Human face recognition based on convolutional neural network

Report submitted to the SASTRA Deemed to be University

as requirement for the course

INT300 – MINI PROJECT

Submitted by

BHARATHRAJ P

(124015016,B.Tech Information Technology)

HARIVARDHAN V

(124015033,B.Tech Information Technology)

SATHIYA PRIYAN R

(124015154,B.Tech Information Technology)

May 2023

SCHOOL OF COMPUTING

THANJAVUR,TAMILNADU,INDIA-613401

SASTRA

ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION

DEEMED TO BE UNIVERSITY

(U/S 3 of the UGC Act,1956)

THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

SCHOOL OF COMPUTING

THANJAVUR-613401

Bonafide Certificate

This is to certify that the report titled "**Human face recognition based on convolutional neural network"** submitted as a requirement for the course, **INT300 : MINI PROJECT** for B.Tech. is a bonafide record of the work done by **Mr.BHARATHRAJ P (124015016,B.Tech IT),Mr.HARIVARDHAN V (124015033,B.Tech IT), Mr.SATHIYA PRIYAN R (124015154,B.Tech IT)** during the academic year 2022-23, in the School of Computing, under my supervision.

Signature of Project supervisor:

Name with affiliation:

Date:

Mini Project *Viva Voce* held on ______________________________

Examiner 1 **Examiner 2**

Acknowledgements

We would like to thank our Honourable Chancellor **Prof. R. Sethuraman** for providing us with an opportunity and the necessary infrastructure for carrying out this project as a part of our curriculum.

We would like to thank our Honorable Vice Chancellor **Dr. S.Vaidhyasubramaniam**and **Dr. S. Swaminathan,** Dean, Planning & Development, for the encouragement and strategic support at every step of our college life.

We extend our sincere thanks to **Dr. R.Chandramouli**, Registrar, SASTRA Deemed to be University for providing the opportunity to pursue the project.

We extend our heartfelt to thanks to **Dr. A. Umamakeswari,** Dean,School of Computing, **Dr. S.GopalaKrishnan,Associate Dean**,Department of Computer Application,**Dr. B.Santhi**, Associate Dean,Research,**Dr.V.S.Shankar Sriram**,Associate Dean,Department of Computer Science and Engineering,**Dr. R. Muthaiah**, Associate Dean, Department of Information Technology and Information & Communication Technology.

Our guide **Mr.Prabahran L,Asst.Professor,** School of Computing was the driving force behind this whole idea from the start. His deep insight in the field and invaluable suggestions helped us in making progress throughout our project work. We also thank the project review panel members for their valuable comments and insights which made this project better.

We would like to extend our gratitude to all the teaching and non-teaching faculties of the School of Computing who have either directly or indirectly helped us in the completion of the project.

We gratefully acknowledge all the contributions and encouragement from my family, friends resulting in the successful completion of this project. We thank you all for

Giving us an opportunity to showcase our skills through this project.

List of figures

Abbreviations

DNN	Deep Neural Network
CNN	Convolutional Neural Network
AI	Artificial Intelligence
PCA	Principal Component Analysis
SVM	Support Vector Machine
ORL	Olivetti Research Laboratory

Abstract

With the development of deep learning,face recognition technology based on Convolutional Neural Network(CNN) has become the main method adopted in the field of face recognition. In this paper, a novel method integrating convolutional neural network (CNN) with augmented data set is created to address the problem of human face identification on tiny original data set. With several face picture adjustments, the small original data set is expandedinto a larger data set. CNN is designed to automatically and adaptively learn spatial hierarchies of features through backpropagation by using multiple building blocks,such as convolution layers,pooling layers and fully connected layers.Convolution preserves the spatial relationships between pixels by learning image features using small squares of the input image.With the help of the clever CNN, it is possible to successfully extract face features from the enhanced face image collection and increase face recognition accuracy. Many tests and comparisons with other popular facial recognition techniques can demonstrate the efficacy and superiority of the proposed strategy.

KEYWORDS: Facial recognition, Convolutional Neural Network (CNN),augmented data set.

