FACE EMOTION RECOGNITION SYSTEM

A project report submitted in the partial fulfillment of the requirements for the

Award of the degree of

BACHELOR OF TECHNOLOGY

In

CSBS

Submitted By

Rich Gedeon

Reg.no: VU21CSEN0200123

Under the Guidance of Mr. Umesh

22 June 2024

DECLARATION BY THE CANDIDATE

I the undersigned solemnly declare that the project report FACE EMOTION RECOGNITION SYSTEM is based on my own work carried out during the course of our study under the supervision of Mr. Umesh. I assert the statements made and conclusions drawn are an outcome of my research work. I further certify that

I. The work contained in the report is original and has been done by me under the general supervision of my supervisor.
II. The work has not been submitted to any other institution for any other degree/diploma/certificate in this university or any other University of India or abroad.
III. We have followed the guidelines provided by the university in writing the report.
IV. Whenever we have used materials (data, theoretical analysis, and text) from other sources, we have given due credit to them in the text of the report and giving their details in the references.

Rich Gedeon

(College Roll No: - VU21CSEN0200123)

(Phoenix ID No: PG2425ETSD535)

ACCEPTANCE/OFFER LETTER

Date: 07-05-2024

Name: Rich Gedeon
Designation: Junior Engineer
Domain: ML - ML with Python
Mode: Hybrid
Phoenix ID: PG2425ETSD535

Dear Rich Gedeon,

PhoenixID:PG2425ETSD535

We are pleased to extend an

invitation for you to join the

Phoenix Global Team!

Phoenix Global is a skill-development company that helps students acquire and master professional and soft skills as per the requirements of the industry benchmarked to world's top firms, trained by top class industry professionals. Phoenix Global is a platform having Industry professionals with esteemed alma mater including the IITs and IIMs to mentor and train students on cutting-edge skills, critical to the emerging industries while also giving them an opportunityto intern on a project under the mentorship of industry professionals from the IITs /IIMs.

Our vision is to be a national leader in skill development and industry readiness training by providing differentiated training from top-class industry experts. The mission is to be a go-to skill development platform for students, imparting skills benchmarked at global standards that help them realize their dream careers profitably

Our core values, the 4Ps – Professionalism, Punctuality, Passion, Perseverance stand for who and what we are as an organization. We are pleased to formalize your relationship with Phoenix Global as a Summer Internship Trainee, details of which are as follows:

General information

Role : Junior Engineer
Department : ML - ML with Python
Location / Mode : Hyderabad / Hybrid

Period : 11th May 2024 – 26th June 2024 (1.5 Months)
Date of Joining : 11-05-2024

1. Appointment: Your date of appointment is effective from the date of joining

2. Benefits:

a. You will be entitled to leave, holidays, benefits, and other allowances as applicable to your category of employees and location of posting, in accordance with the rules of the Company. As an Intern, you are entitled to 2(two) leaves per months allotted on pro-rata basis (these do not include public holidays).

b. You shall receive Internship Completion Certificate, Letter of Recommendation, Verified project documentation report post completion of the internship.

3. Code of Conduct:

a. The Company may require you, at any time, to perform any other administrative, managerial, supervisory, technical or other functions and you will be bound to carry out such functions.

b. You shall maintain proper discipline and dignity of your office/location and so shall deal with all matters.

c. You shall maintain and keep in your safe custody such as Measuring instruments, Safety Equipment and other assets that may be issued to you or may come in your possession and shall return the same when required in good condition.

d. You shall inform the Company of any changes in your personal data within 3 days of the occurrence of such change.

e. Any notice required to be given to you shall be deemed to have been duly and properly given if delivered to you personally or sent by post to you at your address, as recorded in the Company.

f. You shall be solely responsible for any issues that may arise between you and your previous employer with regard to your previous employment and the Company /any of its personnel are not responsible for the same.

g. You shall not apply for any other job outside without the prior written permission from the Management. In response to this communication of appointment you are required to confirm your acceptance by signing the duplicate copy of this order.

If it is found at any time that the information given by you is not correct/true/complete, this appointment may be withdrawn or may be terminated at any time after you have taken up employment with us. Please note that you are governed by all Rules and Regulations of the Company, which are in force from time to time, and the Company shall have the right from time to time to vary or modify any of the terms and conditions of service, which shall be binding on you.

