FACE RECOGNITION USING MACHINE LEARNING

Ch. Nikhila, G. Teja Sukantha, R. Rohith Satya Sai, L. Joshna , Satyanarayana Murty.P
P.Suneetha, Y.Sukanya Department of Electronics and Communication Engineering,
Vignan's Institute of Information Technology, Duvvada, Visakhapatnam, Andhra Pradesh,

Abstract
Face recognition has been considered one of the most fascinating domains over the last few years and has been among the most productive image processing applications. We are going to create a method for face recognition with the use of three convolutional neural networks, namely AlexNet, LeNet, and VGGNet, and their accuracies were compared to get a better understanding of the best network. The three networks were tested on the same dataset, then compared for accuracy. A real-time and a standard dataset were chosen to study the performance of these three CNNs. We have chosen a platform called Spyder to test our model since we have opted for a Python interface. This model is used to look at how well the three neural networks work with both standard datasets and real-time datasets over accuracy.
Keywords: CNN, AlexNet, VGGNet, LeNet, Convolutional Layer, Pooling Layer, Fully Connected Layer, Epochs, Accuracy.

1. Introduction

Facial recognition is a technology that identifies a human face from a digital image or video against a dataset of faces. It has numerous applications, which include automatic attendance monitoring systems in [1], emotion recognition in [2], security and surveillance, verification of identity, and significantly more. Face recognition's great potential in a variety of government and commercial applications makes it popular and has attracted research attention and increased its development over the last 30 years. The very beginning of the classification of faces was first proposed in [3]. Face recognition systems are now involved in numerous real-world applications, indicating that progress has been made [4]. Face recognition has moved quickly due to several things, such as the fact that there are already models that make the process easier and access to large databases of pictures.

The first signs of facial recognition were discovered in psychology in the 1950s, and it entered the engineering literature in the 1960s. Woodrow Wilson Bledsoe was the father of face recognition. He developed a system for categorizing facial images. Face recognition has gotten a lot of attention from pattern recognition and machine learning research organizations ever since the early 1990s. Chellappa et al. [5] proposed a few applications of face recognition technology and discussed their benefits and drawbacks in 1995. They did not, however, evaluate any system that could be employed in real-world applications. At least 25 face recognition systems from 13 different companies were available in 1997 [6]. Due to the obvious emergence of face recognition systems and their involvement in real-time situations, they became popular and drew a lot of attention.

The majority of face recognition systems are built around two major modules: feature extraction and classifiers. Today's facial recognition technology compares facial features to multiple data sets using random (feature-based) and photometric (view-based) features via complicated mathematical representations and matching processes. This is accomplished by comparing the structure, shape, and distribution of the face; the distance between the eyes, nose, mouth, and jaw; the outlines of the eye sockets; the sides of the mouth; the alignment of the nose and eyes; and the nearby area around the cheekbones. The most common methods for facial recognition are feature analysis, networks, eigenfaces, and automatic face processing. Face recognition systems have been designed using various feature extraction and classifier algorithms, such as the Geometric based method, which employs tools such as Support Vector Machines [SVM] [7], and the Statistical Approach, which employs tools such as Discrete Cosine Transform [DCT], and Neural Networks such as Convolutional Neural Network [CNN].

The design of a real-time face recognition system using CNN is suggested in this study, followed by an evaluation process by varying the CNN parameters to improve the system's recognition accuracy. An outline of the Convolutional Neural Network is provided in the following section, followed by experimental observations, a conclusion, and references.

2. CNN's Preliminaries

CNN's origins can be traced back to the first multi-layered artificial neural network, Neocognitron [9], proposed by Kunihiko Fukushima in 1979 [8] for some information processing problems. Convolutional neural networks got their start with Yann LeCun and his team's design of LeNet [10]. It gained importance between 1989 and 1998 for the handwritten digit recognition task. CNN is a type of artificial neural network that falls under the category of machine learning and is primarily composed of algorithms influenced by the patterns and operations of neuron structures in the human brain. These structures, known as neural networks, teach computers to think like humans and to see the world from their perspective. Deep learning models include Artificial Neural Networks (ANN), Autoencoders, Recurrent Neural Networks (RNN), and Reinforcement Learning. CNNs, also known as ConvNets, have made large contributions to the development of computer vision and image processing. The networks differ from others because they can accept image, speech, or audio signals as inputs.

CNN is designed primarily for image processing input. More specifically, the architecture is made up of two main blocks. The first block extract features using one of the convolution layers. The second block is the neural network's endpoint and is used for classification. Classes are used to represent the elements of extracted features and then compute probabilities

