evaluation of the detector's performance. Afterwards, the process of the design of the CNN has been performed. In this phase, the type and size of the input layer are defined. For classification tasks, the input size is chosen as the size of the training images. For detection tasks, CNN should analyze smaller portions of the image, so the input size was chosen as a 32x32 input size, similar to the smallest object in the dataset. Next, the middle layer of the network is defined.. The medium layer created here is made of repeating blocks of convoltional, relu (rectified linear units) and pool layers. Finally, a final layer consisting of fully connected layers and a softmax loss layer was created. Next, the design of the CNN is completed by combining the input, middle and final layers.

The third step is to configuration of training options. At this stage, the training options for the Faster R-CNN method have been configured in four steps. In the first two steps, the region proposals used in the Faster R-CNN and the detection networks are trained. In the last two steps, the first two steps were merged to form a single network. Later, the training options configuration for the R-CNN method were performed in one step. Here, the network training algorithm is configured using an SGDM with an initial learning rate of 0.001.

In the fourth step, training of fast R-CNN and R-CNN object detectors was carried out. At this stage, image patches were extracted by selecting from the training data during the training process. Two kinds of name value pairs are used here. It has been checked which image patch is used with these. Positive training samples here are examples with 0.6 to 1.0 with accuracy boxes, as measured by the bounding box intersection of the unity metric. Negative education examples are examples that overlap between 0 and 0.3. Maximized values for these parameters were selected by testing the trained detector on the verification set.

Finally, the process of evaluation of trained fast R-CNN and R-CNN detectors were carried out. At this stage, the detection results were collected and evaluated by running the trained detectors on the test set.

4 Conclusion

The proposed vehicle detector has been successfully trained by using Faster R-CNN and R-CNN deep learning methods on the sample vehicle datasets and the vehicle detection process has been successfully performed by the trained vehicle detector being tested on the test data set. As the output of the study, the image frames are shown in Figure 1 and Figure 5 for fast R-CNN and in Figure 2 and Figure 6 for R-CNN.

Figure 1 *Result of Frame for Faster R-CNN on test data set 1 [10]*

Figure 2 *Result of Frame for R-CNN on test data set 1 [10]*

Besides, a Precision / Recall (PR) curve is created to highlight the sensitivity state of our detector regarding the degree of recall at various levels, and the mean average precision(map) values are obtained from faster R-CNN and R-CNN respectively, at approximately 0.73, 0.76 and 0.64, 0.65 values. The Precision / Recall (PR) curve is shown in Figure 3 and Figure 7 for fast R-CNN and in Figure 4 and Figure 8 for R-CNN. Furthermore, according to the results, that is shown in Table 1, obtained for the purpose of vehicle detection, results obtained via Faster R-CNN method have higher detection quality and average sensitivity value