Project Report: Anticipating Accident from Dashcam Videos

ANKITA KHADSARE, Rochester Institute of Technology, USA
SRUJAN SHETTY, Rochester Institute of Technology, USA

ACM Reference Format:
Ankita Khadsare and Srujan Shetty. 2020. Project Report: Anticipating Accident from Dashcam Videos. 1, 1 (April 2020), 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 OVERVIEW

In this report we will be describing about our approach to complete the project of predicting accident before it actually happen by making use of videos from dashcam of car as data and applying deep neural network model on this data. Next few sections will consist of explaining the problem statement, the data used for this project. It will then be followed by the approach we took, the difficulties we faced and the final result we achieved. Our final code that is provided with this report makes use of DSA-RNN model as discussed by Yu Xiang et. al. [1]. In the preprocessing section we will describe about the method we used to convert our data in the format which could be made useful in order to work with DSA-RNN model.

2 PROBLEM DESCRIPTION

Auto driving cars are a reality all because of the ability of data and the deep learning models which make use of these data to extract the best use out of it. With deep learning making a huge progress in past few years, scientists and engineers are trying to cover as many area as possible to incorporate deep learning area and make life simpler. Most accidents either occur because of careless decisions by human or some fault in the car. In this project we are trying to cover one aspect of it, the carelessness of human drivers. If in some way we are able to notify the driver that there is probability of occurrence of accident within next few seconds if no preventive measures are taken, then it will make the driver to at least slow down the car which might avoid the accident. In order to build such system, we will be using the clips captured by dashcam videos of the cars, preprocess this data and store the features in a numpy and later use these values in the model.

3 DATA DESCRIPTION

The data provided to us consistent of 1500 clips which had accident occurring in the approximately last 10 frames. There were other clips too which had no accident. Each clip was of length 50 frames and these clips had to be segregated as training and testing sample. We were provided with annotation file consisting array of labels describing if accident occurred at a particular frame or not. Each training and testing samples were further divided into clips consisting of accident and

Authors' addresses: Ankita Khadsare, Rochester Institute of Technology, Rochester, NY, 14623, USA, ak8932@rit.edu; Srujan Shetty, Rochester Institute of Technology, Rochester, NY, 14623, USA, sgs2892@rit.edu.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2020 Association for Computing Machinery.
XXXX-XXXX/2020/4-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

clips with no accident. This labelled data helped us achieve supervised learning to train the model and improve the accuracy of the model.

4 USING LSTM MODEL

4.1 Idea

As per human intuition, we are able to determine when exactly a accident might occur moments before the accident actually taking place. This is achieved by us constantly looking and processing the exact position of car and nearby environment. We tried to approach the same and develop and LSTM model to do so. Our basic idea to use LSTM model is because the data is temporal and varies as per time. The ability of LSTM cell to store the result of previous output makes it a perfect choice to be used with temporal data. To serve as input to LSTM model, we required to detect the objects in the frame and keep record of their position at each frame. To achieve this, we used the file where we manually annotated the clips provided to our group. Then we used these data along with YOLO object detection algorithm to keep track of objects in each frame and predict the probability of accident in next frame using the stored data of past five frame.

4.2 Issues

As everyone might have encountered the same issue, the clips we had annotated the object locations for consisted mainly of non accident frames. This unevenness of data turned about to be a major issue. Usually any model when trained looks to achieve minimum error and maximum accuracy. So with such uneven data, our model was able to achieve a whooping accuracy of 80% even when it predicted that there will be no accident at all. Even though on paper this accuracy seems wonderful, such model is if no use in real world and not even for this project.

5 USING DSA-RNN MODEL

5.1 Idea

After trying the LSTM model as described above and achieving a miserable result, we shifted our attention to make use of model which is already built and ready to use. We came across the model discussed by Yu Xiang et. al. [1] which used DSA-RNN model which assigns weight to every object in the frame and then uses it to make prediction of accident. As discussed by Yu Xiang et. al. [1], they have achieved a really good accuracy with their model able to predict the accident about two seconds of it actually taking place. We studied the model in order to work with it and tried to list down the issues so that we can overcome them.

5.2 Issues

Since the model architecture is actually defined, it required a specific input format to be fed into the system. Yu Xiang et. al. [1] disuses in the paper of what their data are and how they extract feature from their data. We were not able to find any reference in order to extract the same feature format for our data. So, the biggest task was to implement a feature extractor code that will read each frame from each clip of our data and then store these features in a specific format and save them so that we can use these data as input to model during experimental process. We consider this as data preparation and we have provided a detailed explanation in later section.

