Step 10: Study Advanced Units - Computer vision techniques leveraged during the capstone project

I chose to focus on computer vision and image processing as the optional unit to study in the ML Bootcamp, as it relates to my capstone project. This paper will discuss different techniques learned related to the work that was done in the capstone project and the approach taken.

The objective of the capstone project was to be able to detect objects in a video stream and estimate the velocity of the moving objects in the video. In order to do so, the following steps are required:

1. Breakdown video stream into frames
2. Perform object detection and classification in each image frame
3. Perform continuous object tracking by uniquely identifying objects in consecutive frames
4. Estimate object velocity based on movement of the objects between consecutive frames
5. Overlay object detection bounding boxes, and processed info for each object identified in a video stream

The first and last steps in achieving the project objectives was done by utilizing OpenCV for the purpose of image processing. With simple function calls of the OpenCV API, video information such as frame rate and frame dimensions can be extracted. Single video images can be obtained and manipulated as needed. Object detections information can be overlaid and/or highlighted on image frames. New videos can be reconstructed containing the desired image content. Since a fundamental part of the capstone project was to apply computer vision algorithms on single image frames, OpenCV enabled doing this with ease. It also provided an excellent tool which allows visualization of the computer vision algorithm results. As an output of the capstone project, object detection and post processing information is overlaid on the video and presented to the user.

One of the challenges in achieving the goal of object velocity estimation is first to be able to detect and classify the objects in the video. There are several techniques and tools developed to facilitate and enable object detection on a single frame. Tools like scikit-image and OpenCV enable processing of the image to extract the necessary information to make the detection. Algorithms like Histogram of Oriented Gradients (HOG) (N. Dalal; B. Triggs, 2005) help with the classification of objects. With computing power becoming cheaper and more accessible and with development tools that are becoming more readily available, there is a growth in Deep Learning algorithms that help solve the problem of image detection and classification. There are several articles, (like the one found here: cv-tricks.com) that provide a good overview of the various algorithms used for image detection and classification.

In general, There are three primary Deep Learning-based object detection methods:

1. **Region-based Convolutional Neural Networks**

This includes the original Region-based Convolutional Neural Networks (R-CNN) and their variants, Fast R-CNN, and Faster R-CNNs (Ren et al., 2015), Mask R-CNN for Image Segmentation

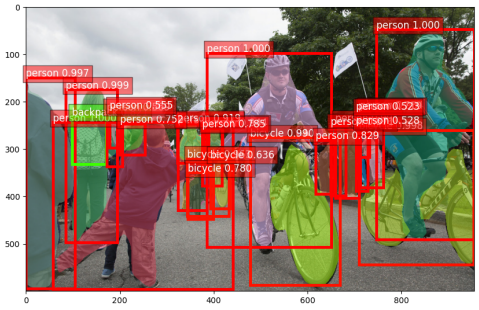

Figure 1: Mask R-CNN (**source**)

These R-CNN based algorithms are considered to be very accurate however could be considered slow for real time applications. Here is a comparison between the different R-CNN methods:

	R-CNN	Fast R-CNN	Faster R-CNN
Test Time per Image	50 Seconds	2 Seconds	0.2 Seconds
Speed Up	1x	25x	250x

2. **Single Shot Detector (SSD)**

The Single Shot Detector (SSD) (Liu et al., 2015) is a regression based object detector. It uses a single deep neural network. It is considered to achieve a good balance between speed and accuracy. SSD runs a convolutional network on input image only once and calculates a feature map. A small 3×3 sized convolutional kernel is run on the feature map and bounding boxes are predicted along with classification probability. SSD uses anchor boxes at various aspect ratios (similar to Faster-RCNN) and learns the off-set rather than learning the box.

3. **You Only Look Once**

The method for image detection and classification chosen for this capstone project is You Only Look Once (YOLO) (Redmon et al., 2015) as it is considered to be extremely fast. YOLO works on the concept of dividing each image into a grid of S x S and each grid predicts N bounding boxes and confidence. The confidence reflects the accuracy of the bounding box and whether the bounding box actually contains an object. YOLO also predicts the classification score of each class it was trained on. The probability of an object of a specific class being in a bounding box can then be calculated. A total SxSxN boxes are predicted. Since most of these boxes have low confidence scores, a threshold is applied on the confidence and boxes with low confidence are removed.

One drawback of YOLO is that it only predicts 1 type of class in one grid therefore it struggles detecting small objects.

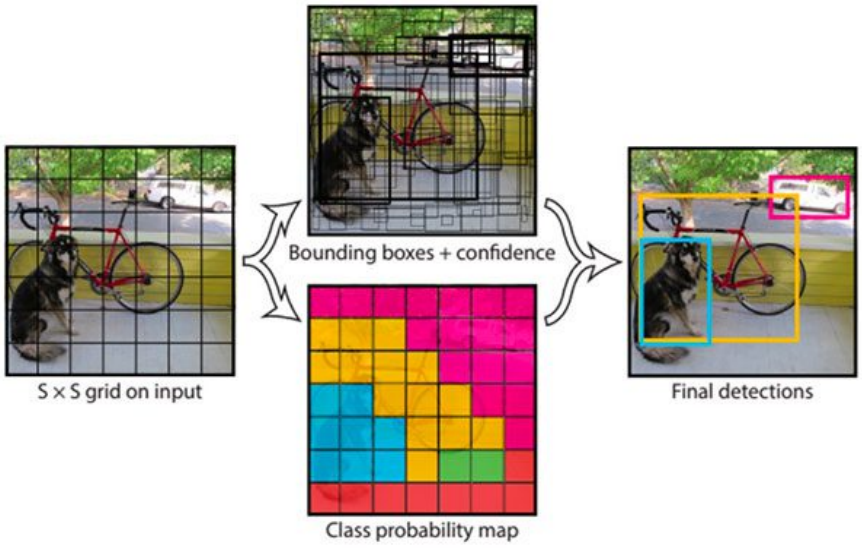

Figure 2: YOLO object detection pipeline (**source**)

YOLO does not apply non-maxima suppression, it needs to be explicitly applied to suppress significantly overlapping bounding boxes, keeping only the most confident ones.

In order to continuously track objects along multiple frames, each object must be uniquely identified between consecutive frames. To accomplish this, the DeepSORT algorithm was chosen since it is suited for real time applications. DeepSORT is based on the Simple Online and Realtime Tracking (SORT) algorithm. Much of the computational complexity is done during the offline pre-training stage where a deep association metric on a large-scale re-identification dataset is learned. When the application is running online, it leverages nearest neighbor queries in visual appearance space and the algorithm performs measurement-to-track associations. With DeepSORT, objects can be tracked through longer periods of occlusions effectively reducing the number of identity switches.