Golden Gate University

MSBA-326 Predictive Analytics and Machine Learning

Convolutional Neural Networks in Python with Keras

Emotion Detection

Made by:

Anthony Philpott,Tanapat Klomjit and Sweta Kumari

Contents

1 Abstract **3**

2 Introduction **4**

3 Use Case **4**

3.1 Facial emotion detection in interviews 5

3.2 Enhancing capability of robot understanding toward human interactions in Hospitality 5

4 Problem Statement **6**

5 Data Description **6**

6 Literature Review **7**

7 Software **8**

7.1 Python Programming Language 8

7.2 Python Libraries 8

8 Process **8**

8.1 Create Dataset 8

8.2 Loading Dataset 14

9 Convolutional Neural Network **16**

9.1 Model Training 18

9.1.1 Compile 18

9.1.2 Model Fit 18

10 Model Results **19**

10.1 Evaluate the Model 19

11 Live Test **21**

11.1 OpenCV Real-Time Emotion Dection 21

12 Recommendations **24**

13 References **25**

1 Abstract

Our facial expression is playing an important role in social life. It conveys the identity and emotion that involves behavior, actions, feelings, and thoughts. It also plays a vital role in any communication. Detecting emotion is not a new concept anymore in the analytics field. Most of the companies are spending tons of money to gauge the customer's mood to know the underlying emotion towards their brands or products. Our purpose of the paper is to identify the emotion by processing the images. Based on various studies in the field of psychology, we have categorized emotion in seven different emotions: happy, surprise, anger, sad, fear, disgust, and neutral. The aim of the study to understand how we can detect emotion in various cases such as HR interviews or Market research. This paper illustrated end to end process from capturing the image of our faces, converting into 60*60 pixels, and then uses a convolutional neural network to train and test the model. Among many deep learning methods available, we used the convolutional neural network among all due to its popularity and accuracy. To accomplish the goal of our analysis, the various machine libraries such as Keras, OpenCV, TensorFlow, Cascade Classifiers are used. The study also includes the live test which has been conducted to check the performance of the model. The paper also discusses the recommendation based on our process, analysis, and model performance.

Keywords: Emotion detection, Convolutional Neural Network

2 Introduction

In a few years, the market of emotion detection has gone beyond the essence of science fiction to a "20 billion industry"(Washington Post, 2019). Organizations use software that can analyze facial emotions and match it to emotions defined, it acts as a competitive advantage that firms could use to identify when consumers respond to a new product or what interviewee felt during the interview. In our analysis, we are using seven universal emotions and their detection process. The advances in the field of machine learning have yielded the ability to capture and interpret images for the purpose of understanding emotional context. The use of computer vision allows for the deciphering of emotion from humans at a faster velocity than accomplished by a human alone. We analyze seven different emotions and their detection.

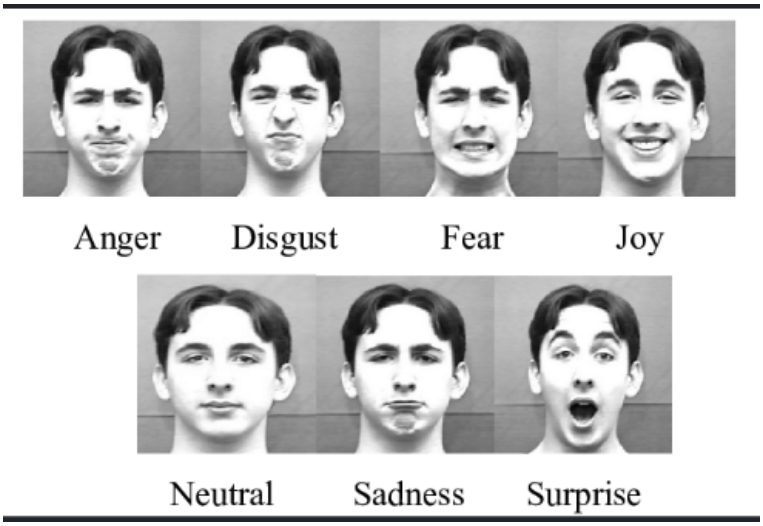


Figure 1: 7 Basic Emotions and their Universal Expressions. Retrieved from https://www.researchgate.net/publicat

3 Use Case

According to Gartner, it predicts that "by 2022, 10 percent of personal devices will have emotion AI capabilities, either on-device or via cloud services." (Moore, 2018). In business, it's undoubtedly essential to understand how customers truly feel about your new features, products, and services. "The emotion detection and recognition market size is estimated to grow from USD 6.72 Billion in 2016 to

USD 36.07 Billion by 2021, at a Compound Annual Growth Rate (CAGR) of 39.9 percent." (Bundela, 2019) With advanced technology capability, emotion analysis is a key path to find out insights that go unsaid. Thus, we demonstrated some use cases of emotion analytics in this section.