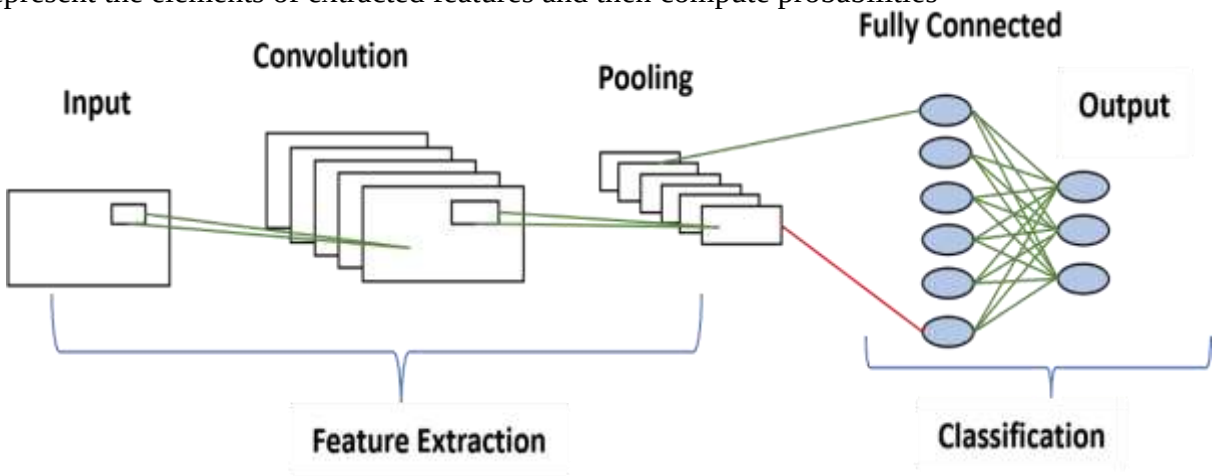


Figure 1: CNN Architecture

2.1. Different layers of the CNN

The structure of a convolutional neural network contains convolutional, pooling, and fully-connected layers.

2.1.1. The convolutional layer:

The very first layer of CNN is the convolutional layer. Its primitive activity is to study the input by extracting different features of these inputs to differentiate one from the other. The layer uses the basic mathematical convolution process between input and a kernel which is sometimes called a filter. By gliding the filter over the input, the dot product is calculated between the filter and the parts of the input.

The obtained result is known as a "feature map," which contains the features and information about the image such as its corners and edges. This feature map is then sent to other layers, which use it to analyze different image features from other layers.

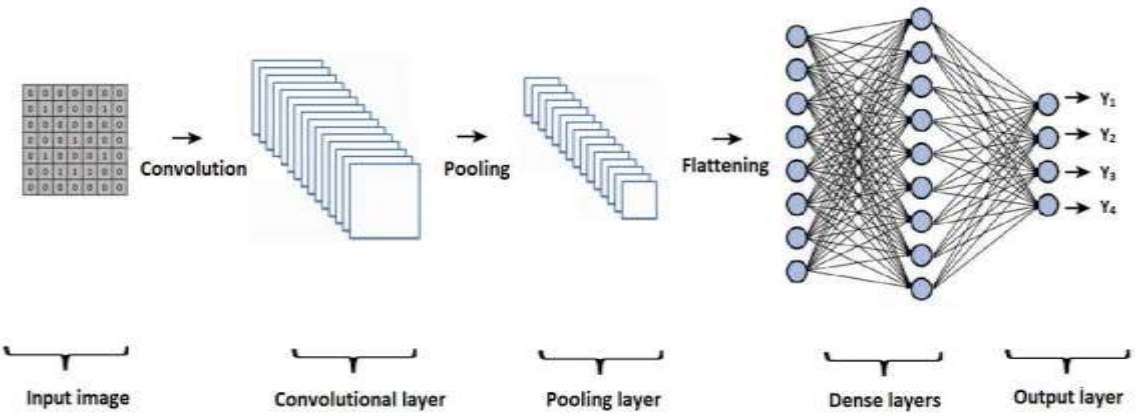


Figure 2: CNN Layers

2.1.2. The Pooling Layer:

A pooling layer follows the convolutional layer. The size of the feature map, obtained by convolution, is reduced in this layer, thereby reducing complexity and computational costs. By minimizing the interconnections between the layers and thereby directly interacting with the feature map. This process is known as pooling and can be of two types, depending on the system used: max pooling and average pooling. When a maximum value from the kernel-covered portion of the input image is returned, it is called max pooling, and when an average value of all the values from the kernel-covered portion of the image is returned, it is said to be average pooling.

2.1.3. A Fully Connected Layer (FC Layer):

Within the fully-connected layer, each node in the output layer is connected to each node in the previous layer. This layer carries out classification tasks using the features extracted by the preceding layers. This layer is also sometimes called "fully connected" or "densely connected" depending on the circumstances. There are all possible connections from layer to layer, indicating every input affects every output. However, not all weights have the same effect on all outputs. The model can be used over several epochs. This layer also uses the SoftMax Classification technique [11], which can differentiate between the important or dominating features and low-level features.

2.2. Pre-Trained Networks

When it comes to selecting a network to deal with a specific problem, pre-trained CNN models stand out. A network's most important characteristics are its accuracy, speed, and size. In general, switching between these functions changes the network selection. Pre-trained models are an important way of testing an organized system [12]. Because of time limitations or computational limitations, it is not always possible to build a model from scratch. This is where pre-trained models come into action. There are several publicly available pre-trained CNN models [13]. In this study, we studied three pre-trained networks: Alex Net, VGG Net, and LeNet-5.

