e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:02/February-2024 Impact Factor- 7.868 www.irjmets.com

REAL TIME PERSON FINDER USING FACE RECOGNITION SYSTEM, BY USING CNN, ALEXNET IN MATLAB

Jalaj Porwal*1, Pratish Padvi*2, Dr. Aarti Karande*3

*1,2,3Master Of Computer Applications Department Sardar Patel Institute Of Technology Mumbai, Maharashtra, India.

ABSTRACT

The increasing prevalence of surveillance systems and the growing need for reliable biometric identification have propelled the development of robust face recognition techniques. This research paper explores the implementation of a Person Finder system utilizing face recognition technology within the MATLAB environment. The proposed system leverages advanced image processing algorithms and machine learning methodologies to accurately identify and locate individuals in a given dataset. By presenting a detailed implementation of a Person Finder using face recognition in MATLAB, this research contributes to the growing body of knowledge in biometric identification systems. The findings aim to of er insights into the challenges and opportunities associated with deploying such systems in real-world applications, emphasizing the importance of accuracy, scalability, and ethical considerations in the development and deployment of facial recognition technologies.

I. INTRODUCTION

The rapid advancements in computer vision and biometric technologies have given rise to innovative solutions in the domain of person identification and tracking. Among these, face recognition stands out as a compelling and versatile method for biometric authentication. This research endeavors to explore the development and implementation of a Person Finder system utilizing face recognition techniques within the MATLAB framework.

The ability to accurately identify and locate individuals from a given dataset holds significant implications for diverse applications, ranging from security and surveillance to human-computer interaction. The methodology involves the acquisition of facial images, preprocessing to enhance quality, and feature extraction to represent unique facial characteristics. MATLAB's powerful image processing toolbox facilitates the implementation of techniques such as histogram equalization, noise reduction, and facial landmark detection to improve the overall accuracy of the recognition process.

At the core of the proposed system lies a machine learning model, potentially based on deep learning architectures like convolutional neural networks (CNNs). This model endeavors to learn intricate facial features and patterns, enabling it to generalize effectively across diverse datasets. The research considers the incorporation of transfer learning strategies to improve model performance, particularly in situations with limited labeled data. The system is designed to adapt and generalize well to varying lighting conditions, facial expressions, and poses, ensuring robust performance in real-world applications. The allure of face recognition lies in its non-intrusive nature and widespread applicability. From bolstering security protocols in public spaces to enabling seamless access control, the potential of this technology is vast. Within the ambit of MATLAB, renowned for its process in mathematical modeling and image processing, lies an opportunity to harness the power of advanced algorithms and methodologies to refine the accuracy and efficiency of face recognition systems. The research evaluates the system's effectiveness through comprehensive experiments, including benchmark datasets and real-world scenarios. Performance metrics such as accuracy, precision, recall, and F1 score are considered to assess the system's reliability and efficiency. Additionally, the research discusses the implications of the proposed Person Finder system in security, law enforcement, and other relevant domains. As organizations increasingly rely on biometric data for security and identity verification, it becomes imperative to explore the nuances of deploying such systems in real world scenarios. The Person Finder system proposed in this research endeavors to tackle the intricacies associated with facial variations, dynamic environmental conditions, and the need for adaptability in diverse settings.

