Car Crash Detection Using YOLO and Siamese Networks on IndyCar

Sanket Patole
School of Informatics, Computing and Engineering
Indiana University
Bloomington, Indiana, US
sspatole@iu.edu

Shivam Thakur
School of Informatics, Computing and Engineering
Indiana University
Bloomington, Indiana, US
spthakur@iu.edu

Manjeet Pandey
School of Informatics, Computing and Engineering
Indiana University
Bloomington, Indiana, US
mkpandey@iu.edu

ABSTRACT

With the advent of Artificial Intelligence, Machine Learning and now established Deep Learning, autonomous self-driving car and object detection research has reached its peak and there is need for nuanced aspects to be taken care of in the field of self-driving cars, one of them being Car crash detection.

We propose a Car crash detection pipeline using the tradition YOLO architecture and further improving it using Siamese networks to improve on reducing False positives, we particularly focus on IndyCar series where pre-emptive Car crash detection has a huge impact.

KEYWORDS

YOLO, Siamese Network, IndyCar.

1 Introduction:

1.1 Indy 500

IndyCar is a car racing league which includes super speed ways, short ovals, road courses, and temporary street circuits. It thrills the drivers with the most diverse challenge in motorsports.

Being one of the premier car racing event in the nation involving numerous participants and the nature of the track with unexpected twists and turns, there is always a possibility of crash. Nearly 60 competitors have perished at Indy, with plenty more spectators and crew also falling at the track. Many of these were in the early days of motorsport, when a seemingly benign crash would suddenly result in drivers ejected from their car and pinned against the wall.[1]

1.2 Crash Detection

Deep Learning and AI have vastly contributed in many real world problems and with the support of parallel computing, where parallel computing makes sure the predictions are real time, we propose an end-to-end Deep learning crash detection pipeline using traditional concepts of Object Detection and Latent space embedding concepts in our project.

2 Dataset:

We scraped YouTube for Indy Car Crash videos, and initially annotated the crash images in YOLO format, with anything other than the normal car labelled as a 'Crash' at first, throughout our iterative process of the project, how the data would be labelled was varied and scrutinised based on our end goal of efficient, and accurate Crash Detection.

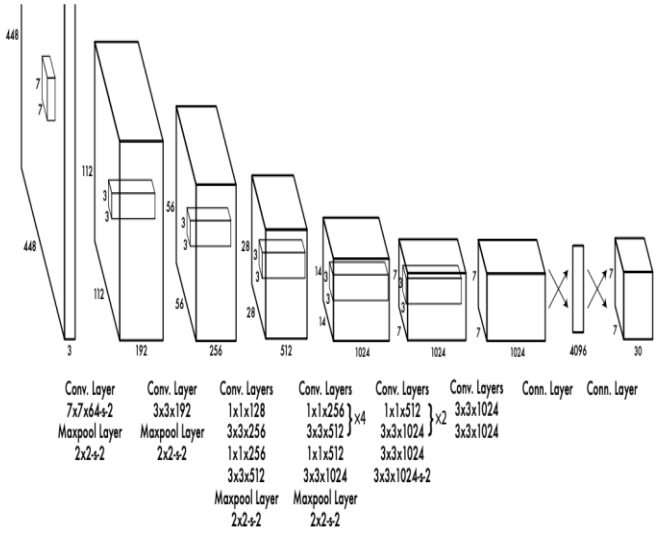
For the model for One-shot prediction using latent vector embeddings, YOLO Car Detection model was run for 1000 images and the cropped predicted images were used for the aforementioned model.

3 Architecture:

The whole pipeline involves two models: YOLO[3] and a network inspired by the Siamese network, the FaceNet model[2].

3.1 YOLO

We chose YOLO for localized object detection since it is the current state-of-the-art Object detection/localization model, compared to RCNN, Fast-RCNN, Faster-RCNN models. YOLO uses the concept of localized bounding boxes at each point called as the anchors. Each anchor box gives the probability of that anchor box containing an object of certain class. Finally, the bounding boxes with high probabilities are merged together using Non-maximal Suppression(NMS) to finally obtain a single bounding box around a detected object. The model architecture is explained in figure 1.



[3] Figure 1: The Architecture. Our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1 × 1 convolutional layers reduce the features space

from preceding layers. We pretrain the convolutional layers on the ImageNet classification task at half the resolution (224 × 224 input image) and then double the resolution for detection.