We take pleasure in welcoming you to our organization and look forward to a mutually beneficial association.

We wish you all the best in your career.

Sincerely.

Harsha Y

Program Director and Head - HR

In the name and seal of

PHOENIX GLOBAL

www.phoenix-global.co.in

Signed and Accepted

Employee Name: Rich Gedeon

Date: 07-05-2024

CERTIFICATE

This is to certify that this project work entitled

"FACE EMOTION RECOGNITION SYSTEM"

is the bonafide work carried out by Rich Gedeon, Reg. No: VU21CSEN0200123 submitted in Partial fulfillment of the requirement for the Award of Degree of Bachelor of Technology in Computer Science Engineering, during May-June 2024.

The results submitted in this project have been verified and are found to be satisfactory. The results embodied in this thesis have not been submitted to any other university for the award of the any other degree/diploma.

Signature of project supervisor

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose constant guidance and encouragement crowned the efforts with success. It is a pleasant aspect that I have now the opportunity to express my gratitude for all of them.

The first person I would like to thank my project guide Mr. Umesh, who had given continuous critical suggestions and extension of proper working atmosphere, abiding interest has finally evolved into this research work.

I am also thankful to all the staff members of the Computer Science Engineering Department for their valuable suggestions. I would like to thank my team mates and parents who extended their help, encouragement and moral support either directly or indirectly in this project.

RICH GEDEON

(VU21CSEN0200123)

CONTENTS

ABSTRACT

This project investigates the application of machine learning for facial emotion recognition. A Convolutional Neural Network (CNN) is implemented to classify emotions displayed in facial expressions into seven categories: Angry, Disgust, Fear, Happy, Neutral, Sad, and Surprise. The system utilizes OpenCV for real-time video capture and face detection.

The report details the inner workings of machine learning, including supervised learning and convolutional neural networks. It then delves into the project by explaining the utilized packages, implementation details, and the code structure of the three Python scripts: test.py, testdata.py, and main.py. Test.py demonstrates real-time emotion recognition on video input, while testdata.py performs emotion recognition on a single image. Finally, main.py showcases the model creation process, including data augmentation and training.

The CNN model is created with Keras incorporating layers for extracting features max pooling layers, for reducing dimensionality dropout layers to prevent overfitting and dense layers for classification. It undergoes training on a dataset containing labelled images and validation on a test dataset.

Real time emotion recognition is accomplished by integrating OpenCV for video capture and face detection. The system processes video frames from a webcam identifies faces using Haar cascade classifiers and predicts emotions using the trained CNN model. These predictions are then displayed on the video frames along with bounding boxes and confidence scores.

The report wraps up by showcasing the project code, which includes scripts, for training the model conducting tests and enabling real time emotion recognition. The systems efficacy is demonstrated through its capability to accurately classify emotions.

ABOUT

Phoenix Global is a skill-development company that helps students acquire and master professional and soft skills as per the requirements of the industry benchmarked to world's top firms, trained by top class industry professionals.

Phoenix Global is a platform having Industry professionals with esteemed alma mater including the IITs and IIMs to mentor and train students on cutting-edge skills, critical to the emerging industries while also giving them an opportunity to intern on a project under the mentorship of industry professionals from the IITs /IIMs.

Our vision is to be a national leader in skill development and industry readiness training by providing differentiated training from top-class industry experts. The mission is to be a go-to skill development platform for students, imparting skills benchmarked at global standards that help them realize their dream careers profitably

Our core values, the 4Ps – Professionalism, Punctuality, Passion, Perseverance stand for who and what we are as an organization.

SCHEDULE OF INTERNSHIP

Day	Activity Plan
1	Induction Program
2	Pre-Readings/Material Distribution
3	Training Session - 1
4	Training Session - 2
5	Training Session - 3
6	Training Session - 4
7	Training Session - 5
8	Teams formation for Project
9	Weekend Off
10	Training Session - 6
11	Training Session - 7
12	Training Session - 8
13	Training Session - 9
14	Training Session - 10
15	Project Title Allocation
16	Weekend Off
17	Project Session - 1
18	Project Session - 2
19	Project Session - 3
20	Project Session - 4
21	Project Session - 5
22	Project Mid Review
23	Weekend Off
24	Project Session - 6
25	Project Session - 7
26	Project Session - 8
27	Project Session - 9
28	Project Session - 10
29-44	Project Working Sessions
45	Project Final Presentation and Thesis Defence

CHAPTER 1 INTRODUCTION

What is Machine learning?