2.2.1. Alex Net

Alex Net was invented in 2012 by Alex Krizhevsky in collaboration with Ilya Sutskever and Geoffrey Hinton to optimize the performance of the ImageNet challenge [14][15]. The Alex Net architecture is comprised of 5 convolutional layers, 3 max-pooling layers, and 2 fully connected layers. Because of the presence of fully connected layers, the input size is fixed at 224x224x3, but because of padding, it appears to be working for 227x227x3. It contains more than 60 million training parameters. AlexNet speeds up the training of larger datasets by distributing the neurons. Alex Net is a powerful model in terms of achieving good accuracy.

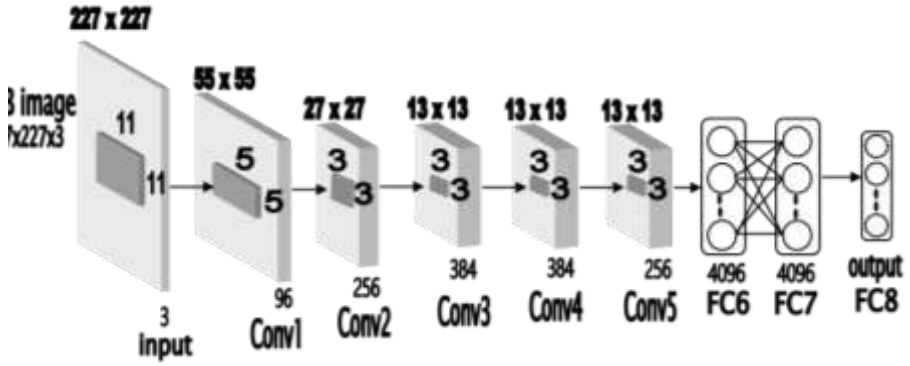


Figure 3: Alex Net Architecture

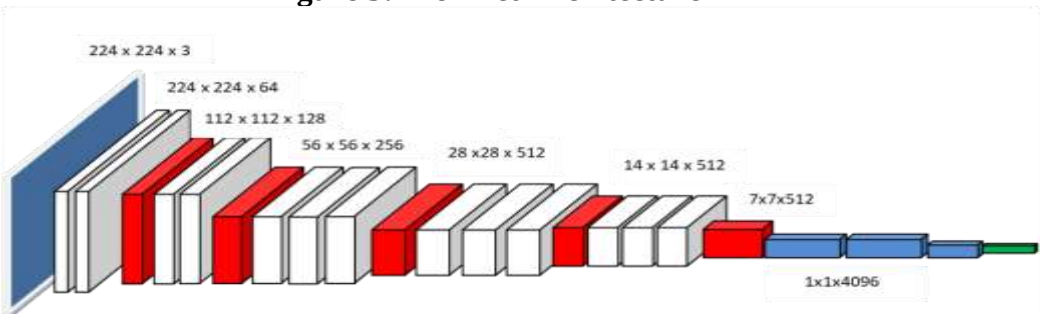


Figure 4: VGG Net Architecture

2.2.3. Le Net

The Le Net architecture is one of the earliest pre-trained models developed by Yann LeCun et al. in 1989 [16]. The architecture of Le Net is very small and simple compared to all other networks. The network has five layers and is hence known as Lenet-5. It consists of three convolution layers and two fully connected layers. The outputs are then made even before being routed to a fully connected layer. The input to this model is a 227x227x3 image. The model has 60,000 trainable parameters. In the United States, it was used a lot to read and sort handwritten characteristics on blank checks.

