16-02-2-2020 – 23-02-2020

Face recognition implementation could be continued as an improvement of the design project. The final state of the design project was over 80% accuracy with features of Siamese network and support vector machine. The next target is to implement a neural network for the classification for the features provided by Siamese network.
Improvements has to be done to improve the accuracy of predictions under lightening changes by adding image augmentations

Started researching for background of number plate detection.
According to the literature [1] the major steps for recognizing number on a number plate 3 major tasks has to be done.

- Licence plate detection

Using object detection number plate has to be detected from the camera. Yolo object detection can be used. Images needed to be captured of number plates and trained.

- License plate segmentation

Characters of the number plate has to be segmented
In order to segment the image input image, need to be

- Converted from RGB to gray
- Binarize image
- Blur image
- Segmentation of letters

Histogram of pixel projection
Horizontal projection is used to find the top and bottom positions of the characters. The value of group of histograms is sum of white pixels along a particular line in the horizontal direction. Get upper and lower limits. After getting all the lines in horizontal direction horizontal projection histogram is obtained. Average of histogram is used as threshold to get lower and upper values. The area having histogram greater than threshold is the area recognized to be in the delimited by the limits. Vertical projection histogram is also determined likewise for vertical lines.

- License plate recognition

The segmented characters have to be recognized using neural network or machine learning characters has to be identified. Each character has to be cropped individually and resized to

the same size with some pre-processing techniques. Then each character needs to be recognized.
23-02-2019 – 08-03-2019

Prepared Project brief
Prepared Project proposal
Participated in viva
Prepared WBS

08-03-2019

Things considered when making rating system [2]

- Number of views
- How favorable the reviews are
- How old or recent the reviews are

Existing 5-star rating systems are working based on the average of the rating provided by users.
Objective of a rating system

- Get the true reflection of user experience
- To get correct decision
- Filter out spams
- Hard to cheat the used

Type of rating
Could be detailed or non-detailed ratings
Length of text, presence of photos, number of photos associated , social sharing enables number of fields filled has to be considered for a rating system on a platform.

User credibility

All users don't contribute ratings correctly. Users has to be segmented (high/mid/low) according to their contribution and usefulness to other users. Contribution over a time period kind of content they add can be used to segment them

Time recency
A businesses' state change for every few months. Their service punctuality may change with the time. Therefore, the older comments have to be given a different weight compared to the present comments.

Cold start problem
Newly added hotels restaurants divers has a problem. When starting with low number of users, can't make rating as no enough number of ratings and also if a place has no rating no one will go to that place. To solve this,
Give average rating to all new entrants. With new comments they will go up or down
Request ratings from experts and critics at the initial stage
Indicate new entrants to the user so they get a chance to serve the user and increase their rating

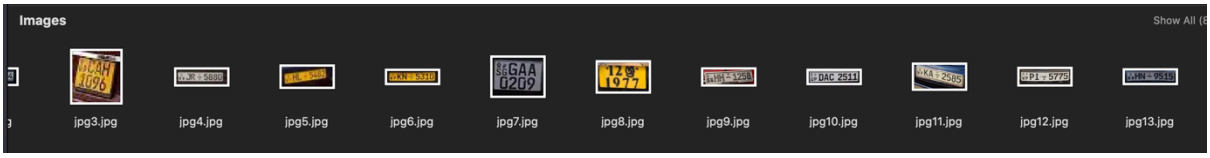
08-03-2020 – 22-03-2020

Study week for mid exam
Studied basics of react-native and android
Studied about number plate detection using YOLO v3 and using it in python.
Studies about other object detection algorithms like R-CNN and haar cascades

22-03-2020 – 29-03-2020

Decided to use YOLO V3 invented. By by Joseph Redmon as,
It's the most cutting-edge object detection network faster than all others
In this, when predicting an image in only once passed thought the network therefore it's very fast than others. it processes images in real-time at 45 frames per second.

Making dataset
Images of number plates were collected from the internet as with the current situation can't collect number plates photos on roads. Only 110 images could be collected. All of them were cropped and taken the only the number plate for creating the dataset.



Each image has to be processed through a software called "LableImage" to make bound boxes around number plates and label them. Then create a text file containing coordinates of the bound box which supports by the Yolo network.

Bounding boxes were drawn around and annotation was saved as “LP” to indicate all the images belong to the same class “LP”. Class names were stored in a configuration file.

