Facial Emotion Recognition (FER) Sentiment Recognition using NLP

AI Project

Team 1
Susan, Sylvia, Ting Er, Yee Nee

Agenda

- Business problem
- Business Value / Industry
- Proof of Concepts
- Tools
- Risks and Assumptions
- Challenges
- Future Enhancements
- Conclusion

Business Problem:

Do you know how exactly poor customer service is costing us?

And what is the cost of not knowing?

Forbes

Businesses Lose $75 Billion Due To Poor Customer Service

Shep Hyken Contributor
Leadership Strategy

Follow

Shutterstock

The numbers are out. NewVoiceMedia's 2018 "Serial Switchers" report reveals that poor customer service is costing businesses more than $75 billion a year. That's up $13 billion since its last report in 2016.

CEOs of companies large and small are recognizing the importance of

Business Problem & Business Value

Business Problem:

- Unhappy customer don't complain, they will just leave and never look back
- Poor customer service is more viral than good customer service
  → lead to loss of potential business
- Reviews are very important for the companies to know more about their customers, and also be aware of the part which needs to be improved

Business Value:

- By the **facial expression** recognition, customer service can spot those signs early and react accordingly, for example
  - adjust the sales pitch to better sell a product
  - understand the satisfaction of the customer and plan how to do after sales service for better customer retention
- Companies depend on **sentiment analysis** to gain a deeper understanding of the consumer mindset.
  → Better return on investment
  → Improved product features, pricing, store locations, customer experience, and overall employee satisfaction

Business Value

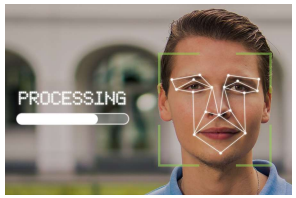
- Customer service interaction
- Facial recognition to detect customer's satisfaction
- For after sales: how happy is the customer with you then plan how to do after sales service for better customer retention
- Beneficial to run analysis based on customer's social media posts to see satisfaction in the products recommended or with company

Industries that will benefit from this technology

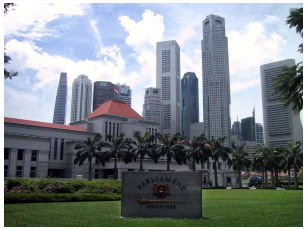
1 Financial Service personalisation

2 Credit/Claim agencies

3 Crime Detection using facial recognition



4 Policy makers leverage of behaviour trends



Proof of Concepts

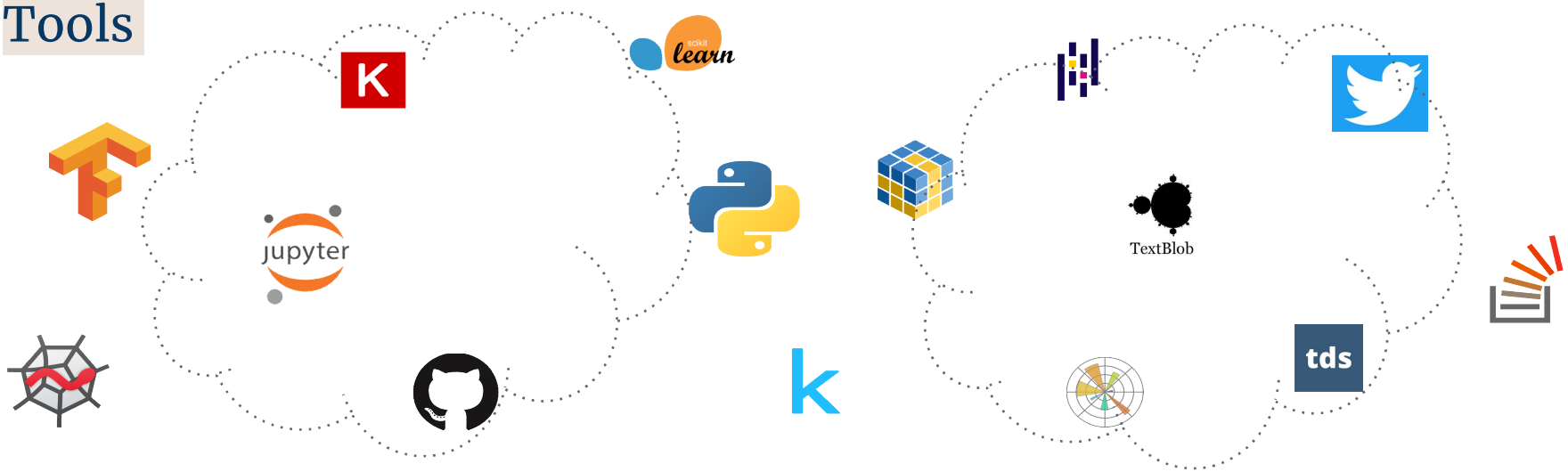
Concept 1:

Emotion recognition through analysis of facial images and videos using machine learning algorithm in Python programming

Concept 2:

Twitter sentiment analysis using Natural Language Processing (NLP) algorithm in Python programming

Tools



- Machine Learning, Tensorflow, Keras, Scikit learn, Facial Emotion Recognition (FER), functions in Python
- Numpy, Pandas & Matplotlib in Python
- Tweepy, Twitter API
- TextBlob, WordCloud, Natural Language Processing in Python
- Jupyter Notebook, Spyder IDE
- GitHub, Kaggle, Stack Overflow, Towards Data Science

- ***Keras*** *is a deep learning API written in Python, running on top of the machine learning platform TensorFlow. It was developed with a focus on enabling fast experimentation.*
- *The* ***Face Emotion Recognizer (FER)*** *is an open-source Python library. It is built on a version that uses a convolutional neural network with weights. This can be overridden by using the FER constructor when the model is called and initiated.*

Risk

There are potential drawbacks to using facial recognition:

- Threats to privacy
- Violations of rights and personal freedoms
- Potential data theft and other crimes
- Risk of errors due to flaws in the technology

Assumptions

- Facial identification with or without masks during the Covid pandemic become the new normal
- Internet penetration and social media websites like Twitter, Reddit, Facebook become more popular and companies rely on them for market study

Challenges

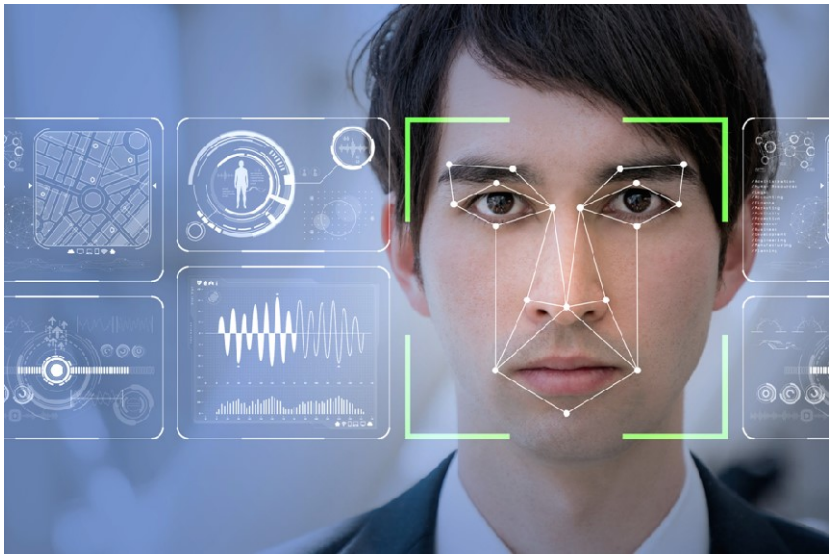
Recognizing the emotions under controlled conditions is simpler. But what makes this task challenging is to account for naturalistic conditions like different poses, illumination, etc and the fact that unexposed expressions are often subtle, can be due system faults like camera distortion, background noise, insufficient storage, improper techniques etc.

https://www.repustate.com/blog/sentiment-analysis-challenges-with-solutions/

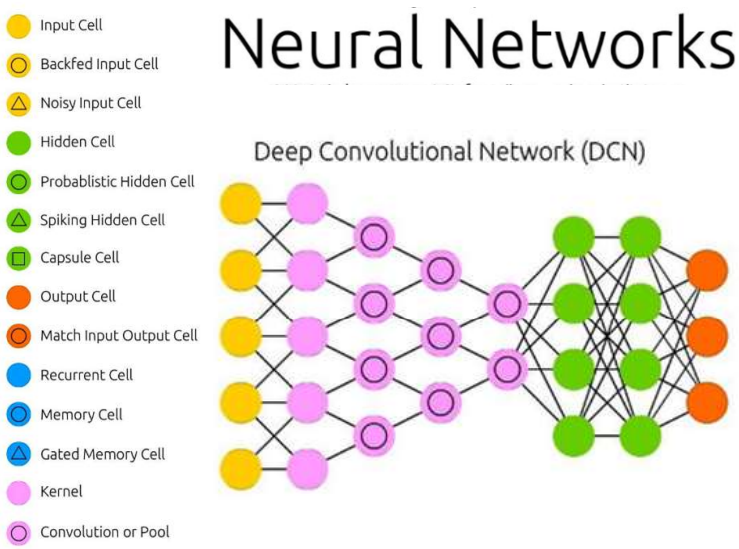
Problem	Solution
• **Tone** can be difficult to interpret verbally, and even more difficult to figure out in the written word.	• The basis of any good sentiment analysis software includes the ability to decipher subjective statements from objective ones and then find the right tone in it.
• There are in-between conjugations of words such as “not so bad” that can mean “average” and hence lie in **mid-polarity**. Sometimes phrases like these get left out, which dilutes the sentiment score.	• Sentiment analysis tools can easily figure out these mid-polar phrases and words in order to give a holistic view of a comment.
• **Sarcasm** or the act of expressing negative sentiment using backhanded compliments can make it difficult for sentiment analysis tools to detect the true context of what the response is actually implying.	• A top-tier sentiment analysis API will be able to detect the context of the language used and everything else involved in creating actual sentiment when a person posts something.
• When **idiom** is used, the sentence can be misconstrued by the algorithm or even ignored.	• The neural networks in an emotion mining API are trained to understand and interpret idioms. Idioms are mapped according to nouns that denote emotions like anger, joy, determination, success, etc.

Proof of Concept 1:

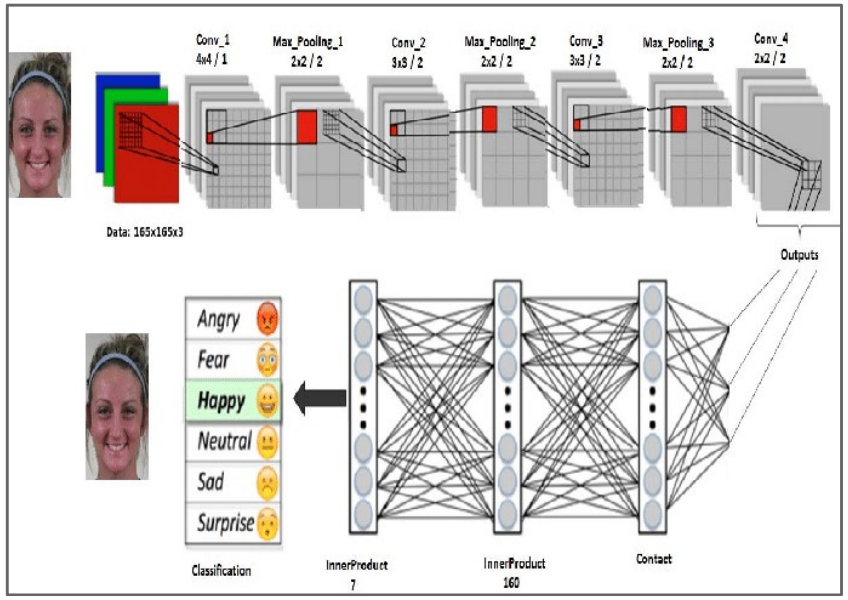
Emotional Recognition using
Facial Images and Videos in Python