3.1 Facial emotion detection in interviews

A well-published use case for facial emotion detection in an interview is from Unilever. This giant consumer goods company already incorporated this emotion, detecting technology into their recruitment process. This technology is embedded in the computer, tablet, and smartphones analyzing the emotional expressions of the prospective candidates during its entry-level interview. Similarly, with our project, there are seven basic emotions that they are trying to capture during the interview. "Your face may register them in milliseconds. Vendors are analyzing video and, from your response to questions, can create personality profiles." (Thibodeau, 2017). This application brings up a different perspective on the table for the company when analyzing their candidates. Moreover, emotion recognition technology helps Unilever to save a significant amount of screening time. This can be beneficial for both the employer and the candidate because it provides more flexibility for both sides. They can proceed interview and review respectively at their schedule. Another advantage is to make the interview process bias-free. Unilever claimed that this new approach has contributed to greater socioeconomic diversity as there was a significant increase in non-white hires (Zetlin, 2018).

3.2 Enhancing capability of robot understanding toward human interactions in Hospitality

Another exciting use case of emotional detection is to enhance the capability of robot understanding toward human interactions. Japan's Henna na Hotel in Tokyo has three lifelike robotic receptionists to meet and greet with guests and proceed with the process of checking in via an automated kiosk. This robot generates positive returns on investment (ROI) due to labor cost saving. They claimed that hotel with this size usually requires about 30 staffs to operate, but Henna na Hotel run with 7 staffs and these robots. (Kikuchi, 2018) Not only can they speak four languages fluently (Japanese,

English, Mandarin Chinses, and Korean), but also they can respond in certain ways while analyzing customer's facial emotions. This added a natural way of how these robots react with guests considering the different emotional expressions.

4 Problem Statement

Sentiment analysis has been popularized as one of the fundamental analyses in terms of gaining more understanding about customers. Since nonverbal components convey two-thirds of human communication, "by carrying emotional meaning, facial expressions are one of the main information channels in interpersonal communication." (Ko, 2018). Thus, we can see that market research applies to various segments of the business in order to retrieve the best insights and ultimately improving the customer experience. For example, Humana, the health insurance company, utilized an emotional intelligence solution expert system called Cogito Corporation to assist their call-center agents. "Here the focus is more on how the customer says something rather than what he or she is saying" (Bundela, 2019). This case proves that emotion can come with different forms verbal, tone, or images.

5 Data Description

In order to provide an accurate representation of an emotional detection system, the creation of a dataset occurred. This consisted of taking 5291 pictures of ourselves in seven distinct facial expressions. The facial expressions consist of seven categories that were classified as 0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, and 6=Neutral. The classified photos were transformed into an aspect ratio of 60X60 with the values of each pixel converted to grey-scale which represents the amount of light that was captured in the photo. Each pixel in the 60X60 photo represented a numeric value between the numbers of 0 and 255. The data was saved into a comma-separated values file with one column represented the labeled emotion, another being the pixel values and the last column representing whether the photo would be used as training or for testing. The training dataset consist of 152 emotions and 48 emotions for testing which follows which is a break out of 70/30.

6 Literature Review

Emotion classification has been debatable topic in psychology, emotion research and cognitive science. Many research is conducted on facial recognition and its classification. They show there is a clear indication for the facial expressions of seven emotions i.e: angry, neutral, disgust, fear, happy, sad, and surprise. According to Friesen's study, the seven type of emotions can be created immediately by people of diverse cultures in response to certain situation. Similar studies have been performed. It proved in over 75 studies that the same seven type of facial expressions are produced when emotions are evoked spontaneously (Matsumoto, Keltner, Shiota, Frank, O'Sullivan, 2008). According to study of National Center for Biotechnology Information, the basic emotion has 7 different type i.e : happiness, surprise, anger, sadness, fear, disgust, and neutral. In our data set, we have seven different facial expression which will help to do the analysis on emotion detection.

According to study of National Center for Biotechnology Information (Chul, 2018), there are several methods used for emotion detection under deep learning methods. CNN (Convolutional Neural Network) is the most widely used method among all. In this method, the data i.e the image is convoluted through a filter collection in the convolution layers. It creates a feature map where it is combined to fully connected networks, and the face expression is recognized as belonging to a class-based the output of the softmax algorithm.