Training the network - first try

After cropping and annotating images few configurations had to be made in network configuration file. Number of classes were changed to 1 and number of filters were changed to 18.
Then using the original c code provided by the implementers of YOLO training was started with 80 training and 25 test images.
Training was continued for 2.5 hours but the accuracy didn’t increase more than 0.5. Therefore, training was cancelled and the first attempt failed.

```
Asiris-MacBook-Air:darknet asirigawesha$ ./darknet detector train custom/darknet.data custom/yolov3.cfg darknet53.conv.74
yolov3
layer     filters    size              input                output
    0 conv     32  3 x 3 / 1   608 x 608 x   3   ->   608 x 608 x  32  0.639 BFLOPs
    1 conv     64  3 x 3 / 2   608 x 608 x  32   ->   304 x 304 x  64  3.407 BFLOPs
    2 conv     32  1 x 1 / 1   304 x 304 x  64   ->   304 x 304 x  32  0.379 BFLOPs
    3 conv     64  3 x 3 / 1   304 x 304 x  32   ->   304 x 304 x  64  3.407 BFLOPs
```

```
Region 82 Avg IOU: 0.123470, Class: 0.333861, Obj: 0.359963, No Obj: 0.494121, .5R: 0.000000, .75R: 0.000000,  count: 4
Region 94 Avg IOU: nan, Class: nan, Obj: nan, No Obj: 0.513186, .5R: nan, .75R: nan,  count: 0
Region 106 Avg IOU: nan, Class: nan, Obj: nan, No Obj: 0.546560, .5R: nan, .75R: nan,  count: 0
Region 82 Avg IOU: 0.251710, Class: 0.405589, Obj: 0.518059, No Obj: 0.493592, .5R: 0.000000, .75R: 0.000000,  count: 4
Region 94 Avg IOU: nan, Class: nan, Obj: nan, No Obj: 0.511069, .5R: nan, .75R: nan,  count: 0
Region 106 Avg IOU: nan, Class: nan, Obj: nan, No Obj: 0.543041, .5R: nan, .75R: nan,  count: 0
^C
Asiris-MacBook-Air:darknet asirigawesha$
```

Reason : must be the cropping which was not necessary. And the quality of the images were very low. Have to go deeper into network and study it

Finding the reason

- As per literature the reasons might be
- Dataset was prepared by cropping all the number plates. But it should not have been done. Images with vehcles having number plates could have been annotated using the “LableImage” software without cropping the number plates. [3]

- Low quality images – Downloader pictures were in different sizes and resolutions and some gets blued when resized to 608*608 as the network accepts only images resized to that size. [3]
- Low number of data – At least 1000 images are required for good results. [3]

Training 608*608 sized 1000 images is not possible to be done on a CPU. It requires at least a mid-range GPU. Therefore, have to leave it behind to be done afterward [3]

As an alternative, a pre-trained model of YOLO v3 for license plates in Australia was found [1] and this could recognize Sri Lankan number plates which are single lined plates that are on most of the cars, accurately.

Correctly recognized images using the pre-trained model.

But it couldn’t recognize the following type number plates

How data is annotated [3]

When annotated a image each line in the created text file contains information of a bounding box

```
<object-class-id> <center-x> <center-y> <width> <height>
```

object-class-id – an integer representing class of object from 0 to any number of classes as we need. For license plate only 1 class named "LP" will be available. Therefore, class is set to 0
center -x,y – coordinates of the center of the bounding box/image width and height
width, height – width and height of bounding box/ image width and height

YOLOv3 configuration parameters

```
# batch=1
# subdivisions=1
# Training
batch=64
subdivisions=16
```

Batch – the subset size of the images to be used in one iteration of training
Batch=64 means 64 pictures will be used at once for an iteration.

Subdivision – As GPU can't take all images in a batch a subset has to be taken at once to the GPU. Subdivision decides how many images from the 64 images will be considered once.
For testing this will be always 1.

```
width=608
height=608
channels=3
```

Width and height is image height and width. Channels is if RGB or grayscale. If 1 grayscale, 3 RGB.