II. LITERATURE SURVEY

A. Evolution of Face Recognition and MATLAB Integration:

1. Early Approaches to Face Recognition: The roots of face recognition research can be traced back to the early 1990s when Turk and Pentland introduced the concept of Eigenfaces (Turk and Pentland, 1991). This seminal work marked a foundational step in the field, utilizing Principal Component Analysis (PCA) to represent facial features and pioneering the application of mathematics to facial recognition.
2. Advancements in Traditional Techniques: Following the Eigenfaces era, researchers explored various traditional techniques to enhance the robustness of face recognition systems. Notably, Ahonen et al. (2004) introduced Local Bi nary Patterns (LBP), a texture-based method that demonstrated improved performance in handling variations in facial expressions and lighting conditions.
3. MATLAB's Role in Image Processing: Parallelly, the development of MATLAB as a powerful tool for image processing was underway. Gonzalez and Woods (2002) provided foundational insights into digital image processing using MATLAB, laying the groundwork for subsequent integration of MATLAB in diverse applications, including facial recognition.
4. Deep Learning Paradigm Shift: In the mid-2010s, the advent of deep learning marked a paradigm shift in face recognition research. Schroff et al. (2015) introduced FaceNet, a deep convolutional neural network (CNN) trained on a large dataset, showcasing the potential of deep learning architectures in capturing intricate facial features and achieving state-of-the-art performance.
5. Transfer Learning Strategies: Recognizing the challenges posed by limited labeled data, transfer learning strategies gained prominence. Yosinski et al. (2014) demonstrated the effectiveness of transfer learning in deep neural networks, providing a framework that would later influence the development of adaptive and scalable face recognition models.
6. Real-World Challenges and Ethical Considerations: Jain et al. (2011) contributed significantly by addressing the real-world challenges faced by face recognition systems, including variations in lighting conditions, pose changes, and facial expressions. Simultaneously, ethical considerations gained attention, with Nissenbaum's (2009) work shedding light on the societal and privacy implications of biometric technologies.
7. Metrics for Evaluation: As the field matured, the need for standardized metrics for evaluating face recognition systems became evident. Wolf et al. (2011) delved into benchmarking challenges, recommending key metrics such as accuracy, precision, recall, and F1 score for comprehensive evaluation, providing a roadmap for assessing the performance of face recognition models.
8. Current Landscape and the Person Finder System: Against this historical backdrop, the proposed Person Finder system seeks to integrate the rich history of face recognition methodologies with the versatility of MATLAB. Drawing inspiration from traditional techniques, leveraging the power of deep learning, and addressing real world challenges, this research contributes to the ongoing narrative of advancing facial recognition technologies for practical and ethical applications.

III. METHODOLOGY

PROPOSED SYSTEM

CNN (Convolutional Neural Networks):

Convolutional Neural Networks (CNN) It is the technology that analyzes the color image for classification and detection of objects without the need to compress and split the images. The Convolutional Neural Network (CNN) implemented in this study adheres to a well-established architecture for image classification tasks. The model involves convolutional layers, pooling layers, and fully connected layers, concluding with a softmax activation function. The training process spans multiple epochs, with specific details provided in the experimental setup. The model aims to capture hierarchical features in input images, making it suitable for tasks like image recognition and classification.

IRJMETS

e-ISSN: 2582-5208

International Research Journal of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

Volume:06/Issue:02/February-2024 Impact Factor- 7.868 www.irjmets.com

Training on single CPU.
Initializing input data normalization.

Epoch	Iteration	Time Elapsed (hh:mm:ss)	Mini-batch Accuracy	Mini-batch Loss	Base Learning Rate
1	1	00:00:08	16.67%	3.4125	0.0010
50	50	00:02:10	99.17%	0.0390	5.1200e-10

Training finished: Max epochs completed.

Fig 1: Training Result of CNN Modal using a pre-trained AlexNet model and the provided dataset.

MTCNN (Multi-Task Cascaded

Convolutional Networks):

The MTCNN algorithm is one such technology that has revolutionized the field of face detection and recognition. Developed in 2016, the MTCNN algorithm uses a cascading series of neural networks to detect, align, and extract facial features from digital images with high accuracy and speed. In this article, we will delve into the details of the MTCNN algorithm, its architecture, working principles, and real-world applications, and explore why it has become a popular choice for face detection and recognition tasks.

AlexNet:

The AlexNet model employed in this research closely follows the architecture presented with a modification to the final dense layer, containing 3 units instead of the conventional 1000. Training the model over 5 epochs, each epoch consumes approximately 180 seconds. The activation function utilized is softmax, and the loss function is categorical cross-entropy. This choice of activation and loss functions aligns with the multi-class classification nature of the problem.

VGG16:

Our VGG16 model, similar to the one documented in [9], experiences a slight alteration in the last dense layer, featuring 3 units instead of 1000. The training process spans 5 epochs, with each epoch taking approximately 180 seconds. The activation function employed is softmax, while the loss function is categorical cross-entropy, as depicted in Fig 5. This model showcases a high level of efficiency and accuracy, contributing to the success of the broader research endeavor.