Convolutional Neural Network

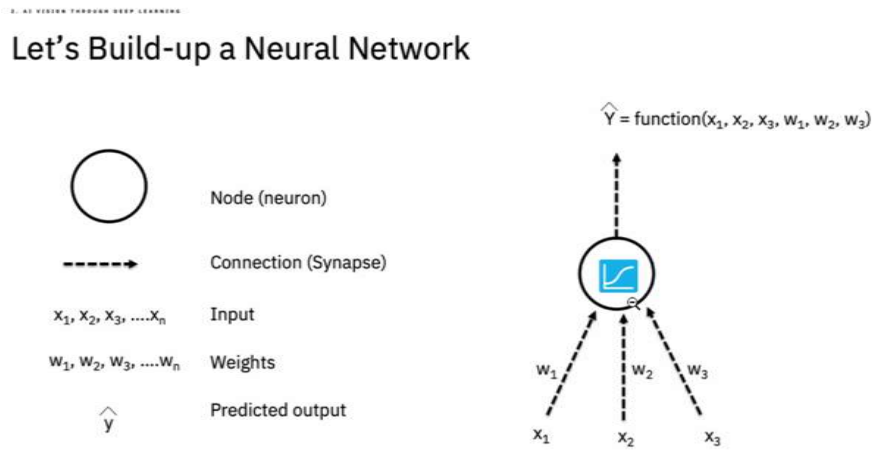


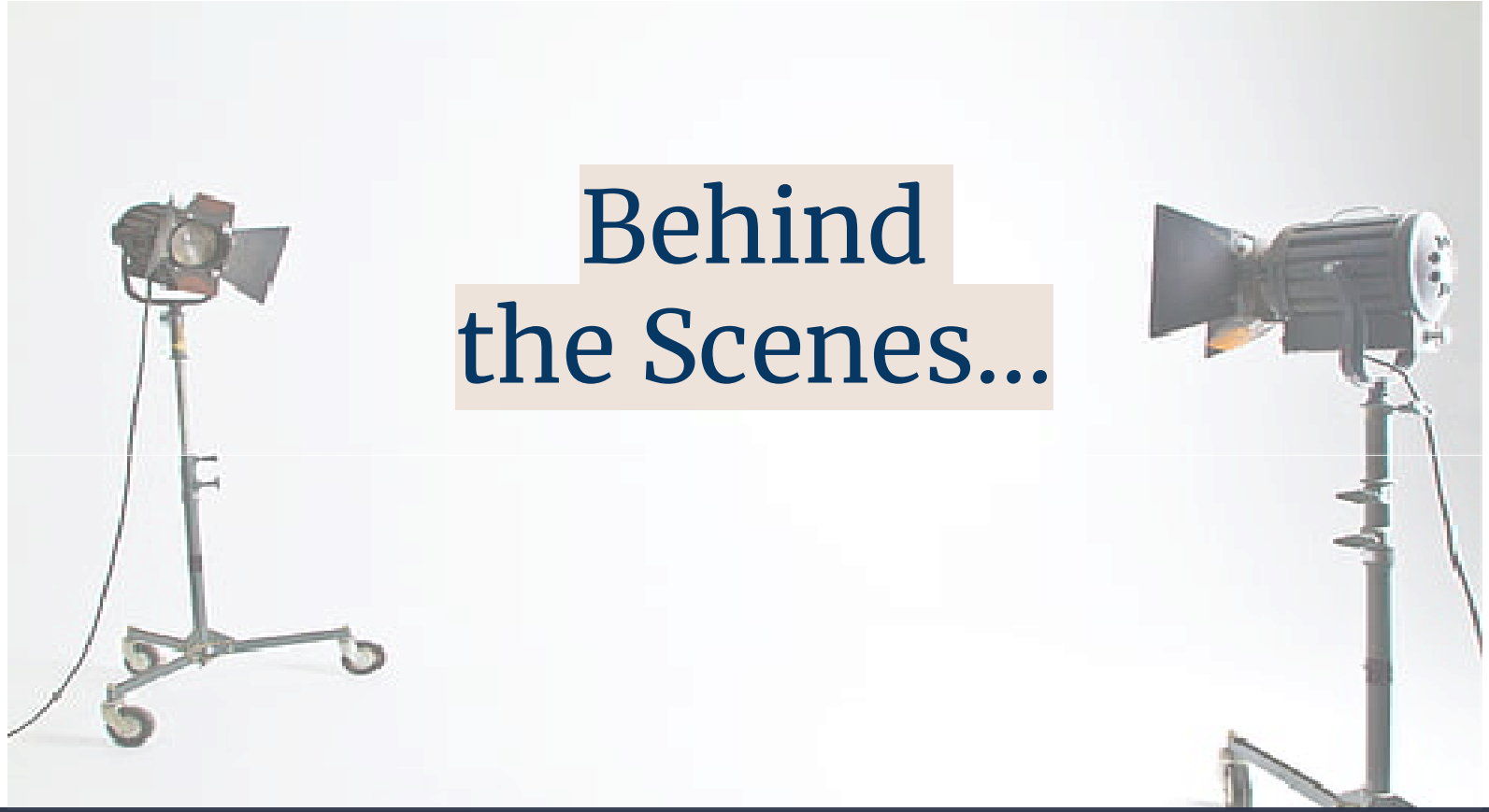
CNN example



Neural network

The neural networks underlying computer vision are fairly straightforward. They receive an image as input and process it through a series of steps. They first detect pixels, then edges and contours, then whole objects, before eventually producing a final guess about what they're looking at.These are known as “feed forward” systems because of their assembly-line setup.





Behind the Scenes...

Building our own Machine Learning Model to Predict Emotion

Preparing our dataset- Computer sees images as pixels

Images

```python
def pixels_to_array(pixels):
    array = np.array(pixels.split(),'float64')
    return array

def image_reshape(data):
    image = np.reshape(data['pixels'].to_list(),(data.shape[0],48,48,1))
    return image
```

Codes

Dataset

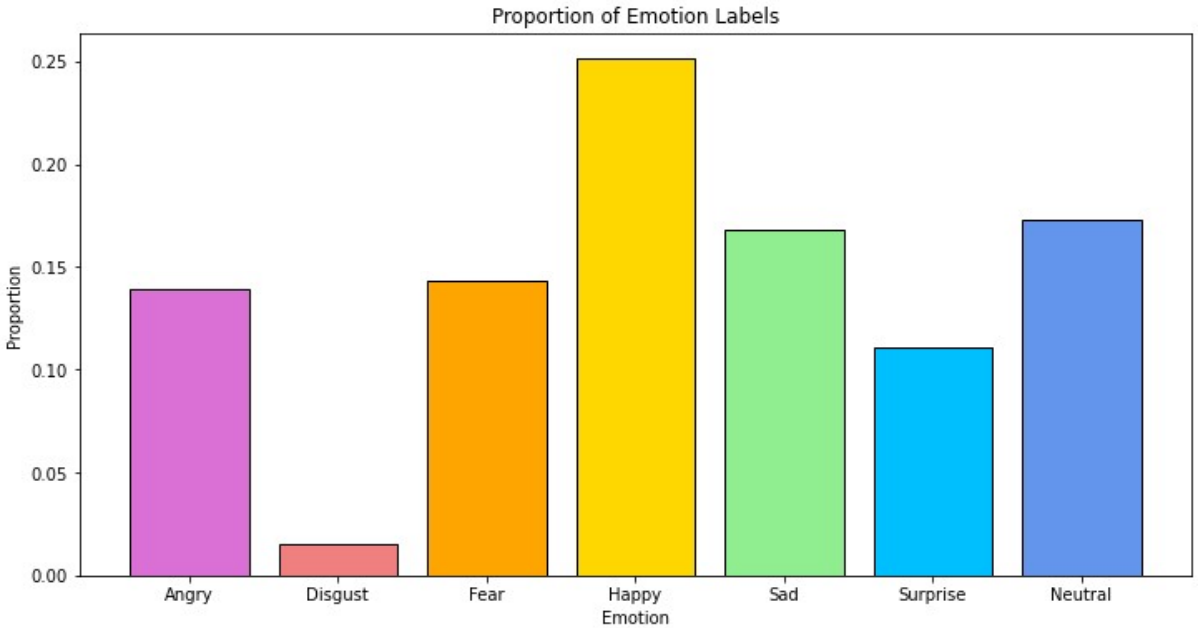
	emotion	pixels
0	0	70 80 82 72 58 58 60 63 54 58 60 48 89 115 121...
1	0	151 150 147 155 148 133 111 140 170 174 182 15...
2	2	231 212 156 164 174 138 161 173 182 200 106 38...
3	4	24 32 36 30 32 23 19 20 30 41 21 22 32 34 21 1...
4	6	4 0 0 0 0 0 0 0 0 0 0 0 3 15 23 28 48 50 58 84...

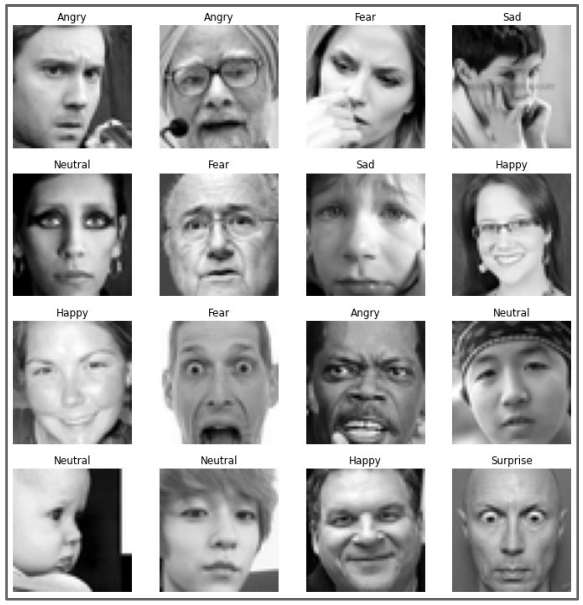
Our code will convert images to pixels and each image classification has a numeric number to represent it.

(0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral.)

Proportion of our dataset used for training model

- Images from our dataset
- Total no of data: 28,710





Building CNN

Layers *early* in the network architecture (i.e., closer to the actual input image) learn *fewer* convolutional filters while layers *deeper* in the network (i.e., closer to the output predictions) will learn *more* filters.

Layers and Filters

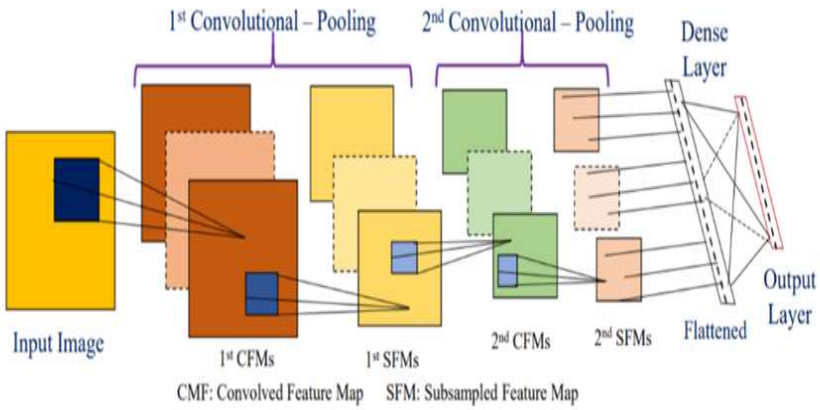
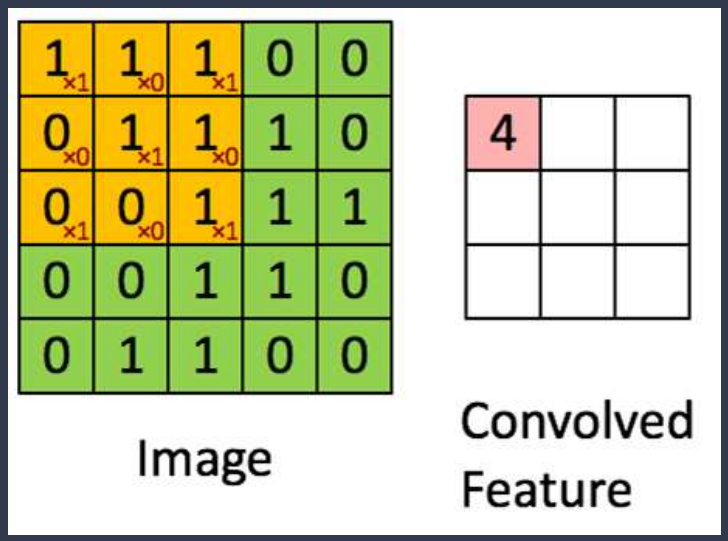


Figure 1. The generic architecture of a convolutional neural network with two convolutional-pooling layers.