3.2 Siamese Network

Siamese networks use a base convolution network to which pairs of images are sent. The pairs of images can be from same domain or different domains. Siamese network makes sure the images that are from same domain lie closely in the latent space, while the dissimilar ones far apart, subsequently in the process learning to embed domain specific images. These embeddings can then be used in a K-means classifier or a SVM classifier. These final classifiers give us one shot classification of a given image. The loss function used is Contrastive loss.

4 Experiments:

4.1

At first, data was annotated with tight bounding boxes with not much care given to surroundings like off track/on grass, smoke, broken car etc. Our yolo model consisted of only ‘aeroplane’ and ‘crash’ labels, for this model the MAP was 45% although the predictions were pretty good but we assume the bounding boxes were not exactly accurate to the dot compared to the actual labels but close enough to give visual results close to state-of-the-art object detection.

4.2

Crash context was given to the images with a larger bounding box including the aforementioned surroundings to make sure the context is included for a given car and the model jumped to 71% MAP accuracy. At this point, ‘Car’ labels were not used and hence some normal cars were also detected as ‘Crash’, although with less probability. To fix this we ran our experiment again in the next step.

4.3

For our third iteration on this model, we included the ‘Car’ dataset as well with a class distribution of 25% crash images and rest 75% car images, At this point the MAP increased to 81% but still in the visual results, some normal cars were detected as ‘Crash’. To battle these False Positives, we tried another experiment in the fourth iteration.

4.4

We finally used a Siamese network to use a One-shot model to classify between a normal car v/s a totalled/off-track/smoky car, we propose not to use another Deep Learning model over the YOLO model in the whole pipeline since it’ll increase the inference time, time being extremely critical in our use-case. Training a Siamese network on the contrary and then creating a K-means model on the learned embeddings was the better choice. For this model, the dataset was created from the

original YOLO model to fetch cropped out ‘Crash’ and ‘Car’ images(cropped) and resized to 256x256 and then used the Siamese model finally.

The model uses a VGG16 backend with some added Convolution and Maxpool layers followed by a dense layer for final classification. The dataset is created as pairs of ‘Car-Car’, ‘Car-Crash’, ‘Crash-Crash’, with the same domain pair images labelled as 1 and dissimilar ones as 0.The traditional contrastive loss was used with single margin in the loss function in Figure 2.

The exact loss function is

$$L(W, Y, \vec{X}_1, \vec{X}_2) = (1-Y)\frac{1}{2}(D_W)^2 + (Y)\frac{1}{2}\{max(0, m - D_W)\}^2 \quad (4)$$

Figure 2: The Contrastive loss uses a margin to keep dissimilar pairs apart while keeping the similar ones close together[4].

With this model, using K-means clustering the accuracy achieved was ~93%. We further investigated the loss function and came across the double-margin loss [5]. Instead of just focusing on keeping the dissimilar images apart, it works both on positive pairs collectively which worked much better for our dataset with ~98% accuracy. The loss function is stated below

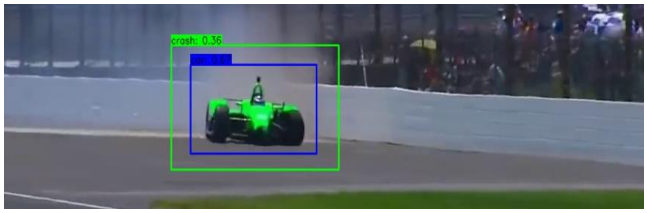
L (Ip, Iq) = 1 2 [y max (d − α1, 0)2 + (1 − y) max (α2 − d, 0)2], where now we have margins for both positive and negative samples.

5 Results :

Stage	Accuracy (MAP)
Stage – 1 (Data Annotation with tight bounding boxes)(Crash Data)	45% ±5%
Stage – 2 (Precise Annotation) (Crash Data)	71% ±3%
Stage – 3 (Crash and Car data mixed)	81% ±3%
Stage – 4 (Conjugation of YOLO and Siamese network)	93% ±2%

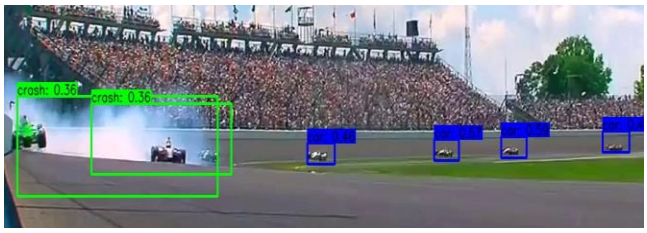
6 Predictions :