Consistent with this approach, according to research conducted by sensors (a peer reviewed article), CNN is the powerful technique for image classification and detection (sensors, 2019). In our analysis, we have similar approach and using Convolutional Neural Network method which will help to detect the emotion with better accuracy rate.

According to frontier in psychology, though CNN require large amount of processing unit, it is most acceptable method due to its advantages. "CNN is to completely remove or highly reduce the dependence on physics-based models and/or other pre-processing techniques by enabling "end-to-end" learning directly from input images" (Front. Psychol., 2018). Due to this, CNN is widely used in object recognition or scene understanding etc. Due to its heterogeneous layers which include convolution, max pooling, and fully connected layers, most of the image processing projects have adopted

it. Similarly, our analysis uses CNN to train and test the model and achieve the desired results.
It is becoming increasingly to understand the deep learning method and the technique used that align with the goal of our analysis. By using CNN, we will be able to achieve and detect the emotion correctly. The result of the analysis can be helpful in various fields such as HR interview or market research.

7 Software

7.1 Python Programming Language

The Python programming language was used to complete this program. This includes data manipulation, analyzing, visualizations and the live test.

7.2 Python Libraries

Glob: is a library that "finds all the pathnames matching a specified pattern according to the rules used by the Unix shell"(Glob,2020)
Pandas: is a library that " data analysis and manipulation tool"(Pandas,2020)
NumPy: is a library that provides "scientific computing" (Numpy,2020)
Matplotlib: is a "library for creating static, animated visualizations in Python"(Visualization,2020)
OpenCV: is an "open source computer vision and machine learning software library, built to provide a common infrastructure for computer vision applications"(About,2020)
Keras: Is "high-level neural networks API"(Keras,2020)

8 Process

8.1 Create Dataset

The start of the process was to create the dataset which included creating a program that was named *Take Picture*. The program was constructed using OpenCV to take a picture of the user and save the

said picture into a folder. This process allowed for the quick taking of pictures with different emotions. Once the photos were taken they were transferred to their represented labeled folder according to the type of emotion.

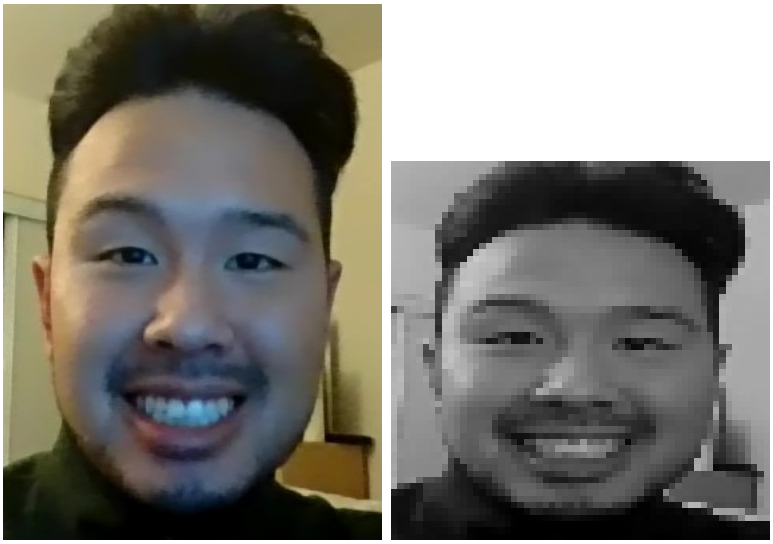
```
# import the necessary packages
from imutils.video import VideoStream
import argparse
import imutils
import time
import cv2
import os

cam = cv2.VideoCapture(0)
cv2.namedWindow("test")
img_counter = 0

while True:
    ret, frame = cam.read()
    cv2.imshow("test", frame)
    if not ret:
        break
    k = cv2.waitKey(1)
    if k%256 == 27:
        # ESC pressed
        print("Escape hit, closing ...")
        break
    elif k%256 == 32:
        # SPACE pressed
        img_name = "opencv_frame_{}.png".format(img_counter)
        cv2.imwrite(img_name, frame)
```

```
        print("{} written!".format(img_name))
        img_counter += 1
cam.release()
cv2.destroyAllWindows
```