Momentum and decay

```
momentum=0.9
decay=0.0005
```

Momentum controls the fluctuation of weights changing between iterations.
Overfitting a network means, its well trained on training data but can't predict test data well.
To control this, penalize large value for weights.
Decay controls this penalty term. If model keeps overfitting this could be changed.

```
learning_rate=0.001
policy=steps
steps=3800
scales=.1
```

```
burn_in=400
```

Learning rate – controls how aggressively the learning should be done. Normally 0.01 and 0.0001.
At the beginning as information is zero a higher learning rate should be applied and then should decrease over the time.
Policy – decides how often learning rate should decrease.
Steps – decides for how many iterations learning rate should be constant.

Scales – after the mentioned no of steps learning rate will be multiplied by the scale value and change the learning rate
Burn in – Makes the learning rate lower for a short period. If its 400, learning rate will decrease at 400^{th} step and increase it after sometime.

Next week's plan

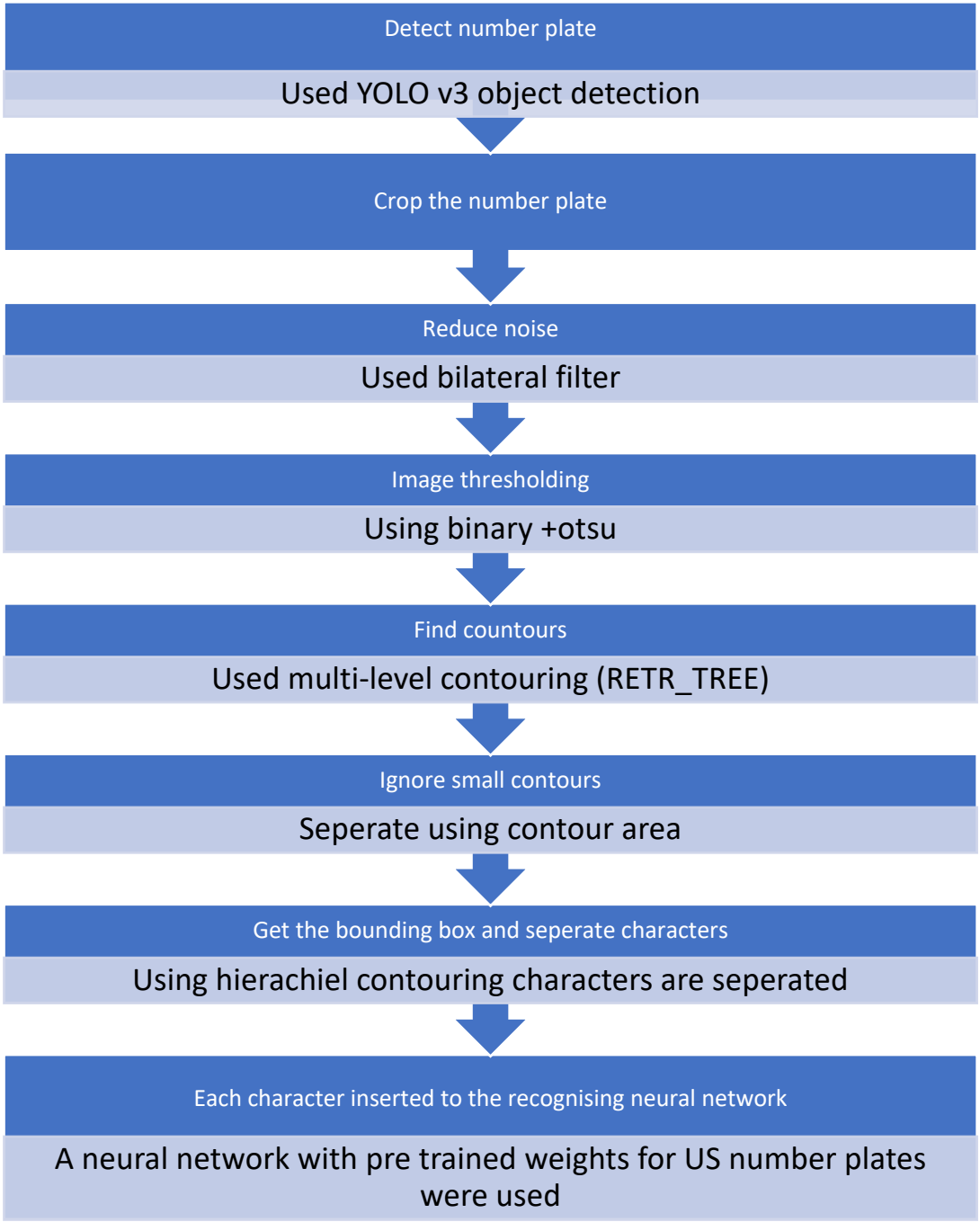
Work on making the implemented number plate detection code as a python backend and make a android front end and make communication between them

Work on implementing number plate recognition.