Table of contents

CHAPTER 1

INTRODUCTION

Face Recognition:

Biometrics are an important part of the cutting-edge technology. These biometric systems can add great ease by replacing passwords in security sector, catch criminals in the law sector and many more. One such type is Face Recognition In Face Recognition, we measure the unique features of a person’s face by comparing and analysing the facial contours. It is not only used in law enforcement and security but it is also used to authenticate an identity and unlock devices like smartphones and laptops.

Artificial intelligence (AI) is becoming more and more common in our daily lives as a result of the quick advancement of computer science and technology. Deep neural networks (DNN), among others, have been used to great effect in a number of applications, including speech, picture, and character recognition. Particularly, facial recognition based on DNN has been the subject of extensive research in recent years.Numerous face recognition Experiments were conducted on public face databases,and the findings shown that DNN can successfully extract facial features based on a variety of picture preprocessing techniques. To address the problem of face recognition, an approach integrating principal component analysis (PCA) and support vector machine (SVM) was introduced. PCA may not only lighten the computational load but also increase the recognition accuracy. A new method for recognising human faces using convolutional neural networks and enhanced datasets has been created in response to the discoveries and considerations discussed above.

2. Summary of the base paper:

The paper presents that a small original data set is augmented by operations like rotation,flip,shift and rotation.Based on the augmented face dataset,the face recognition can be effectively implemented via an ingenious convolutional neural network(CNN).

The use of such CNN model will result in higher accuracy thus by effectively recognizing the face in various applications.This paper represents that the augmented data set can introduce abundant sample features,which can enhance the training of network and result in high face recognition accuracy.It also represents that with larger number of training samples,the face recognition accuracy increases.

2.1. Base Paper details:

Title: Human face recognition based on convolutional neural network and augmented dataset.

Year: 2020

Journal Name: Systems Science & Control Engineering

Indexed in: Scopus

MERITS AND DEMERITS OF THE BASE PAPER:

MERITS:

- Increased accuracy: Augmenting the data set helped to improve the accuracy of face recognition using CNNs. By adding more samples to the data set, the model learnt torecognize a wider range of variations in facial features and lighting conditions.

- Better generalization: Augmenting the data set helps the CNN generalize better to newfaces. By including more diverse faces in the training data, the model can learn to recognize common features and patterns in different types of faces, making it more robust to variations.

- Reduced overfitting: Augmenting the data set helps reducing overfitting in the CNNmodel. Overfitting occurs when the model becomes too specialized to the training data and performs poorly on new data. By adding more samples to the data set, the model learnt to generalize better and avoid overfitting.

DEMERITS:

- Computational complexity: Augmenting the data set increased the computationalcomplexity of training the CNN model. This made it more difficult and time- consuming to train the model.

- Data quality: Augmenting the data set can introduce noise or irrelevant features thatmay degrade the performance of the model. Careful selection and pre-processing ofthe augmented data are important to ensure that only high-quality data is used for training.

3. IMPLEMENTATION:

In this paper, a novel approach for human face identification using enhanced data set with convolutional neural network is proposed. The main contribution can be summarized into two steps:

- Data set Augmentation: The modest original data set is expanded into a larger data set by employing several face image adjustments.
- Building the CNN model: Face identification is carried out using an inventive CNN thatis strong in images alterations and is based on the data set for enhanced human faces.

3.1. Data set Augmentation:

Data augmentation is a method employed in machine learning to artificially expanding the size of a data set by creating modified copies of existing data. It is a method for growing adataset's size and variety by transforming already-existing data. By performing various alterations on existing photos, data set augmentation can be utilised in the context of human face recognition to produce more images of faces. A face recognition model's performance and robustness against changes in image quality, lighting conditions, and position may be improved by utilising data set augmentation to expand the amount and variety of the training data set. The operations used for data set augmentation in the paper are as follows:

- Rotation
- Shift
- Scaling
- Flipping

	ROTATION
	FLIP
	SHIFT

By augmenting the data set tremendously with these operations, there will be more number oftraining samples there by introducing abundant samples. Due to this, the face recognition model precision will go up. Now the augmented data set is passed to the CNN model to evaluate the the effectiveness of the suggested model.

