CO 542 project contribution report

Group 07 - Intelligent Parking Space Detection System

E/16/076 – Deshan L.A.C.

Problem

Identification of free parking spaces in a parking area. Parking areas don't want to be a well labeled paring area.

Fig 01 : well labeled parking area

Fig 02 : unlabeled parking area

Previous work

We did refer several resources before the project proposal about the intelligent parking space detection systems. In most of those projects, what people had done was they had manually drawn the areas where the cars can park and if there is no car in that marked slot then the system gives an indication mentioning there is a free space. Problem with this is we have to manually draw the parking slots. When developing this system in real world, manual annotation is not suitable (if we are going to establish these systems to different parking yards we have work in isolate for each park). And also if parking slots are not marked properly then manually annotation is difficult (in Sri Lanka most of car parks are not marked correctly). To detect a car they have used object detection models like Yolo, MaskRCNN etc.

Our solution

Our idea was a different one. To develop this idea I gave a big contribution. My idea was, to identify a free slot, we can check two frames of the video feed and in first frame there is a car in a specific location and in the other frame car is at different location then we can say that there is a free space in first location (as pixel values are changed in the first frame of video feed). Problem came with this was, if a car moving in a lane in the park, then in two different frames the location get changed, then the system will detect the first place as a free space which is not correct. I introduced to others the semantic segmentation concept that I had read before, as a remedy for this problem, from which we can identify the parking slots. In countries like Sri Lanka, parking areas are not marked. So to identify the parking areas what we want to do is to observe the behavior of cars and mark those areas as parking slots. Then after we identifying the parking slots, then we can get the moving or pixel differences and can say that the vehicle has moved from a parking slot. If the moving is happened in a lane it gets rejected. This is the overall idea of the project.

My Contribution

To do semantic segmentation we had to go for a deep neural network and we choose **UNet Architecture** (*Chen, W., Liu, B., Peng, S., Sun, J., & Qiao, X. (2018, September). S3D-UNet: separable 3D U-Net for brain tumor segmentation. In International MICCAI Brainlesion Workshop (pp. 358-368). Springer, Cham*). In this project my task was to develop, train and tune the UNet model architecture. For this we used KITTI data set (http://www.cvlibs.net/datasets/kitti/index.php) which has been segmented on 20 different classes of objects and for this purpose I trained only on 6 different classes ('building', 'land', 'road', 'vegetation', 'water', 'unlabeled') where I put other classes as unlabeled. So we can show all the other 14 objects in one color. KITTI data set is for autonomous cars and it is not well suited for our task. But as they are captured form top camera of a vehicle that elevation of photos was normally suited for our task.

Fig 03 : KITTI train image

Fig 04 : KITTI train mask

Original data set consisted of 200 images and I split the dataset in 9:1 ratio for train and validation sets. I test the data set from another data set which was taken from CCTV camera of a real car park. As train data is not for car park I did this approach. As training data set was finally 180 images, I used data augmentation where I did 8 transformations on the train set. So finally there were 1710 train and mask images. I trained the architecture from scratch and due to overfitting problem I introduced dropout layers. Original architecture was designed for 512x512x3 images. I trained this for several occasions (for 521x512, 256x256, 128x128, 64x64 image sizes). E16221 also helped me to train this. So finally from the outputs 256x256 gave some considerable accuracy. It was 67% on the test set. With the time limitations and not having proper segmented dataset for this task, I couldn’t be able to train this for some high accuracy. This is my part actually.