29-03-2020 – 05-04-2020

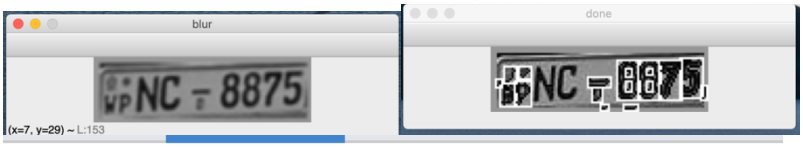
A code was implemented to read an image and detect the number plate crop the number plate segment the numbers and recognize the numbers.

The flow of the code as follows



Results were not bad even though the network was never trained for Sri Lankan number plates. After many tries with some preprocessing the results were up to an acceptable accuracy. A major weakness of the network is that it can't recognize letter "w".

Before noise reduction




```
Prediction : K , Precision : 0.56691813
Prediction : P , Precision : 0.99996567
Prediction : N , Precision : 0.9999852
Prediction : C , Precision : 0.5246444
Prediction : 8 , Precision : 0.4466316
Prediction : 8 , Precision : 0.8161647
Prediction : 7 , Precision : 0.66791165
Prediction : 5 , Precision : 0.49112922
```

After noise reduction

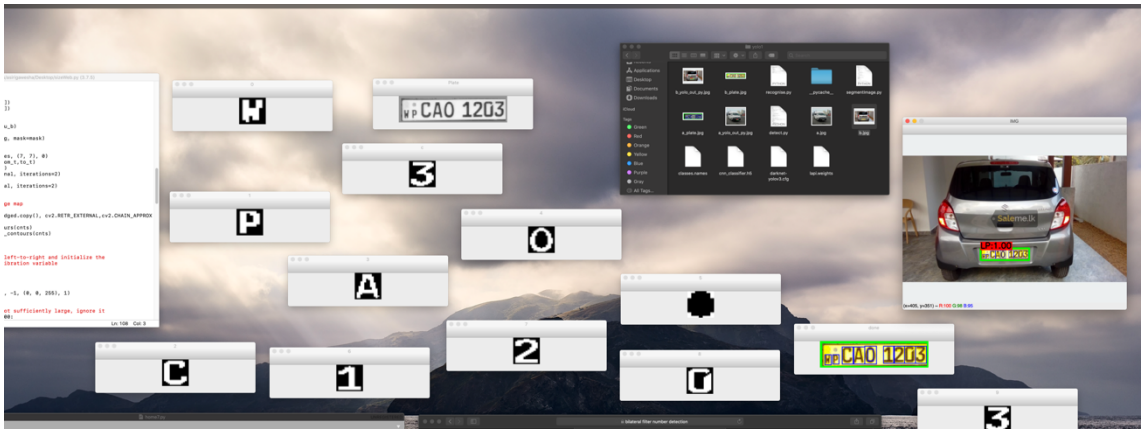

```
Prediction : K , Precision : 0.4622897
Prediction : P , Precision : 0.9993656
Prediction : N , Precision : 0.99999154
Prediction : C , Precision : 0.5246444
Prediction : 8 , Precision : 0.66641563
Prediction : 8 , Precision : 0.8812852
Prediction : 7 , Precision : 0.66791165
Prediction : 5 , Precision : 0.49112922
```

After reducing noise, contours drawn over unwanted parts were removed and the accuracy has improved.

Results of recognition


```
Prediction : K , Precision : 0.9999654
Prediction : L , Precision : 0.6472955
Prediction : G , Precision : 0.28558326
Prediction : S , Precision : 0.4134014
Prediction : G , Precision : 0.32874984
```