3.2.Convolutional Neural Network:

A deep learning neural network called a convolutional neural network (CNN) is frequently employed for image recognition and processing tasks. It is intended to learn spatial feature hierarchies automatically and adaptively from raw image data by leveraging the concept of convolution. A CNN architecture consists of the following layers:

- Input Layer
- Convolutional Layer
- Pooling Layer
- Fully connected Layer
- Output Layer

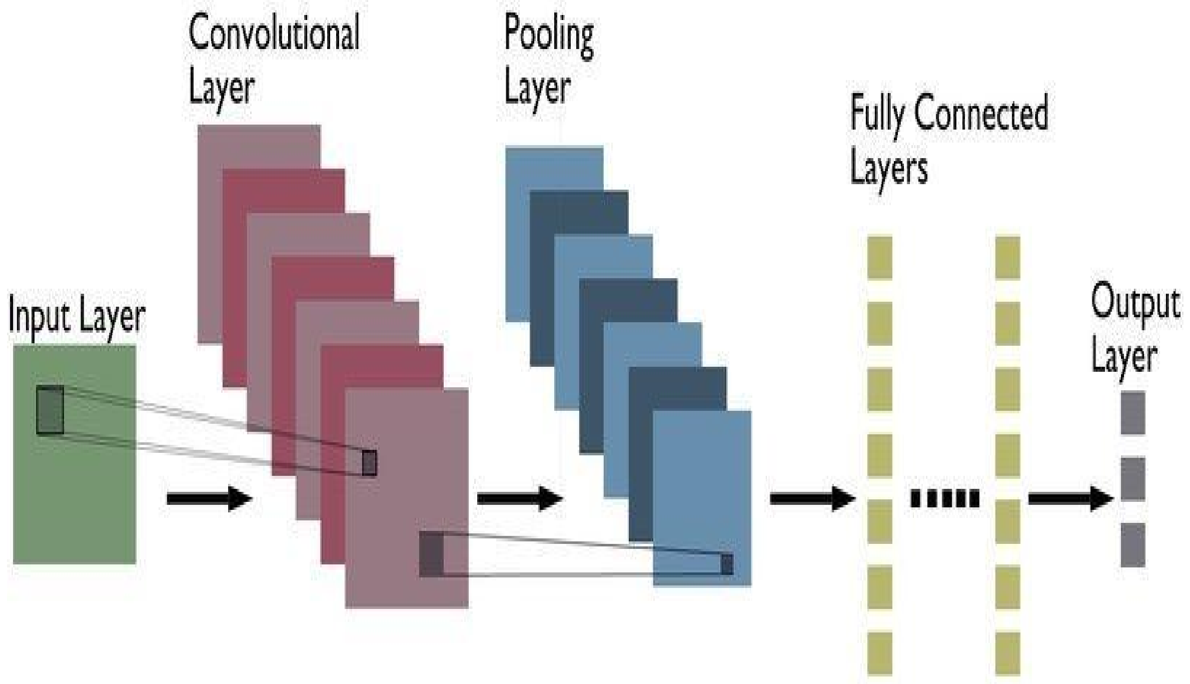


Fig 1.1

INPUT LAYER:

The input layer of a Convolutional Neural Network (CNN) is the first layer of the network that receives the input image. The input layer is in charge of accepting the raw pixel values of the source picture and preparing them for further processing by the network. A group of neurons that individually represent a pixel from the input image make up the input layer in most cases. The size of the input image affects how many neurons are present in the input layer. Typically, pre-processing occurs on the input image before it is sent to the network. Common pre-processing methods for enhancing network performance include data augmentation and normalisation. To avoid the network being extremely sensitive to changes in pixel intensity, normalisation entails scaling the pixel data to have a mean of 0 and a variance of 1. By performing arbitrary modifications to the input images, such as rotations, translations, and zooms, data augmentation includes creating more training instances. This contributes to expanding the amount and diversity of the training data set, which can enhance the network's robustness.