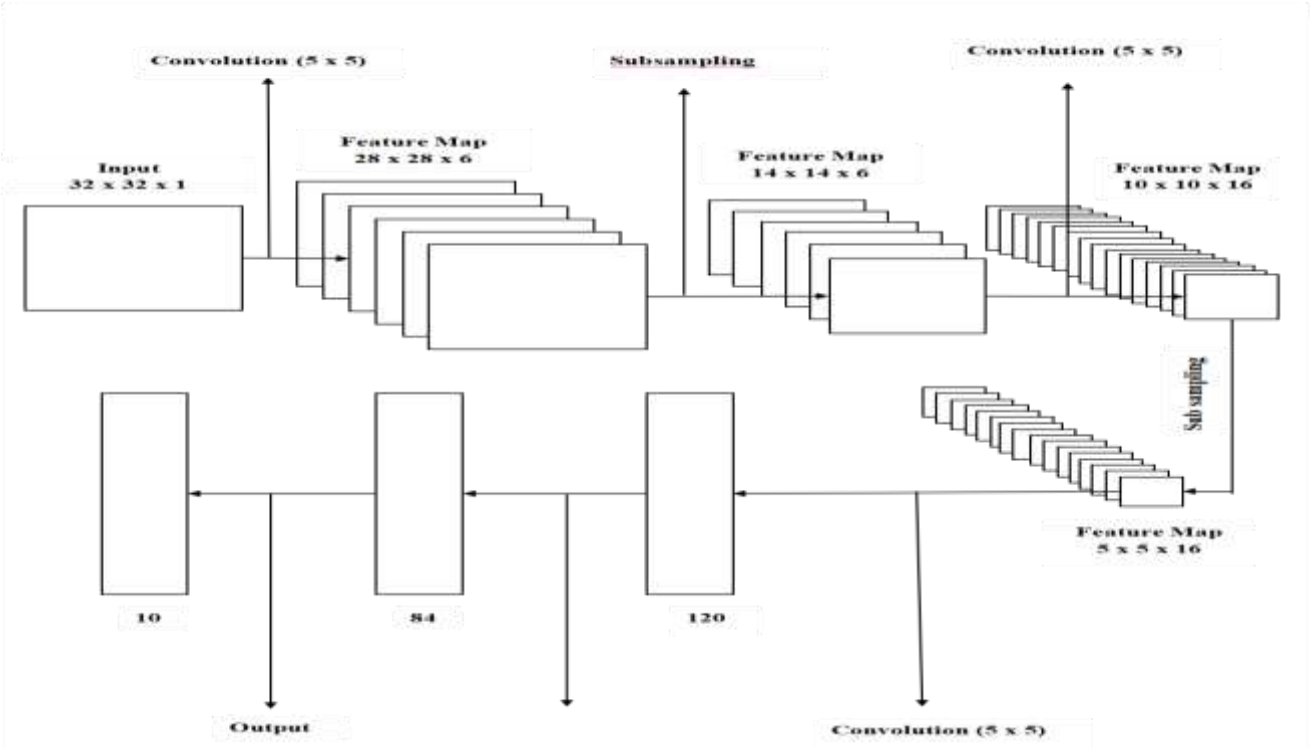


Figure 5: LeNet Architecture

Proposed Model

3.1. Working

The working of the model can be described with the aid of the below flowchart. The proposed model is based on the convolutional neural network methodology. When an image is processed through the neural network, features are extracted to a greater extent to attain accuracy in face recognition. This face recognition proves to be more advantageous than other systems in overcoming the drawbacks of fraudulent usage and other means. To test the model, a database is collected using the system's webcam, which runs automatically based on code. The number of images to be collected for creating a database is done manually from the designed code, which controls the operation of the webcam to turn off when the desired number of images is collected for database formation. The collected database is then trained with the help of these pre-defined neural networks, whose outcome is the extracted features set of input images, which are further classified into various classes. An image from the testing dataset is compared with the existing classified model during the output generation. The test image will be compared with every class of the classified model, and if the features of the test image are similar to any of the class features, then the output will be displayed as the corresponding name of the class (the class name will be taken as the name of the person). In these experimental results, AlexNet, VGGNet, and LeNet were used to create three different convolutional nets, and the accuracy, validation accuracy, and validation loss were measured and compared. This showed that face recognition works.

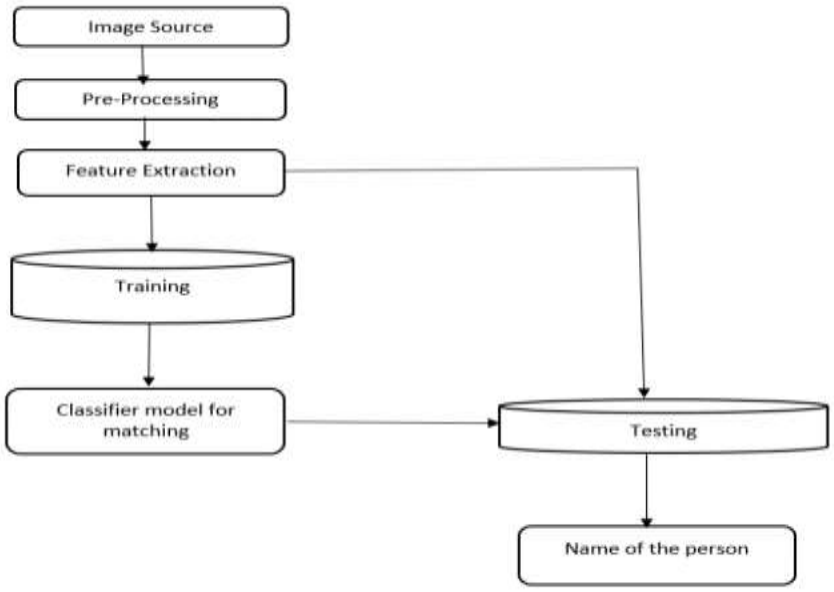


Figure 6: Block diagram

3.2. Methodology

The proposed model is divided into different phases:

- Data collection
- Data pre-processing
- Feature extraction
- CNN training and validation
- Testing

3.2.1. Data collection

Figure 7: Standard dataset

The efficient process of collecting the data is a major factor here. We collected two different datasets, one a standard dataset and the other a real-time dataset. A real-time dataset is collected using the webcam of the system. For the real-time dataset, data were classified into 10 classes, each class consisting of 100, 150, and 200 images. The classes represent the number of people, and in both of the datasets, people of both genders are considered. In the standard dataset, data were classified into 16 classes, with each class consisting of 20 images. The collected datasets are stored in the form of folders in a database, with the folder name being the name of the individual. Figures 7 and 8 show the sample collection of the standard and real-time datasets, which contain 10 classes in each of their datasets.

Figure 8: Real-time dataset

3.2.2. Data pre-processing

The collected dataset has images of sizes ranging from a few KB to a few MB. As each net is predefined, the input size of each net is different, and hence data needs to be pre-processed where the inputs are resized accordingly. AlexNet and VGGNet use 224x224, while LeNet uses 227x227.