Machine learning is a branch of artificial intelligence (AI) that focuses on teaching computers to learn from data and make decisions or predictions based on that data. Instead of being explicitly programmed to perform specific tasks, a machine learning model improves its performance by learning from examples. To explain machine learning in simpler terms, imagine you have a robot, and you want this robot to be good at something, like recognizing your friends in photos. Instead of telling the robot exactly what your friends look like (which is hard to do because everyone looks a bit different in each photo), you show it lots of pictures of your friends and letit figure out the patterns on its own. This process of learning from lots of examples to get better at a task is what we call machine learning.

Steps involved in Machine Learning:

Learning from Data: Machine learning entails providing a computer with amounts of data and utilizing algorithms (special rules and calculations) to help the computer recognize patterns or connections, within the data.

Algorithms: These mathematical models and processes are utilized by the computer to learn from the data. Various algorithms are suitable for types of tasks such, as classification (e.g., identifying spam emails) regression (e.g., forecasting house prices) and clustering (e.g., groupingitems).

Testing: The data is typically split into two parts; a training set and a testing set. The training set is used to educate the model while the testing set is employed to assess how well the model has learned. This ensures that the model can apply its knowledge to data.

Making Predictions: Once trained the model can generate predictions or decisions based on data.

For instance, a trained system could forecast the weather recommend products you may enjoy from your purchases or identify faces, in pictures.

Types of Machine Learning:

1.Supervised Learning:

The model is trained on labelled data, meaning the input data is paired with the correct output. The goal is for the model to learn to predict the output from new input data.

Examples of Algorithms: Linear regression, logistic regression, decision trees, support vector machines (SVM), and neural networks.

Applications:

- **Spam detection**: Classifying emails as spam or not spam.
- **Image classification**: Recognizing objects in images (e.g., cats vs. dogs).
- **Medical diagnosis**: Predicting disease based on patient data.

2. Unsupervised Learning:

The model is given data without labelled responses and must find patterns and relationships in the data on its own.

Examples of Algorithms: K-means clustering, hierarchical clustering, principal component analysis (PCA), and anomaly detection.

Applications:

- **Customer segmentation**: Grouping customers based on purchasing behaviour.
- **Market basket analysis**: Finding associations between products bought together.
- **Anomaly detection**: Identifying unusual data points in datasets (e.g., fraud detection).

Types of Neural Networks:

1. **Feedforward Neural Networks**: The simplest type, where connections do not form cycles.
2. **Convolutional Neural Networks (CNNs)**: Used primarily for image and video recognition tasks. They employ convolutional layers that automatically and adaptively learn spatial hierarchies of features.
3. **Recurrent Neural Networks (RNNs)**: Designed for sequential data, such as time series or natural language. They have connections that form directed cycles, allowing them to maintain a memory of previous inputs.
4. **Long Short-Term Memory Networks (LSTMs)**: A type of RNN that can learn long-term dependencies and handle vanishing gradient problems.

5. **Generative Adversarial Networks (GANs)**: Consist of two networks, a generator, and a discriminator, that compete against each other to create realistic synthetic data.

What is CNN?

Convolutional Neural Networks, also known as CNNs stand out as a form of learning neural network specifically adept, at handling tasks related to images and videos. Drawing inspiration from the workings of the visual cortex CNNs mirror how the brain processes visual data in a step-by-step manner. They start by identifying features like edges and lines before piecing them together to recognize more intricate shapes and objects. This layered approach in CNNs enables them to extract features from data, coupled with their design making them a popular choice, for a wide range of applications that involve analyzing visual content.

