**Real-time Social Distance Detection System**

Team 0x29A

Member: Yexiong Xu, Qinxi Yu, Hanlai Chen

1.

Due to the outbreak of COVID-19, many researchers have concluded that maintaining a social distance of at least 6ft is a very effective measure to slow down or even stop the spread of the virus. Many cities have introduced rules to limit their citizens to interact with each other. **Social Distance Detector is a program that automatically detects and raises an alarm when people are too close to each other.** The application of this program can help to slow down the spread of the virus and at the same time increase people’s self-awareness.

2.

In the Social Distance Detector, the most important feature is object detection and perspective transformation. For example, in order to measure the distances between pedestrians, object detection is applied to each frame of the video to locate every pedestrian. Then distances are calculated using the coordinate of each person on the transformed frame. Also, concepts of Affine Transformation and Ransac could also be used for the perspective transformation.

Eventually, object detection is still relevant and will continue to contribute to this course.

3.

Social Distance Detection is a new topic since the outbreak in late 2019, there is not so much solution out there. However, object detection and perspective transformation are no longer new topics, and they have been studied for years. There are many object detection algorithms such as R-CNN, SPP, Fast R-CNN, Faster R-CNN, Feature Pyramid networks, RetinaNet(Focal loss), SSD, and Yolo(v1,v2,v3). The most popular algorithms are R-CNN and Yolo. R-CNN is known as [Region-based Convolutional Neural Networks](https://arxiv.org/pdf/1311.2524.pdf) published by Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik in 2014. R-CNN is a combination of regional proposals with CNN. It has the capability to scale without resorting to approximate techniques. Another well-known algorithm is called Yolo (You look only once), the architecture of Yolo would first split the input into m by m grid and for each grid, it generates two bounding boxes and class probabilities for those boxes.

For perspective transformation, there are algorithms like [Bird’s Eye View Transformation](https://www.ijser.org/researchpaper/A-Simple-Birds-Eye-View-Transformation-Technique.pdf). In this algorithm, first, we have to represent the image in a shifted coordinate system, next we perform a rotation of the image, and then project the enlarged image on a 2d plane.

4.

We will use Yolo v1 as a starting point. According to the paper first published in 2016: [You Only Look Once: Unified, Real-Time Object Detection](https://arxiv.org/pdf/1506.02640.pdf) by Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi. Yolo has developed to version 3 and its performance has been greatly improved. But for the simplicity of learning, Yolov1 is always a great starting point. By learning Yolo v1, we will try to implement Yolo v3 and train models with data from COCO. The reason why we prefer Yolo over other object detection algorithms is that we need a fast and robust classifier. We can apply filters to unconfident predictions and labels that are not people. After detecting all people, we would calculate the euclidean distance between every pair of people. Starting with finding the ground plane by ground detection algorithm that we might implement or simply by input the plane boundary or plane corners. Through the perspective transformation algorithm, we could obtain a bird’s eye view diagram with all the people’s location marked. In this way, we can calculate the distance between each pair.

5.

* set up Github repository for the project, include all the starter code, references, readme and other documents.
* Read all the papers related to the topics as a team, and hold a meeting for any adjustment on our proposal and design.
* Find the most reliable and suitable dataset for training and testing our model. We would more likely to be choosing our data from [COCO](https://cocodataset.org/#home).
* Find videos that contain pedestrians. All videos should be preprocessed so that their dimensions and resolutions are favourable (not too high).
* Implement the Yolo v3 algorithm and make sure it works, if that is not the case, we would consider referencing code from other sources.
* Start training our model and tuning the required hyperparameters. If we decide to change our training dataset we would go back to step 3.
* According to our studies, derive the required formulas and equations for perspective transformation so that we can implement them later.
* At the same time, other members of the group may start implementing the interface of the program and the perspective transformation algorithm.
* Combine the two parts of the work and make sure it works.
* Examine our results using some test sets. Conclude our results and try to optimize our program.
* Hold a meeting to plan our demo and at the same time write our project report. Share the individual part with partners and write down a list of Q&A for the oral defence. According to all the above information, we would record our presentation video.

Reference:

[1] UC Berkeley, 2014. *Rich Feature Hierarchies For Accurate Object Detection And Semantic Segmentation*. [online] Available at: <<https://arxiv.org/pdf/1311.2524.pdf>> [Accessed 4 November 2020].

[2] M.Venkatesh, P.Vijayakumar, 2012. *Transformation Technique*. [online] Available at:<<https://www.ijser.org/researchpaper/A-Simple-Birds-Eye-View-Transformation-Technique.pdf>> [Accessed 4 November 2020].

[3] Redmon, J., Divvala, S., Girshick, R. and Farhadi, A., 2016.*You Only Look Once: Unified, Real-Time Object Detection*. [online] Available at: <<https://arxiv.org/pdf/1506.02640.pdf>> [Accessed 4 November 2020].