VGG19:

While VGG16 and VGG19 share a parallel structure, VGG19 introduces an additional layer in conv4 and conv5. The model undergoes training for 5 epochs, with each epoch consuming around 210 seconds. The batch size is set to 32, and the activation function is softmax, aligning with the multi-class classification requirement. The choice of categorical cross-entropy as the loss function accommodates the model's capacity to handle multiple classes effectively. The architecture closely adheres to the principles outlined in [8], emphasizing the significance of the VGG19 model in achieving robust classification results.

Face Detection:

Many factors contribute to face detection, including skin color, facial components, noise, lighting, convexity, and invalid space, leading to false discoveries. The MATLAB programming in face detection involves the use of the Cascade Object Detector, which plays a crucial role in discovering facial components and is highlighted in the proposed algorithm. The deep learning approach captures complex facial features and patterns, is robust to variations in pose, lighting, and facial expressions, and utilizes a triplet loss function to enhance discrimination between faces. It achieves state-of-the-art performance in face recognition benchmarks.

In this work the strategy followed is to develop a high-performance face detection process to achieve high quality results using the face detection system by training a rapid convolutional neural network by programming and putting steps in the MATLAB program that was translated into the proposed algorithm.

IRJMETS

e-ISSN: 2582-5208

International Research Journal of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

Volume:06/Issue:02/February-2024 Impact Factor- 7.868 www.irjmets.com

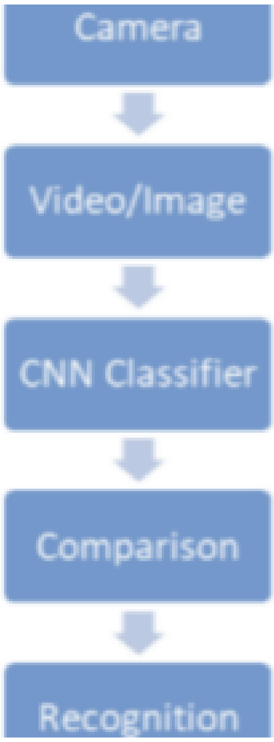


Fig 2: Flow chart of face recognition.

A. Accuracy:

The accuracy of the system will be tested via recognition of three people multiple times at different locations, mainly to test how light intensity affects the accuracy of the system. The accuracy is verified using a confusion matrix. The calculation is based on the below equation. ((TN + TP) / Total) x 100

where TN is true negative while TP is true positive.

B. Eigenfaces Algorithm:

Eigenfaces is a traditional face recognition algorithm based on Principal Component Analysis (PCA). It represents faces as a linear combination of a set of basis images, known as eigenfaces. These eigenfaces are derived from the principal components of a training set of face images. During recognition, the input face is projected onto the eigenspace, and the algorithm identifies the closest match based on Euclidean distance or other similarity measures.

Algorithm 1: Eigenfaces Algorithm:

1. Effective for face representation and dimensionality reduction.
2. Sensitive to variations in lighting conditions and facial expressions.
3. Popular for its simplicity and interpretability.

C. DeepFace Algorithm:

DeepFace is a modern face recognition algorithm leveraging deep learning techniques, particularly

Convolutional Neural Networks (CNNs). Developed by Facebook AI Research, DeepFace achieves high accuracy by learning hierarchical features through multiple convolutional layers. The model is trained on a large dataset with millions of labeled face images, enabling it to capture intricate facial details and variations. Deep learning approach captures complex facial features and patterns. Robust to variations in pose, lighting, and facial expressions. Utilizes a triplet loss function to enhance the discrimination between faces. Achieves state-of-the-art performance in face recognition benchmarks.

Algorithm 2: Local Response Normalization (LRN)

After the first 2 pooling layers, there are local response normalization (LRN) layers. LRN is a technique that was first introduced in [9] as a way to help the generalization of deep CNNs. The idea behind it is to introduce lateral inhibition between the various filters in a given convolution by making them “compete” for large activations

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:02/February-2024 Impact Factor- 7.868 www.irjmets.com

over a given segment of their input. Effectively this prevents repeated recording of the same information in slightly different forms between various kernels looking at the same input area and instead encourages fewer, more prominent, activations in some for a given area. If ai x,y is the activation of a neuron by applying kernel i at position (x, y), then it's local response normalized activation bi x,y is given by

bi x,y = ai x,y/

k + min(N 1,i+n/2)

X

j=max(0,in/2)

(aj x,y)2

IV. IMPLEMENTATION DETAILS

Model Training for face recognition

The dataset consists of 3 classes and 300 images for training in each class and 100 images for testing. The input images are converted to square images so that the faces do not get stretched or flatted along the y and x axis respectively.