Because all of the images in this set are in RGB colors, the feature extraction for comparison will be adequate.

3.2.3. Feature Extraction

The feature extraction network is composed of convolutional and pooling layers in various configurations. The convolutional layer is made up of a collection of digital filters that perform the convolution operation on the input data and transform the image. The pooling layer is used to reduce dimensionality. These features are then classified using a classifier network.

3.2.4. CNN Training and Validation

A training set is a collection of data used to train the model and assist it in learning the hidden features in the data. The neural network is constantly fed with the same training data in each epoch, and the model continues to learn more and more features from the data. In an epoch, the whole dataset is used exactly once. The epoch indicates the number of passes of the entire training dataset the algorithm has trained. If we feed a neural network for more than one epoch in different positions, we can hope for better abstraction when given a new unseen input (test data). The training set should have a variety of inputs so that the model can be trained in all situations and can predict any data sample that was not known before. A validation set is a completely separate set of data from the training set. It is used to evaluate the performance and optimize our model during training. It indicates whether or not the training was done accurately. The model will be trained on the training dataset while the model is tested on the validation dataset after each epoch.

3.2.5. Testing

The test data set is a separate set of data that is used to test the model after it has been trained. It provides objective final model conduct measures such as accuracy, perfection, and so on. Simply put, it answers the question "*How well does the model perform?"* The main objective of testing is to compare outputs from the neural network against testing instances. In the results session, we look at the results of testing our model with three neural networks and look at how accurate they are and how often they make mistakes.

4. Results and Discussion

The system was built using three different neural networks: Alexnet, VGGNet, and LeNet, all of which used the same data set.

AlexNet is trained with 50 epochs, with each epoch having 30 iterations to obtain great accuracy. After 1500 iterations, the validation accuracy for the standard dataset was 33.3%. For the real-time dataset with 10 classes and 100, 150, and 200 images in each class, the validation accuracy was 83.3%, 100%, and 100%, respectively.

VGGNet is trained with 50 epochs, with each epoch having 30 iterations to obtain great accuracy. After 1500 iterations for the standard dataset, a validation accuracy of 6.67% was achieved, and for the real-time dataset with 10 classes and each class having 100 images, 150 images, and 200 images, an accuracy of 16.67%, 33.3%, and 33.3% was acquired, respectively.

LeNet is trained with 50 epochs, with each epoch having 30 iterations to obtain great accuracy. After 1500 iterations for the standard dataset, a validation accuracy of 16.67% was achieved, and for the real-time dataset with 10 classes and each class having 100 images, 150 images, and 200 images, an accuracy of 50%, 66.7%, and 66.7 % was acquired, respectively.

AlexNet has the highest accuracy when compared to other networks.

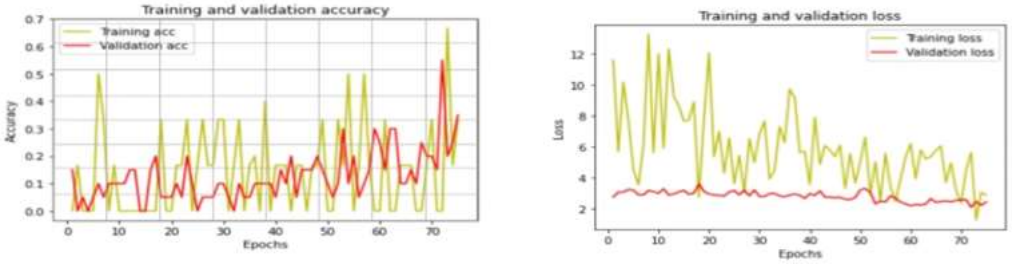
Table 1: For the Standard dataset

Type Of Net	Accuracy	Validation-Loss	Validation-Accuracy
ALEXIE	0.3333	2.424	0.35
VGGNET16	0.0667	2.6684	0.0889
LENET	0.1667	2.7713	0.1

Validation Accuracy: The accuracy one calculates on a data set that isn't used for training but used (during the training process) for validation.

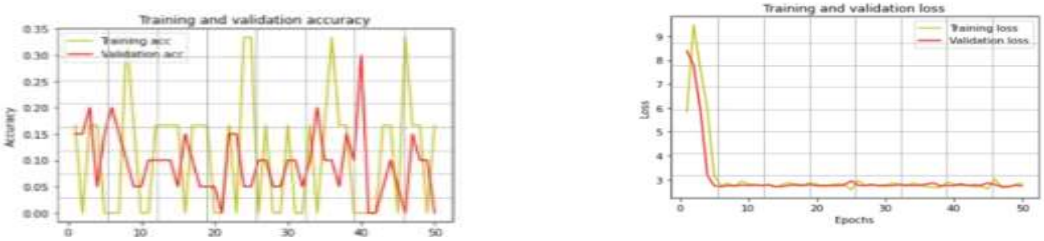
Validation Loss: When data is split into train/validate/test sets, this is the loss calculated on the validation set.

Training and Testing Accuracy: Training accuracy is the accuracy obtained when we apply a model to training data, while testing accuracy is the accuracy obtained when we apply the model to testing data.



