Capstone Project 2 Milestone Report 1, Springboard

Problem statement:

Detecting and identifying objects on videos has many useful applications. For example, it is crucial for self driving cars to accurately detect and identify objects on roads for safe driving. By applying deep learning algorithms against the video dataset, we can build accurate models for the detection and identification. In this second capstone project, I am going to use YOLOv3 algorithm for object detection on videos.

The videos that will be used for object detection must include cars and people on streets. The model will identify these two categories. Later it could be modified and trained to identify more categories. The techniques used in the project can be potentially applied to other videos for security, AI powered grocery solutions and others for which object detection and identification is important.

The videos will be uploaded into docker container, which will start based on pre-build docker image with python3.5 environment and installed darknet open source neural networks. The video processing will involve braking it into frames. Then, the YOLOv3 algorithm will be applied to each frame to detect the object within the frame and then identify it. All processed frames will be combined again into new video with added boxes around objects.

Overall, advancements in hardware and deep learning architecture creates new possibilities for AI powered solutions that involve real time video processing and identifications in videos. This include speech recognition and summary creation about the video content.

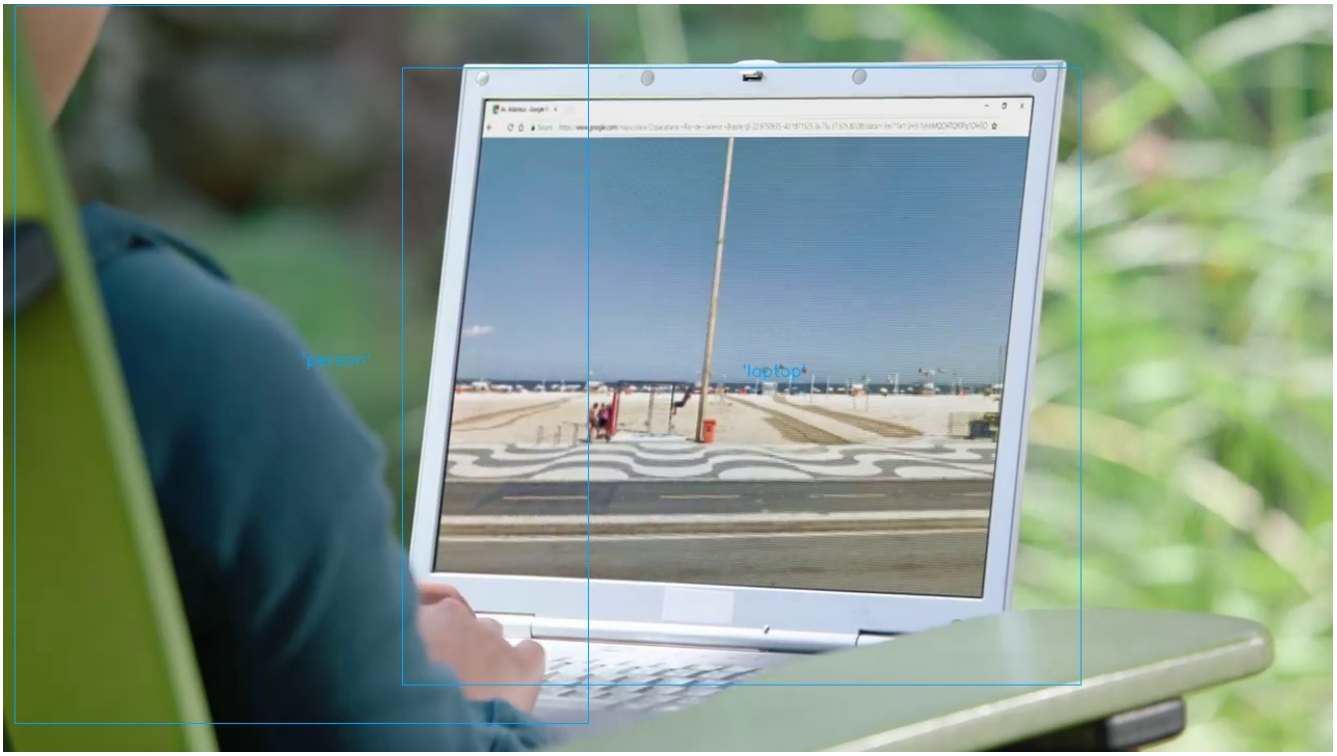
Description of the dataset:

Videos with mp4 format that include cars and persons will be used as an input to docker container where python application will run to detect and identify objects on the given video frame. The resulting processed video with detected and tagged objects will be produced as an output video in avi format.

As an example, video with mp4 format can be downloaded from the youtube using the following online video grabber tool: https://www.videograbber.net/, and then used for testing purposes. We recommend to create new directory, rename newly downloaded video to test.mp4 and store the video in that new directory. We will use the path to the directory as an input docker volume mapping into container. Thus, the video will be available inside the container for the analyses.

Initial findings:

Source: https://www.youtube.com/watch?v=6FZH652qYkA



Source: https://www.youtube.com/watch?v=6FZH652qYkA

Summary of findings:

We can clearly see accurately detected and tagged objects on the video frames. These are car, truck, person and laptop objects. We used the open cv library to brake the video into frames and analyze them with darknet open source neural networks. Then, we combined processed frames into resulting video.

The processing took place in docker container. The docker image for the container was build with python3.5 base. Also, it includes all darknet files downloaded from the source and installed with make command.

Running video processing container requires the following steps:

1. Download the Dockerfile and two additional files (darkent.py & requirements.txt), cd to the directory and run:

docker build -t <iamge-name> .

where <image-name> will be used for the image tag

2. Run the container:

docker run -it -v <path-to-video-dir>:/usr/src/app/video --name <container-name> <image-name>

where <path-to-video-dir> is the directory were test.mp4 file is stored for processing.
<container-name> can be specified to tag the container.

Once the container finishes image processing, newly created video with processed frames will be written in the <path-to-video-dir> directory.