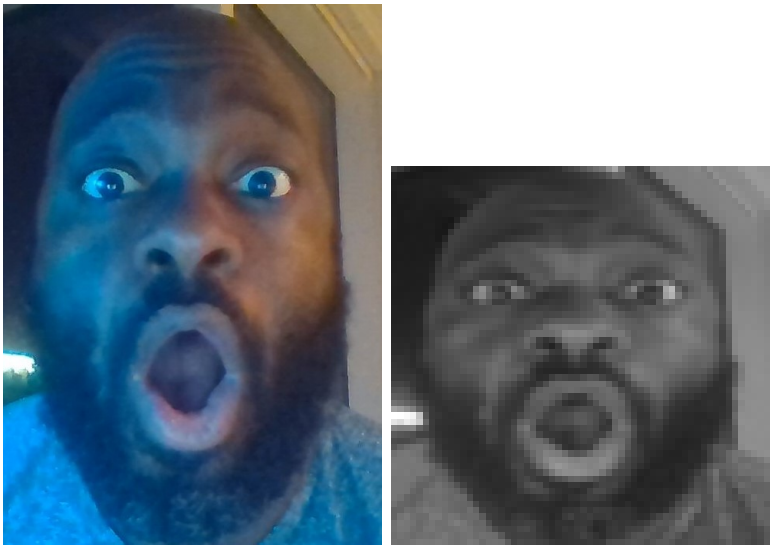
Another program was created called *PNG to Pixel* which consisted of a for loop searching for all png files in a designated emotion folder. The program then creates a csv file based on converting each photo into a separate row with its pixel representation. Each row was assigned a label to correspond to the emotion and was added to an aggregate csv file.

```
import numpy as np
import pandas as pd
from PIL import Image
import glob
import csv
#resize image
newsize = (48,48)
#Open a file to write the pixel data
with open('surprise.csv', 'w', newline='') as f_output:
    csv_output = csv.writer(f_output)
    for filename in glob.glob(r'C:\Users\philp\Desktop\Face\surprise\*.png'):
        im=Image.open(filename).convert('L')
        img1 = im.resize(newsize)
        pixels = list(img1.getdata())
        csv_output.writerow([pixels])
```

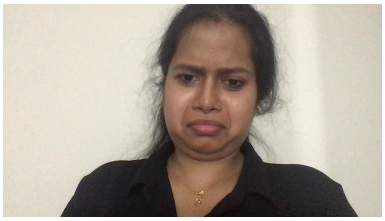
Happy



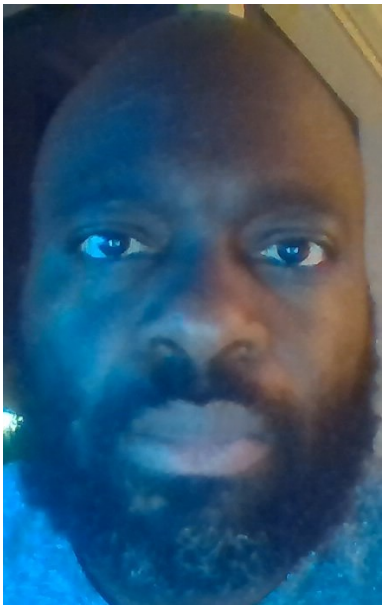
Surprise



Sad



Neutral



8.2 Loading Dataset

First, we imported the necessary libraries into our program and uploaded the dataset using pandas with the following code:

```
import sys, os
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
from keras.models import Sequential
from keras.layers import Dense, Dropout, Activation, Flatten
from keras.layers import Conv2D, MaxPooling2D, BatchNormalization,AveragePooling2D
from keras.losses import categorical_crossentropy
from keras.optimizers import Adam
from keras.regularizers import l2
from keras.utils import np_utils

df = pd.read_csv('new_dataset.csv')
```

Then we created a for loop which assigns the value as the column pixels and to look for either Training or Test in the Usage column in order to split the dataset up between training and test within the program. Both the training and test datasets predictors were assigned the emotion column.

```
X_train,train_y,X_test,test_y=[ ],[ ],[ ],[ ]

for index, row in df.iterrows():
    val=row['pixels'].split(" ")
    try:
        if 'Training' in row['Usage']:
            X_train.append(np.array(val,'float32'))
```

```
            train_y.append(row['emotion'])
        elif 'Test' in row['Usage']:
            X_test.append(np.array(val,'float32'))
            test_y.append(row['emotion'])
    except:
        print(f"error occured at index :{index} and row:{row}")
```

The assignment of the number of labels, batch size, and epochs were done to be fed into the model later on in the process. We set the batch size to 64 to feed the algorithm 64 sample sets at a time. The choice of 500 epochs which means that there are 500 full training cycles on the training data set was decided as the optimal amount of runs.

```
num_labels = 7
batch_size = 25
epochs = 50
```