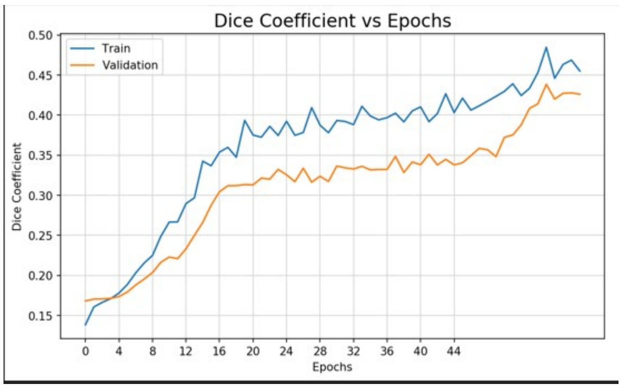


Fig 04

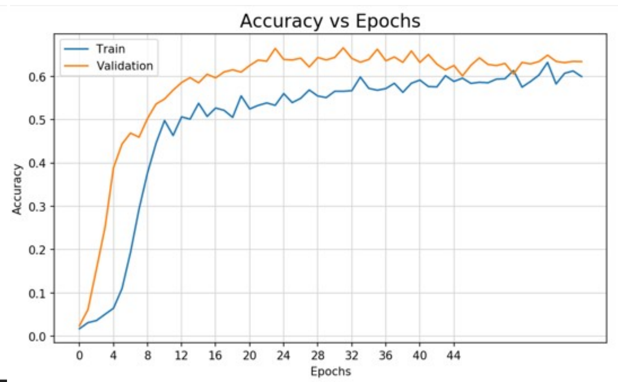


Fig05

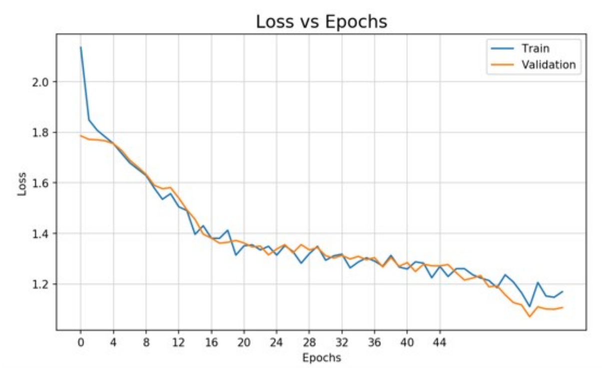


