Emotion Recognition

With Facial Feature Extraction

A Summer Internship Project by:

19071A1222 - Joshua Kingston Silveri
19071A1230 - Kavya Muktha
19071A1247 - Mohanvenkat Patta
19071A1255 - Sri Rachana Narasu Baditha

Under the Guidance of Dr. D Srinivasa Rao

Contents

About The Project

Field Chosen: Image Processing

Project Chosen: Image Classification using Image Processing

Title Chosen: Emotion Recognition using Facial Feature Extraction

Language Used: Python

Abstract

This project aims to create a program which is able to recognize different emotions and classify them appropriately from a given image.

This is done using Convolutional Neural Networks (CNN), a type of artificial neural network used in image recognition and processing that is specifically designed to process pixel data.

Introduction

Emotion recognition can be captured through different mechanisms such as speech, facial expression, body gesture, etc. According to some psychologists, communication happening through facial expressions account for about 55% of communication.

Facial expressions are key to conveying feelings, attitude, intentions, etc. and these are pivotal in recognizing the emotions.

Emotion recognition through facial expression is becoming popular due its various applications like robotics, biometrics and in surveillance.

Objectives

The objective of this project is to see if a deep learning model can be built to recognize and identify one among seven different emotions from an image.

The seven different emotions are:

Happy, Sad, Angry, Disgusted, Fearful, Surprised, and Neutral

Datasets Used

The dataset which is used to train and test the sequential model is FER-2013.It consists of approximately 30,000 facial images of size 48x48 pixels.The faces are labeled as any of the six cardinal expressions as well as neutral.

FER has more variation in the images, including facial occlusion (mostly with a hand), partial faces, low-contrast images, and eyeglasses.

The disgust expression has the minimal number of images – 600, while other labels have nearly 5,000 samples each.

Angry

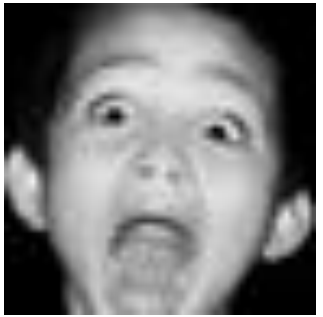
Disgust

Neutral

Sad

Fear

Happy



Surprised

Sample of Datasets Used

Libraries and Modules Used

Tensorflow:

Tensorflow is an open source library for numerical computation and large-scale machine learning. It can train and run deep neural networks for image classification and recognition.

Keras:

Keras is an Open Source Neural Network library which handles the way we make models, defining layers, or set up multiple input-output models.

Keras leverages various optimization techniques to make high level neural network API easier and more performant.

About Deep Learning

Deep Learning is self-capable of learning from the extraordinary amount of data provided to it and doesn’t require a human programmer to intervene.

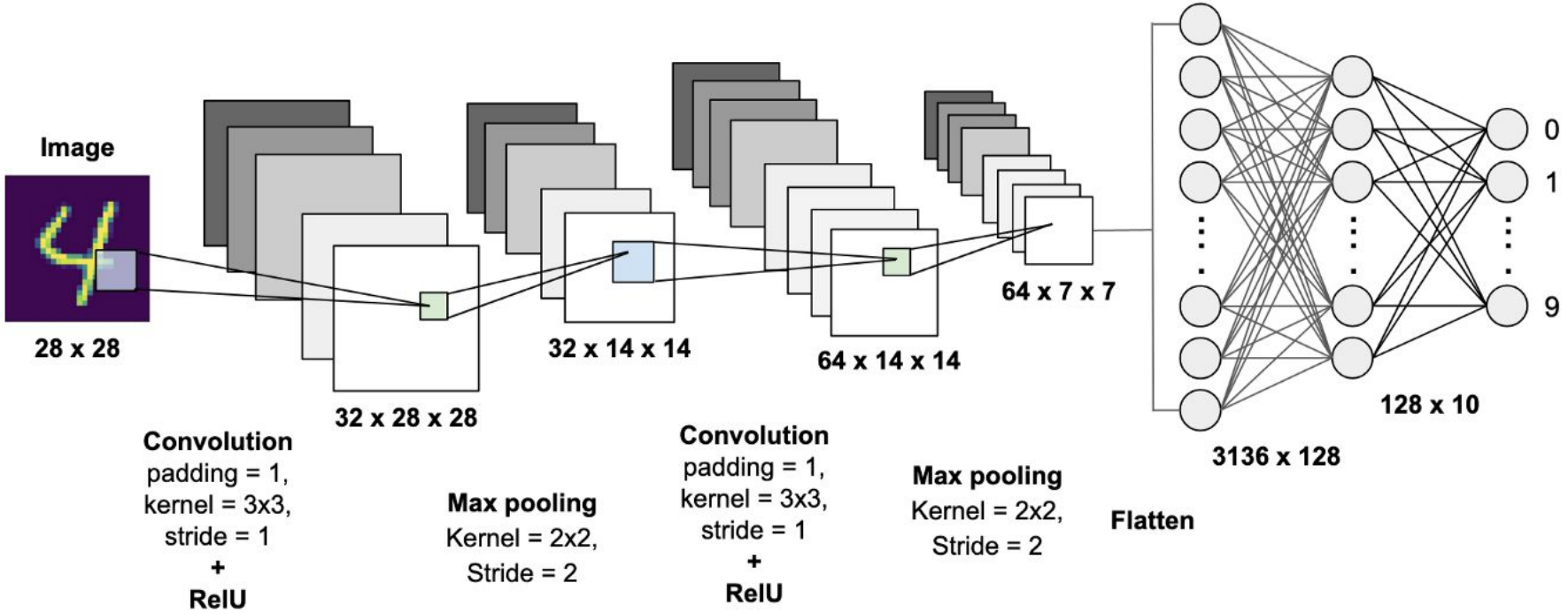
It uses enormous neural networks to teach machines how to automate the tasks performed by human visual systems.