Both the training and test datasets were then converted into arrays and changed to floats and the categorical labels are converted into a binary matrix in order for the algorithm to understand.

```
X_train = np.array(X_train,'float32')
train_y = np.array(train_y,'float32')
X_test = np.array(X_test,'float32')
test_y = np.array(test_y,'float32')

train_y=np_utils.to_categorical(train_y, num_classes=num_labels)
test_y=np_utils.to_categorical(test_y,
    num_classes=num_labels)test_y=np_utils.to_categorical(test_y, num_classes=num_labels)
```

The training and text data set is then normalized by taking the z-score of the data set and reshaped into a 48, 48,1 matrix to match our dataset.

```
#normalizing data between 0 and 1
X_train -= np.mean(X_train, axis=0)
X_train /= np.std(X_train, axis=0)

X_test -= np.mean(X_test, axis=0)
X_test /= np.std(X_test, axis=0)

X_train = X_train.reshape(X_train.shape[0], 60, 60, 1)
X_test = X_test.reshape(X_test.shape[0], 60, 60, 1)
```

9 Convolutional Neural Network

The next stage is to create and train and test a convolutional neural network (CNN) model based on the dataset that was created. The first step is to call the necessary libraries in order to create the model.

The Keras Sequential model is a "linear stack of layers"(Getting,2020) which allows us to add layers in the order that we want to compute. This program we decided to use three layers due to the complexity of the classification program we are solving. When conducting the experiment with three layers we experienced a mixture of overfitting and unrepresentative validation when we graphed our loss. The decision was made to simplify our model and to only include one layer.

Each layer is a Conv2D that uses "spatial convolution over images"(convolutional,2020) with a kernel size of (3,3) which details the "height and width of the 2D convolution window"(convolutional,2020). The activation of Rectified Linear Unit was chosen due to its computationally efficiency. We then added MaxPooling2D after each layer in order to decrease the chance of over-fitting occurring. Max-

Pooling sets out to reduce the number of parameters by taking the max from each region of the input matrix to create a new output matrix. In order to regulate the CNN we chose to add a dropout at .5 to help prevent overfitting. This will ignore 50 percent of the forward pass and any weight will not update during the backward pass.

```
#designing the cnn
#1st convolution layer
model = Sequential()
model.add(Conv2D(64, kernel_size=(3, 3), activation='relu', input_shape=(X_train.shape[1:])))
model.add(Conv2D(64,kernel_size= (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2), strides=(2, 2)))
model.add(Dropout(0.5))
```

After the data has been through the CNN layer the model is flattened in order to have the output of the layers to be a large piece of unstructured data in order to be classified. The fully connected layer is using 1024 neurons in the first layer with a dropout of 20 percent. The activation choice was Rectified Linear Unit and the process is repeated to the final layer which uses the seven numeric emotional labels with the Softmax activation. The Softmax activation was used due to our multi-class classification challenge. The Softmax activation turns the output of the last layer of our neural network into probabilities that sum to 1. This is accomplished by getting the probabilities of the distribution of all of the labels.

```
model.add(Flatten())
#fully connected neural networks
model.add(Dense(1024, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(1024, activation='relu'))
model.add(Dropout(0.2))
```

```
model.add(Dense(num_labels, activation='softmax'))
```

9.1 Model Training

9.1.1 Compile

Next we compiled the model which specifies the loss function, type of optimizer and the metrics. These specifications will be used to train our model. The loss function we chose was the categorical crossentropy which was used because our classification challenge can only have one correct result. The optimization algorithm that was selected was Adam which is " computationally efficient, has little memory requirements and is well suited for problems that are large in terms of data"(Kingma,BA,2017). Lastly, the metric we want to track is the accuracy thus that was selected to evaluate the performance of our CNN model.

```
#Compliling the model
model.compile(loss=categorical_crossentropy,
              optimizer=Adam(),
              metrics=['accuracy'])
```

9.1.2 Model Fit

The last part in the CNN process is to fit the model which feeds the training dataset and labels based on our assignment of epochs and batch size.

```
#Training the model
history = model.fit(X_train, train_y,
          batch_size=batch_size,
          epochs=epochs,
          verbose=1,
          validation_data=(X_test, test_y),
```

```
        shuffle =True)
```

10 Model Results

10.1 Evaluate the Model

After the model finished compiling it produced an accuracy score of 97.92 percent.