Network has a problem in recognizing some numbers



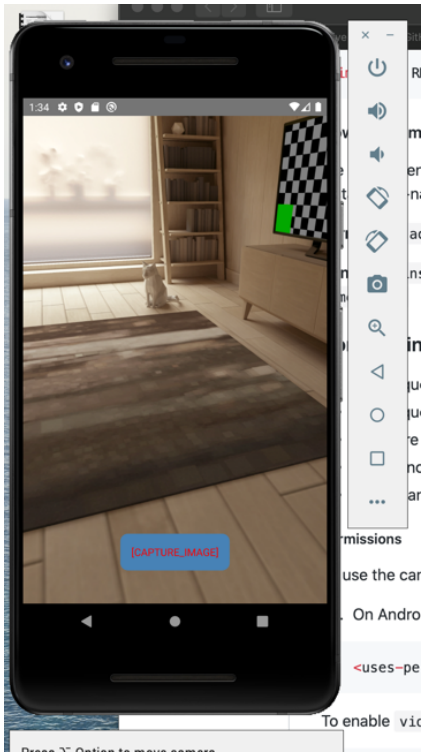

```
Prediction : H , Precision : 0.55186814
Prediction : P , Precision : 0.9959998
Prediction : C , Precision : 0.9641992
Prediction : A , Precision : 0.85150486
Prediction : O , Precision : 0.85735446
Prediction : L , Precision : 0.44148165
Prediction : 1 , Precision : 0.99918073
Prediction : 2 , Precision : 0.999246
Prediction : U , Precision : 0.7339003
Prediction : 3 , Precision : 0.9997789
```

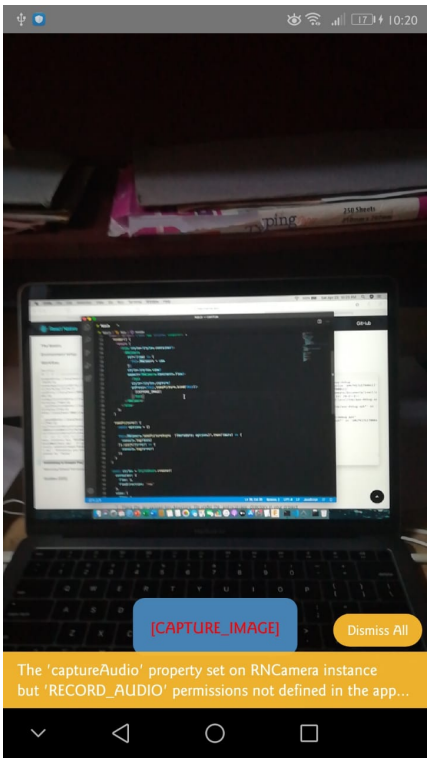
For further improvements of the accuracy it has to be trained for Sri Lankan number plate characters.

06-04-2020- 26-4-2020

Amidst the mid and assignments studied basics of react native and android.
Also worked on Flask server development on python as a method to need to be implemented to open the camera of the mobile phone, upload image to a server, process it with python code, upload the result to server, send the result to mobile phone and display it.

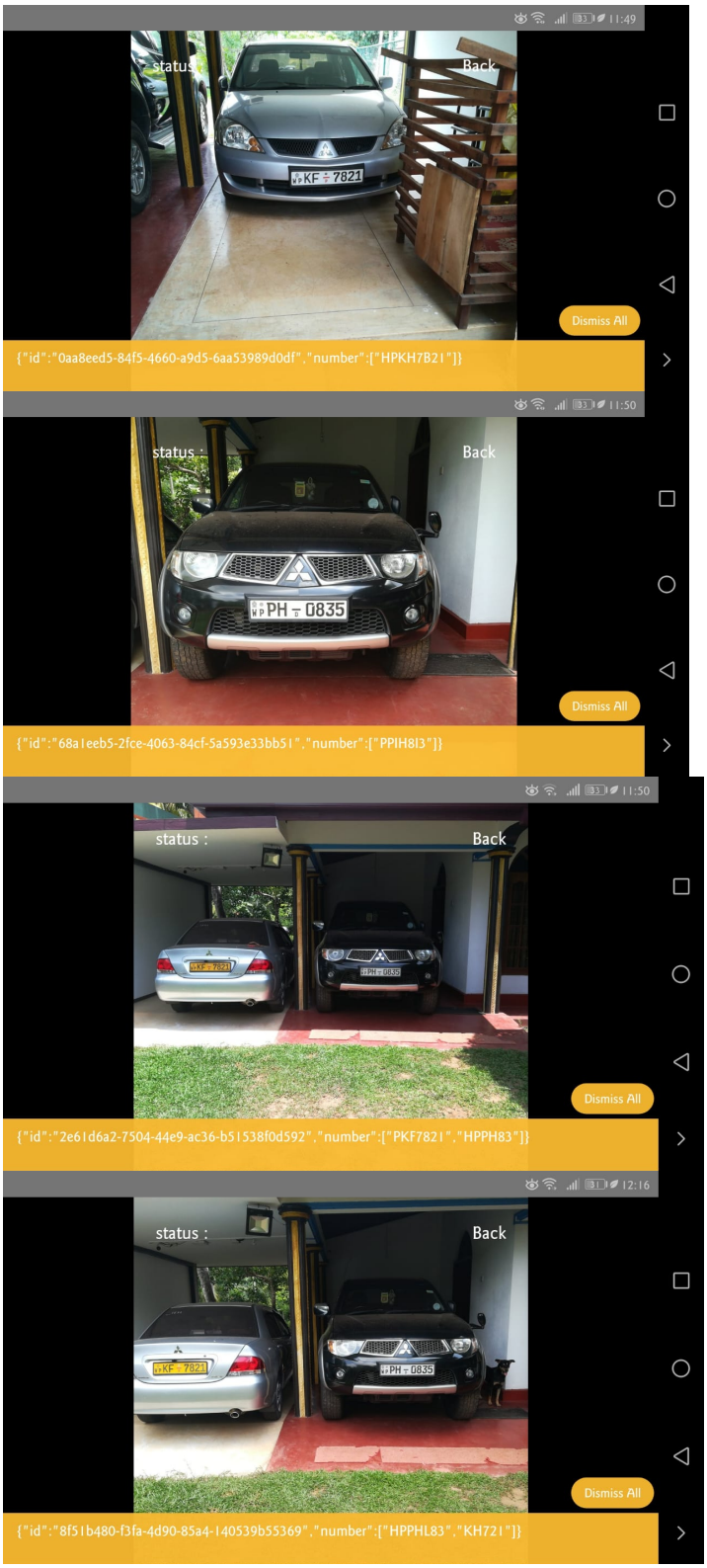
After a lot of struggle managed to develop an app to open camera and capture a image