6 DATA PREPARATION

Every clip provided to us consisted of 50 clips with accident taking place at approximately last 10 frames. The DSA-RNN model required two set of inputs. The first one being of length 4096 for 20 focus, 1 frame + 19 objects at each frame of a clip. Ten such clips and their data comprised as one

batch input for the model. The format for second set of input required the coordinates of the 19 objects from each frame. For this purpose we made use of pre-trained YOLO model. For extracting 4096 feature values out of each frame, we used the VGG16 model from keras as discussed in the paper by Yu Xiang et. al. [1]. For data preparation process, we pass every frame and detected object

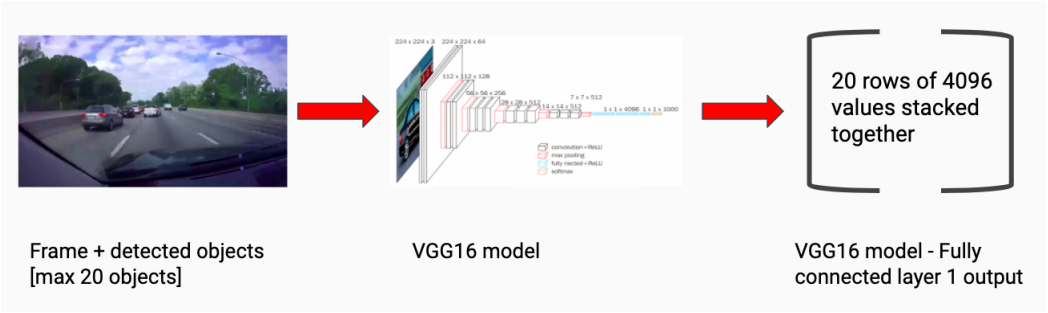


Fig. 1. Data preparation process for one frame

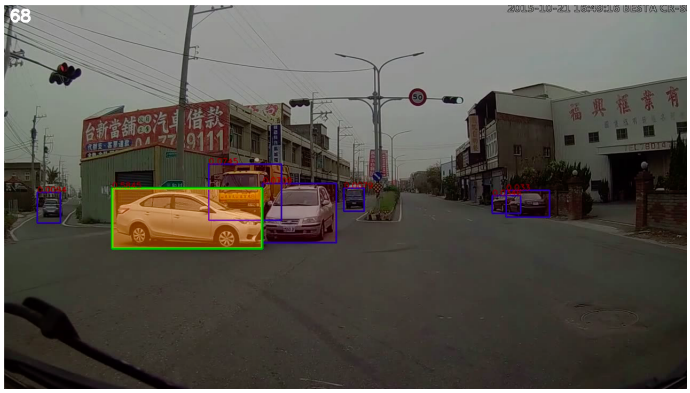
through VGG16 model. We extract the values from first fully connected layer of VGG16 model and store one set of values as one row. We stack up values from each detected object in similar manner. If the total rows do not add up to 20, we append the remaining rows with zeros to reach a length of 20. Code for data preparation is provided inside the submission folder.

7 EXPERIMENT

For experimental purpose, we combined the 1500 clips of accidents and 3000 clips of no accident. We randomly drop few data and make use of 2000 clips in total. Out of this 2000 clips we use 1500 clips for training process and the remaining 500 clips for validation purpose. We used the GPU resource provided by Google colab and the entire training process took about a total of six hours to complete. In order to avoid going through this entire training process, we have saved the values of weights and other hyperparameters for our model in checkpoint named final_model. We have included this file in the submission.

8 RESULT

We have determined the performance of our model, by comparing it with the output video provided by authors in their YouTube video. We have attached expected and actual output in Fig. 2. Figure *a* is the output that authors have achieved. On the other hand, you can see the output we get on our data set represented as figure *b*. You can notice that the car is about to crash, but our model predicts the crash just in time to notify the driver.



(a) Output achieved by authors

(b) Output achieved by us

Fig. 2. Output of DSA-RNN model

While working on our data set, we have noticed that there are few instances where the model is unable to predict the accident happening. This is because it is actually not able to recognize a object being in front of it. This is due to us making use of tiny version of YOLO model which consists of fewer layers of convolution to detect objects. This accuracy could be increased with use of fuller version of YOLO.

9 CONCLUSION

During entire phase of completing this project, we came across multiple challenges which helped us to learn a lot and keep moving forward. As this project has reached it's final phase, we have achieved a good outcome out of this and it can be further worked upon to get a better result.

REFERENCES

[1] Fu-Hsiang Chan, Yu-Ting Chen, Yu Xiang, and Min Sun. 2016. Anticipating accidents in dashcam videos. In *Asian Conference on Computer Vision*. Springer, 136–153.