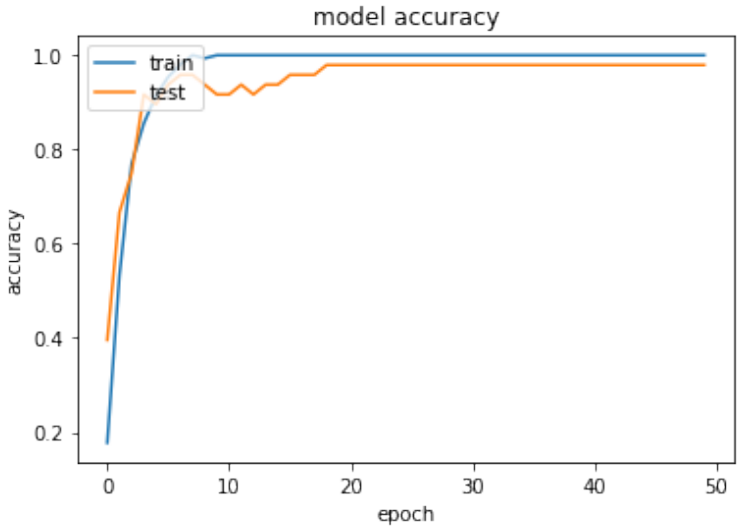
```
# evaluate the model
scores = model.evaluate(X_test, test_y, verbose=0)
print("%s: %.2f%%" % (model.metrics_names[1], scores[1]*100))
```

We then graphed our model's performance by utilizing the callback function to visualize our accuracy and loss before our training and test datasets.

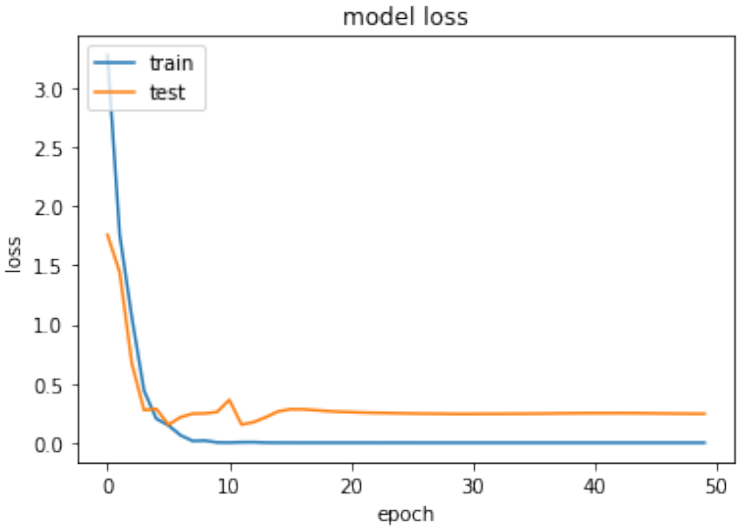
```
# summarize history for accuracy
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
# summarize history for loss
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
```

```
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```

The model accuracy for the train and test curve showed that it is a good fit. The plot shows that it increased as the epochs increased and eventually stabilized with a small gap between the training and testing curves. There was a slight period between epoch 10 and 20 of noise in the test dataset.



The model loss plot depicts a good fit where both the training and fit loss decreased over a period of epochs. There was a slight period between epoch 10 and 20 that noise was recorded but then evened out. There is a slight generalization gap between the training and test loss curves.



11 Live Test

11.1 OpenCV Real-Time Emotion Dection

After the completion of the training and testing of the model the application of the program can occur. Using OpenCV we constructed a live face tracking video feed using object detection. To tackle this program first we use Haar feature-based cascade classifiers. The Haar feature-based cascade classifier is a "machine learning based approach where a cascade function is trained from a lot of positive and negative images"(Cascade,2020). The algorithm works by using positive images in this case our faces and negative images which are images without faces to train the classifier. OpenCV provides pertained models that we used for this program which we utilized the frontal-face trained classifier .

First we imported our necessary libraries in order to run the program. We also uploaded our model which we trained using the CNN model and the Haar Cascade Classifier.