Key Components of a CNN:

- **Convolutional Layers:** These play an important role, in CNNs. They utilize filters, known as kernels that move across the input image. The filter adjusts weights to emphasize characteristics in the image, such as edges or textures. As the filter moves it generates a feature map that highlights these characteristics. Multiple filters can be employed to identify features.
- **Pooling Layers:** These layers decrease the size of feature maps by using techniques like max pooling, which retains the value from a pixel grid. This reduces data volume. Allows the network to concentrate on features while making it robust against minor shifts, in the image.
- **Activation Layers:** These layers bring in a sense of non-linearity, to the network.Popular activation functions such, as ReLU (Rectified Linear Unit) incorporate a threshold effectively deactivating neurons and enabling the network to grasp intricate patterns.
- **Fully Connected Layers:** In the phases of the CNN the connected layer's function much like regular neural networks. They receive the condensed data from the layers. Work on it to produce the ultimate result like determining whether an image depicts a "cat" or a "dog", in a classification task.

Types of layers:

In constructing machine learning models' different layers are incorporated, with the types of layers chosen based on the model's nature and the objective, at hand. Here is a breakdown of some common layer types:

Basic Layers:

- **Dense Layer (Fully Connected Layer):** This layer is, like the backbone in a lot of models. It receives an input vector multiplies it with a weight matrix. Then adds a bias vector. After that an activation function is applied to each element of the outcome. Dense layers are employed for making changes and understanding connections, among features.

Layers for Convolutional Neural Networks (CNNs):

- **Convolutional Layer:** The layers use filters (kernels) that move across the input typically images. The filters are trained to identify characteristics in the data.
- **Pooling Layers:** The subsequent layers compress the information, from layers cutting down on expenses and enhancing the model's resilience, to minor changes. Popular pooling methods comprise max pooling and average pooling.

Layers for Recurrent Neural Networks (RNNs):

- **Recurrent Layer (e.g., LSTM, GRU):** These specialized layers are created to work with data that follows a sequence, such, as text or time series. Their built-in memory enables them to analyse and utilize information from preceding stages within the sequence.

Other Layers:

- **Activation Layers:** These layers bring in a sense of non-linearity, to the network.Popular activation functions such, as ReLU (Rectified Linear Unit) incorporate a threshold enabling the network to grasp patterns.
- **Pooling Layers (General):** Pooling layers are versatile. Can be applied to types of data not limited to just images. They help decrease the complexity of the data while retaining characteristics.
- **Normalization Layers:** The purpose of these layers is to standardize the neuron activations within a layer typically done to enhance the training process.
- **Dropout Layers:** During training these layers randomly deactivate a portion of neurons. This technique is used to avoid overfitting and enhance the model's ability to generalize.
- **Output Layer:** The last component of the system that produces the result, like a likelihood of classification or a numerical value.

The selection of layers and how they are set up play a role, in shaping the abilities of a machine learning model. It is essential to grasp the types of layers to construct and comprehend machine learning models.

Image Processing:

Image processing used to require people to identify and extract features from images manually for tasks, like object recognition. Machine learning has made this process much simpler. Convolutional Neural Networks (CNNs) a type of learning model excels at extracting features. By being trained on datasets of labelled images CNNs can. Recognize important features automatically. This enables the model to identify patterns and objects when there are changes, in lighting pose or background.

Beyond Basic Image Manipulations:

Image processing, in the past primarily dealt with activities such, as filtering, reducing noise and sharpening images. However, machine learning has the potential to elevate these functions to a level of capability. Machine learning models can now perform complex tasks like:

- **Object Detection and Recognition:** Detecting and pinpointing items, in a picture when they are partially hidden or in surroundings.
- **Image Segmentation:** Distinguish objects from the surroundings. Categorize areas, within a picture. This is essential for tasks such, as analysing images or enhancing the perception of self-driving cars when objects are partially hidden or located in busysettings.
- **Image Captioning:** Generate captions automatically that explain the content of a picture
- **Image Generation:** Generate images, from the ground up. Adjust current visuals to meet requirements.

Generalization for Real-World Applications:

Machine learning algorithms have the capability to learn from datasets enabling them to apply their knowledge to images effectively. This feature is crucial, for uses where images exhibit characteristics. For example, a facial recognition technology powered by machine learning can identify faces under varying lighting conditions, poses. When there are obstructions, like glasses or masks.

