**Final Project Report: Collision Avoidance using Video Processing.**

Group 02: Izza Godinez & Sravan Pingali

* **Project Definition**

Over 1.35 million people fall victim to accidents each year. It is estimated that each day about 3700 people are killed globally due to some form of road accidents. Vehicular collisions are the most common road accidents that happen in the world and form the biggest part of this number. With the advent of consumer AI and the advancements in sensor technology, collisions can be drastically reduced. Some of the common measures taken by car manufacturers are audio, visual feedback when a car is going off a lane or when it is approaching a road railing. We are existing in a time when self-driving cars are on the cusp of being commercially available for use. Many tech giants have already developed cars that use advanced sensors and high level computing to have cars drive through cities on its own. Waymo is a front runner in the field of self-driving cars and it has been testing its cars in Arizona since October 2017. A vital part in the success is in the car’s ability to detect objects (here, cars, pedestrians, animals, or any obstacle). Without a good object detection system, self-driving cars are doomed to fail and these companies know it. Object detection isn’t simply used in self-driving cars, it can also be used to develop a system that can help visually impaired individuals with navigation.

Object detection isn’t simply used in self-driving cars, it has also found use cases in many other industries. It can be used for tracking objects, or optical character recognition, face detection and recognition, object extraction, medical imaging, ball tracking etc. The potential applications of object detection software is humongous and are potential business opportunities as well.

* **Project Scope and Dataset**

Given the time constraints and the other commitments that the team members have, it has been decided that the team will only be able to concentrate on one aspect of the project. Since object detection is the core of a collision avoidance system, the team will be focusing on this aspect of the project alone. The dataset used for this project will be from Udacity (Udacity Self Driving Car Dataset). This dataset contains a series of images where the object detection software will be used to detect and classify objects in an image. Since, a video stream is a bunch of images and video processing is done for each frame of the video, it can be considered as image processing. Therefore, implementing an object detection software for video will be very similar to the object detection software for images.

Having information from sensors such a lidar, or a radar are also crucial in determining the collision, but this information is difficult to find and we are using only images as our source of information.

* **EDA**

For the EDA, it was important to study the two parts of the dataset that were being used, the images and the annotations. When looking at the annotations file, it was a .json file which included information about each of the images. The information from the .json file was imported into the notebook and transferred into a dictionary data structure. When analyzing the dataset, the annotation file has information on the image ID, the image file name, the bounding boxes of objects in the images, as well as the category that the object falls under. To clean up the data, the initial .json file information was divided into different data structures for the image information, the category information, and the annotation information. The image information contains the image ID as well as the file path name. The category information contains the category ID as well as the category name. The annotation structure contains the image ID, the category ID, the bounding box information, the area, if there are multiple people, the image path, and the category name. In order to see the bounding box, the image was loaded using cv2 package. The bounding box information was used to draw a rectangle on the image so that when it was displayed, it would show what object the annotation was referring to.



Fig 1: Bounding box around obstacle

* **Solution**

The solution was primarily developed using python using a Keras Sequential Probability Model, as well as a pretrained ImageAI object detection model. The python package cv2 was used to load the image into the notebook and it was displayed using matplotlib. Using the bounding box, images were cropped from the main image to the size of the bounding box. This created images that only showed the objects in the images. The category name is also displayed.



Fig 2: Cropped image of an obstacle

When training the Sequential CNN, functions were created that allowed the program to crop the bounding box image from each of the given images. Then, the categories were one-hot encoded into an array where each row represented a category. A function was then created that would put the category information of each sample image into a new dictionary, using the one-hot encoding array. The following image shows the annotation and the corresponding one-hot encoding array based on the category ID:

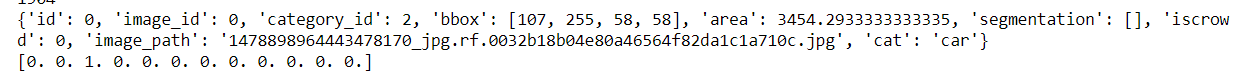


Fig 3: Converting category information into one hot encoding

The information of the features and labels is then split using sklearn to create a training dataset and testing dataset. Using the features and labels as the x and y variables, a Sequential model was trained using 16 different layers and an Epoch value of 20. When comparing the trained model against the testing data, the model was found to have a validation accuracy of 81%.



Fig 4: Training the CNN model with an accuracy of 81%

Alternatively, a pre-trained model was then used as a comparison point. This model was the ImageAI pre-trained model. When used on an image that was not in the original training data, it found multiple obstacles:

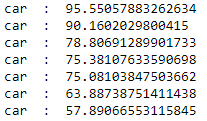


Fig 5: Information on each obstacle an the confidence percentage of accuracy

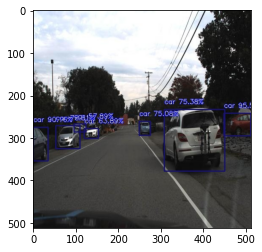


Fig 6: Image with bounding box and accuracy percentages overlayed

* **Future Improvements**

There are quite a few improvements that can be done to this system. For starters, this system can be expanded to work with real-time video. Incorporating this system to a livestream will open up the many potential use cases. Since we do not have the resources to actually test this system in a car, a simple game can be created to test this system. This game will feature a road with 4-5 lanes and a car in the center lane. Objects can be randomly generated in a lane and if the car is in the same lane, it needs to detect it and switch lanes accordingly.

* **References**

[1] Olafenwa, Moses. *Medium*, Towards Data Science, 16 June 2018, towardsdatascience.com/object-detection-with-10-lines-of-code-d6cb4d86f606.

[2] Roboflow. “Udacity Self Driving Car Object Detection Dataset.” *Roboflow*, 11 Feb. 2020, public.roboflow.com/object-detection/self-driving-car.