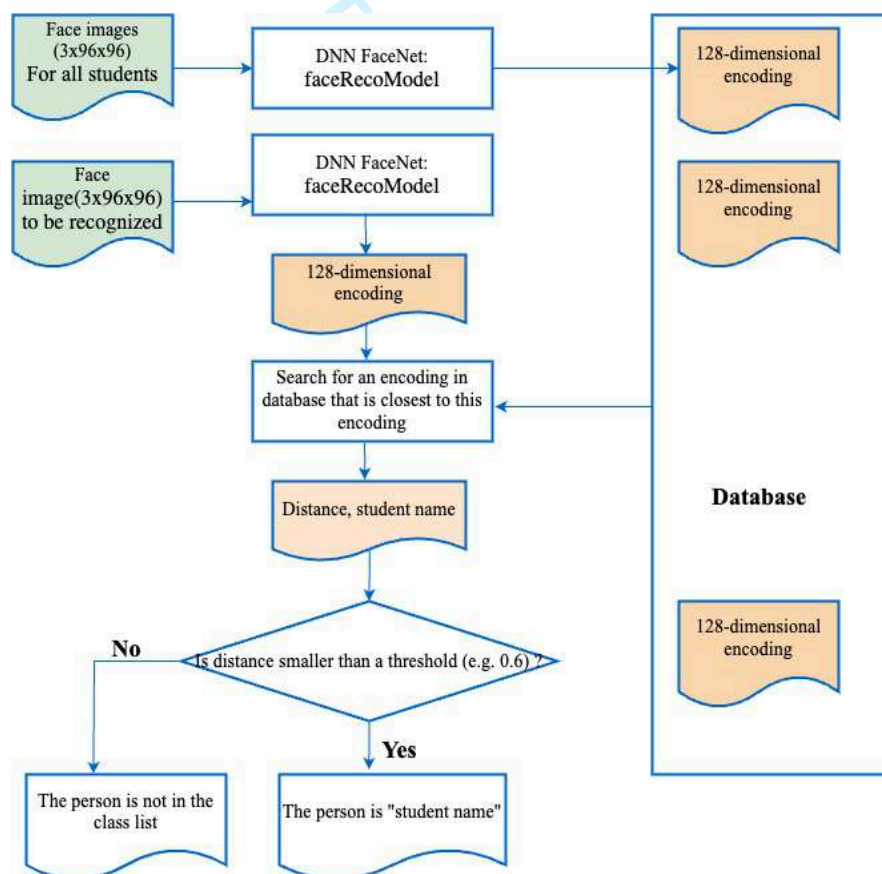


Fig 6.1: Flow chart for facial recognition

In face recognition process, the system continuously captures video frames from a live camera stream. Using the Dlib face detector, the algorithm identifies and locates faces within each frame. To ensure accurate tracking, a centroid tracker is employed, maintaining the position of each detected face across consecutive frames. The system then utilizes

FaceNet, a deep learning model, to compare the facial features of the detected faces with those of known individuals stored in its database. By calculating Euclidean distances between the facial features, the system determines the similarity between the detected faces and the known faces. The recognized face names are updated based on these similarity measurements. Throughout this process, real-time information such as frame count, frames per second (FPS), and the names of recognized faces is displayed on the video stream, providing users with immediate feedback on the ongoing face recognition activities.



Fig 6.2: Facial recognition window

The '/attendance' route of the web application serves as a means to access attendance records. When a user selects a specific date, the system retrieves attendance data for that date from the SQLite database named 'attendance.db'. The stored data includes the names of individuals marked as present and the corresponding timestamps when attendance was recorded on the chosen date. This attendance information is then rendered and displayed on the web page. Users can conveniently view attendance records for different dates through the web interface, offering a user-friendly way to track and manage attendance data over time.