Deep learning in neural networks has wide applications in the area of image recognition, classification, decision making, pattern recognition etc.

About CNN

CNN or Convolutional Neural Network is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other.

CNN’s were first developed and used around the 1980s. The most that a CNN could do at that time was recognize handwritten digits. It was mostly used in the postal sectors to read zip codes, pin codes, etc.



Use of CNN to Recognize Handwritten Digits

Components of a CNN

Convolutional neural networks are distinguished from other neural networks by their superior performance with image, speech, or audio signal inputs. They have three main types of layers, which are:

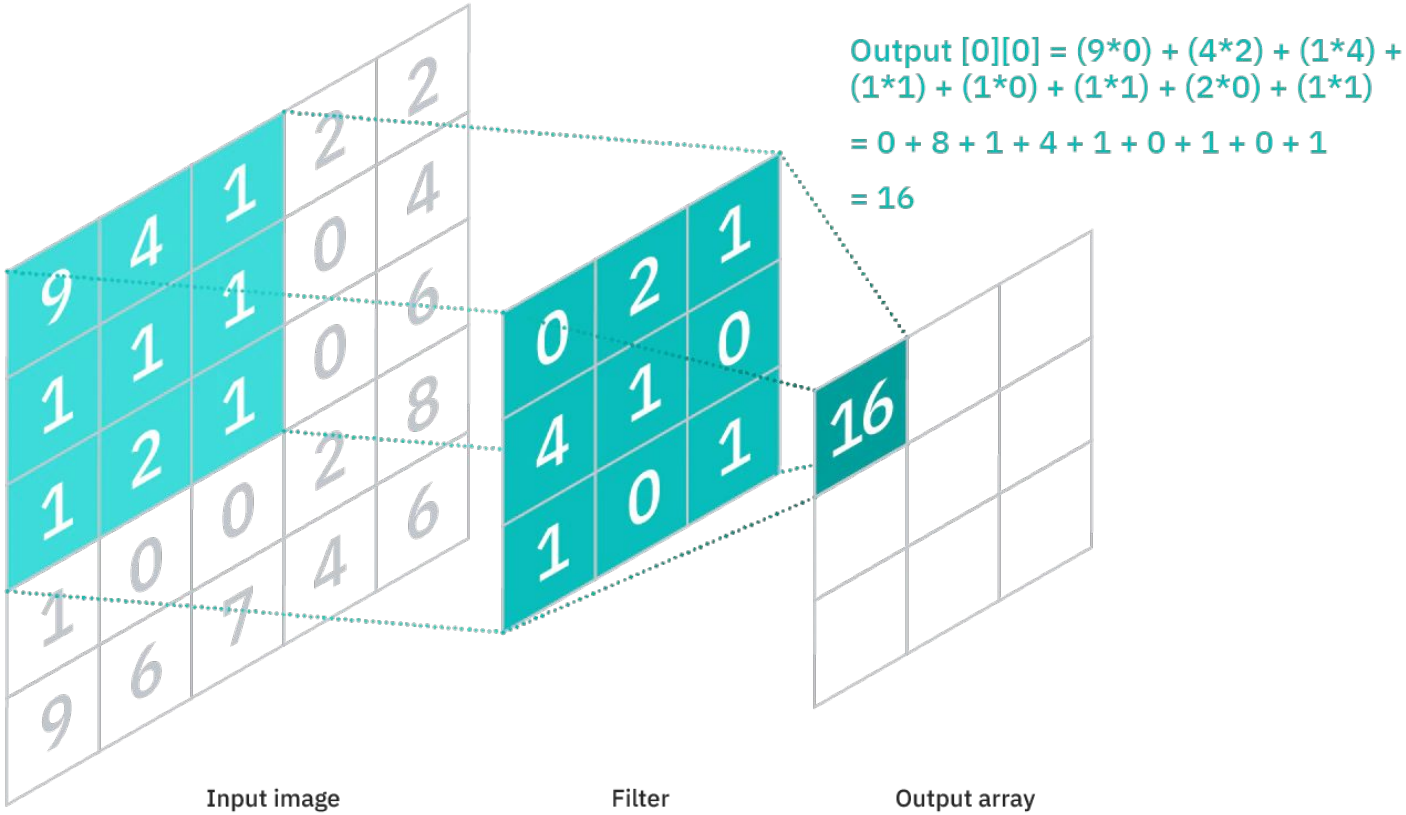
1) Convolutional layer
2) Pooling layer
3) Fully-connected (FC) layer