Fig 06

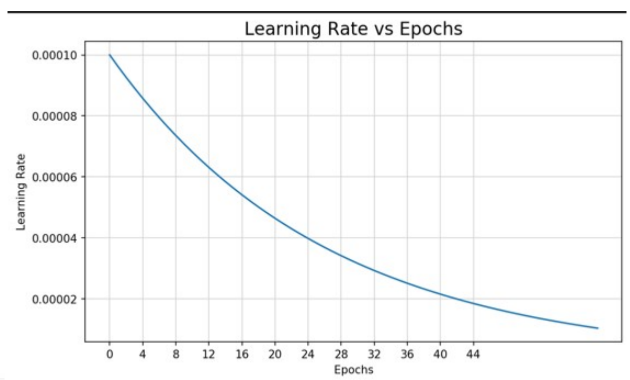
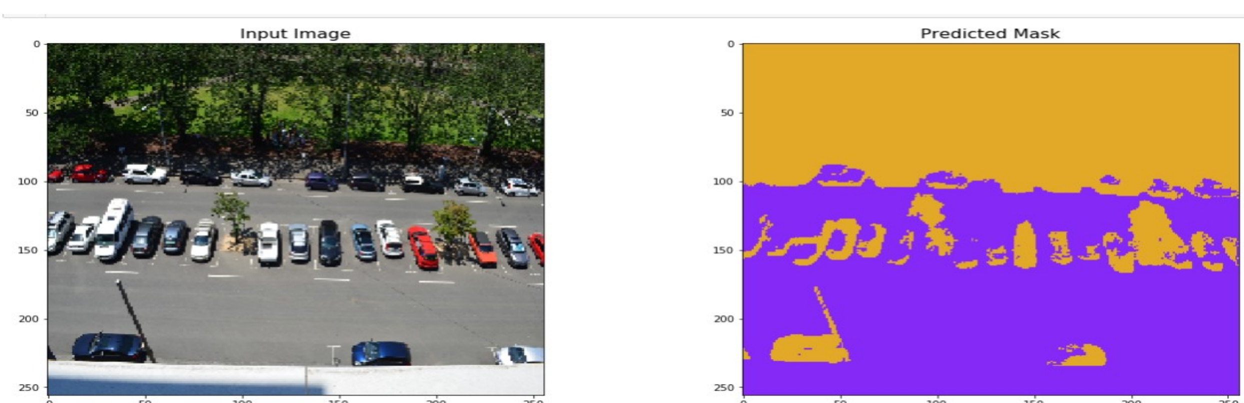
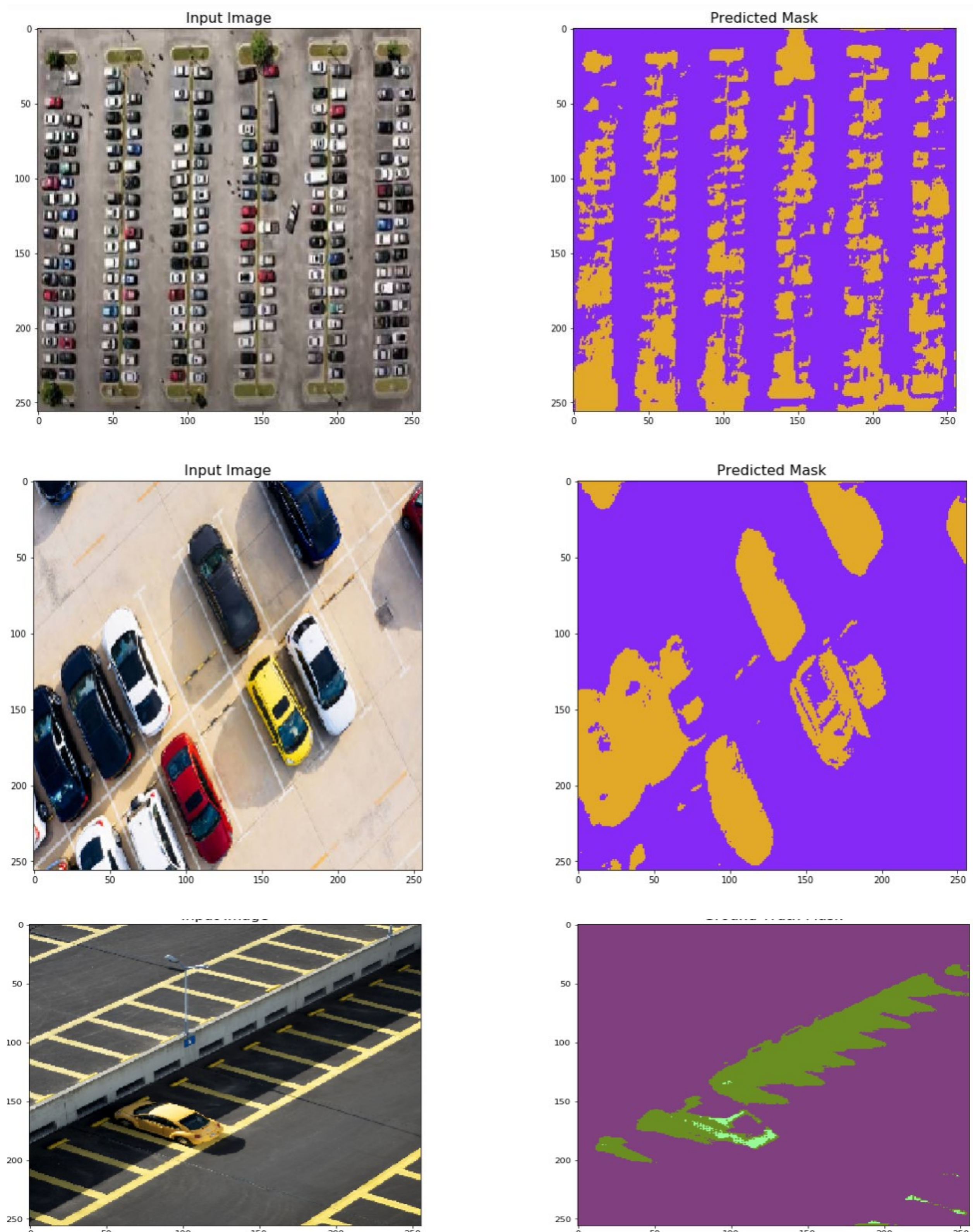


