IntersectIQ Project Report

Detecting Unruly/ Aggressive behaviour in Drivers

Anurag Kurle
19110111

Gautam Vashishta
19110035

Yash More
19110123

20 April, 2021

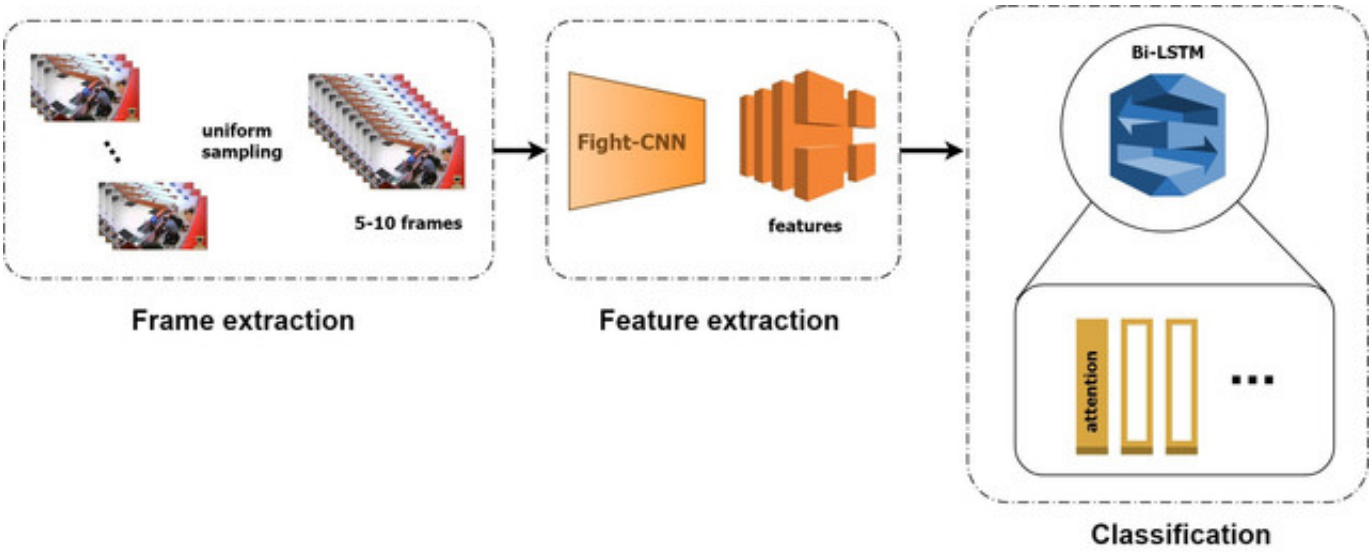
Contents

Problem Statement

The majority of transportation in and out of a plant occurs through trucks and rails; 100s of trucks pass in and out of the plants, holding 100s of drivers and support personnel, as well as as many mentalities/mind sets. Some of these drives exhibit violent actions against guards and other staff, initiating some form of physical assault, entering the control room, and even shouting excessively.
Our objective was to create a Deep Learning Model that comprised of individual pipelines that prove to be auxiliary in detecting the rage and rudeness in the behaviour of drivers to curb accidents due to this. The goal was to minimize human surveillance and create an all independent system that can prove to be extremely useful in many different real world applications

Data set

2.1 Video Model



[1] Objectives while choosing a data set for training the Violence Detection Model:

- Collecting Videos from Surveillance Cameras.
- Choosing low resolution videos as the input would be from a CCTV camera at the toll point.
- Short length models due to computation limitations.

On the basis of the following objectives, we trained the model on the following data set:

2.2 Audio Model

As auditory analysis for unruly behaviour detection is a fairly novel concept, there were no appropriate data sets to train corresponding models for this. We thus, created a baseline model based on the State of the Art language recognition detection APIs and packages for the auditory model.