LAYER OF CONVOLUTIONAL:

An interpolation layer is an essential component of a Convolutional Neural Network (CNN). It is intended to pull out features from the image input by applying a group of filters or kernels to the image. The filters in a convolutional layer are typically tiny and have a specific spatial range, such 3x3 or 5x5. By sliding the filters over the input image and conducting a convolution operation at each position, the filters are applied to the image. This process entails multiplying the filter values by the matching input image pixel values, adding the results, and producing a single output value for that particular place. The feature map that results from stacking the output values together depicts how that specific filter responded to each area in the input image. By changing the weights of the filters to minimise the loss function during training, the network discovers the ideal values for each filter. The filters in the network become more adept at identifying specific aspects of the input image, such as edges, corners, and textures, as the network is trained. Each convolutional layer's output is frequently subjected to a nonlinear activation function, such as ReLU, to add nonlinearity to the network and boost its expressive capacity.

POOLING LAYER:

Neural networks with convolutions (CNNs) frequently employ the layer type called a pooling layer to scale down the dimensions of feature maps in space produced by the convolutional layer technology. The pooling layer's main goal is to decrease the amount of computation needed in following layers while simultaneously controlling overfitting. It is often added after one or more convolutional layers. The most common pooling functions are,

- Average pooling
- Maximum pooling

➢ The maximum value of the input values contained in each pooling zone is all that is used to determine the output value in max pooling.

➢ In average pooling, the output value is average values inside the given range.

The fundamental benefit of pooling is that it contributes to lowering the dimensionality of the feature maps, which in turn lowers the quantity of parameters needed in future layers. This lessens the network's computational complexity and aids in avoiding overfitting. Additionally, pooling can help the network be more resilient by lessening its sensitivity to minute changes in the input image.

Pooling layers and convolutional layers are frequently combined in CNNs, with several layers of convolution being then a layer that pools follows. As a result, the input image isrepresented hierarchically, with lower layers detecting more basic elements and higher layersrecognising more intricate ones. The network can capture progressively more complicated patterns in the input image while keeping the computational cost under control by shrinking the spatial parameters with respect to each pooling layer's feature maps.

FULLY CONNECTED LAYER:

A fully connected layer in a Convolutional Neural Network (CNN) is a type of layer that performs the classification task by taking the extracted features from the revolving layers and transforming them into a set of output values, which represent the class probabilities of the input image. The completely interconnected layer is typically placed at the endof the CNN, after the layers of convolution and pooling. Its primary purpose is to classify the input image using the characteristics learned using convolutional layers. The fully linked layer connects every neuron to every other neuron in the previous layer, making it analogousto the dense layers in a conventional neural network. In a CNN, the result of the last pooling layer, which is a one-dimensional array of features, is typically flattened and fed into the layer that is totally connected. The input features are then subjected to a set of learnable weights by the totally interconnected layer, which results in a set of output values that correspond to the input image's class probabilities. Fully connected layers have demonstrated success in a number of image classification applications, including scene categorization, object recognition, and face and object recognition. However, especially with smaller datasets, theirvast number of factors can also render them susceptible to overfitting. Techniques like dropout and regularisation are frequently employed to overcome this issue and stop overfitting in fully connected layers.

OUTPUT LAYER:

A Convolutional Neural Network's (CNN) top layer that generates output predictions for a specific input image is called the final layer. According to the features that the convolutional and complete layer connections of the network have learned, the output layer is intended to present the network's final outputs for classification or regression. Back propagation and gradient descent techniques are used during training to modify the weights in the output adding a layer to cut down the loss function. The error signal is backpropagated through the network to change the weights and enhance the accuracy of the output predictions when loss function compares the expected output with the actual output. Overall, the output layer is a crucial part of a CNN since it generates the network's final output predictions based on the information that the convolutional and totally interconnected layers have learnt. The output layer is configured according to the requirements of the particular task, andits weights are changed during training to reduce the loss function and enhance the network's functionality.