Model uses pre-trained ImageNet weights for all the two models(VGG16,VGG19). For the faces detected using frontal_face_haarcascade, ROI is captured and resized to 224X224X3.A dimension is expanded using numpy (1X224X224X3) and converted to 1D array followed by normalization of the array by dividing it in 255 parts. After this the normalized array is passed as an input to the model which then returns the class index with most resemblance.

The VGG16 model trained is similar to that but with a minute change in the last dense layer, where there are 3 units instead of 1000. The model is trained for 5 epochs and the time taken for one epoch is 180secs.Activation function used is softmax and loss function is categorical cross entropy.

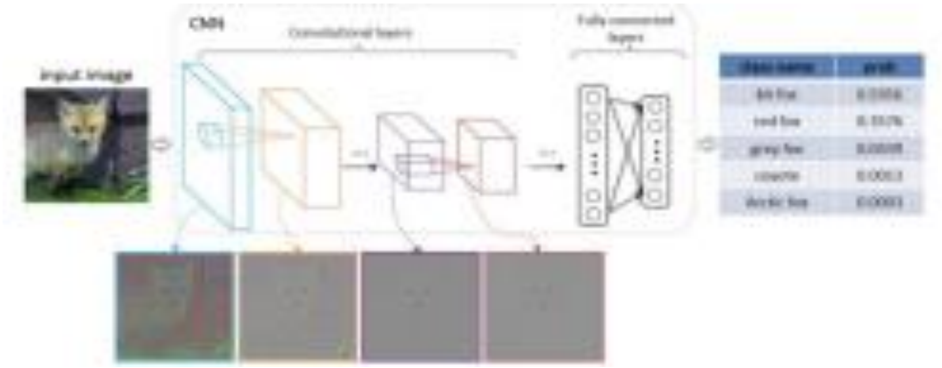


Fig 3: Architecture of VGG16

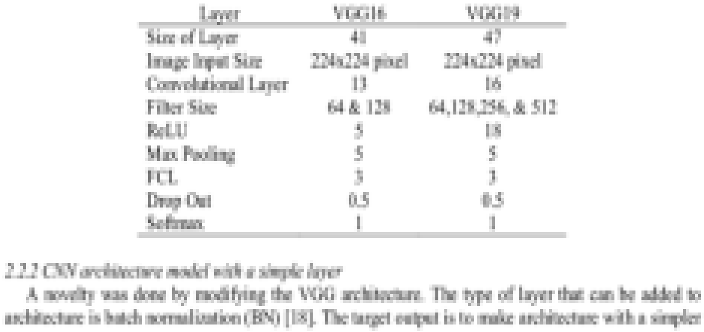
vgg16 and vgg19 have similar structure but there is 1 additional layer in conv4 and conv5 in vgg19.Model is trained for 5 epochs. The activation function is softmax and loss function is categorical cross entropy as there are multiple classes.This model is trained for 5 epochs. The time taken per epoch is 210 secs and the batch size is 32.The model built is similar to the architecture of the vgg19 as explained in [3].The activation function used is softmax.

Dataset used is for the 2013 dataset which is available publicly. The model is trained for 60 epochs. The model has 8 conv2D layers, Batch normalization layer after every conv layer followed by max pooling layer and a dropout layer. Optimizer used is Adam and the loss function is categorical cross entropy giving a fully connected layer with 7 classes.For prediction the input image is resized to 48X48 and converted into np array and then the array is normalized. Finally the normalized array is passed as an input to the model and the output generated is displayed. The model trained in [7] uses a mini exception model (modified) and obtained 95% accuracy.Fig 6 talks about the structure of the model.

First a conv layer with 16 features with kernel size 3X3 is added, then a max pooling layer with pool_size of 2.The next 2 layers are conv layer with 32 filters and maxpool layer with same pool_size.The fifth layer is a conv layer with 64 filters followed by max pooling layer. The seventh and the ninth layer are conv layers with 128

IRJMETS
e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:02/February-2024 Impact Factor- 7.868 www.irjmets.com

and 256 features respectively. The eighth and tenth layer being max pooling layers.