Recognition - Google Speech Recognition API
Translation - Google Trans API
Sentiment Analysis - vaderSentiment package

Work Flow

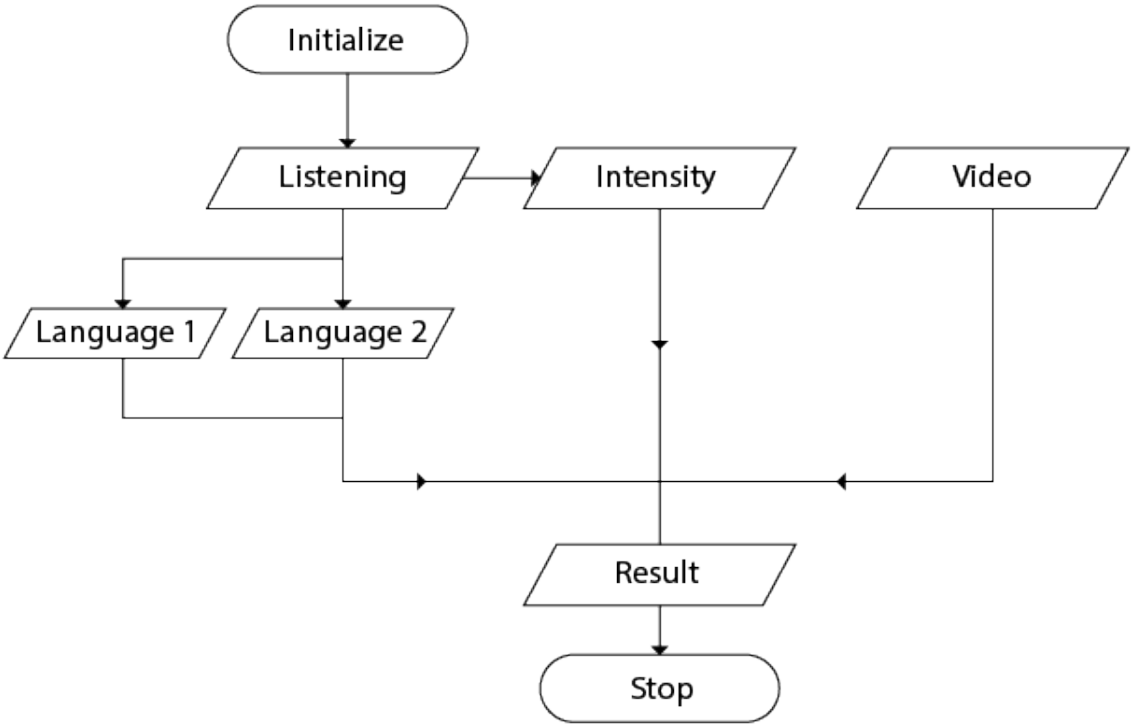


Figure 3.1: Flow chart describing the workflow

The program starts by a one time initialization by utilizing the novel Geo spatial Language Identification Algorithm based on the geographic position of the user. This fetches us the various languages spoken in the vicinity of the area in which the user is. We now begin running the following threads in parallel:

- Audio - N number of channels are started based on the number of languages detected in the vicinity of the user. For each language, we run the recognition model and obtain a sentiment score which is the maximum out of all N languages.
- Intensity - This is a quintessential aspect in determination of rudeness and aggression in the tones of the drivers. We obtain the maximum value of the intensity in a particular time frame and normalize it to obtain the intensity score.
- Video - This pipelines runs the trained model to determine aggression in the feed captured by the CCTV camera to obtain the vision metric

Specifics

4.1 Auditory Recognition Algorithms

The following are the Auditory Recognition Algorithms that are used in the Audio Pipeline :

4.2 Geo Spatial Language Detection algorithm

Figure 4.1: Algorithm in action

As most toll point lie at the junction of different states, it was crucial to create an algorithm that accurately determines the languages spoken in that region covering all the corner cases. We thus came up with a novel algorithm that uses radial arcs to find all neighbouring states of the user's location. We start at the position of the user and choose a value of radius according to the precision requirements and start sweeping at uniform increment of angle. At each increment, we check the state in which the current point falls and save the value to an array.
After the sweep has been completed, we then find the corresponding languages spoken in each state using a robust State Language Pair Dictionary and store all the languages in a final Array.

4.3 Sentiment Analysis

Upon obtaining the Languages, we use the Vader bag of words model for sentiment analysis on the sentence obtained after translating it into English Language using State of the Art APIs mentioned in the above chapter. We return a sentiment metric that is normalized between 0 and 1 based on the sentiment of the user input where 1 represents absolute anger and 0 represents passive conditions.