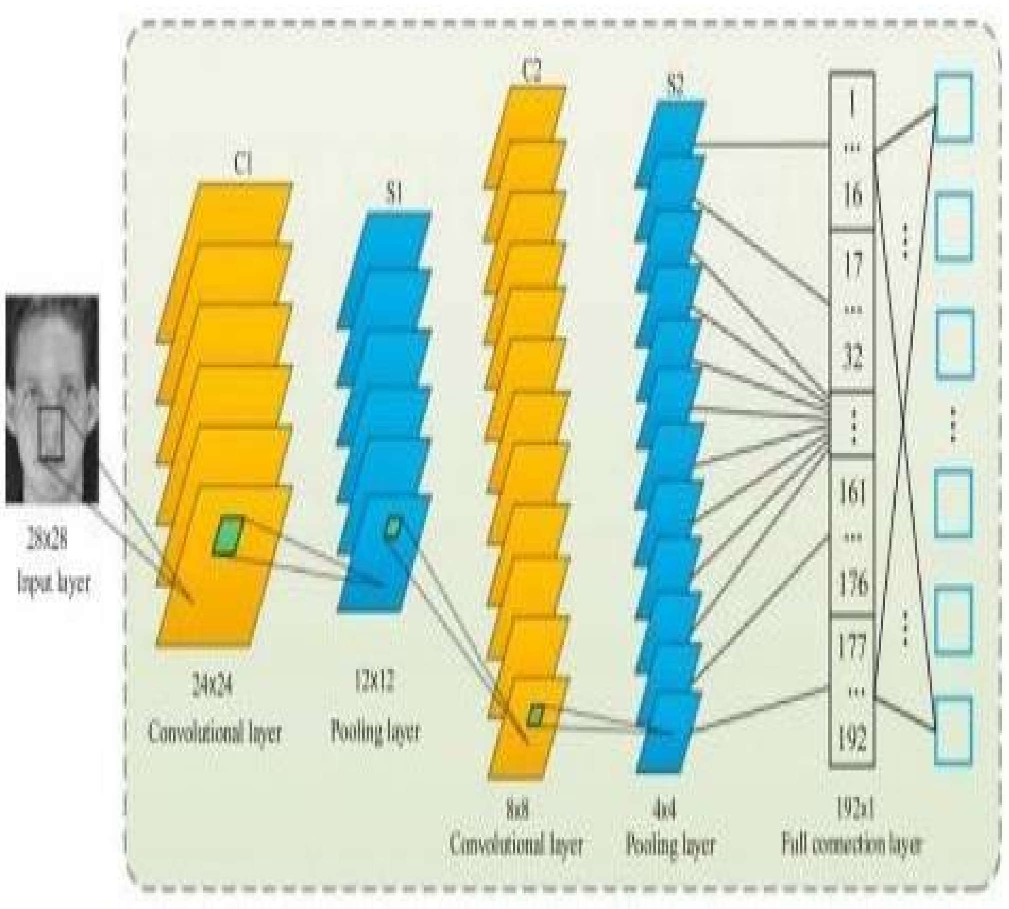


Fig 1.2

WORKFLOW:

A CNN model is created in this study to increase the precision of facial picture classification. The model's structure is comparable to that of the traditional LeNet-5 model, but several of its parameters—including input information, network breadth, and entire connection layers are distinct.

Two convolutional layers(C1 and C2),Two pooling layers, (S1 and S2) make up the newly constructed CNN. As seen in Figure 1.2, the layers placed in thepattern alternately C1-S1-C2-S2.

The input layer has just one feature map, which is used to feed the CNN model with the normalised facial picture.A convolution that is created randomly is applied to each neuron in the kernel of size 5 5 in the first convolutional layer, C1, with six feature maps.The output of the first pooling layer, S1, is six feature maps that were produced using the results of the preceding layer. Each feature map element is linked to the an average convolution kernel of the associated feature map in the C1 layer, preventing overlap between the elements' receptive fields. The second pooling layer (S2), along with the second convolutional layer (C2), respectively, each include 12 feature maps and follow a similar computation process to their forerunners.Between the output layer and S2 layer is also a completely connected single-layer perceptron. Figure 1.2 illustrates the output, which is a 40-dimensional vector to identify 40 different faces and multi-label classification using the sigmoid function.

Then the ORL data set is expanded tremendously by using four augmentation techniques:

- Scaling
- Rotation
- Horizontal flip
- Shift

It goes without saying that a sizable original data set can offer a wealth of visual attributes for the challenge of human face identification. Unfortunately, real applications make it difficult to obtain such a perfect database. The modest picture collection can be massively increased in size by employing the facial image modification. In order to improve the classifier's face recognition performance, more picture features can be extracted.