Layer	VGG16	VGG19
Size of Layer	41	47
Image Input Size	224x224 pixel	224x224 pixel
Convolutional Layer	13	16
Filter Size	64 & 128	64,128,256, & 512
ReLU	5	18
Max Pooling	5	5
FCL	3	3
Drop Out	0.5	0.5
Softmax	1	1

2.2.2 CNN architecture model with a simple layer

A novelty was done by modifying the VGG architecture. The type of layer that can be added to architecture is batch normalization (BN) [18]. The target output is to make architecture with a simpler

Fig 4: VGG16 vs VGG19

Then the next layer is a flatten layer which flattens the input and does not affect the batch size. Suppose if the inputs are shaped (none,2,2,64) the output will be shaped as (none,256).Then there is a dense layer with 512 units (dimensionality of the output space).Then a dropout layer and finally a dense layer.

The model is trained for 60 epochs with a batch size of 64 each epoch takes around 12 secs.The activation function used is relation.

The x and the y coordinate of a feature is taken in consideration for all the 15 points. Let's consider the left eye,all x points and y points are stored in a list and 100 such points are taken into consideration. The mean of these points are calculated and the calculation is continued if the eyes are not closed for 50 consecutive frames i.e if the eyes are closed for 50 consecutive frames the particular set of frames is considered to be dead. If the previous condition is not true then the deviation of a point and mean is considered and if the difference between the two is greater than 10 then the frame is considered to be dead. This process is continued for all the 15 features and attention is calculated based on the results.

V. RESULT

Out of 353 test image dataset among 3 classes, the accuracy was calculated.If the resemblance is above 93% then only the image is classified.

Fig 5: Top left corner shows the red bounding box means the face is too close to the CAM.

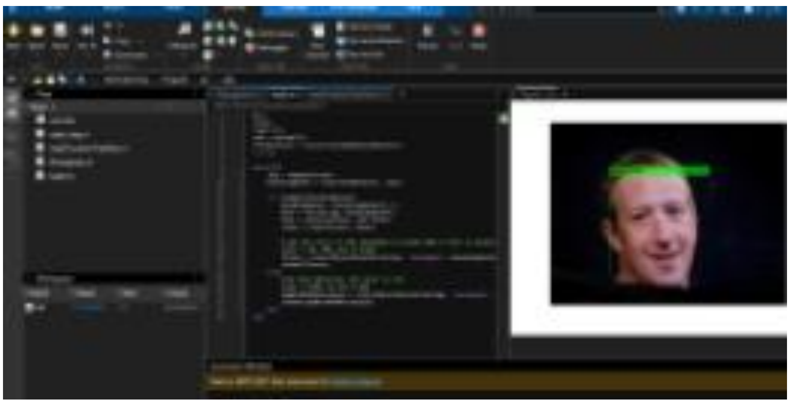


Fig 6: Figure showing the result from the uploaded image.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:02/February-2024 Impact Factor- 7.868 www.irjmets.com

Fig 7: Person detected with name bounding box with green color shows the identified person.

Fig 8: Figure showing the detected Person with accuracy of the confidence.