4.4 Intensity Detection

While analysing the sentiment of the sentences, it is also extremely crucial to incorporate the intensity with which the sentences were spoken to get a more robust model that understand the tonality of the input speech. For this, we keep a track of the intensity of the speakers' voice, this coupled with Sentiment analysis gives us a strong metric to judge upon.

4.5 Violence Detection by CNN + LSTM

[2]

The network architecture is shown in Fig. 4.1. In addition to adding the LSTM (which is supposed to extract global temporal features) after the CNN, the local temporal features that can be obtained from the optical flow is also important .The effect of optical flow is supposed to be mimicked by taking two video frames as input. The two input frames are processed by the pre-trained CNN. The two frames outputs from the bottom layer of the pre-trained model are concatenated in the last channel and then fed into the additional CNN (labeled by orange color in Fig. 4.1). Since the outputs from the bottom layer are regarded as the low-level features, the additional CNN is supposed to learn the local motion features as well as the appearance invariant features by comparing the two frames feature map. The two frames outputs from the top layer of the pre-trained network are also concatenated and fed into the other additional CNN to compare the high-level features of the two frames. The outputs from the two additional CNN are then concatenated and passed to a fully-connected layer and the LSTM cell to learn the global temporal features. Finally, the outputs of the LSTM cell are classified by a fully-connected layer

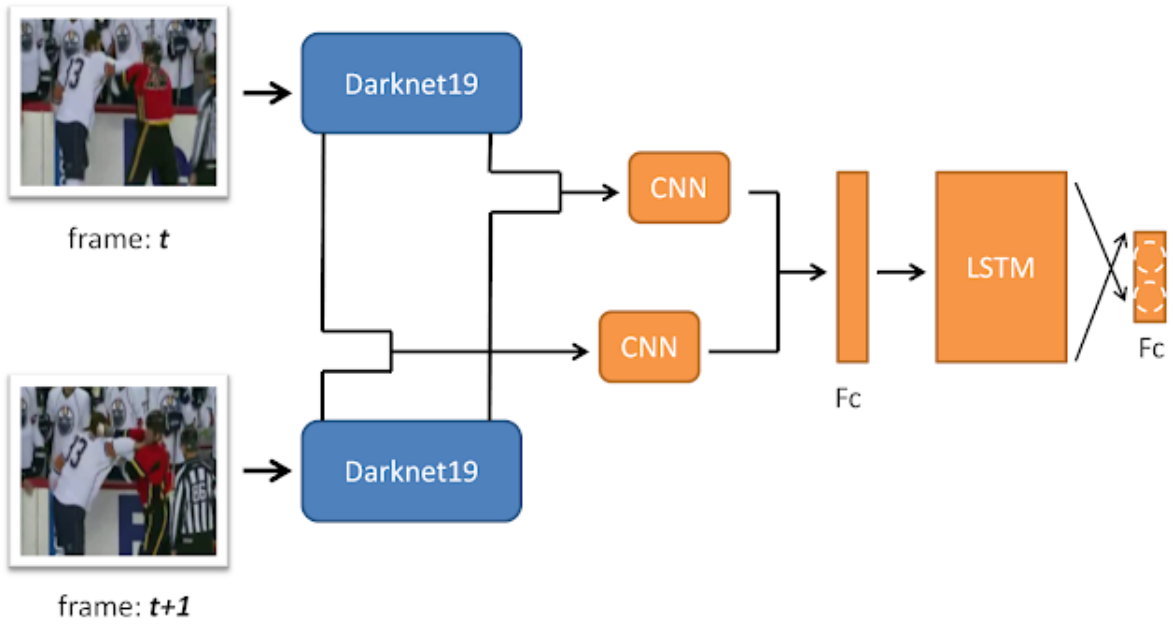


Figure 4.2: The network architecture. The layers that labeled by blue color are pre-trained on the ImageNet dataset and are frozen during training. The layers that labeled by the orange color are trained on the video dataset.

which contained two neurons that represent the two categories (fight and non-fight), respectively.

4.6 Unimetric For Unruly Analysis

Upon finding Normalized scores for both Sentiment analysis and Intensity channels, we then take a weighted average of both and come up with a auditory unimetric for further computation.
This combined with the metric obtained from the video channel is used to compute a weighted average to finally come up with the verdict on the current input.