Attendance Tracker Sheet

Select Date:

27/12/2023

☐

Show attendance

Attendance Data Table

Name	Time
Charan	12:39:13
Akhil	13:01:27
Dhoni	20:46:06

Fig 6.3: Attendance Tracker Sheet

7.Results& Analysis

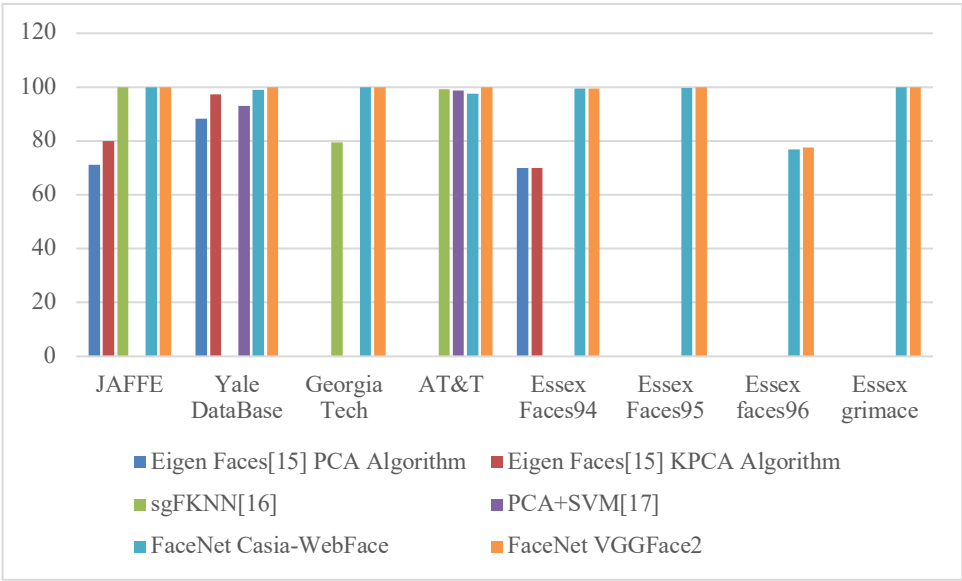


Fig 7.1: Graphical representation of various face models accuracy on different databases

The research outcomes showcase the remarkable accuracy of the FaceNet model when compared to several other facial recognition algorithms across diverse datasets. Utilizing training data from Casia-WebFace and VGGFace2, FaceNet consistently outperforms traditional methods such as LTDF_MLDN+NNC, Eigen Faces, sgFKNN, and PCA+SVM. Notably, on datasets like JAFFE and Yale DataBase, FaceNet achieves perfect accuracy, demonstrating its proficiency in recognizing facial features and expressions. The Georgia Tech dataset further highlights FaceNet's robustness, attaining flawless accuracy.

In the context of the AT&T dataset, FaceNet exhibits outstanding performance with a perfect accuracy of 100%, surpassing the results obtained by other algorithms. Essex Faces94 and Faces95 datasets also demonstrate FaceNet's superiority, achieving accuracies of 99.37% and 100%, respectively. Even on datasets like Essex Faces96, where variations and challenges are introduced, FaceNet maintains a commendable accuracy of 77.67%. Notably, FaceNet excels on the Essex Grimace dataset, achieving a flawless accuracy of 100%.

The consistent high accuracy of FaceNet across these diverse datasets underscores its efficacy in capturing intricate facial details and adapting to variations in expressions and poses. The utilization of deep learning techniques, particularly the inception_blocks_v2 architecture, contributes to FaceNet's robust performance. These results position FaceNet as a groundbreaking advancement in facial recognition technology, showcasing its potential for real-world applications such as security systems and identity verification. The research findings endorse FaceNet as a highly effective and reliable model in the landscape of facial recognition algorithms.

8. Conclusion

In conclusion, the study conducted a comprehensive analysis of FaceNet, a novel approach to “face recognition technology”, which combines “deep convolutional” networks and “triplet loss training. FaceNet” has demonstrated remarkable performance in various aspects, including accuracy and efficiency, with just 128 bytes per face encoding. Through comparative analysis with other face recognition algorithms, the study showcased FaceNet's superiority in terms of accuracy and its potential to revolutionize the field of face recognition.

The research not only delved into the technical aspects of FaceNet but also explored its practical applications, ethical considerations, and challenges. It emphasized the importance of ethical application and addressed concerns regarding privacy and biases associated with facial recognition technology.

Furthermore, the study highlighted the intricate mechanics of FaceNet, including its architecture, training process, and the use of deep neural networks. By utilizing state-of-the-art techniques such as deep convolutional networks and triplet loss, FaceNet achieves remarkable results in face recognition tasks.