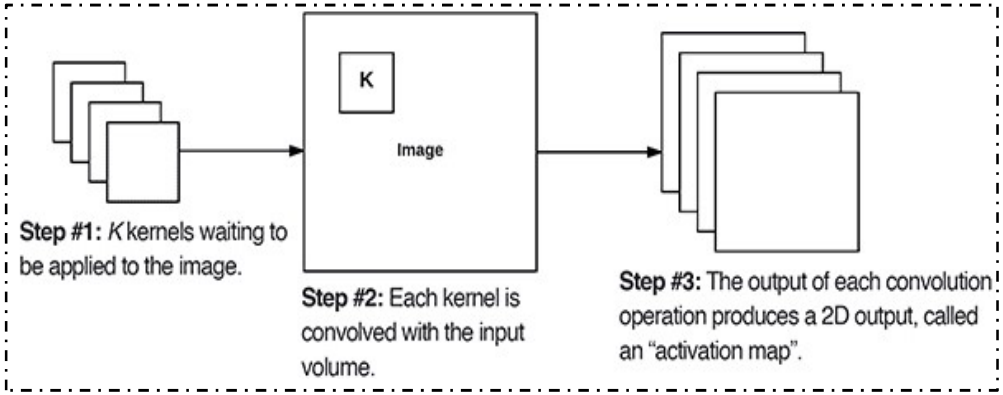
Our neural network has 3 convolution layers

```
#Build Network
np.random.seed(1)
tf.random.set_seed(1)

cnn = Sequential([
    Conv2D(64, (3,3), activation = 'relu', padding = 'same', input_shape=(48,48,1)),
    Conv2D(64, (3,3), activation = 'relu', padding = 'same'),
    MaxPooling2D(2,2),
    Dropout(0.25),
    BatchNormalization(),

    Conv2D(128, (3,3), activation = 'relu', padding = 'same'),
    Conv2D(128, (3,3), activation = 'relu', padding = 'same'),
    MaxPooling2D(2,2),
    Dropout(0.5),
    BatchNormalization(),
```

Building CNN



Choosing Kernel Size



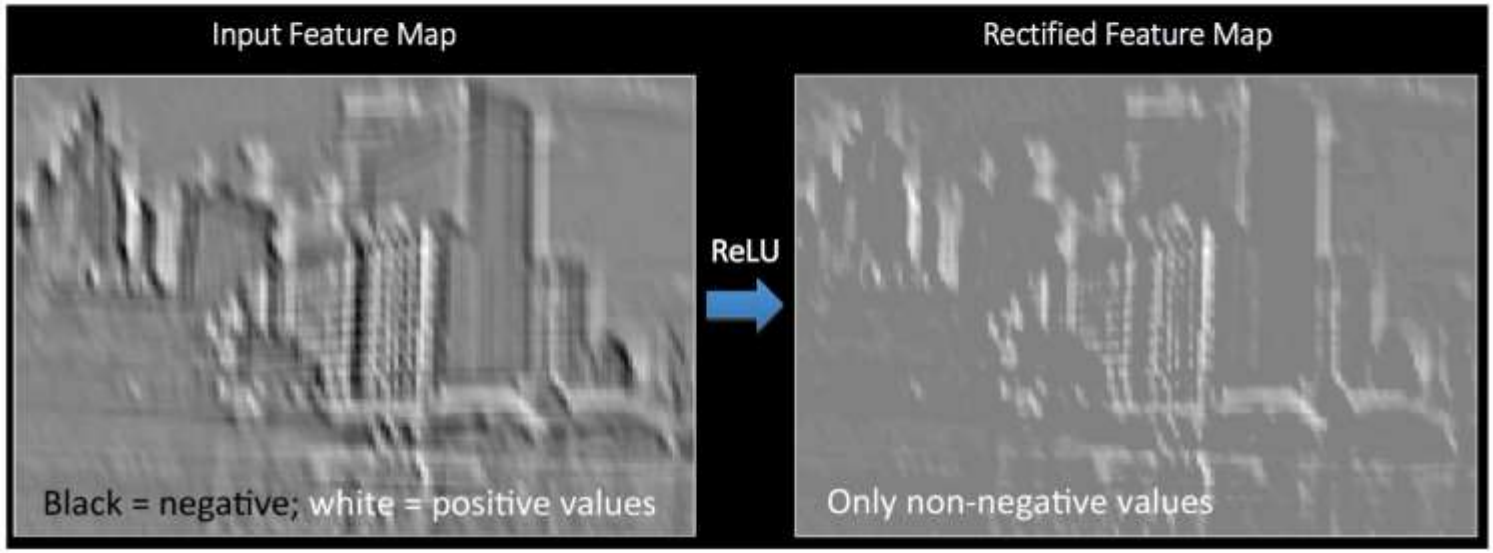
- Since our data image size is not big, we choose Kernel size of 3x3
- We want to preserve the spatial dimensions of the volume such that the output volume size matches the input volume size, so we apply a value of same for the padding

Building CNN

Applying ReLU

- ReLU stand for Rectified Linear Unit and is a non-linear operation
- ReLU is an element wise operation (applied per pixel) and replaces all negative pixel values in the feature map by zero

Example of Relu applied to image



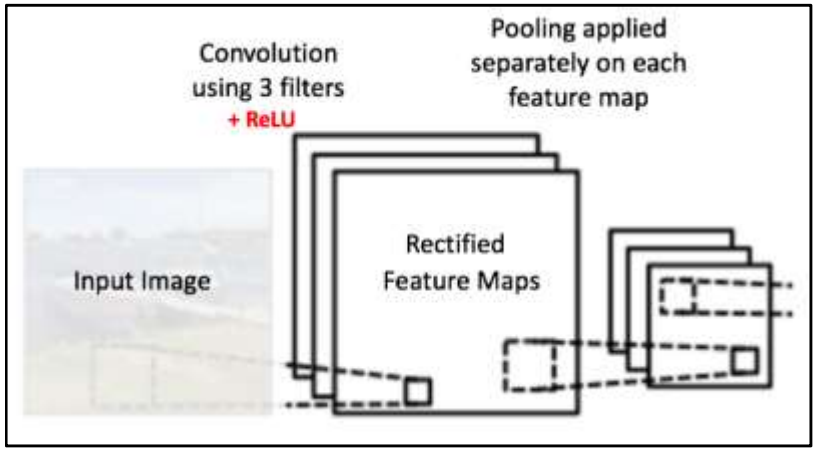
Building CNN

Function of pooling:

- makes the input representations (feature dimension) smaller and more manageable
- Control overfitting
- makes the network invariant to small transformations, in new input image
- helps us detect objects in an image no matter where they are located

Pooling

- Spatial Pooling (also called subsampling or downsampling) reduces the dimensionality of each feature map but retains the most important information



- In the network shown above, pooling operation is applied separately to each feature map (notice that, due to this, we get three output maps from three input maps)

Training the model

```
#split data
X_train, X_valid, y_train, y_valid = train_test_split(X, y, test_size=0.2, random_state=1)
print(X_train.shape)
print(y_train.shape)
print(X_valid.shape)
print(y_valid.shape)
```

(22967, 48, 48, 1)
(22967,)
(5742, 48, 48, 1)
(5742,)

- 80% of dataset (22,967 images) used to train the model, 20%(5,742 images) used to testing the model
- Random state set to 1 to prepare the model for different data received

28,709 facial images in total.

Training the model

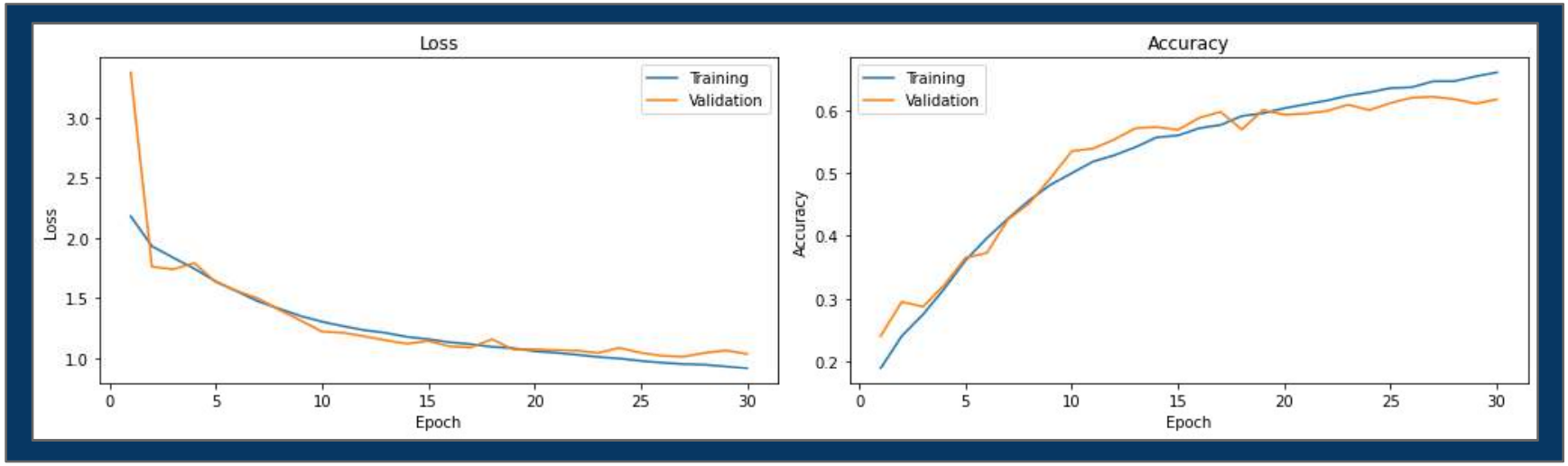
30 epochs, 90 iterations to complete each epochs. (means batch size=28709/90)

- **Stochastic gradient descent** is an iterative learning algorithm that uses a training dataset to update a model
- The iterations, also known as batch size is a hyperparameter of gradient descent that controls the number of training samples to work through before the model's internal parameters are updated
- The number of epochs is a hyperparameter of gradient descent that controls the number of complete passes through the training dataset

And so we proceeded to train our model:

```
Training in progress…
Epoch 1/30
90/90 [==============================] - 622s 7s/step - loss: 2.1809 - accuracy: 0.1887 - val_loss: 3.3745 - val_accuracy: 0.2402
Epoch 2/30
90/90 [==============================] - 710s 8s/step - loss: 1.9323 - accuracy: 0.2405 - val_loss: 1.7611 - val_accuracy: 0.2948
Epoch 3/30
90/90 [==============================] - 640s 7s/step - loss: 1.8367 - accuracy: 0.2747 - val_loss: 1.7403 - val_accuracy: 0.2870
Epoch 4/30
90/90 [==============================] - 625s 7s/step - loss: 1.7447 - accuracy: 0.3156 - val_loss: 1.7921 - val_accuracy: 0.3218
Epoch 5/30
90/90 [==============================] - 617s 7s/step - loss: 1.6398 - accuracy: 0.3608 - val_loss: 1.6340 - val_accuracy: 0.3650
```