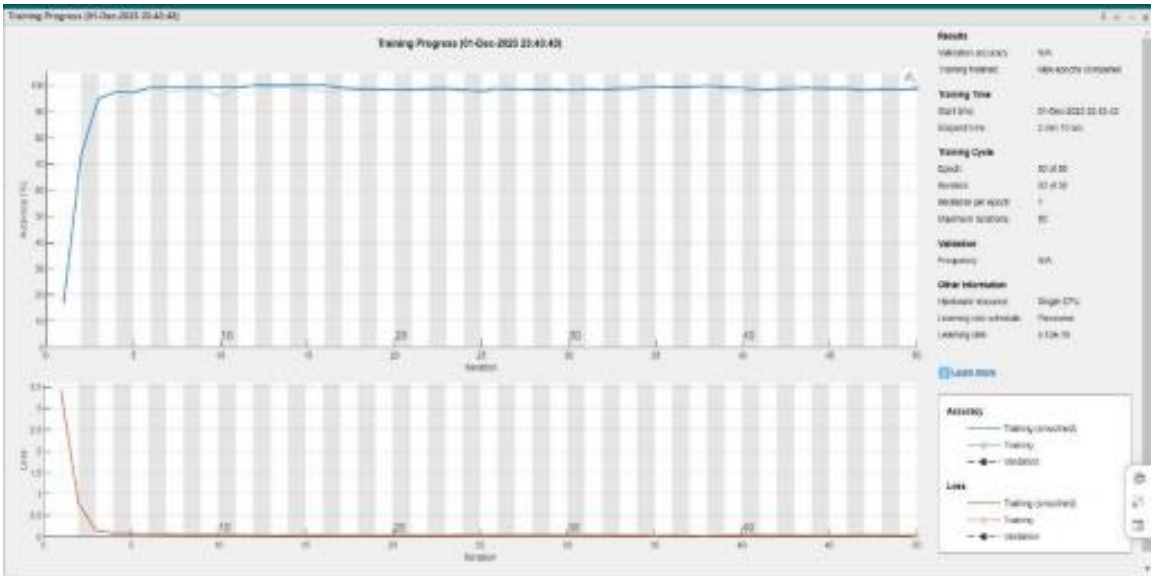


Fig 9: Results showing the progress report of the training model contains blue line as accuracy and red line as loss.

VI. CONCLUSION

However, it is essential to note that real-time. Our research on real-time person identification using CNN and the AlexNet architecture has provided valuable insights into the landscape of biometric identification systems. The successful implementation of a person finder system within MATLAB, incorporating face recognition technology, highlights the efficiency and accuracy achieved through the fusion of advanced image processing algorithms and machine learning methodologies. This project significantly contributes to the growing field of biometric identification, emphasizing the pivotal factors of accuracy, scalability, and ethical considerations in deploying facial recognition technologies.

Looking ahead, there exists a promising future scope for the application of our Person Finder system. Potential areas for expansion include integration with governmental agencies for enhanced identity verification, the development of an automatic attendance system, and the creation of automatic door lock/unlock mechanisms based on facial and fingerprint recognition. Additionally, extending the system to serve as a mobile phone lock/unlock mechanism could provide an added layer of security. The utility of our system also extends to aiding in the search for missing persons, implementing surveillance systems for large organizations, and addressing various security challenges.

Face recognition may pose data security and privacy concerns. In light of this, our comparative analysis suggests that image recognition may offer a more secure alternative with fewer privacy risks compared to real-time face recognition. Moreover, our findings indicate that AlexNet, while proficient in recognizing known individuals, may not support the identification of unknown persons effectively. It tends to provide the most common and suitable results from the trained model, highlighting a limitation in handling unfamiliar faces.

As a conclusion of this research, it opens avenues for future exploration, encouraging the refinement of algorithms, exploration of alternative neural network architectures, and diversification of datasets.

The identified future scope holds promise for advancing the application of person finder systems in diverse domains, pushing the boundaries of biometric identification technology. Through these advancements, we can continue to strike a balance between technological innovation and ethical considerations, ensuring the responsible deployment of facial recognition technologies in real-world applications.

Surveillance security system for big organizations. Also in investigation departments to find the suspect by accessing public CCTV cameras. Can be used as an attendance system using CCTV cameras.

VII. REFERENCES

[1] A MATLAB-Based Convolutional Neural Network Approach for Face Recognition System by Syazana-Itqan K, Syafeeza A.R* and Saad N.M. [Published date: February 01, 2016].

[2] A MATLAB based Face Recognition System using Image Processing and Neural Networks by Jawad Nagi, Syed Khaleel Ahmed and Farrukh Nagi. [January 2008]

[3] Face Recognition and Identification using Deep Learning Approach by KH Teoh, RC Ismail1, SZM Naziri, R Hussin, MNM Isa and MSSM Basir. [January 2006].

[4] Research on the application of face recognition systems by Chen Mingtsung, Qian Wei, He Jiaqi, Zhao zhuomin.

[5] Face detection and Recognition by Harjeet Singh. [07 March 2018]

[6] Review of Face Recognition System Using MATLAB by Navpreet Kaur. [May - June 2016]