(a). Training and validation accuracy (b). Training and validation loss

Figure 9: AlexNet results for the standard dataset



(a). Training and validation accuracy (b). Training and validation loss

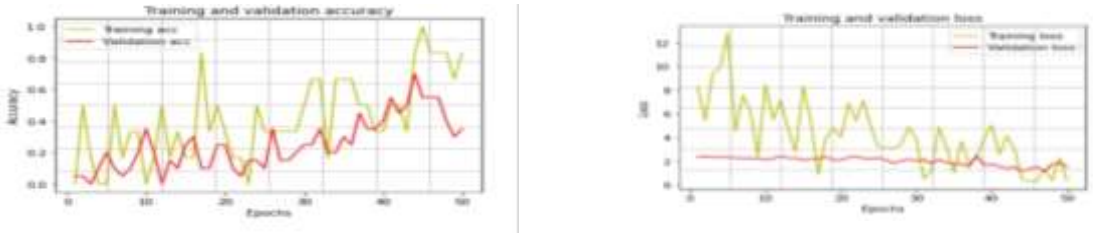
Figure 10: VGGNet results for the standard dataset

(a). Training and validation accuracy (b). Training and validation loss

Figure 11: LeNet results for the standard dataset

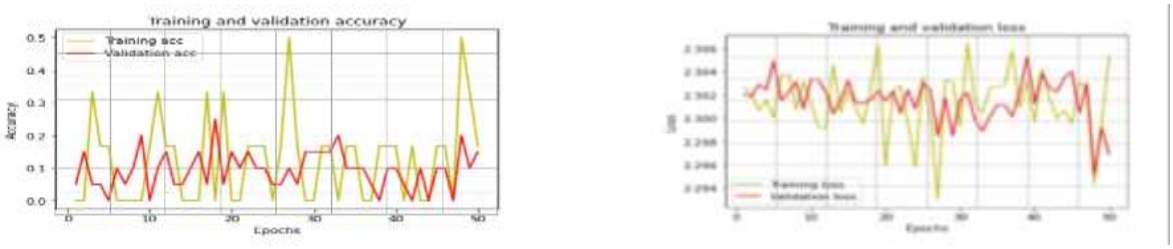
From Figures 9a, 9b, 10a, 10b, 11a, and 11b plots of training and validation accuracy, training and validation results are studied for AlexNet, VGGNet, and LeNet networks respectively.

Table 2: Data size is 100 images per face			
Type Of Net	Accuracy	Validation-Loss	Validation-Accuracy
ALEXNET	0.8333	1.537	0.35
VGGNET16	0.1667	2.2969	0.15
LENET	0.5	0.7717	0.7



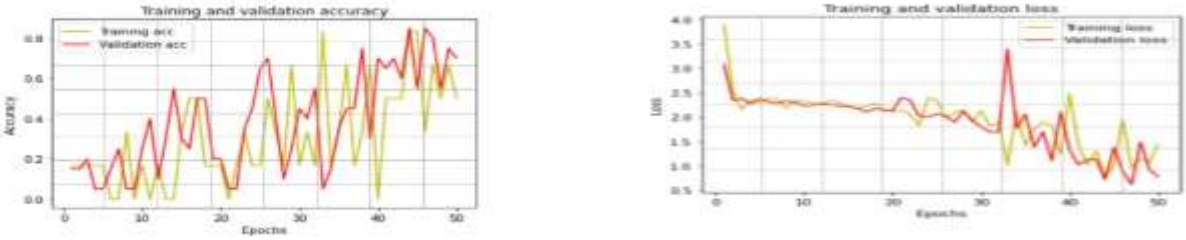
(a). Training and validation accuracy (b). Training and validation loss

Figure 12: AlexNet results for 100 images per face (real-time dataset)



(a). Training and validation accuracy (b). Training and validation loss

Figure 13: VGGNet results for 100 images per face (real-time dataset)

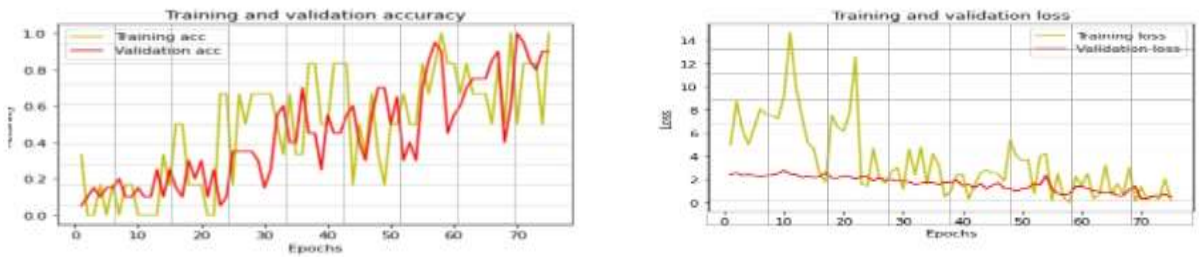


(a). Training and validation accuracy (b). Training and validation loss

Figure 14: LeNet results for 100 images per face (real-time dataset)

From Figures 12a, 12b, 13a, 13b, 14a, and 14b plots of training and validation accuracy, training and validation results are studied for AlexNet, VGGNet, and LeNet networks respectively.