Convolutional Layer

The convolutional layer is the core building block of a CNN.

Convolutional layers apply a convolution operation to the input, passing the result to the next layer. A convolution converts all the pixels in its receptive field into a single value.

The most common type of convolution that is used is the 2D convolution layer and is usually abbreviated as conv2D. A filter or a kernel in a conv2D layer “slides” over the 2D input data, performing an element-wise multiplication. As a result, it will be summing up the results into a single output pixel.



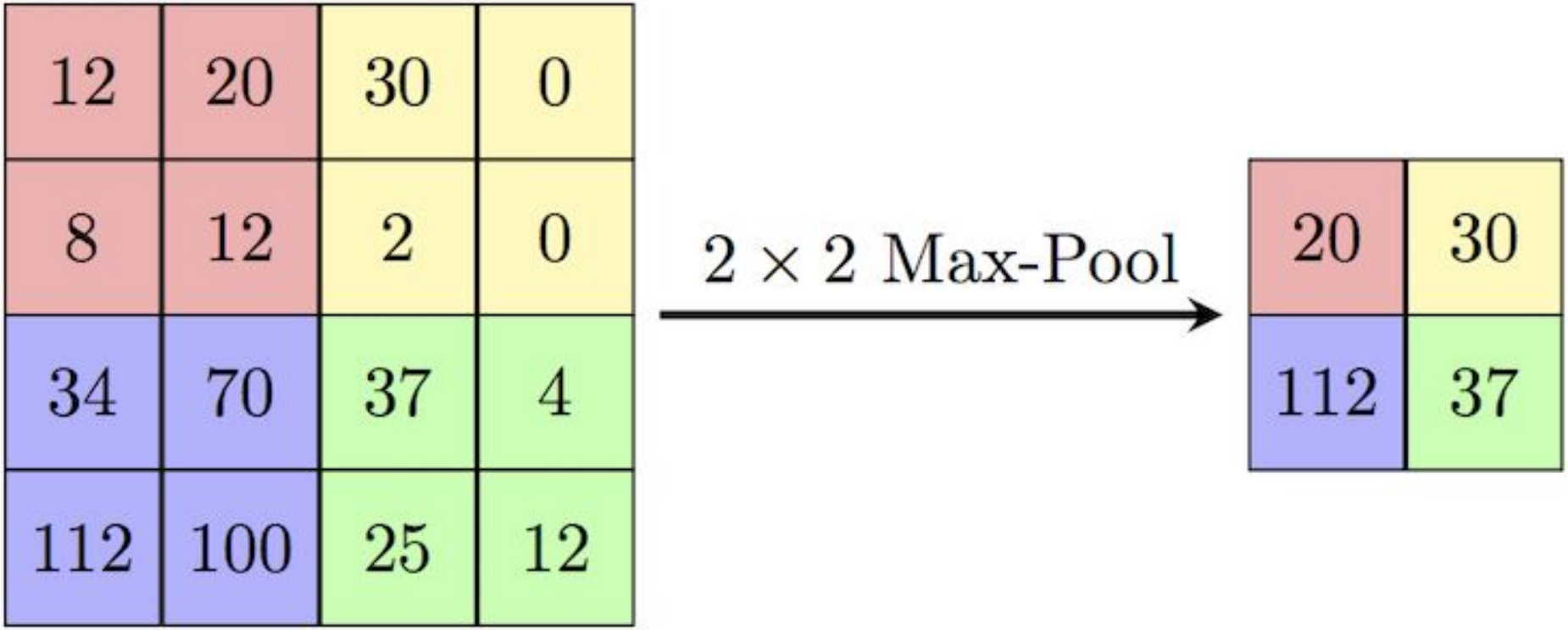
Working of Convolutional Layer of CNN

Pooling Layer