CHAPTER 2

SOURCE CODE

CODE FOR DATASET AUGMENTATION:

```
from keras.preprocessing.image import ImageDataGenerator

import os

from tensorflow.keras.utils import load_img

from tensorflow.keras.utils import img_to_array

from PIL import Image


original_dataset_path = "G:\\Mini\\samples\\C40"


augmented_dataset_path = "G:\\Mini\\augmented\C40"


n_augmentations = 1000
```

#Image data generator with the desired augmentation parameters

```
datagen =

    ImageDataGenerator(rotation

    _range=20,

    width_shift_range=0.1,

    height_shift_range=0.1,

    zoom_range=0.2,

    horizontal_flip=True,
    fill_mode='nearest')


# Loop through each image in the original dataset and generate n_augmentations new images
for filename in os.listdir(original_dataset_path):
    # Load the original image
    img = load_img(os.path.join(original_dataset_path, filename))
    # Convert the image to a numpy array
    x = img_to_array(img)
    x = x.reshape((1,) + x.shape)
    i = 0
    for batch in datagen.flow(x, batch_size=1, save_to_dir=augmented_dataset_path, save_prefix=filename[:-4], save_format='jpg'):
        i += 1
        if i >= n_augmentations:
            break
```

CODE FOR CNN MODEL:

#Importing necessary libraries

```
import numpy as np

import cv2

from keras.models import Sequential

from keras.layers import Dense, Dropout, Flatten, Conv2D, MaxPooling2D

from keras.preprocessing.image import ImageDataGenerator

import pandas as pd


data_path = 'G:\\Mini\\augmented'

img_height = 70

img_width = 80

num_classes = 40

batch_size =50


train_datagen = ImageDataGenerator(rescale=1./255,

                                   rotation_range=10,

                                   width_shift_range=0.1,

                                   height_shift_range=0.1,

                                   zoom_range=0.1,

                                   horizontal_flip=True,

                                   vertical_flip=False,

                                   fill_mode='nearest')
```

```
test_datagen = ImageDataGenerator(rescale=1./255)
train_generator = train_datagen.flow_from_directory(data_path,
                                                    target_size=(img_height, img_width),
                                                    batch_size=batch_size,
                                                    class_mode='categorical',
                                                    shuffle=True)
test_generator = test_datagen.flow_from_directory(data_path,
                                                  target_size=(img_height, img_width),
                                                  batch_size=batch_size,
                                                  class_mode='categorical',
                                                  shuffle=False)
```

Defining the model:

```
model = Sequential()
model.add(Conv2D(32, (5,5), activation='relu', input_shape=(img_height, img_width,3)))
model.add(MaxPooling2D(pool_size=(2, 2)))


model.add(Conv2D(64, (5,5), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))


model.add(Flatten())
model.add(Dense(num_classes, activation='sigmoid'))
model.summary()
```

Compiling the model:

```
model.compile(loss='categorical_crossentropy',
              optimizer='adam', metrics=['accuracy'])
```