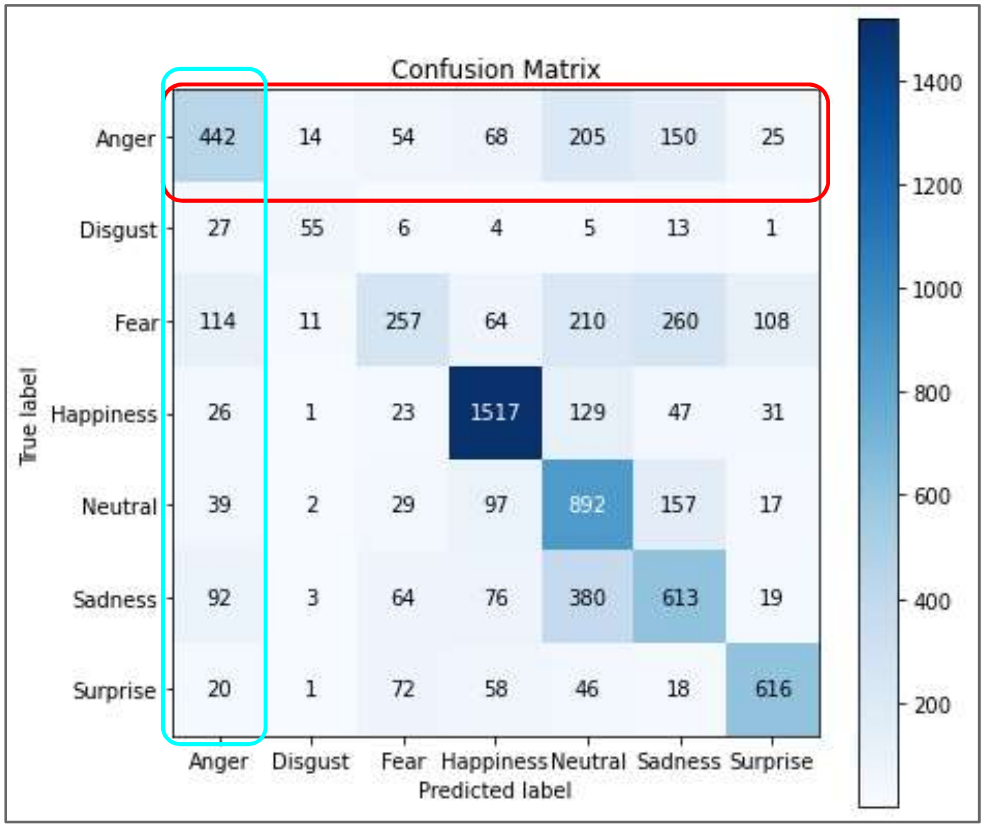
Machine learning model ready!



- **Loss decrease and accuracy increase as each epoch runs (which is good!).**

- Final results on our machine learning model
  loss: 0.9162 - accuracy: 0.6599 - val_loss: 1.0378 - val_accuracy: 0.6170

- Other things that we have tried:
  - Ran another model with two more convolutional layers but found the accuracy level is about the same
  - Second run but the accuracy only improved a little
  - Added more filters but accuracy didn't improve much as well



Confusion Matrix

Precision = TP/(TP+FP)
= positive which predicted correctly / total predicted
Eg. Anger = 0.58 (442/760)

Recall = TP/(TP+FN)
= positive which predicted correctly / total correct positive
Eg. Anger = 0.46 (442/958)

- **The Confusion Matrix allows us to see a breakdown of the predicted labels and the true labels**
- Our model had a hard time classifying images of Fear. Correctly predicted 257 images where the true emotion was Fear. However, it wrongly predicted that the emotion as Sadness when it was actually 260 Fear images
- In the future, we may decide to combine the Fear and Sadness classes since they are very similar emotions.

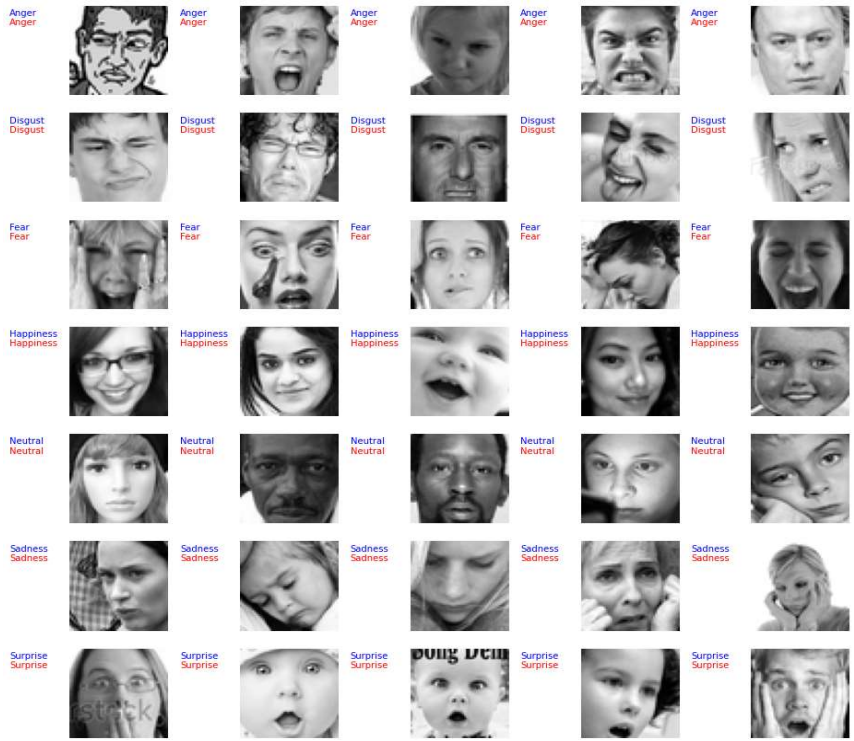
```
(7178, 48, 48, 1)
              precision    recall  f1-score   support

       Anger       0.58      0.46      0.51       958
     Disgust       0.63      0.50      0.56       111
        Fear       0.51      0.25      0.34      1024
   Happiness       0.81      0.86      0.83      1774
     Neutral       0.48      0.72      0.58      1233
     Sadness       0.49      0.49      0.49      1247
    Surprise       0.75      0.74      0.75       831

    accuracy                           0.61      7178
   macro avg       0.61      0.57      0.58      7178
weighted avg       0.61      0.61      0.60      7178

Total Wrong Predictions: 2786
Anger top accuracy: 0.0
Disgust top accuracy: 0.35
Fear top accuracy: 0.54
Happiness top accuracy: 0.66
Sadness top accuracy: 0.76
Surprise top accuracy: 0.84
Neutral top accuracy: 0.93
```

Evaluation Report

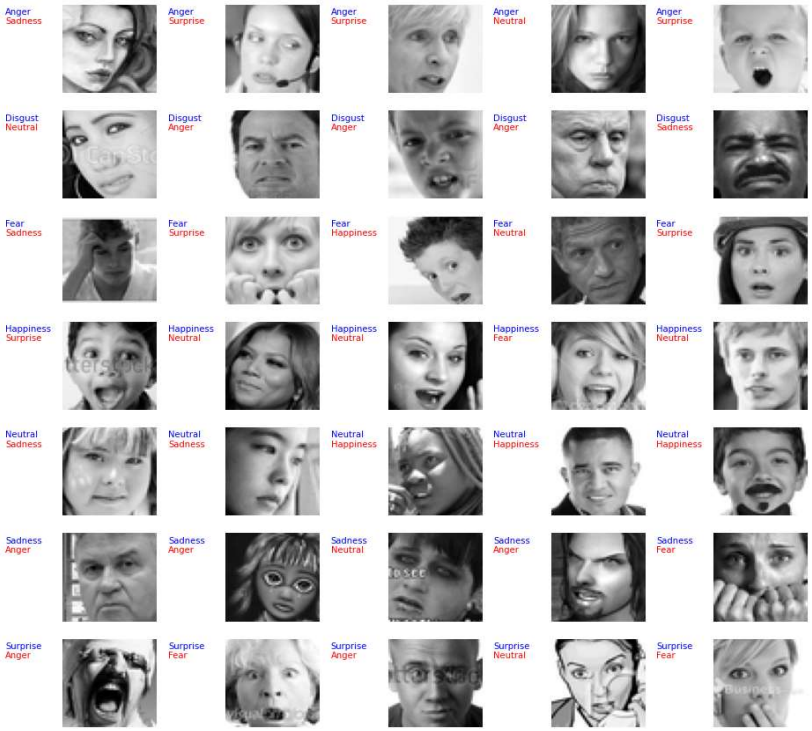
- Total number of images used for evaluation is 7178.
- Emotion 'Anger' has 0.0 top-k accuracy score which is a generalisation of the accuracy score of our machine learning model
- Emotion 'Neutral' has 0.93 top-k accuracy score, meaning higher accuracy of the classification/labelling

The blue text next to each image is the observed emotion and the red text is the predicted emotion.

Correct Predictions



Incorrect Predictions



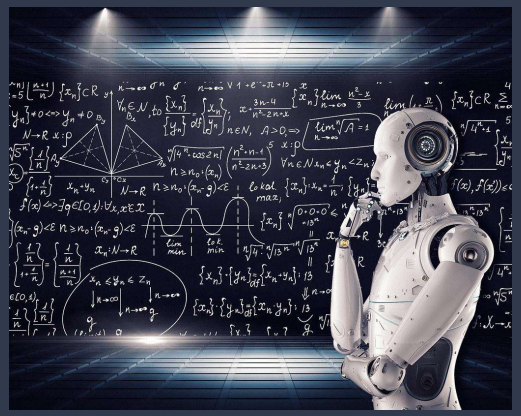
Future Enhancements

1) To feed more data to improve accuracy. We only have an average of 3000+ data for each category and just a few hundred data for category 'disgust'

2) Running more epochs would increase the accuracy

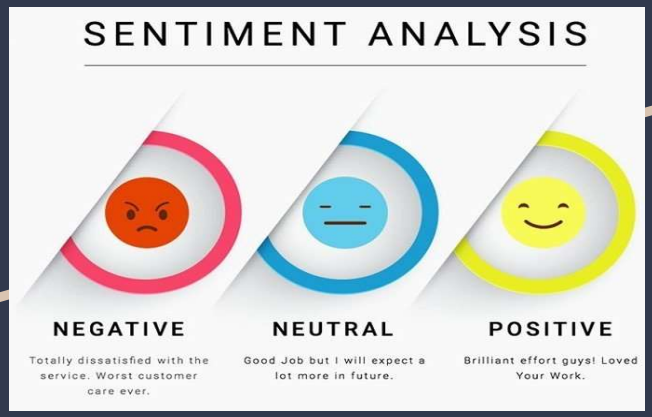
1) To use image data augmentation by applying random transformations to the training images such as horizontal flipping or small random rotations

Actual Usage of the Machine Learning Model



Use Convolutional Neural Network (CNN) on facial expressions to predict the emotions

A study by Rahulraj Singh who works at Columbia University Data Science and also Microsoft, researching Artificial Intelligence and implementing Big Data Analytics on Xbox Games)



Marketing research has proven that predicting sentiments correctly can be a huge source of growth for businesses and that's what we will be working on today — **Reading Emotions**. In the world of data and machine learning, this concept falls under the umbrella of cognitive systems

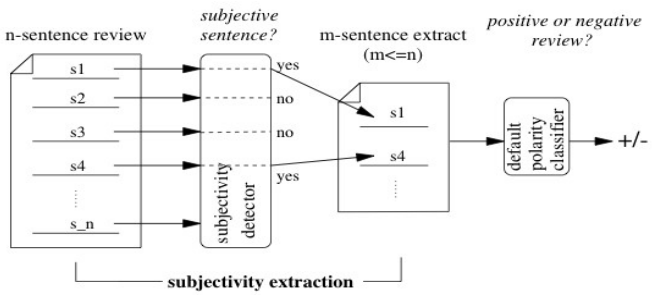
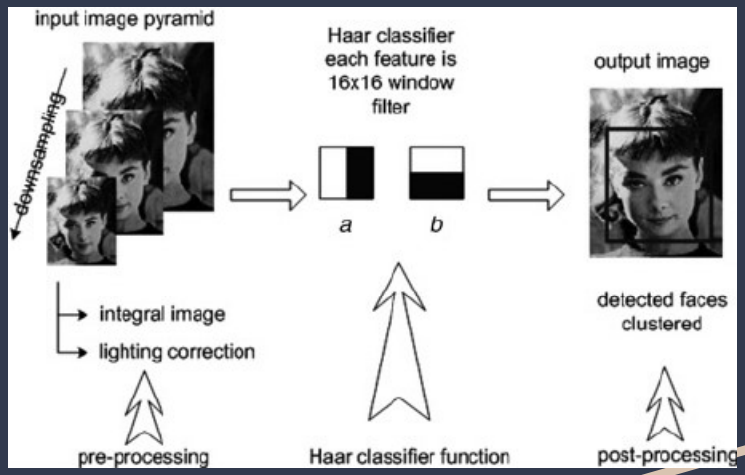


Figure 1: Polarity classification via subjectivity detection.