The Pooling layer is responsible for reducing the spatial size of the Convolved Feature. This is to decrease the computational power required to process the data by reducing the dimensions.

In Max Pooling, we find the maximum value of a pixel from a portion of the image covered by the kernel.

Max Pooling also performs as a Noise Suppressant. It discards the noisy activations altogether and also performs de-noising along with dimensionality reduction.



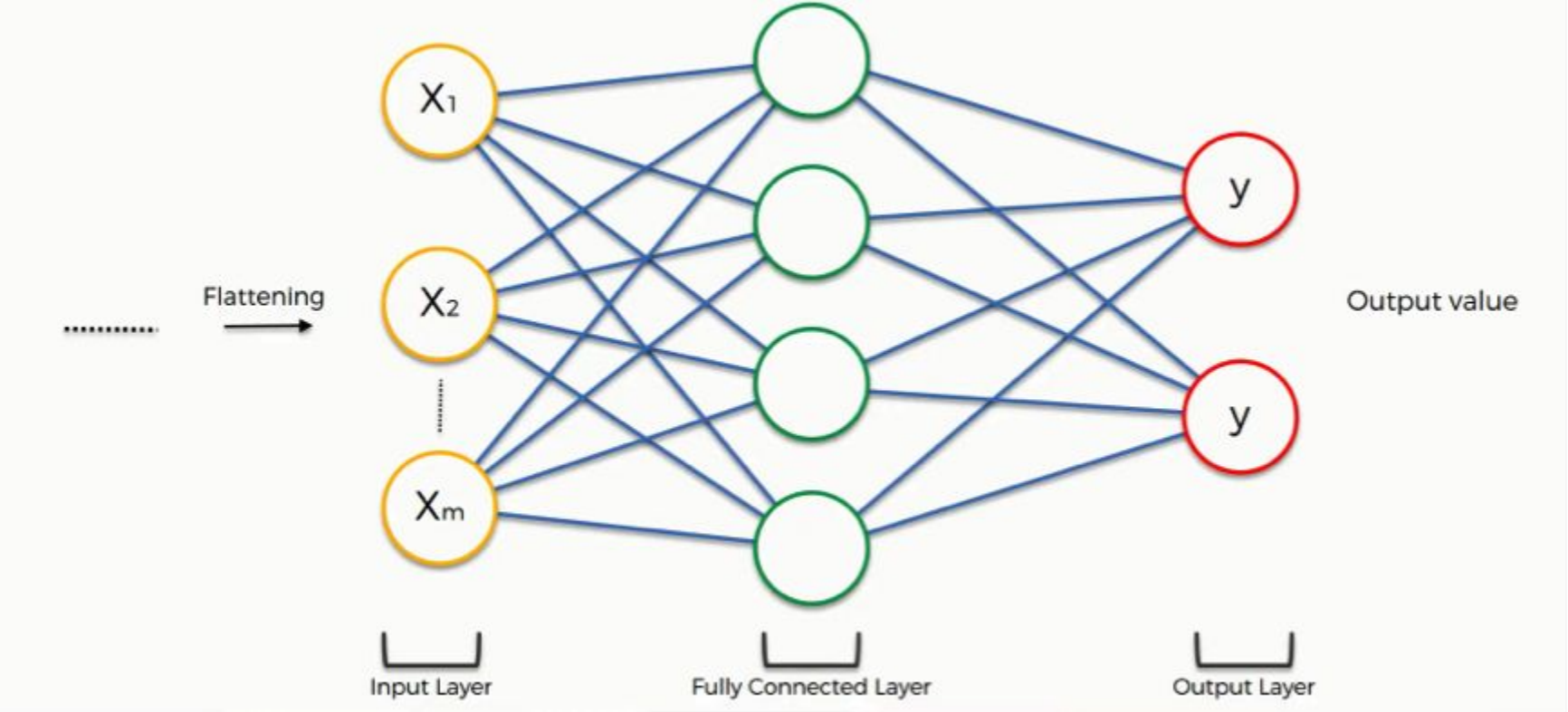
Working of Max Pooling Layer of CNN

Fully Connected Layer

This layer performs the task of classification based on the features extracted through the previous layers and their different filters.