27-04-2020- 3-04-2020

- Established the connection between mobile application and flask server and image captured from camera can be sent to the server.
- Implemented the number plate detection and recognition in flask server.
- Implemented method to read the received image and process image and send the detected number back to the mobile phone.
- Tested system on physical device.



- Can detect multiple number plates in the same frame.
- Recognition model needs to be trained for Sri Lankan number plates as the model can't recognize some of the characters and number plate detection won't even detect number plates on three wheelers as they are having a different shape.
- Response time is about 3 seconds using the wireless router

Future improvements

- Collect data and retrain detection and recognition models
- GUI improvements to show the number
- Connect to SQL database and get driver information

Sentiment analysis

Sentiment analysis is to interpret and classify emotions like negative, positive and neutral

By reading a comment of a user. It helps to get the feedback of customer from their own words automatically by analyzing their comments. There are some types of sentiment analysis.

Polarity

- Very positive
- Positive
- Neutral
- Negative
- Very negative

Emotions

- Happiness
- Frustration
- Anger
- Sadness

Detecting these types will be harder as killing word might be contained in sentences used to express both happiness and sadness

Aspect based

To get a review about a product's feature for example "Battery life of this is too short" the predictor should be able to recognize the feature battery life is having a negative opinion.

Multilingual

Recognition of entire sentences in any language.

3 types of algorithms used
Rule based – manually crafted rules
Automatic – using machine learning
Hybrid – both rule based and automatic

Plan for the next week

Implement code to do analyze comments in twitter then if possible make a data set based on comments for taxi drivers and hotel visit comments

Bibliography

[1] A. KHAZRI, "Automatic License Plate Detection & Recognition using deep learning," 1 Aug 2019. [Online]. Available: https://towardsdatascience.com/automatic-license-plate-detection-recognition-using-deep-learning-624def07eaaf. [Accessed 10 Feb 2020].

[2] P. Arora, "Building a ratings system," 11 Feb 2017. [Online]. [Accessed 20 Feb 2020].

[3] S. Nayak, "Training YOLOv3 : Deep Learning based Custom Object Detector," 14 Jan 2019. [Online]. Available: https://www.learnopencv.com/training-yolov3-deep-learning-based-custom-object-detector/. [Accessed 26 March 2020].

[4] K. Bhanot, "Create a complete Machine learning web application using React and Flask," 16 April 2019. [Online]. Available: https://towardsdatascience.com/create-a-complete-machine-learning-web-application-using-react-and-flask-859340bddb33. [Accessed 3 April 2020].