Fig 07

Predicted images




Input Image
Predicted Mask
Input Image
Predicted Mask

<u>Following diagram shown the modified UNet architecture.</u>

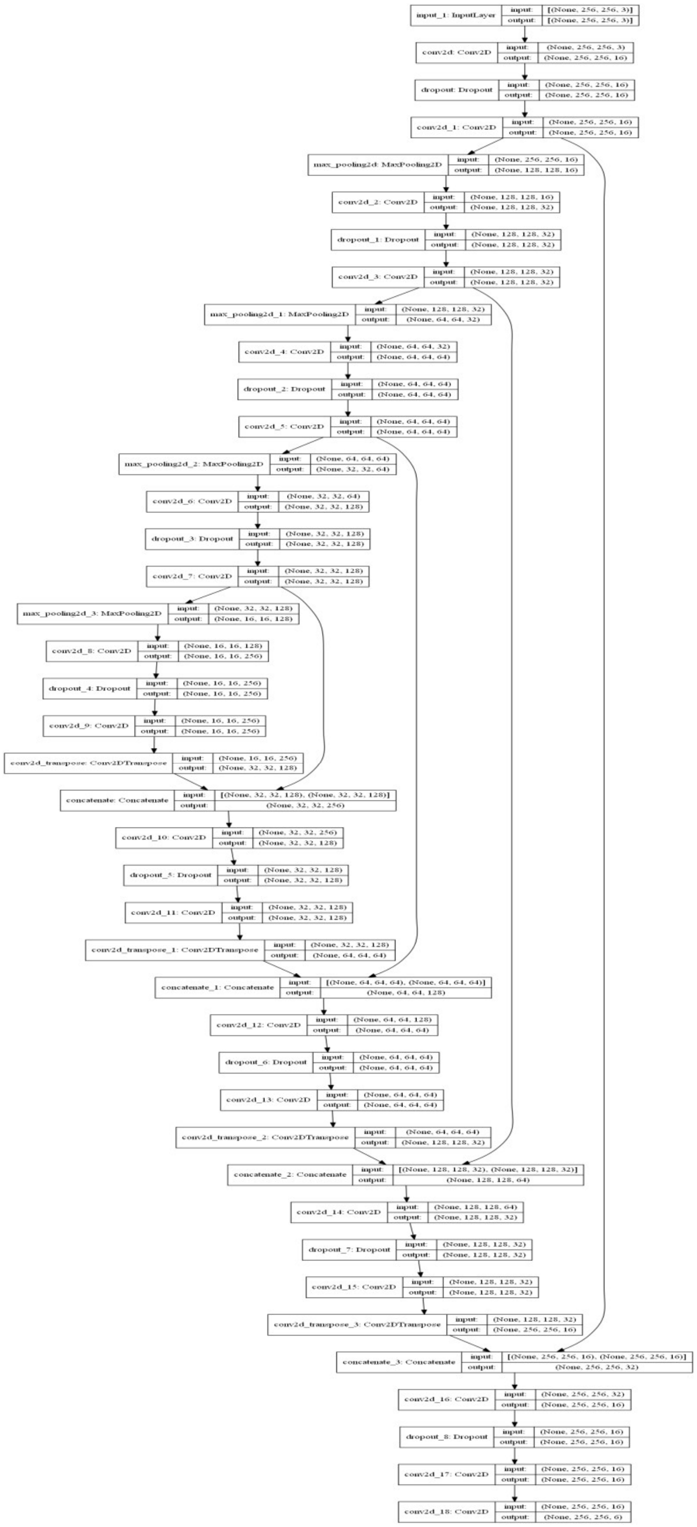


Fig 08 : modified UNet architecture

Other part of this project is to identify cars and identify they have moved. For this we used MaskRCNN pretrained model(https://github.com/matterport/Mask_RCNN) and identify the car motion in two frames and detect the empty area. If that empty area is within the parking slots that my part identified, then system say there is an empty space. If the empty area is not in the parking slots (empty area detected is in the lane or any other area. This area is detected by the classes that I gave to UNet when training), then system will reject that event even a motion is happened. So our system can identify the parking slots that get free after car has gone.

Fig 09 : incorrectly predicted without using semantic segmented mask

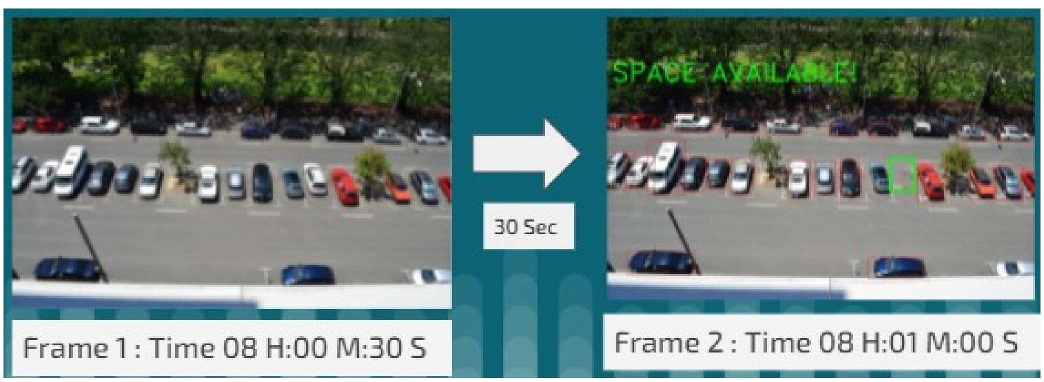


Fig 10 : correctly predicted when apply semantic segmented mask

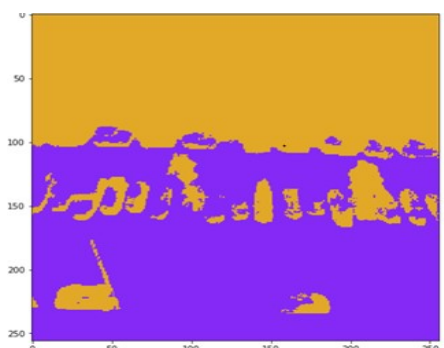


Fig 11 : Applied semantic segmented mask