Table 3: Data size is 150 images per face			
Type Of Net	Accuracy	Validation-Loss	Validation-Accuracy
ALEXNET	1	0.4829	0.9
VGGNET16	0.3333	1.1315	0.6
LENET	0.6667	2.3043	0.1

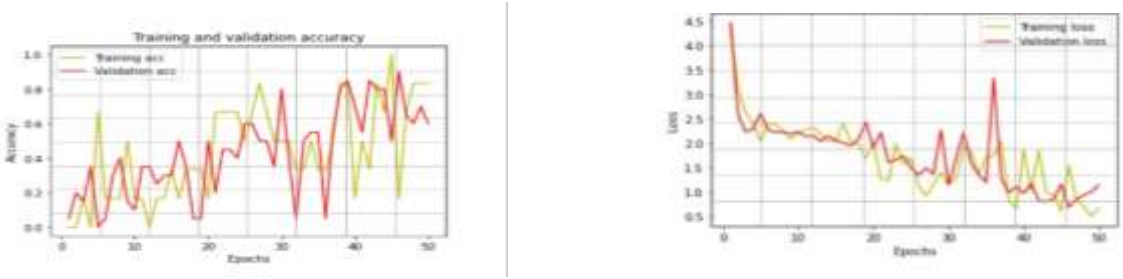


(a). Training and validation accuracy (b). Training and validation loss

Figure 15: AlexNet results for 150 images per face (real-time dataset)

(a). Training and validation accuracy (b). Training and validation loss

Figure 16: VGGNet results for 150 images per face (real-time dataset)



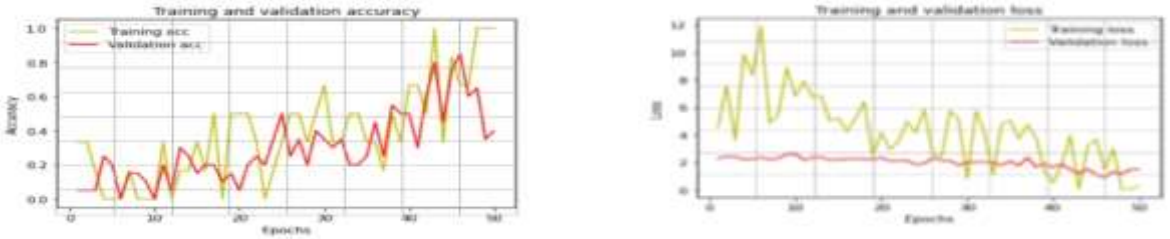
(a). Training and validation accuracy (b). Training and validation loss

Figure 17: LeNet results for 150 images per face (real-time dataset)

From Figures 15a, 15b, 16a, 16b, 17a, and 17b plots of training and validation accuracy, training and validation results are studied for AlexNet, VGGNet, and LeNet networks respectively.

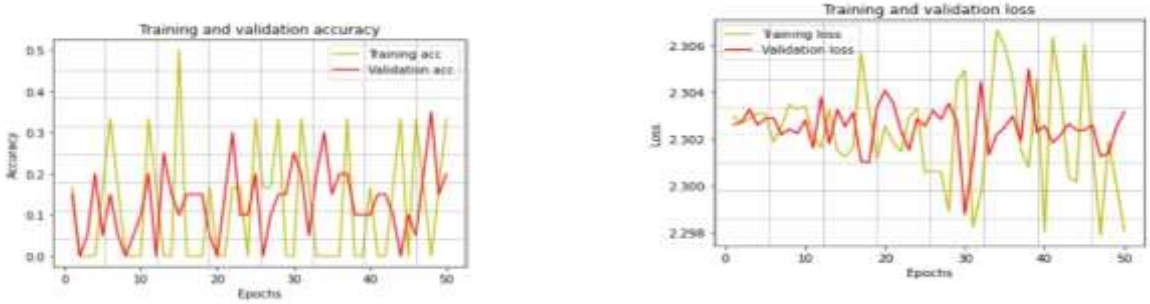
Table 4: Data size is 200 images per face

Type Of Net	Accuracy	Validation-Loss	Validation-Accuracy
ALEXNET	1	1.4958	0.4
VGGNET16	0.3333	2.3032	0.2
LENET	0.6667	0.8975	0.75



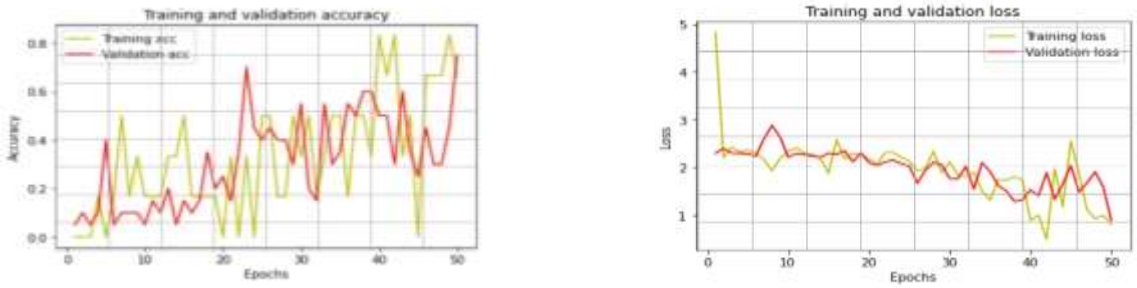
(a). Training and validation accuracy (b). Training and validation loss

Figure 18: AlexNet results for 200 images per face (real-time dataset)



(a). Training and validation accuracy (b). Training and validation loss

Figure 19: VGGNet results for 200 images per face (real-time dataset)



(a). Training and validation accuracy (b). Training and validation loss

Figure 20: LeNet results for 200 images per face (real-time dataset)

From Figures 18a, 18b, 18a, 18b, 19a, and 19b plots of training and validation accuracy, training and validation results are studied for AlexNet, VGGNet, and LeNet networks respectively.