Analysing Facial Images

Haar Cascade Algorithm



- *Dependencies for installing FER are OpenCV version 3.2 or greater, TensorFlow version 1.7 or greater, and Python 3.6.*
- OpenCV (**Open Source Computer Vision Library**) is an open source computer vision and machine learning software library.
- MTCNN (**multi cascade convolutional network**)- It is a technique to detect faces. When it is set to 'True' the MTCNN model is used to detect faces, and when it is set to 'False' the function uses the default OpenCV Haar Cascade classifier.

```
# We install the FER() library to perform facial recognition
# This installation will also take care of any of the above dependencies if they are missing
pip install FER

from fer import FER
import matplotlib.pyplot as plt
%matplotlib inline

test_image_one = plt.imread("/content/Image-One.jpeg")
emo_detector = FER(mtcnn=True)
# Capture all the emotions on the image
captured_emotions = emo_detector.detect_emotions(test_image_one)
# Print all captured emotions with the image
print(captured_emotions)
plt.imshow(test_image_one)

# Use the top Emotion() function to call for the dominant emotion in the image
dominant_emotion, emotion_score = emo_detector.top_emotion(test_image_one)
print(dominant_emotion, emotion_score)
```

Analysing Videos

*Install **FER** and **Tensorflow** functions and show the dominant emotion and assign emotion score.*

The flow of Logic: Although the underlying algorithm is similar for both images and videos, there are a few key changes that we will follow for videos.

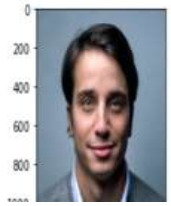
- Video_analyze(): This function is responsible for extracting the individual image frames from a video and then analyze those independently.
- Each frame analyzed by this function is stored as a separate image by the algorithm in the root directory folder where the code is running. Also, this function later creates a replica of the original video by placing a box around the face and showing live emotions within the video.
- We then create a Pandas DataFrame from these analyzed values and plot this dataframe using matplotlib. In this plot, we can see every emotion plotted against time.
- We can further analyze this dataframe by taking individual emotion values that were recognized by the model and finding which sentiment was dominant across the entire video.

```
]: pip install FER
```

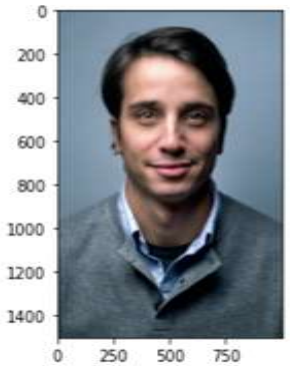
```
]: pip install tensorflow
```

```
]: # This installation will also take care of any of the above dependencies if they are missing
from fer import FER
import matplotlib.pyplot as plt
%matplotlib inline

test_image_one = plt.imread(r"\Project\FACE\Image-Three.jpg")
emo_detector = FER(mtcnn=True)
# Capture all the emotions on the image
captured_emotions = emo_detector.detect_emotions(test_image_one)
# Print all captured emotions with the image
print(captured_emotions)
plt.imshow(test_image_one)

# Use the top Emotion() function to call for the dominant emotion in the image
dominant_emotion, emotion_score = emo_detector.top_emotion(test_image_one)
print(dominant_emotion, emotion_score)
```

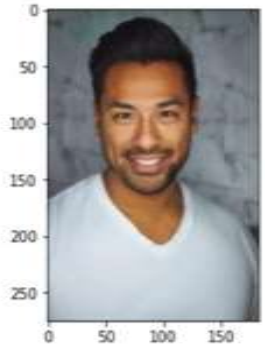
```
[{'box': (155, 193, 676, 676), 'emotions': {'angry': 0.0, 'disgust': 0.0, 'fear': 0.0, 'happy': 0.58, 'sad': 0.02, 'surpris
e': 0.0, 'neutral': 0.4}}]
happy 0.58
```



```
[{'box': (155, 193, 676, 676), 'emotions': {'angry': 0.0, 'disgust': 0.0, 'fear': 0.0, 'happy': 0.58, 'sad': 0.02, 'surpris
e': 0.0, 'neutral': 0.4}}]
happy 0.58
```

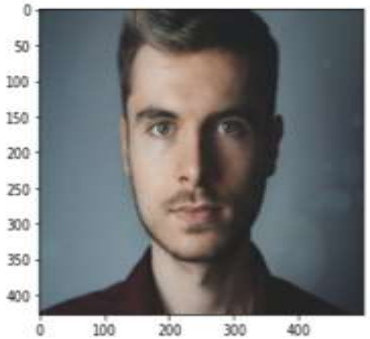


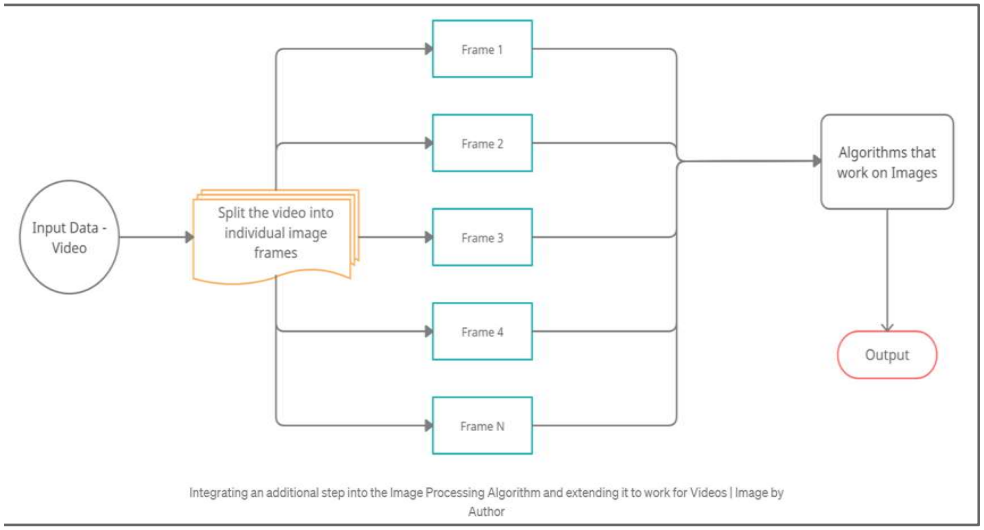
```
[{'box': (30, 48, 110, 110), 'emotions': {'angry': 0.0, 'disgust': 0.0, 'fear': 0.0, 'happy': 0.98, 'sad': 0.0, 'surprise':
0.0, 'neutral': 0.01}}]
happy 0.98
```



```
[{'box': (150, 75, 237, 237), 'emotions': {'angry': 0.78, 'disgust': 0.0, 'fear': 0.14, 'happy': 0.0, 'sad': 0.03, 'surpris
e': 0.01, 'neutral': 0.04}}]
angry 0.78
```

```
[{'box': (94, 47, 293, 293), 'emotions': {'angry': 0.01, 'disgust': 0.0, 'fear': 0.0, 'happy': 0.01, 'sad': 0.01, 'surpris
e': 0.0, 'neutral': 0.97}}]
neutral 0.97
```





Integrating an additional step into the Image Processing Algorithm and extending it to work for Videos | Image by Author

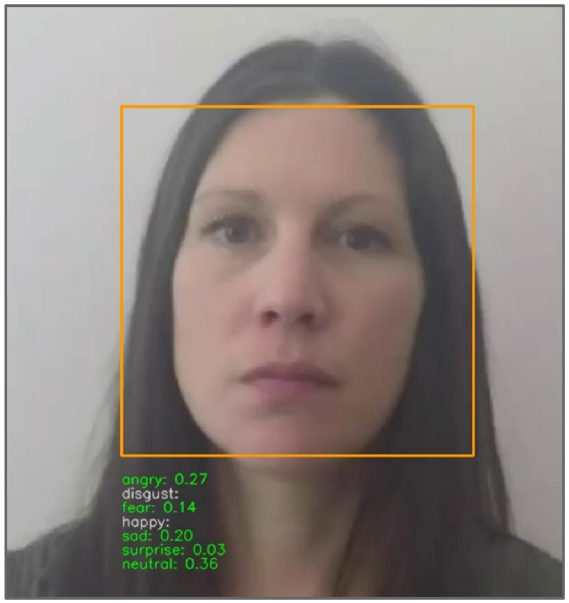
- The OS module in Python provides **functions for creating and removing** a directory (folder), fetching its contents, changing and identifying the current directory etc

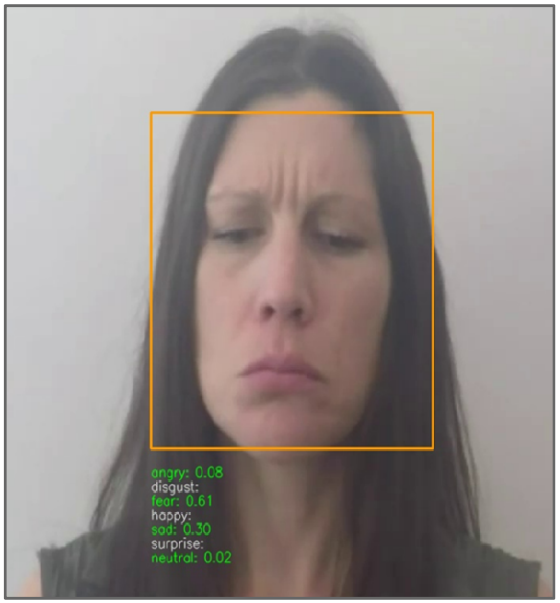
- The sys module in Python provides various functions and variables that are **used to manipulate different parts of the Python runtime environment**. It allows operating on the interpreter as it provides access to the variables and functions that interact strongly with the interpreter.

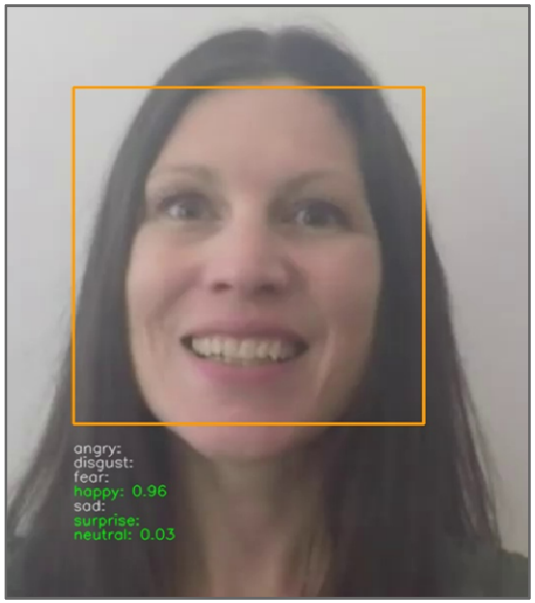
```
In [9]: from fer import Video
        from fer import FER
        import os
        import sys
        import pandas as pd
        # Put in the location of the video file that has to be processed
        location_videofile = "/content/Video_One.mp4"

In [4]: # But the Face detection detector
        face_detector = FER(mtcnn=True)
        # Input the video for processing
        input_video = Video(location_videofile)

WARNING:tensorflow:From /usr/local/lib/python3.7/dist-packages/tensorflow/python/keras/layers/normalization.py:534: _colocate_with (from tensorflow.pyth
on.framework.ops) is deprecated and will be removed in a future version.
Instructions for updating:
Colocations handled automatically by placer.
23-07-2021:11:46:34,263 WARNING  [deprecation.py:336] From /usr/local/lib/python3.7/dist-packages/tensorflow/python/keras/layers/normalization.py:534: _
colocate_with (from tensorflow.python.framework.ops) is deprecated and will be removed in a future version.
Instructions for updating:
Colocations handled automatically by placer.
/usr/local/lib/python3.7/dist-packages/tensorflow/python/keras/engine/training.py:2426: UserWarning: `Model.state_updates` will be removed in a future v
ersion. This property should not be used in TensorFlow 2.0, as `updates` are applied automatically.
  warnings.warn('`Model.state_updates` will be removed in a future version. '
```