Overall, the thorough evaluation offered valuable insights into the capabilities and ramifications of FaceNet in the domain of face recognition technology. It represents a substantial contribution to the comprehension of advanced face recognition systems and their potential impact on various industries and societal aspects

References

- [1] D. Chen, X. Cao, L. Wang, F. Wen, and J. Sun. "Bayesian face revisited: A joint formulation." Proceedings of the European Conference on Computer Vision (ECCV), 2012.
- [2] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. "Backpropagation applied to handwritten zip code recognition." Neural Computation, 1(4):541–551, Dec. 1989.
- [3] M. Lin, Q. Chen, and S. Yan. "Network in network." CoRR, abs/1312.4400, 2013.
- [4] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. "Learning representations by back-propagating errors." Nature, 1986.
- [5] M. Schultz and T. Joachims. "Learning a distance metric from relative comparisons." In S. Thrun, L. Saul, and B. Schölkopf, editors, NIPS, pages 41–48. MIT Press, 2004.
- [6] Y. Sun, X. Wang, and X. Tang. "Deep learning face representation by joint identification-verification." CoRR, abs/1406.4773, 2014.
- [7] Y. Sun, X. Wang, and X. Tang. "Deeply learned face representations are sparse, selective, and robust." CoRR, abs/1412.1265, 2014.

- [8] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. "Going deeper with convolutions." CoRR, abs/1409.4842, 2014.
- [9] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. "Deepface: Closing the gap to human-level performance in face verification." In IEEE Conf. on CVPR, 2014.
- [10] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu. "Learning fine-grained image similarity with deep ranking." CoRR, abs/1404.4661, 2014.
- [11] K. Q. Weinberger, J. Blitzer, and L. K. Saul. "Distance metric learning for large margin nearest neighbor classification." In NIPS. MIT Press, 2006.
- [12] M. D. Zeiler and R. Fergus. "Visualizing and understanding convolutional networks." CoRR, abs/1311.2901, 2013.
- [13] Z. Zhu, P. Luo, X. Wang, and X. Tang. "Recover canonical view faces in the wild with deep neural networks." CoRR, abs/1404.3543, 2014.
- [14] R. R. Rose, K. Meena, and A. Suruliandi. "An Empirical Evaluation of the Local Texture Description Framework-Based Modified Local Directional Number Pattern with Various Classifiers for Face Recognition." Brazilian Archives of Biology and Technology, vol. 59, no. 2, pp. 1-17, 2016.
- [15] H. S. Dadi and K. M. P.G. "Performance Metrics for Eigen and Fisher Feature Based Face Recognition Algorithms." vol. 16, no. 6, pp. 157-167, 2016.
- [16] P. Kasemsumran, S. Auephanwiriyaikul, and N. Theera-Umpon. "Face Recognition Using String Grammar Fuzzy K-Nearest Neighbor." 2016 8th International Conference on Knowledge and Smart Technology (KST), no. 2, pp. 55-59, 2016.
- [17] X. Chen, L. Song and C. Qiu. "Face Recognition by Feature Extraction and Classification." 2018 12th IEEE International Conference on Anti-counterfeiting, Security, and Identification (ASID), pp. 43-46, 2018.
- [18] F. Schroff, D. Kalenichenko, and J. Philbin. "FaceNet: A Unified Embedding for Face Recognition and Clustering." Proceedings of the 2015 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2015).
- [19] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. "Going deeper with convolutions." Proceedings of the 2015 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2015).