When images were tested, the three CNN models successfully recognized faces with good accuracy. The performances of these three CNN models are compared among different parameters like accuracy, validation loss, and validation accuracy for both standard and real-time datasets. Table 5 illustrates these comparisons among three different CNN models, and it can be observed that Alexnet achieved good accuracy for the standard dataset as well as for real-time datasets. As a result, we can say that AlexNet is the most accurate network, LeNet is the second most accurate, and VGGNet is the least accurate of the three.

Table 5: Comparison table

Types Of Net	Standard Data			Real Data		
	Accuracy	Validation-Loss	Validation-Accuracy	Accuracy	Validation-Loss	Validation-Accuracy
ALEX NET	0.3333	2.242	0.35	1	1.498	0.4
VGG16	0.0667	2.6684	0.0889	0.3333	2.3032	0.2
LENET	0.1667	2.7713	0.1	0.6667	0.8975	0.75

5. Conclusion

The proposed system meets the objective of achieving greater accuracy and less validation loss. We used machine learning by training three pre-trained convolutional neural networks on our data. In terms of accuracy, this model performed the best. The three used networks are AlexNet, LeNet, and VGGNet. The proposed system could be extended for applications like attendance systems, emotion recognition, door access systems, and many other fields comprised of privacy in data.

6. References

1. L. C. De Silva, T. Miyasato, and R. Nakatsu, "Facial emotion recognition using multi-modal information," Proceedings of ICICS, 1997 International Conference on Information, Communications, and Signal Processing.
2. Francis Galton, “Personal identification and description,” In Nature, pp. 173-177, June 21, 1888
3. W. A. Barrett, "A survey of face recognition algorithms and testing results," Conference Record of the ThirtyFirst Asilomar Conference on Signals, Systems and Computers (Cat. No.97CB36136), 1997, pp. 301-305 vol.1, DOI: 10.1109/ACSSC.1997.680208.
4. S. Sharma, M. Bhatt and P. Sharma, "Face Recognition System Using Machine Learning Algorithm," *2020 5th International Conference on Communication and Electronics Systems (ICCES)*, 2020, pp. 1162-1168, doi: 10.1109/ICCES48766.2020.9137850.R. Chellappa, C. L. Wilson and S. Sirohey, "Human and machine recognition of faces: a survey," in Proceedings of the IEEE, vol. 83, no. 5, pp. 705-741, May 1995, DOI: 10.1109/5.381842.

5. C. Bunney. Survey: face recognition systems. Biometric Technology Today, pages 8–12, 1997.
6. D Cherifi, R Kaddari, H Zair, and A Nait Ali. (2019) “Infrared Face Recognition Using Neural Networks and HOG- SVM.” Third International Conference on Bio-engineering for Smart Technologies, Paris, France, 24−26 April, IEEE Press, pp. 1−5.
7. Fukushima, Kunihiko (April 1980). "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position". Biological Cybernetics. **36** (4): 193– 202. doi:10.1007/bf00344251. PMID 7370364. S2CID 206775608.
8. Fukushima, Kunihiko; Miyake, S.; Ito, T. (1983). "Neocognitron: a neural network model for a mechanism of visual pattern recognition". IEEE Transactions on Systems, Man, and Cybernetics. SMC-13 (3): 826– 834. doi:10.1109/TSMC.1983.6313076. S2CID 8235461.
9. LeCun, Y.; Kavukcuoglu, K.; Farabet, C. Convolutional networks, and applications in vision. In Proceedings of the 2010 IEEE International Symposium on Circuits and Systems, Paris, France, 30 May–2 June 2010; pp. 253–256.
10. Andrian Rosebrock. (2016). Softmax classifier explained [Pyimagesearch].
11. P. Marcelino, “Transfer learning from pre-trained models,” Towards Data Science, 2018.
12. A. Gandhi, “Data Augmentation | How to use Deep Learning when you have Limited Data—Part 2,” 2018.
13. Gershgorn, Dave. "The data that transformed AI research—and possibly the world".
14. Krizhevsky, Alex; Sutskever, Ilya; Hinton, Geoffrey E. (2017-05-24).
15. LeCun, Y.; Boser, B.; Denker, J. S.; Henderson, D.; Howard, R. E.; Hubbard, W.; Jackel, L. D. (December 1989). "Backpropagation Applied to Handwritten Zip Code Recognition". *Neural Computation*. **1** (4): 541–551. doi:10.1162/neco.1989.1.4.541. ISSN 0899-7667. S2CID 41312633.