Recognizing Facial Emotions from Live Videos

19.43 seconds video output
converted into 583 image frames



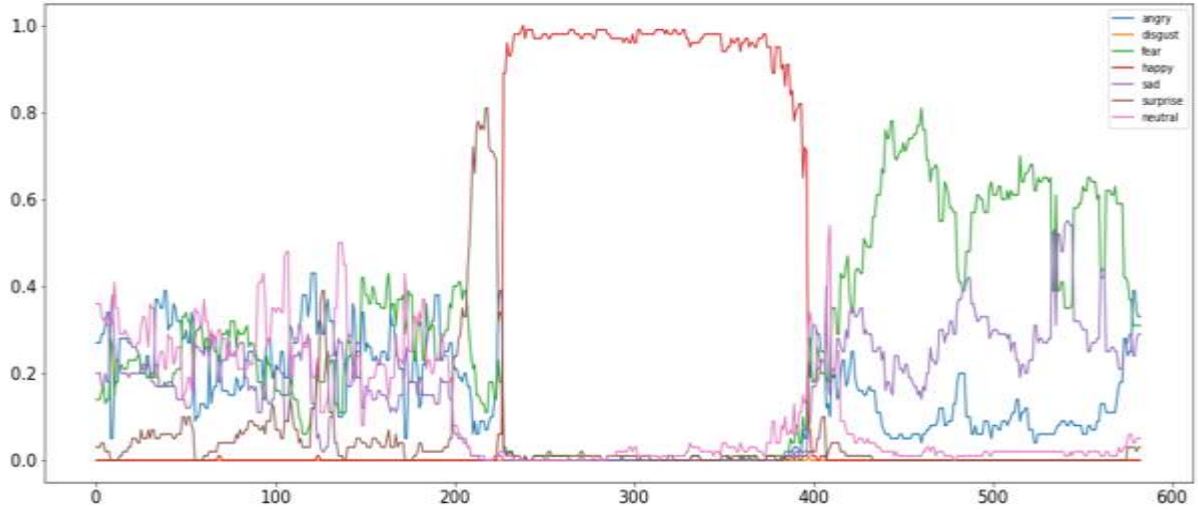




```
score_comparisons['Emotion Value from the Video'] = emotions_values
score_comparisons
```

```
  0%|                                                                                                    | 0/583 [00:00<?, ?frames/s]

05-01-2022:16:55:33,128 INFO     [classes.py:199] 30.00 fps, 583 frames, 19.43 seconds
05-01-2022:16:55:33,141 INFO     [classes.py:207] Making directories at output

100%|[redacted]| 583/583 [19:13<00:00,  1.98s/frames]

05-01-2022:17:14:46,333 INFO     [classes.py:320] Completed analysis: saved to output\Video_One_output.mp4
05-01-2022:17:14:46,333 INFO     [classes.py:327] Starting to Zip
05-01-2022:17:14:46,594 INFO     [classes.py:338] Compressing: 8%
05-01-2022:17:14:46,834 INFO     [classes.py:338] Compressing: 17%
05-01-2022:17:14:47,101 INFO     [classes.py:338] Compressing: 25%
05-01-2022:17:14:47,344 INFO     [classes.py:338] Compressing: 34%
05-01-2022:17:14:47,594 INFO     [classes.py:338] Compressing: 42%
05-01-2022:17:14:47,844 INFO     [classes.py:338] Compressing: 51%
05-01-2022:17:14:48,83 INFO     [classes.py:338] Compressing: 60%
05-01-2022:17:14:48,353 INFO     [classes.py:338] Compressing: 68%
05-01-2022:17:14:48,604 INFO     [classes.py:338] Compressing: 77%
05-01-2022:17:14:48,864 INFO     [classes.py:338] Compressing: 85%
05-01-2022:17:14:49,113 INFO     [classes.py:338] Compressing: 94%
05-01-2022:17:14:49,284 INFO     [classes.py:339] Zip has finished
```

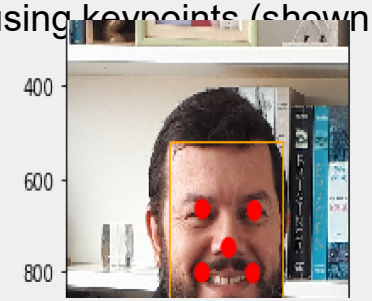
	Human Emotions	Emotion Value from the Video
0	Angry	76.78
1	Disgust	0.01
2	Fear	156.73
3	Happy	163.22
4	Sad	89.77
5	Surprise	28.33
6	Neutral	67.22



Future Enhancements

1) Improve the speed of analysing the video (currently Python takes about 30.00 fps).

Using boundary box displaying eyes, mouth, nose around the face using keypoints (shown below).



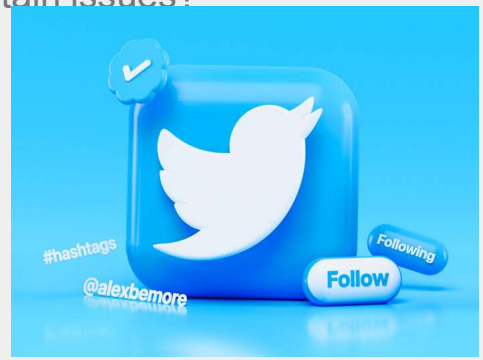
Run MTCNN on a GPU and use the sped-up version, it will achieve around 60–100 pictures/frames a second. That is a boost of up to **100 times**! To use MTCNN on a GPU you will need to set up CUDA, cudnn, pytorch and so on. Reference:

https://towardsdatascience.com/face-detection-using-mtcnn-a-guide-for-face-extraction-with-a-focus-on-speed-c6d59f82d49

2) Analyse public sentiments on the topic instead of just individual sentiment

Topic Modeling and Sentiment Analysis on Twitter Data Using Spark

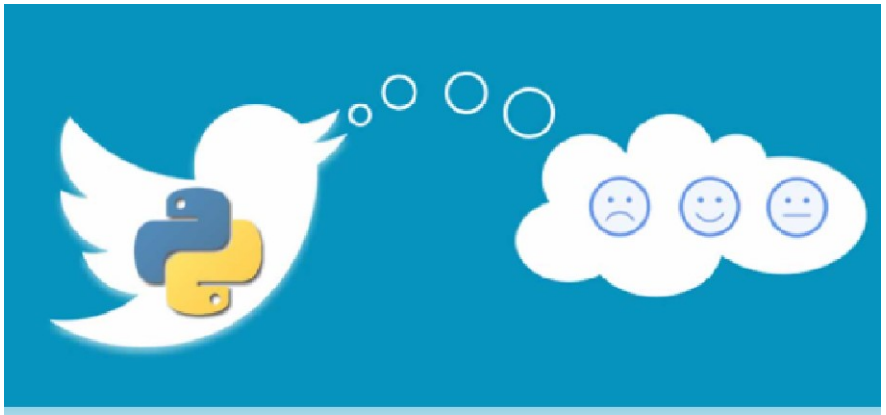
What are twitter users from a certain region concerned about and what are their reactions towards certain issues?



Recent study in June 2021. Reference: https://towardsdatascience.com/topic-modeling-and-sentiment-analysis-on-twitter-data-using-spark-a145bfcc433

Proof of Concept 2:

Twitter Sentiment Analysis
using Natural Language Processing (NLP)
in Python

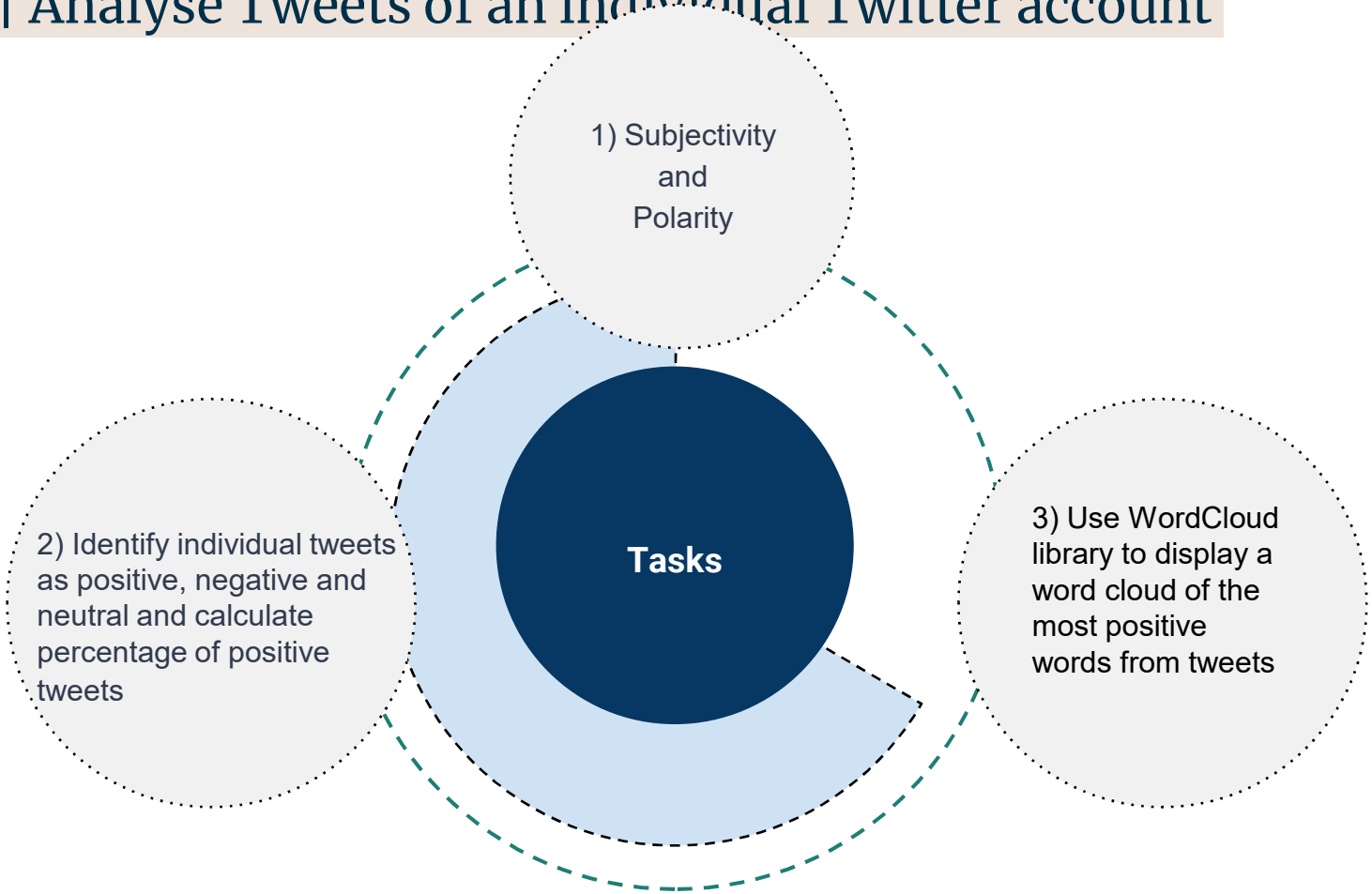


Twitter Sentiment Analysis| Tools

	Python
	Tweepy *Python library for accessing the Twitter API*
	TextBlob *Powerful natural language processing library for Python*
	WordCloud Visualise findings, display word cloud of relevant words from the tweets

	Pandas
	Numpy
	Matplotlib

Task | Analyse Tweets of an individual Twitter account



Steps by step solution

1

Using Python and install and import the following libraries: TextBlob, Tweepy, WordCloud, Pandas, Numpy and Matplotlib

```
(twitter_venv) $ conda install tweepy

(twitter_venv) $ conda install textblob

(twitter_venv) $ conda install wordcloud
```

2

Create Twitter Developer account to use the Twitter API (to access information from twitter)

3

Retrieve the last 50 Tweets & replies from the specified Twitter account. Eg. using "**elonmusk**" twitter account. Run on 3 Jan 2022, 9:30pm and create Pandas Data Frame

	Tweet
0	RT @CRAdvocacy: "The future of cars can and wi...
1	RT @MotorTrend: We put the 2020 @Tesla Model Y...
2	RT @SpaceX: Starman, last seen leaving Earth, ...
3	RT @SpaceX: Falcon 9 launches 60 Starlink sate...
4	RT @SpaceX: Deployment of 60 Starlink satellit...