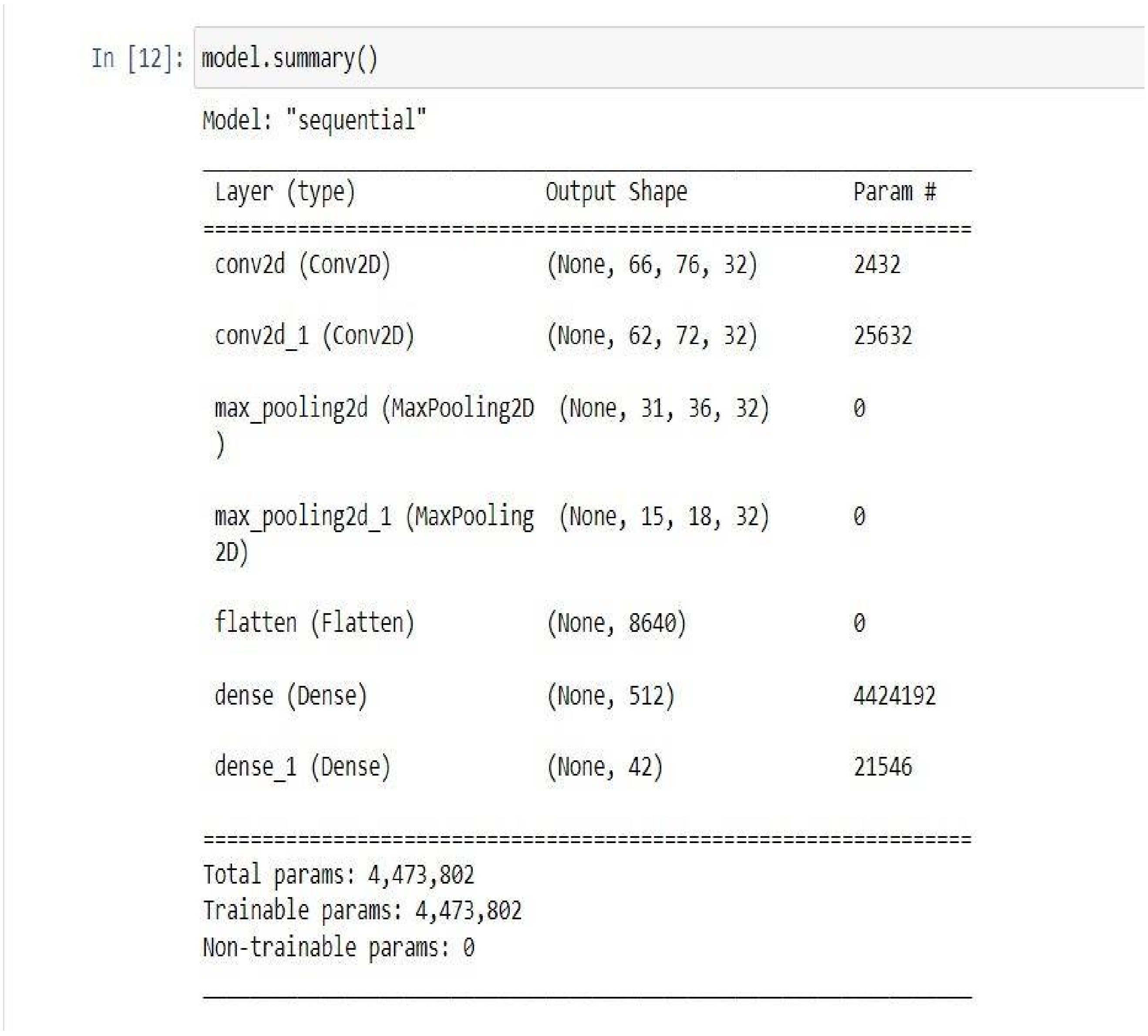
Evaluating the model:

```
epochs = 2
history = model.fit_generator(train_generator,
                              steps_per_epoch=train_generator.samples//batch_size,
                              epochs=epochs)
```