We found the threshold value by running it over multiple epochs with values ranging between 0.2 to 0.49 (As we wanted to ensure that the Intensity score is given a higher weight in the overall metric). Upon running it over several discretized values in this range, we found out that the overall system worked the best at threshold = 0.355

Thus, Threshold Value = 0.355
U = 0.355 * SentimentScore + (1-0.355) * IntensityScore

4.7 Role of Multi threading

Multi threading plays a crucial role in this project. As described in the workflow, the whole flow would give a true positive result only when the audio and video streams would run in parallel.
With the use of powerful GPU like the NVIDIA Jetson, it is possible to run multiple threads that would be auxiliary in ensuring that the sentiment thread, intensity thread as well as the video component run simultaneously and work on the same data. This ensures that we ignore any false positives, such as slang used in daily language which if computed without the intensity and video measures would confuse the program into thinking that its a stress-full situation.

Results

5.1 Video Results

Figure 5.1: Red signifies fight detected

5.2 Audio Results

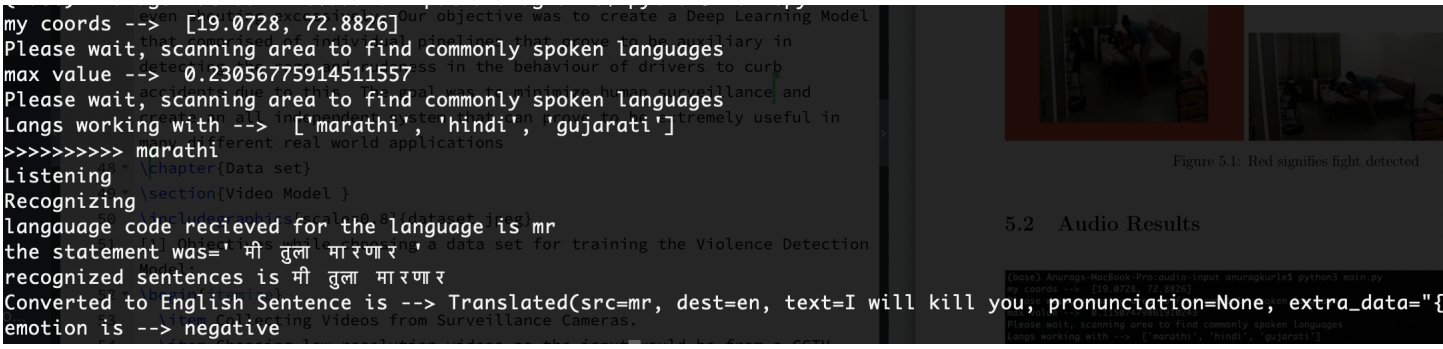


Figure 5.2: Voice input from marathi translated to english, sentiment detection performed on it

Further scope

6.1 Advancing visual detection model using YOLO (you only look once) model

Another approach to make the workflow robust would be to incorate "common sense" into it. For example, suppose a car arrived at a toll plaza. Ideally the driver, while sitting in the car, should pay the fee and leave in about 10 to 15 seconds, but on case It stands there for more than the ideal duration, then It raises some red flags.
Consider another case where the driver of the car got out out the vehicle after at the toll booth, this also raises some red flags.
These non ideal conditions could be programmed through a YOLO model for making it more robust.

6.2 Audio Signal Limiting

Along with de-noising the input audio signal, we can limit the value of the amplitudes in to the range of human communication. This in turn removes the false positives that occur due to high decibel background noise, for example - loud ambient traffic noise, which may confuse the program into taking the wrong intensity values into account, hence giving rise to false positives.

6.3 Threshold Peak Removal Algorithms

In the real world there would be a lot of high pitch noise that could though the system off, for example a loud horn and the system is blind to any other input. For overcoming this problem, we could identify the available frequencies through FFT (Fast Fourier transform) and then remove the peaks that exceed the threshold. This way we would then have to deal with realistic sound input.

References

1 Aktı, G.A. Tataroğlu, H.K. Ekenel, "Vision-based Fight Detection from Surveillance Cameras", IEEE/EURASIP 9th International Conference on Image Processing Theory, Tools and Applications, Istanbul, Turkey, November 2019.

2 The blog referred to implement the Video model (hyperlinked)