Steps by step solution (Cont')

4

Clean the tweets and remove mentions, hashtags, retweets, URLls ; apply it for all Tweets in Pandas data frame

```
# Cleaning the tweets

def cleanUpTweet(txt):
    # Remove mentions
    txt = re.sub(r'@[A-Za-z0-9_]+', '', txt)
    # Remove hashtags
    txt = re.sub(r'#', '', txt)
    # Remove retweets:
    txt = re.sub(r'RT : ', '', txt)
    # Remove urls
    txt = re.sub(r'https?:\/\/[A-Za-z0-9\.\/]+', '', txt)
    return txt
```

```
df['Tweet'] = df['Tweet'].apply(cleanUpTweet)
```

5

Build a couple more functions to calculate the subjectivity and polarity of our tweets using TextBlob and apply these functions to out data frame and create two new data frame: Subjectivity and Polarity

```
def getTextSubjectivity(txt):
    return TextBlob(txt).sentiment.subjectivity

def getTextPolarity(txt):
    return TextBlob(txt).sentiment.polarity
```

```
df['Subjectivity'] = df['Tweet'].apply(getTextSubjectivity)
df['Polarity'] = df['Tweet'].apply(getTextPolarity)
```

TextBlob

Steps by step solution (Cont')

This is how the data frame looks like.

Score is calculated for subjectivity and polarity.

	Tweet	Subjectivity	Polarity
0	"The future of cars can and will be electric i...	0.450000	0.100000
1	We put the 2020 Model Y Long Range electric S...	0.400000	-0.050000
2	Starman, last seen leaving Earth, made its fir...	0.200000	0.125000
3	Falcon 9 launches 60 Starlink satellites to or...	0.166667	-0.166667
4	Deployment of 60 Starlink satellites confirmed	1.000000	0.400000
5	4th flight & landing for this booster	0.000000	0.000000
6	5 minutes from launch. Looks good so far.	0.800000	0.400000
7	Turn volume to 11 & play Powerglide in you...	0.000000	0.000000
8	Music volume on a Tesla goes to 11, because it...	0.000000	0.000000
9	Rewatched Young Frankenstein this weekend. Sti...	0.700000	0.550000
10	Hollywood should just start with the sequel &a...	0.750000	0.375000
11	New SpaceX droneship will be called "A Shortfa...	0.454545	0.136364
12	Nothing can escape gravity, not even black hol...	0.316667	0.041667
13	Call of Booty, great game	0.575000	0.200000
15	Once this Dragon docks to the , there will be ...	0.000000	0.000000
16	SpaceX training of astronauts that will fly on...	0.544444	0.391667
17	After launch of Crew Dragon's first operationa...	0.166667	0.125000
18	NEWS: We're now targeting 2:40am ET Saturday, ...	0.000000	0.000000
20	Thanks Tesla Team for great work on deliveries...	0.517803	0.417424

Image by author

Steps by step solution (Cont')

6

Build function and categorise tweets into Positive, Negative, Neutral and create another feature called Score

The following is the data frame with our tweets, subjectivity, polarity and score for all tweets.

Visualise tweets using Matplotlib

	Tweet	Subjectivity	Polarity	Score
0	"The future of cars can and will be electric i...	0.450000	0.100000	Positive
1	We put the 2020 Model Y Long Range electric S...	0.400000	-0.050000	Negative
2	Starman, last seen leaving Earth, made its fir...	0.200000	0.125000	Positive
3	Falcon 9 launches 60 Starlink satellites to or...	0.166667	-0.166667	Negative
4	Deployment of 60 Starlink satellites confirmed	1.000000	0.400000	Positive
5	4th flight & landing for this booster	0.000000	0.000000	Neutral
6	5 minutes from launch. Looks good so far.	0.800000	0.400000	Positive
7	Turn volume to 11 & play Powerglide in you...	0.000000	0.000000	Neutral
8	Music volume on a Tesla goes to 11, because it...	0.000000	0.000000	Neutral
9	Rewatched Young Frankenstein this weekend. Sti...	0.700000	0.550000	Positive
10	Hollywood should just start with the sequel &a...	0.750000	0.375000	Positive
11	New SpaceX droneship will be called "A Shortfa...	0.454545	0.136364	Positive
12	Nothing can escape gravity, not even black hol...	0.316667	0.041667	Positive
13	Call of Booty, great game	0.575000	0.200000	Positive
15	Once this Dragon docks to the , there will be ...	0.000000	0.000000	Neutral
16	SpaceX training of astronauts that will fly on...	0.544444	0.391667	Positive
17	After launch of Crew Dragon's first operationa...	0.166667	0.125000	Positive
18	NEWS: We're now targeting 2:40am ET Saturday, ...	0.000000	0.000000	Neutral
20	Thanks Tesla Team for great work on deliveries...	0.517803	0.417424	Positive

Steps by step solution (Cont')

4

Clean the tweets and remove mentions, hashtags, retweets, URLls ; apply it for all Tweets in Pandas data frame

```
# Cleaning the tweets

def cleanUpTweet(txt):
    # Remove mentions
    txt = re.sub(r'@[A-Za-z0-9_]+', '', txt)
    # Remove hashtags
    txt = re.sub(r'#', '', txt)
    # Remove retweets:
    txt = re.sub(r'RT : ', '', txt)
    # Remove urls
    txt = re.sub(r'https?:\/\/[A-Za-z0-9\.\/]+', '', txt)
    return txt
```

```
df['Tweet'] = df['Tweet'].apply(cleanUpTweet)
```

5

Build a couple more functions to calculate the subjectivity and polarity of our tweets using TextBlob

```
def getTextSubjectivity(txt):
    return TextBlob(txt).sentiment.subjectivity

def getTextPolarity(txt):
    return TextBlob(txt).sentiment.polarity
```

```
df['Subjectivity'] = df['Tweet'].apply(getTextSubjectivity)
df['Polarity'] = df['Tweet'].apply(getTextPolarity)
```

TextBlob

6

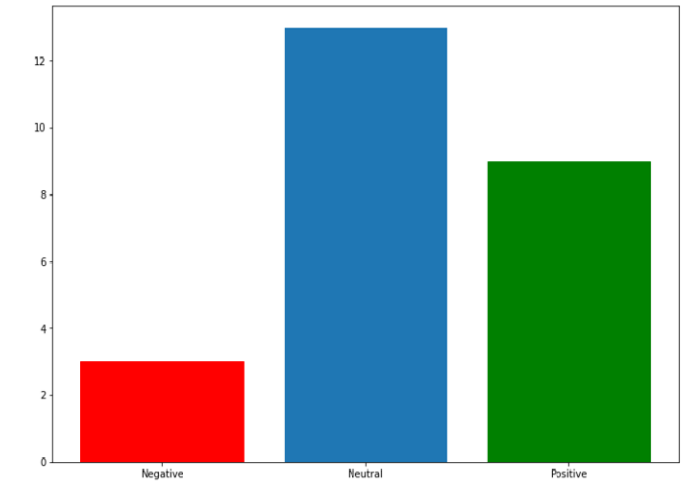
Visualise Positive, Negative, Neutral tweets using Matplotlib

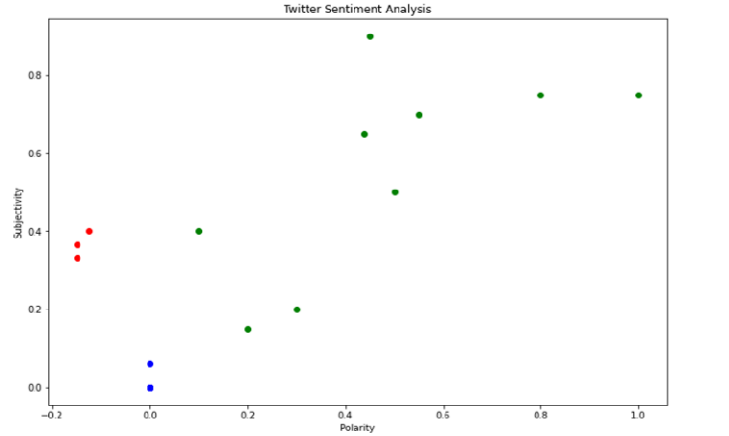
36.0 % of positive tweets

48.0 % of objective tweets

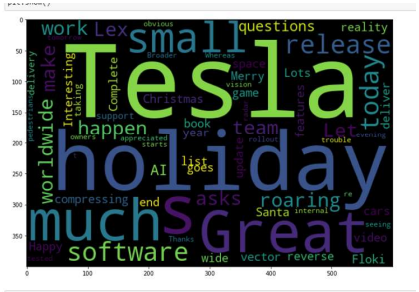
	Tweet	Subjectivity	Polarity	Score
1	Let's make the roaring 20's happen!	0.000000	0.0000	Neutral
2	Great work by Tesla team worldwide!	0.750000	1.0000	Positive
5	Lex asks great questions	0.750000	0.8000	Positive
6	So much of AI is about compressing reality to ...	0.333333	-0.1500	Negative
8	Floki Santa	0.000000	0.0000	Neutral
9	Merry Christmas 🎄🎁🎅⭐😃	0.000000	0.0000	Neutral
10	Interesting\n	0.500000	0.5000	Positive
11	Complete list of features here. Happy holidays...	0.700000	0.5500	Positive
12	Tesla holiday software update goes to wide rel...	0.400000	-0.1250	Negative
13	The book is 👍	0.000000	0.0000	Neutral
14	Lots of Tesla cars to deliver before year end!...	0.150000	0.2000	Positive
15	Whereas radar has trouble seeing small pedestr...	0.366667	-0.1500	Negative
16	Tesla holiday software release being tested wi...	0.000000	0.0000	Neutral
17	Stage separation confirmed! The Dragon is now...	0.900000	0.4500	Positive
18	100th SpaceX rocket landing was today	0.000000	0.0000	Neutral
19	Has anyone seen web3? I can't find it.	0.000000	0.0000	Neutral
20	🤣🤣 True. Ask your airline for Starlink!	0.650000	0.4375	Positive
21	For those wondering, I will pay over $11 billi...	0.000000	0.0000	Neutral
22	Given the almost unimaginable nature of the pr...	0.062500	0.0000	Neutral
23	Should be taught to all at a young age	0.400000	0.1000	Positive
24	Two Falcon 9 launches completed in ~15 hours	0.000000	0.0000	Neutral
25	traceroute woke_mind_virus	0.000000	0.0000	Neutral
26	Watch Falcon 9 launch 52 Starlink satellites t...	0.000000	0.0000	Neutral
27	Silco & Jinx headline Coachella?	0.000000	0.0000	Neutral
29	Tesla FSD 10.8 plus holiday fun software relea...	0.200000	0.3000	Positive

```
plt.show()
```