CHAPTER 3

SNAPSHOTS

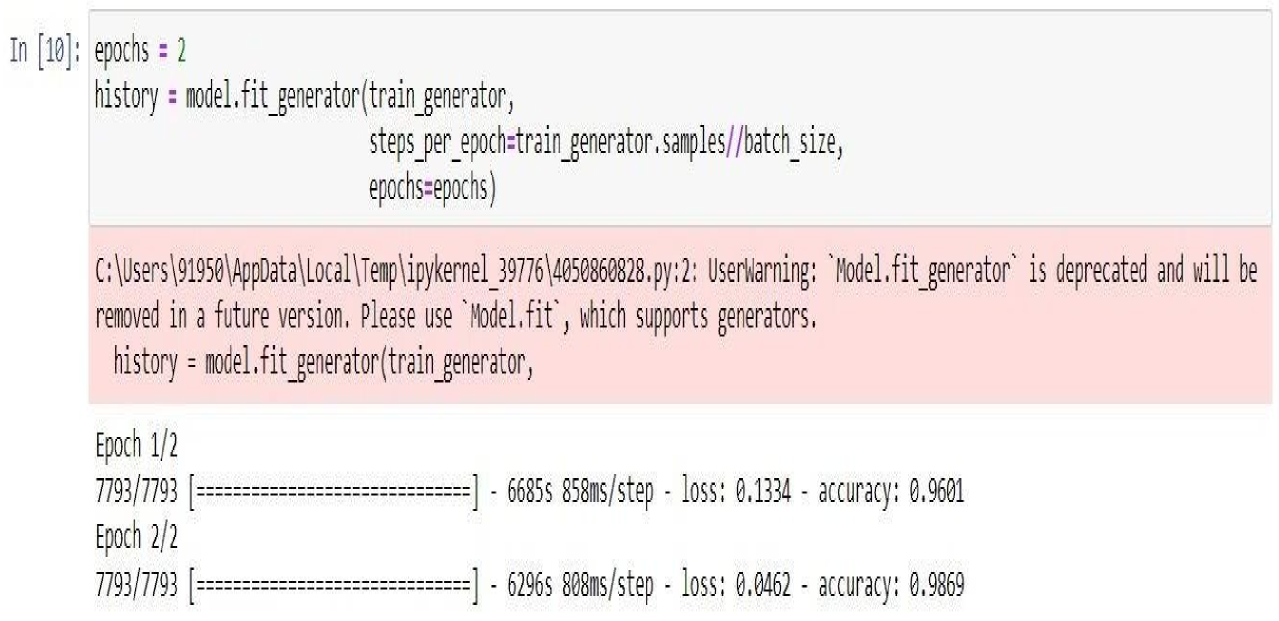
SUMMARY OF THE MODEL SUGGESTED:



```
In [12]: model.summary()

Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 66, 76, 32)        2432

 conv2d_1 (Conv2D)           (None, 62, 72, 32)        25632

 max_pooling2d (MaxPooling2D  (None, 31, 36, 32)       0
 )

 max_pooling2d_1 (MaxPooling  (None, 15, 18, 32)       0
 2D)

 flatten (Flatten)           (None, 8640)              0

 dense (Dense)               (None, 512)               4424192

 dense_1 (Dense)             (None, 42)                21546

=================================================================
Total params: 4,473,802
Trainable params: 4,473,802
Non-trainable params: 0
_________________________________________________________________
```

Fig 3.1

ACCURACY OF THE PROPOSED MODEL:



```
In [10]: epochs = 2
         history = model.fit_generator(train_generator,
                                       steps_per_epoch=train_generator.samples//batch_size,
                                       epochs=epochs)
```

```
C:\Users\91950\AppData\Local\Temp\ipykernel_39776\4050860828.py:2: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which supports generators.
  history = model.fit_generator(train_generator,
```

```
Epoch 1/2
7793/7793 [==============================] - 6685s 858ms/step - loss: 0.1334 - accuracy: 0.9601
Epoch 2/2
7793/7793 [==============================] - 6296s 808ms/step - loss: 0.0462 - accuracy: 0.9869
```

Fig 3.2

CONCLUSION AND FUTURE PLANS

In this research, a novel method for handling the challenge of identification of human faces on a limited first data set is presented. Through the use of face image adjustments like rotation, scaling, shift, and flip the initial tiny data set is expanded into a large data set.

Face recognition can be successfully applied using a clever CNN according to the impressively data set of enhanced faces. The effectiveness of the supplemented data set is tested in a number of trials, and the new approach's superiority can be demonstrated when compared to several of the most popular face identification techniques.

In the fact, the suggested approach is a practical way to increase the data set that may be used ina number of contexts for data-based training and learning. The use of the data augmentation method to bit more complicated problems, such as processing of signals, picture identification, and detecting faults in images, will be the main goal of our future study.

REFERENCES

- Lin, G., & Shen, W. (2018). Research on convolutional neural network based on improved Relu piecewise activation function. Procedia Computer Science,
- Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., & Alsaadi, F. E. (2016). A survey of deep neural network architectures and their applications.Neurocomputing,
- Ptucha, R., Petroski Such, F., Pilla, S., Brockler, F., Singh, V., & Hutkowski, P. (2019). Intelligent character recognition using fully convolutional neural networks. Pattern Recognition,
- Ranganathan, S., Gribskov, M., Nakai, K., & Schönbach, C. (2019). Delta rule and backpropagation. Encyclopedia of Bioinformatics and Computational Biology,
- Zhao, F., Li, J., Zhang, L., Li, Z., & Na, S. G. (2020). Multi-view face recognition using deep neural networks. Future Generation Computer Systems,