Continuous Improvement and Automation:

Continuous enhancements of machine learning models are achieved by providing them with data. This enables them to adjust to evolving situations and enhance their efficiency gradually. Furthermore, ML streamlines image processing assignments resulting in more effective outcomes, with reduced chances of human mistakes.

CHAPTER 2 Face Emotion Recognition

The Face Emotion Recognition project is designed to classify emotions displayed in facial expressions into one of seven categories: Angry, Disgust, Fear, Happy, Neutral, Sad, and Surprise. The system uses a Convolutional Neural Network (CNN) for classification and OpenCV for real-time video capture and face detection.

Packages Used

1. Keras: Keras is a high-level neural networks API, written in Python, and capable of running on top of TensorFlow. It simplifies the process of building and training deep learning models.

keras.preprocessing.image: Provides tools for image data augmentation and preprocessing.

keras.models.Sequential: A linear stack of layers that allows the creation of a neural network model layer-by-layer.

keras.layers: A module containing various neural network layers such as Dense, Dropout, Flatten, Conv2D, and MaxPooling2D.

2. OpenCV (cv2): OpenCV is an open-source computer vision library that provides tools for real-time image and video processing.

cv2.CascadeClassifier: Loads pre-trained Haar cascade classifiers for face detection.

cv2.VideoCapture: Captures video from a camera or video file.

cv2.cvtColor: Converts images between different color spaces, such as from BGR to grayscale.

cv2.resize: Resizes images to the specified dimensions.

cv2.rectangle: Draws rectangles around detected faces.

cv2.putText: Places text on images or video frames.

3. NumPy (np): NumPy is a library for numerical computations in Python. It provides support for arrays and matrices, along with a collection of mathematical functions to operate on these data structures.

4. OS (os): The os module provides functions for interacting with the operating system, such as reading file paths and counting files in a directory.

Implementation Details:

Image Data Augmentation: ImageDataGenerator is used to apply random transformations to the training images (rescaling, rotation, shear, zoom, horizontal flip) to improve the generalization of the model.

Data Loading: The flow_from_directory method loads images from the specified directories, applies the augmentations, and prepares batches for training and validation.

Model Architecture: A Sequential CNN model is created with several convolutional layers for feature extraction, followed by max-pooling layers to reduce the spatial dimensions. Dropout layers are added to prevent overfitting by randomly setting a fraction of input units to zero during training. A Flatten layer converts the 2D matrix into a 1D vector, which is fed into a fully connected (Dense) layer. The final Dense layer uses the softmax activation function to output probabilities for each of the seven emotion classes.

Model Compilation and Training: The model is compiled using the Adam optimizer and categorical crossentropy loss function. The model is trained on the augmented training data and validated on the validation data for a specified number of epochs.

Model Saving: The trained model is saved to a file for later use in emotion detection.

Model Loading: The pre-trained model is loaded from the saved file.

Video Capture: The script captures video frames from the webcam.

Face Detection and Emotion Prediction: Each video frame is converted to grayscale. Haar cascade classifiers detect faces in the frame. Detected face regions are resized, normalized, and reshaped to match the input format of the model. The model predicts the emotion for each detected face, and the label with the highest probability is chosen. A rectangle is drawn around each detected face, and the predicted emotion label is displayed with the confidence score.

Display and Exit: The processed video frame is displayed in a window. The script runs in a loop, updating the display in real-time until the user presses the 'q' key to exit.

CHAPTER 3 The Code

Test.py: -

```
import cv2
import numpy as np
from keras.models import load_model
model = load_model('model_file_30epochs.h5')
video = cv2.VideoCapture(0)
faceDetect = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
labels_dict = {0: 'Angry', 1: 'Disgust', 2: 'Fear', 3: 'Happy', 4: 'Neutral', 5: 'Sad', 6: 'Surprise'}
while True:
    ret, frame = video.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = faceDetect.detectMultiScale(gray, 1.3, 3)
    for (x, y, w, h) in faces:
        sub_face_img = gray[y:y + h, x:x + w]
        resized = cv2.resize(sub_face_img, (48, 48))
        normalize = resized / 255.0
        reshaped = np.reshape(normalize, (1, 48, 48, 1))
        result = model.predict(reshaped)
        label = np.argmax(result, axis=1)[0]
        confidence = np.max(result) * 100
        print(f"Label: {label}, Confidence: {confidence:.2f}%")
 cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 0, 255), 2)
```

```
        cv2.rectangle(frame, (x, y - 40), (x + w, y), (0, 0, 255), -1)
        cv2.putText(frame,f"{labels_dict[label]}({confidence:.2f}%)",(x,y-10),
cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0, 255, 0), 2) # Green color for text
    cv2.imshow("Frame", frame)
    k = cv2.waitKey(1)
    if k == ord('q'):
        break
video.release()
cv2.destroyAllWindows()
```