WordCloud

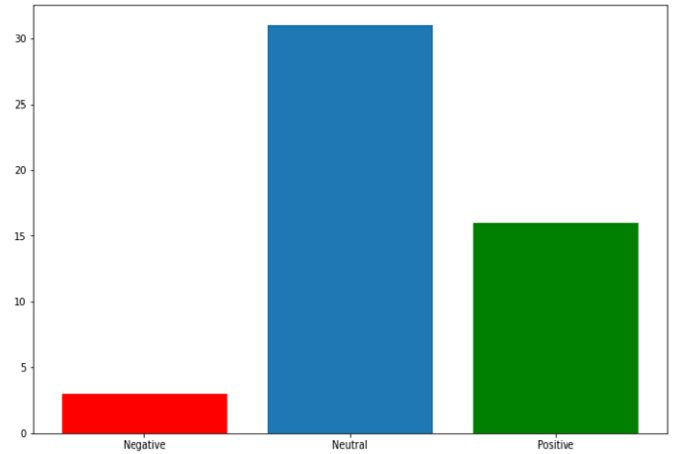


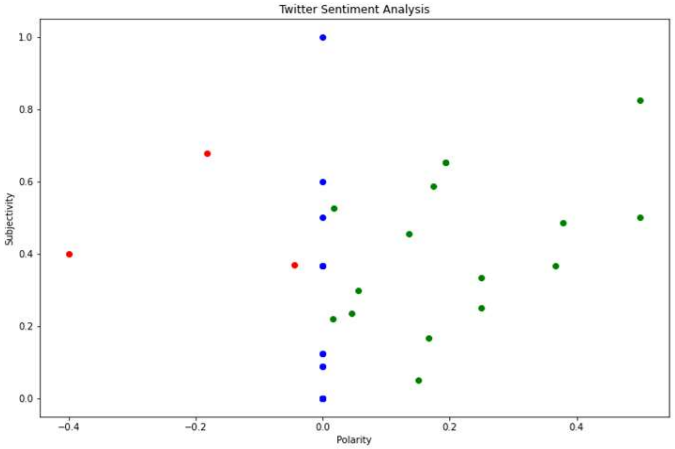
- twitterAccount eg. "**crypto**".
- Run on 6 Jan 2022, 4:29pm

32.0 % of positive tweets

42.0 % of objective tweets

	Tweet	Subjectivity	Polarity	Score
0	UPDATE: Bitcoin slumps to the lowest level sin...	0.000000	0.000000	Neutral
1	JUST IN: Security concerns — not the volatilit...	0.000000	0.000000	Neutral
2	A pair of scions from Singapore's wealthiest f...	0.220833	0.016667	Positive
3	BTCS, a small Nasdaq-listed blockchain company...	0.366667	0.000000	Neutral
4	The system of stock ownership runs on differen...	0.600000	0.000000	Neutral
5	Bitcoin mining stocks are taking a beating as ...	0.125000	0.000000	Neutral
6	BTCS, a small Nasdaq-listed blockchain company...	0.366667	0.000000	Neutral
7	Ethereum's dominance in one of the hottest cor...	0.500000	0.000000	Neutral
8	NEW: A pair of scions from Singapore's wealthi...	0.298737	0.056566	Positive
9	Bitcoin will continue to take market share fro...	0.000000	0.000000	Neutral
10	Bitcoin's rut is explained in three key charts...	1.000000	0.000000	Neutral
11	Bitcoin mining stocks are taking a beating as ...	0.125000	0.000000	Neutral
12	Bloomberg Opinion's Matt Levine weighs in on t...	0.000000	0.000000	Neutral
13	Opinion | The critics of the cringeworthy Matt...	0.500000	0.500000	Positive
14	Cursed images from CES, a thread	0.000000	0.000000	Neutral
15	Bitcoin slumped to the lowest level since its ...	0.000000	0.000000	Neutral
16	Barry Sternlicht, the chairman of Starwood Cap...	0.050000	0.150000	Positive
17	Bitcoin is plunging	0.000000	0.000000	Neutral
18	Miami's Francis Suarez takes over leadership o...	0.000000	0.000000	Neutral
19	NEW: Bitcoin mining stocks are taking a beatin...	0.234848	0.045455	Positive
20	Bloomberg's Eddie van der Walt says 2021 was a...	0.000000	0.000000	Neutral
21	New investment funds aim to piggyback on the h...	0.652273	0.193182	Positive
22	Bitcoin may be about to rally again, after lan...	0.000000	0.000000	Neutral
23	The world's first public-company Bitcoin divid...	0.333333	0.250000	Positive
24	At TheYearAhead don't miss 's conversation wit...	0.000000	0.000000	Neutral
25	Bitcoin miners were lured to Kazakhstan by the...	0.825000	0.500000	Positive





Future Enhancements

After python coding is done…
Step 1: Authentication with twitter API

```python
class TwitterClient(object):
    '''
    Generic Twitter Class for sentiment analysis.
    '''
    def __init__(self):
        '''
        Class constructor or initialization method.
        '''
        # keys and tokens from the Twitter Dev Console
        consumer_key = 'VZ6wUmKm67OzDzcYVOLbyMq0b'
        consumer_secret = 'QLCj1PNP4mT1a2rmIT0j0OJnvqTo275EZiyNJejlklF08aZgyU'
        access_token = '1473518480825618434-dZoFIbJB5xd38Is3uDjYJxWrpyBD0J'
        access_token_secret = 'fmnXeOjbVwyvfsLrzyENcKYCMdCrPBsbR1Qb1Y5lUbuxn'

        # attempt authentication
        try:
            # create OAuthHandler object
            self.auth = OAuthHandler(consumer_key, consumer_secret)
            # set access token and secret
            self.auth.set_access_token(access_token, access_token_secret)
            # create tweepy API object to fetch tweets
            self.api = tweepy.API(self.auth)
        except:
            print("Error: Authentication Failed")
```

Step 2

- Enter requirement for search eg. date created, search term.
- Analyse sentiment of tweets with textblob (NLP)

TextBlob

```python
def get_tweet_sentiment(self, tweet):
    '''
    Utility function to classify sentiment of passed tweet
    using textblob's sentiment method
    '''
    # create TextBlob object of passed tweet text
    analysis = TextBlob(self.clean_tweet(tweet))
    # set sentiment
    if analysis.sentiment.polarity > 0:
        return 'positive'
    elif analysis.sentiment.polarity == 0:
        return 'neutral'
    else:
        return 'negative'

    #df['Subjectivity'] = df['Tweet'].apply(getTextSubjectivity)
    #df['Polarity'] = df['Tweet'].apply(getTextPolarity)
```

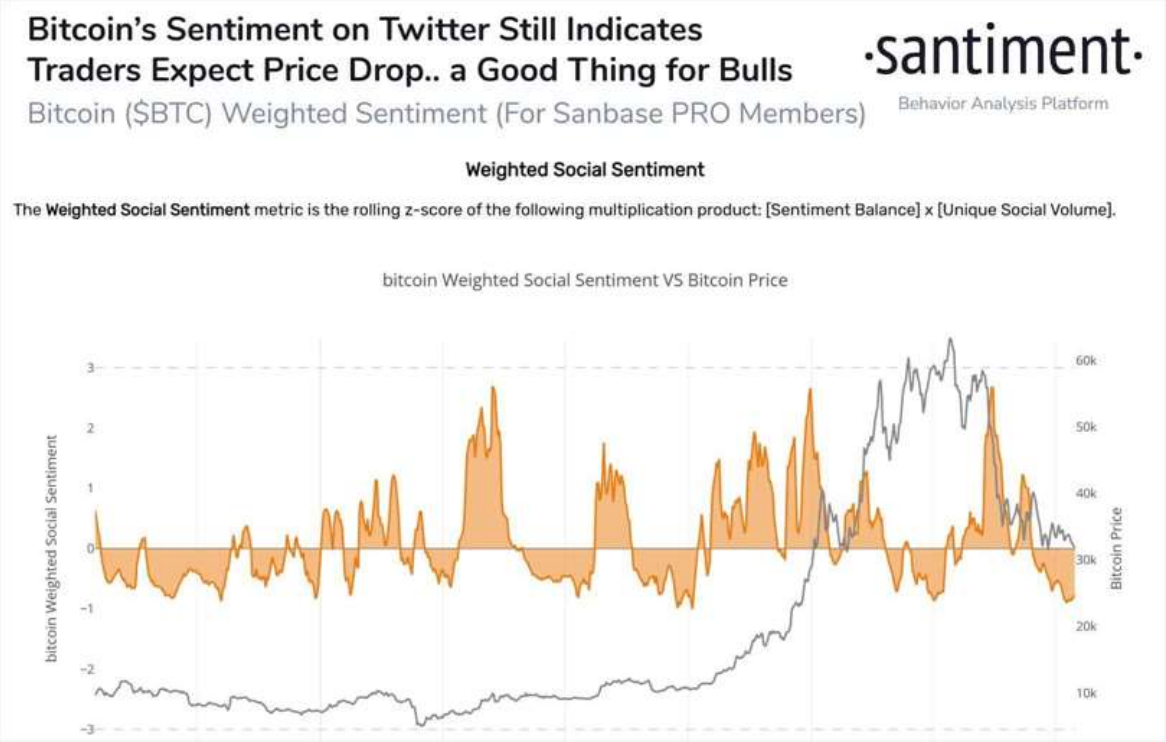
```python
def get_tweets(self, query, lang, until, count = 100000):
    '''
    Main function to fetch tweets and parse them.
    '''
    # empty list to store parsed tweets
    tweets = []

    try:
        # call twitter api to fetch tweets

        #fetched_tweets = self.api.search_tweets(q = query, count = count)
        fetched_tweets = self.api.search_tweets(q=query, lang=lang,until=until,count=count)
```

Step 3:

- Extract data from semi-structured data (Json file) received via twitter api
- Export results to excel for data analysis

	Sentiment	Tweet	Date Created
2	neutral	RT @VLaunchCOM: ðŸ"£INVESTOR ANNOUNCEMENTðŸ"£	2021-12-18 23:59:59+00:00
3	positive	Interesting read on the rise of the networked peer2peer state	2021-12-18 23:59:58+00:00
4	positive	@CryptoSenorita Invest in #BITCOIN safely & securely with the world's leading crypto exchange #BITGO. Kindly inbox me	2021-12-18 23:59:57+00:00
5	positive	RT @VLaunchCOM: ðŸ"£ INVESTOR ANNOUNCEMENT ðŸ"£	2021-12-18 23:59:57+00:00
6	neutral	RT @elainedtaylor: 3 Minute Gold News -	2021-12-18 23:59:56+00:00
7	neutral	RT @VLaunchCOM: ðŸ"£INVESTOR ANNOUNCEMENTðŸ"£	2021-12-18 23:59:56+00:00
8	positive	Thanks to China Antifragile #Bitcoin experienced, overcame and became more resilient after another failed attack. https://t.co/jzgbvy2C7z	2021-12-18 23:59:56+00:00
9	negative	RT @saylor: #Turkish currency has lost 55% of its value this year & 37% in the last 30 days. What follows is a parade of horribles - capitaâ€¦	2021-12-18 23:59:55+00:00
10	neutral	RT @NftDropsCal: â° #GIVEAWAY for 48 HOURS	2021-12-18 23:59:55+00:00
11	neutral	RT @Brovvneee: Giving away an @adidasoriginals NFT token from todays drop with @BoredApeYC and @pixelvault_ ðŸ¥³	2021-12-18 23:59:54+00:00
12	neutral	RT @MMCrypto: Sometimes I think I should take a break from making content and just relax and take time for myself. Then I wake up and I feeâ€	2021-12-18 23:59:54+00:00
13	neutral	Surrounded by ðŸ' in the ðŸŒ ðŸŒŽ	2021-12-18 23:59:54+00:00
14	positive	RT @APompliano: This is pretty funny. You can automate bitcoin purchases every time @PeterSchiff tweets about #bitcoin	2021-12-18 23:59:53+00:00
15	neutral	RT @MMCrypto: Sometimes I think I should take a break from making content and just relax and take time for myself. Then I wake up and I feeâ€	2021-12-18 23:59:53+00:00
16	neutral	RT @gaborgurbacs: $46,000 Bitcoin in February vs $46,000 Bitcoin in December https://t.co/pesrDKyQMA	2021-12-18 23:59:52+00:00
17	neutral	RT @40BC40: Giving away an @adidasoriginals NFT token from todays drop with @BoredApeYC and @pixelvault_ ðŸ¥³	2021-12-18 23:59:51+00:00
18	neutral	Bitcoin: Puell Multiple	2021-12-18 23:59:49+00:00
19	neutral	Current Bitcoin transaction fees:	2021-12-18 23:59:49+00:00

Such as trend analysis and prediction on a company's product



Conclusion of the Twitter S .entiment analysis using Python

- Doing sentiment analysis of the tweets enabled us to calculate numerical values of subjectivity and polarity → This could help us to understand better this Twitter account in terms of the language that is being used

- Combining this with additional information about likes and comments can be very useful from marketing point of view and enable us to find some correlation between subjectivity, polarity and the engagement our the users for a specified Twitter account

- We can experiment more with this example and come up with some more ideas how this can be used in practice

Thank you

End of Presentation