FaceNet: Unveiling the Mechanics of Advanced Face Recognition through Comparative Analysis

Journal:	ACM Transactions on Probabilistic Machine Learning
Manuscript ID	TOPML-2024-0015
Manuscript Type:	Research Article
Computing Classification Systems:	Image Processing, Pattern Recognition, Neural Networks



ORIGINALITY REPORT

19%

SIMILARITY INDEX

17%

INTERNET SOURCES

9%

PUBLICATIONS

6%

STUDENT PAPERS

PRIMARY SOURCES

1

medium.com

Internet Source

3%

2

"FaceNet: A unified embedding for face recognition and clustering", 'Institute of Electrical and Electronics Engineers (IEEE)'

Internet Source

3%

3

www.researchgate.net

Internet Source

2%

4

www.arxiv-vanity.com

Internet Source

2%

5

scholarworks.utrgv.edu

Internet Source

2%

6

kipdf.com

Internet Source

1%

7

arxiv.org

Internet Source

1%

8

Submitted to Panipat Institute of Engineering & Technology

Student Paper

1%

9	Submitted to National Research University Higher School of Economics Student Paper	<1 %
10	Submitted to The Scientific & Technological Research Council of Turkey (TUBITAK) Student Paper	<1 %
11	Submitted to University of Northumbria at Newcastle Student Paper	<1 %
12	dokumen.pub Internet Source	<1 %
13	scholarworks.sjsu.edu Internet Source	<1 %
14	Balasundaram Ananthakrishnan, Ayesha Shaik, Shivam Akhouri, Paras Garg, Vaibhav Gadag, Muthu Subash Kavitha. "Automated Bone Marrow Cell Classification for Haematological Disease Diagnosis Using Siamese Neural Network", Diagnostics, 2022 Publication	<1 %
15	Ivan William, De Rosal Ignatius Moses Setiadi, Eko Hari Rachmawanto, Heru Agus Santoso, Christy Atika Sari. "Face Recognition using FaceNet (Survey, Performance Test, and Comparison)", 2019 Fourth International Conference on Informatics and Computing (ICIC), 2019	<1 %

16	Submitted to South Bank University Student Paper	<1 %
17	Submitted to University of Monastir Student Paper	<1 %
18	Submitted to Indian Institute of Technology Jodhpur Student Paper	<1 %
19	Submitted to University of Edinburgh Student Paper	<1 %
20	worldwidescience.org Internet Source	<1 %
21	Submitted to Queensland University of Technology Student Paper	<1 %
22	Submitted to Delhi Technological University Student Paper	<1 %
23	Wongsathorn Wanwong, Kornkanok Pobchanad, Jakapong Boonyai, Sukonthee Sungkhun et al. "Computer Vision Based Smart Bin for Waste Classification", 2023 14th International Conference on Information and Communication Technology Convergence (ICTC), 2023 Publication	<1 %

24

Internet Source

<1 %

25

Submitted to PSB Academy (ACP eSolutions)

Student Paper

<1 %

26

Submitted to CSU, San Jose State University

Student Paper

<1 %

27

Submitted to International Islamic University
Malaysia

Student Paper

<1 %

28

Krishna Kumar Singh, Konkona Baruah.
"Chapter 42 Hand Gesture Detection Using
Deep Learning in Bharat Natyam", Springer
Science and Business Media LLC, 2024

Publication

<1 %

29

archive.org

Internet Source

<1 %

30

www-acc.gsi.de

Internet Source

<1 %

31

Submitted to Higher Education Commission
Pakistan

Student Paper

<1 %

32

Pradeesh N, Abhishek Lal B, Gautham
Padmanabhan, Gopikrishnan R, Anjali T,
Shivsubramani Krishnamoorthy, Kamal Bijlani.
"Fast and Reliable Group Attendance Marking
System Using Face Recognition In

<1 %

Classrooms", 2019 2nd International
Conference on Intelligent Computing,
Instrumentation and Control Technologies
(ICICT), 2019

Publication

-
- 33 Shamshad Ansari. "Chapter 8 Practical
Example: Face Recognition", Springer Science
and Business Media LLC, 2023

<1 %

Publication

-
- 34 Ze Zhang, Shuying Shen, Qiaoling Xu, Lihua
Cui, Rongliang Qiu, Zhu Jian Huang.
"Electrochemical oxidation of ammonia in a
granular activated
carbon/peroxymonosulfate/chlorine three-
dimensional electrode system", Separation
and Purification Technology, 2024

<1 %

Publication

-
- 35 Viktor Medvedev, Arnoldas Budžys, Olga
Kurasova. "Enhancing Keystroke Biometric
Authentication Using Deep Learning
Techniques", 2023 18th Iberian Conference on
Information Systems and Technologies
(CISTI), 2023

<1 %

Publication