In the fully-connected layer, each node in the output layer connects directly to a node in the previous layer.

Fully Connected Layers form the last few layers in the network. The input to the fully connected layer is the output from the final Pooling or Convolutional Layer, which is flattened and then fed into the fully connected layer.



Structure of Fully-Connected Layer of CNN

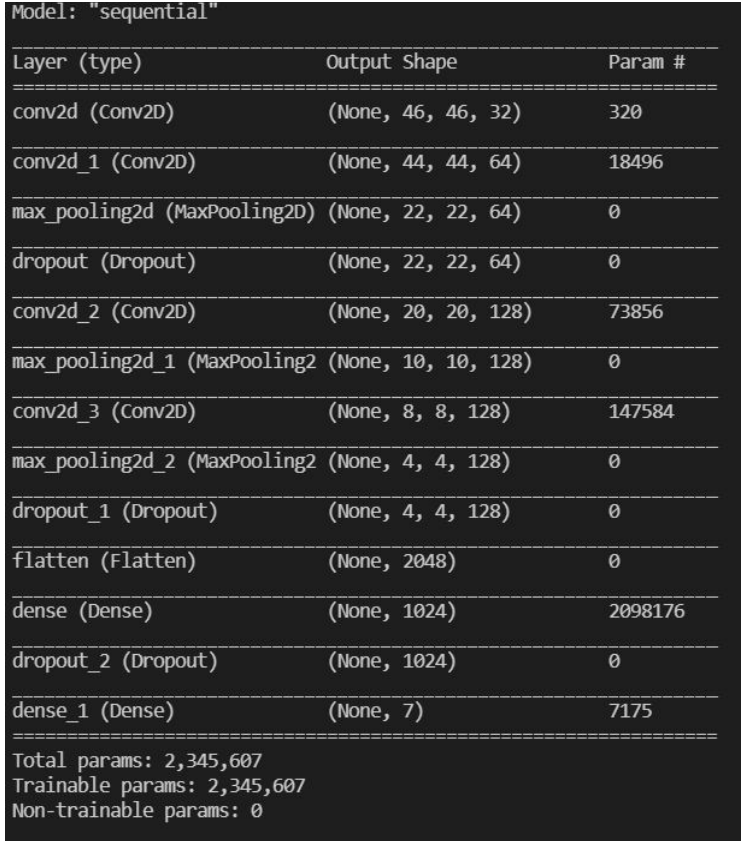
Results

In employing these techniques, we were able to create a CNN model which is able to recognize the seven emotions.

We trained the sequential model with several images and then used the test images to see how the results match up. We trained the model through 50 epochs.

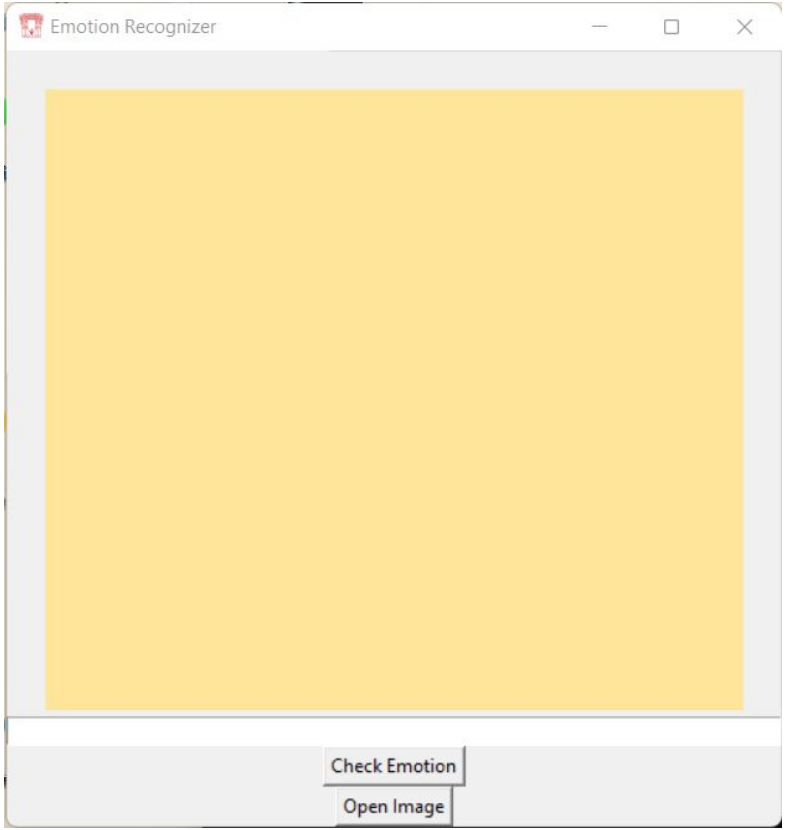
For this model, the accuracy that we achieved for the validation set is 81.7% .