```
import os
import cv2
import numpy as np
from keras.models import model_from_json
```

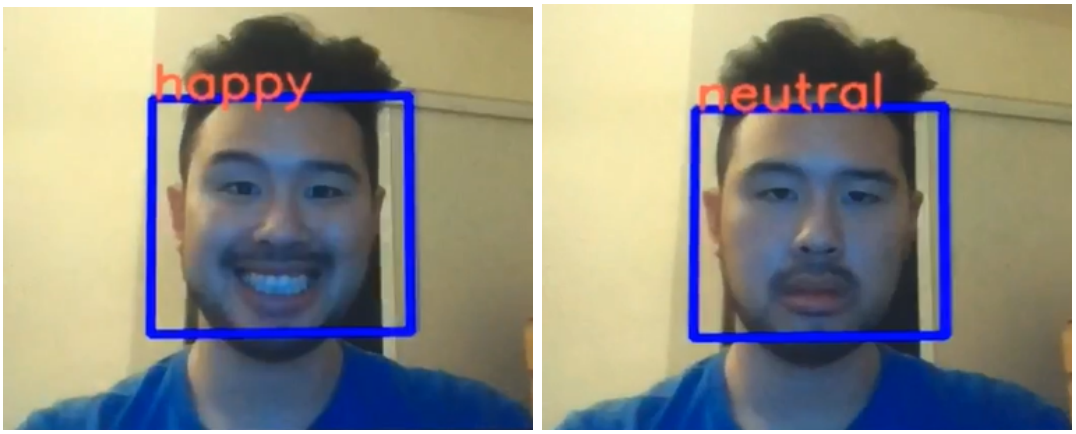
```
from keras.preprocessing import image
#load model
model = model_from_json(open("msba326.json", "r").read())
#load weights
model.load_weights('msba326.h5')
face_haar_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
```

To capture video from a camera we utilized the *cvs.VideoCapture()* function in conjunction with a while loop which lasts forever in a while True expression. This operation will convert the video into grayscale video. The *.detectMultiScale()* method creates a list of rectangles for all of the faces that are in the video. It creates a list of pixels in each face in order to depict the location of the face. Next the code then calls the *.rectangle()* which creates the rectangle around the detected faces. The code uses a for loop to go through the pixel locations from the previous function to create the rectangle. The last step was to call the emotions that were saved in the model and apply the probability of an emotion matching the real-time face in the video.

```
cap=cv2.VideoCapture(0)

while True:
    ret,test_img=cap.read()# captures frame and returns boolean value and captured image
    if not ret:
        continue
    gray_img= cv2.cvtColor(test_img, cv2.COLOR_BGR2GRAY)

    faces_detected = face_haar_cascade.detectMultiScale(gray_img, 1.32, 5)

    for (x,y,w,h) in faces_detected:
        cv2.rectangle(test_img,(x,y),(x+w,y+h),(255,0,0),thickness=4)
```

```
            roi_gray=gray_img[y:y+w,x:x+h]
            #cropping region of interest i.e. face area from image
            roi_gray=cv2.resize(roi_gray,(48,48))
            img_pixels = image.img_to_array(roi_gray)
            img_pixels = np.expand_dims(img_pixels, axis = 0)
            img_pixels /= 255

            predictions = model.predict(img_pixels)
            #find max indexed array
            max_index = np.argmax(predictions[0])

            emotions = ('angry', 'disgust', 'fear', 'happy', 'sad', 'surprise', 'neutral')
            predicted_emotion = emotions[max_index]
            cv2.putText(test_img, predicted_emotion, (int(x), int(y)),
                cv2.FONT_HERSHEY_SIMPLEX, 1, (0,0,255), 2)

        resized_img = cv2.resize(test_img, (1000, 700))
        cv2.imshow('Facial emotion analysis ',resized_img)