Stage-1:



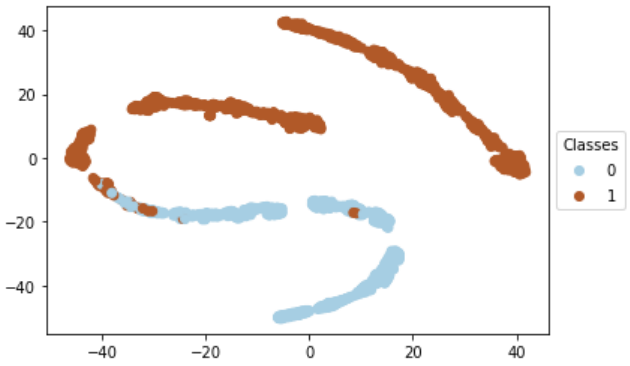
Stage-2:

Stage-3:

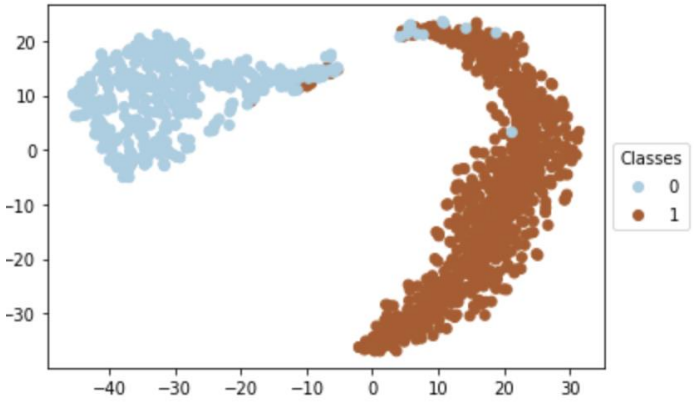


Stage-4: K-Means clustering on embeddings from Siamese

i. Single Margin



ii. Double Margin



7 Conclusion:

1. YOLO performs well in detecting cars in general, irrespective of its condition.
2. When trained to differentiate and predict Crash and normal car, it gets confused and sometimes detects a crashed car as a normal car.
3. Also, since the idea behind YOLO to detect a crash is to detect smoke, off-road or a car with broken parts, sometimes a normal car in the vicinity of smoke or broken parts is detected as crash.
4. Adding a Siamese network in the pipe-line along with YOLO boosts the accuracy a lot.
5. Performing clustering using K-Means on the embeddings of the two classes (crash and car) clearly visualizes two separate clusters, which supports our boost in the accuracy with Siamese network.

6. However, apart from the above results, there is still room for improvement.
7. Amongst our experiments, we realized that the ideal architecture to predict crash should involve recurrence, such as a RNN.
8. With recurrence we can use data from multiple frames of the live stream and then make meticulous predictions, since a recurrent network will use data along a time-line to increase the accuracy of the prediction. Such as multiple frames of a crash one-after the other.

9 Team Contribution:

Team Member	Task	Contribution
Shivam Thakur	Labelling/Annotation of training data, Siamese Network, Documentation, K-Means clustering	**35%**
Sanket Patole	Labelling/Annotation of training data, Yolo architecture, Documentation, K-Means clustering	**35%**
Manjeet Kumar Pandey	Labelling/Annotation of training data, Yolo architecture, Integration of Live stream prediction, Documentation, Presentation	**30%**

10 REFERENCES

[1] Miller, A. (2015, March 12). The 8 Deadliest Race Tracks In The World. https://www.thrillist.com/cars/the-8-deadliest-tracks-on-earth-most-dangerous-races.

[2] Schroff, Florian, et al. "FaceNet: A Unified Embedding for Face Recognition and Clustering." *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, doi:10.1109/cvpr.2015.7298682.

[3] Redmon, Joseph, et al. "You Only Look Once: Unified, Real-Time Object Detection." *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, doi:10.1109/cvpr.2016.91.

[4] Hadsell, R., Chopra, S., & Lecun, Y. (n.d.). Dimensionality Reduction by Learning an Invariant Mapping. *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2 (CVPR06)*.

[5] Hao, J., Dong, J., Wang, W., & Tan, T. (2018). DeepFirearm: Learning Discriminative Feature Representation for Fine-grained Firearm Retrieval. *2018 24th International Conference on Pattern Recognition (ICPR)*.