Testdata.py: -

```
import cv2
import numpy as np
from keras.models import load_model
model=load_model('model_file_30epochs.h5')
faceDetect=cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
labels_dict={0:'Angry',1:'Disgust', 2:'Fear', 3:'Happy',4:'Neutral',5:'Sad',6:'Surprise'}
# len(number_of_image), image_height, image_width, channel
frame=cv2.imread("faces-small.jpg")
gray=cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
faces= faceDetect.detectMultiScale(gray, 1.3, 3)
for x,y,w,h in faces:
    sub_face_img=gray[y:y+h, x:x+w]
    resized=cv2.resize(sub_face_img,(48,48))
```

```
    normalize=resized/255.0
    reshaped=np.reshape(normalize, (1, 48, 48, 1))
    result=model.predict(reshaped)
    label=np.argmax(result, axis=1)[0]
    print(label)
    cv2.rectangle(frame, (x,y), (x+w, y+h), (0,0,255), 1)
    cv2.rectangle(frame,(x,y),(x+w,y+h),(50,50,255),2)
    cv2.rectangle(frame,(x,y-40),(x+w,y),(50,50,255),-1)
    cv2.putText(frame, labels_dict[label], (x, y-
10),cv2.FONT_HERSHEY_SIMPLEX,0.8,(255,255,255),2)
cv2.imshow("Frame",frame)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Main.py: -

```
from keras.preprocessing.image import ImageDataGenerator
from keras.models import Sequential
from keras.layers import Dense,Dropout,Flatten
from keras.layers import Conv2D,MaxPooling2D
import os
train_data_dir='data/train/'
validation_data_dir='data/test/'
train_datagen = ImageDataGenerator(
                    rescale=1./255,
```

```
                rotation_range=30,
                shear_range=0.3,
                zoom_range=0.3,
                horizontal_flip=True,
                fill_mode='nearest')
validation_datagen = ImageDataGenerator(rescale=1./255)
train_generator = train_datagen.flow_from_directory(
                train_data_dir,
                color_mode='grayscale',
                target_size=(48, 48),
                batch_size=32,
                class_mode='categorical',
                shuffle=True)
validation_generator = validation_datagen.flow_from_directory(
                        validation_data_dir,
                        color_mode='grayscale',
                        target_size=(48, 48),
                        batch_size=32,
                        class_mode='categorical',
                        shuffle=True)
class_labels=['Angry','Disgust', 'Fear', 'Happy','Neutral','Sad','Surprise']
img, label = train_generator._next_()
model = Sequential()
model.add(Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=(48,48,1)))
```

```
model.add(Conv2D(64, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.1))
model.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.1))
model.add(Conv2D(256, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.1))
model.add(Flatten())
model.add(Dense(512, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(7, activation='softmax'))
model.compile(optimizer = 'adam', loss='categorical_crossentropy', metrics=['accuracy'])
print(model.summary())
train_path = "data/train/"
test_path = "data/test"
num_train_imgs = 0
for root, dirs, files in os.walk(train_path):
    num_train_imgs += len(files)
num_test_imgs = 0
for root, dirs, files in os.walk(test_path):
    num_test_imgs += len(files)
print(num_train_imgs)
```

```
print(num_test_imgs)
epochs=30
history=model.fit(train_generator,
        steps_per_epoch=num_train_imgs//32,
        epochs=epochs,
        validation_data=validation_generator,
        validation_steps=num_test_imgs//32)
model.save('model_file.h5')
```

CHAPTER 4 Outputs