        if cv2.waitKey(10) == ord('q'):#wait until 'q' key is pressed
            break

cap.release()
cv2.destroyAllWindows
```

Live Test Results



Live test results correctly identify the human emotions. We captured image of Happy and Neutral face.

12 Recommendations

The recommendation for the emotion detection program is to continue to increase the dataset with different faces. The larger the variety and number of quality faces for training the increase of likelihood that the accuracy score of our program will increase. This is due to the number of challenges we faced creating a dataset from the beginning. The data quality of emotions could have been attributed to the low accuracy score; a few pictures could have included incomplete emotions such as particle smiles. Another key finding while creating the dataset was to make sure the camera that is taking the picture does not include a black border. Through testing we found that using Windows default camera created a black bar on the top and bottom of the phone which resulted in pixels being generated with zeros. The zeros subsequently caused an error when the program tried to fit and discarded those observations.

13 References

About CNET - What We Do. (2020). Retrieved from https://www.cnet.com/about/

Bundela, V. (2019, February 7). Next 5 best examples of emotion analytics in the real world. Retrieved from https://www.softwebsolutions.com/resources/5-examples-of-emotion-analytics.html

Chul, B. (2018). A Brief Review of Facial Emotion Recognition Based on Visual Information. Retrieved from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5856145/

Ekman, P., Friesen, W. V., Ellsworth, P. (1972). Emotion in the human face: guide-lines for research and an integration of findings. New York: Pergamon Press.

Front. Psychol. (2018). Human Observers and Automated Assessment of Dynamic Emotional Facial Expressions: KDEF-dyn Database Validation. Retrieved from https://www.frontiersin.org/articles /10.3389/fpsyg.2018.02052/full

Fuller, S. (2020). Topic: Digital Publishing Industry. Retrieved from https://www.statista.com/topics /1453/digital-publishing/

Google Colaboratory.(2020). Retrieved from https://colab.research.google.com

Gupta, Sneha Chaudhari, Manoj S.(2016, Jan-Mar). An Approach to Predictive Analytics of Website Visitors Traffic and Pageviews

Kikuchi, T. (2018, February 22). Robot staff make Japan's Henn na Hotels quirky and efficient. Retrieved from https://asia.nikkei.com/Business/Robot-staff-make-Japan-s-Henn-na-Hotels-quirky-and-efficient

Ko, B.C. (2018, January 10). A Brief Review of Facial Emotion Recognition Based on Visual Information. Sensors(Basel), 18(2), 401. Retrieved from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5856145/

Li, J., Moore, A. (1970, January 01). Forecasting Web Page Views: Methods and Observations. Retrieved from https://research.google/pubs/pub34912/

Matsumoto, D. and Hwang, H.S. (2011). Reading facial expressions of emotion. Retrieved from https://www.apa.org/science/about/psa/2011/05/facial-expressions

Matsumoto, D., Keltner, D., Shiota, M. N., Frank, M. G., O'Sullivan, M. (2008). What's in a face? Facial expressions as signals of discrete emotions. In M. Lewis, J. M. Haviland L. Feldman Barrett (Eds.), Handbook of emotions (pp. 211-234). New York: Guilford Press.

NumPy¶. (2020 casting procedure implemented in R and Python. It is fast and provides completely automated forecasts that can be tuned by hand by data scientists and analysts. (2020). Retrieved from https://facebook.github.io/prophet/

Pandas. (2020). Retrieved from https://pandas.pydata.org/

Quick Start. (2020). Retrieved from https://facebook.github.io/prophet/docs/quickstart.htmlpython-api

Sensors. (2019). A Case Study of Facial Emotion Classification Using Affdex. Retrieved from https://europepmc.org/article/med/31075816

Subashini, A., Sandhiya, K., Saranya, S., Harsha, U. (2019). Forecasting Website Traffic Using

Prophet Time Series Model. International Research Journal of Multidisciplinary Technovation, 1(1), 56-63.

Taylor, S. J., Letham, B. (2017, September 27). Forecasting at scale.
Retrieved from https://peerj.com/preprints/3190/

Thibodeau, P. (2017, December 4). Emotion analytics may expose your true feelings to HR. Retrieved from https://searchhrsoftware.techtarget.com/feature/Emotion-analytics-may-expose-your-true-feelings-to-HR

Van Gelder, P., Beijer, G., Berger, M.: Statistical analysis of pageviews on web sites, Information and Communication Technologies, vol. 28. Witpress, Southampton (2002)

Video Games Reviews News. (2020). Retrieved from https://www.gamespot.com/?tag =brand-slist;brandgamespot

Visualization with Python. (2020). Retrieved from https://matplotlib.org/

Washington Post. (2019). 'Emotion detection' AI is a 20 billion industry. New research says it can't do what it claims. Retrieved from https://www.washingtonpost.com/business/2019/07/31/ emotion-detection-ai-is-billion-industry-new-research-says-it-cant-do-what-it-claims

Welcome to pandas-gbq's documentation!. (2020). Retrieved from https://pandas-gbq.readthedocs.io/en/latest/

Zetlin, M. (2018, February 28). AI Is Now Analyzing Candidates' Facial Expressions During Video Job Interviews. Retrieved from https://www.inc.com/minda-zetlin/ai-is-now-analyzing-candidates-facial-expressions-during-video-job-interviews.html

7 Basic Emotions and their Universal Expressions. Emotion Recognition via Facial Expression: Utilization of Numerous Feature Descriptors in Different Machine Learning Algorithms. Retrieved from https://www.researchgate.net/publication/329290966$_{E}motion_{R}ecognition_{v}ia_{F}acial_{E}xpression_{U}tilization_{o}f_{N}um$