To further increase the accuracy of the model, we can either expand the training dataset we have or increase the step size for the model. Through these parameters, we can increase the model accuracy for this model.



```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 46, 46, 32)        320
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 44, 44, 64)        18496
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 22, 22, 64)        0
_________________________________________________________________
dropout (Dropout)            (None, 22, 22, 64)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 20, 20, 128)       73856
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 10, 10, 128)       0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 8, 8, 128)         147584
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 4, 4, 128)         0
_________________________________________________________________
dropout_1 (Dropout)          (None, 4, 4, 128)         0
_________________________________________________________________
flatten (Flatten)            (None, 2048)              0
_________________________________________________________________
dense (Dense)                (None, 1024)              2098176
_________________________________________________________________
dropout_2 (Dropout)          (None, 1024)              0
_________________________________________________________________
dense_1 (Dense)              (None, 7)                 7175
=================================================================
Total params: 2,345,607
Trainable params: 2,345,607
Non-trainable params: 0
```

Summary of Sequential Model Built



Demonstration of Emotion Recognizer Application

Advantages

Emotion recognition is already widely used by different companies to gauge consumer mood towards their product or brand. The opportunities brought by this technology goes further than market research and digital advertising.

Detecting emotions with technology is quite a challenging task, yet one where machine learning algorithms have shown great promise. By using Facial Emotion Recognition, businesses can process images, and videos in real-time for monitoring video feeds or automating video analytics, thus saving costs and making life better for their users.

Emotion recognition can be shown to have significant usage in various different industries to boost their marketing and research, or aid in their ongoing services.

Automotive industry and emotion recognition:

The automotive industry is applying emotion recognition technology as car manufacturers around the world are increasingly focusing on making cars more personal and safe for people to drive. Using facial emotion detection smart cars can alert the driver when he is feeling drowsy.

Emotion recognition in Health Care:

An industry that’s taking advantage of this technology is Health Care, with AI-powered recognition software helping to decide when patients necessitate medicine or to help out physicians determine who to see first.

Emotion recognition in video game testing:

Using facial emotion recognition can aid in understanding which emotions a user is experiencing in real-time as he or she is playing without analyzing the complete video manually.

Limitations

Poor quality of images limits the effectiveness of emotion recognition:The image quality of scanning video is quite low compared with that of a digital camera.

Different face angles can throw off emotion recognition reliability:The movement of head or different camera positions can cause changes of facial texture and it will generate the wrong result.

A slight change in lighting conditions or the background of the object can make a major impact on the results.

Conclusion

Hence, we were able to gain a deeper understanding of Neural Networks and Deep Learning, and were able to create our own application using the same technologies.

Image recognition is a powerful tool, and emotion recognition could have a game-changing effect on many different industries if researched further and utilised effectively.

Sources

Theoretical Working of CNN:

DeepLizard - Youtube

A Friendly Introduction to CNN - Luis Serrano - Youtube

But What Is A Neural Network - 3Blue1Brown - Youtube

Datasets:

Facial Emotion Recognition - Kaggle

CNN Programming References:

Conv2D - Tensorflow

MaxPool2D - Tensorflow

ImageDataGenerator Class - PyImageSearch

CNN Model References:

Image Classification using CNN - CodeBasic - Youtube

Emotion Investigator - Sanjay Marreddi - GitHub

Any Questions?

THANK YOU!