MiniProject_20231499

Introduction

- This mini project is done for the classification and detection of the speed limit signs.
- The outcome expected was to correctly detect the sign on a set of images and classify it for the speed limit.
- A Dataset was provided for testing and training the model for the task.

Dataset

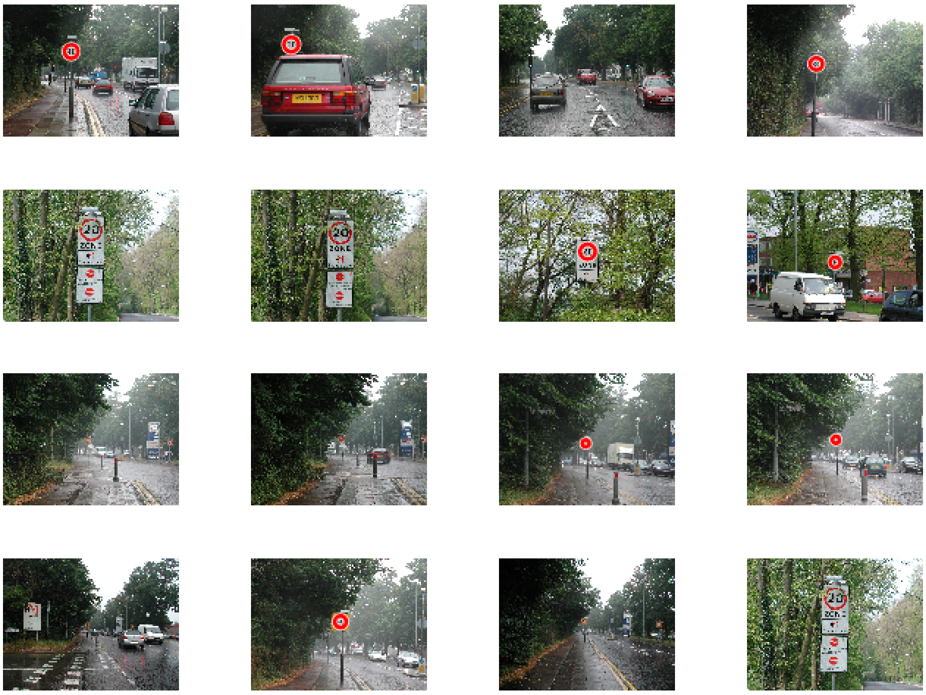
- The dataset provided consisted of 5 label categories and one stress data along with gold standard data.
- To process the data and help with training the model, manually and randomly the dataset was split into training and testing, **the .rar file for the whole dataset used is attached in the submission.** For the validation the train dataset was then split into 80:20 ratio.

Preprocessing

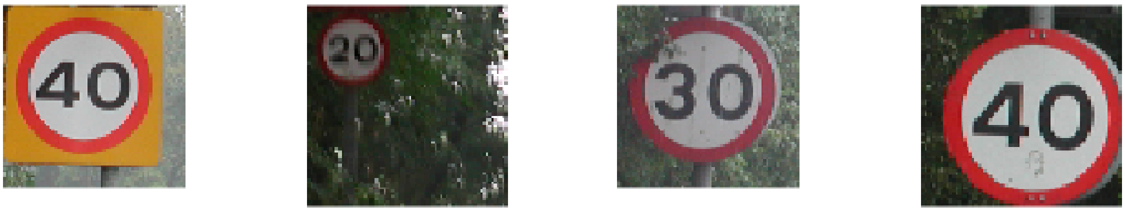
- The imageDataStore was used to load the images from the folder and to automatically detect the class label for training the data.
- The images were resized to 32 x 32 x 3 , before passing the data to the training data augmentation was done.
- For data augmentation imageDataAugmentor was used and the image was transformed for reflection and translation to flip and tilt image before training.

Object Detection

- For object detection the inbuilt function of matlab imfindcircles was used.
- This function works on the basis of hough transform that detects any circles in binary image with the specifies property to then give the radius and center points in x, y axis of the detected circle.
- For this specific dataset after multiple trail and errors the parameters selected were bright object polarity as the images are mainly in daylight and the size of the pixel ranging from 9 to 70 pixels, finally the sensitivity was set to 0.82.
- After this around 70% images from the dataset correctly identified the signs. The following image shows the images detected and the bright circle that is the exact position of the detected circle in image that was drawn using viscircles function.



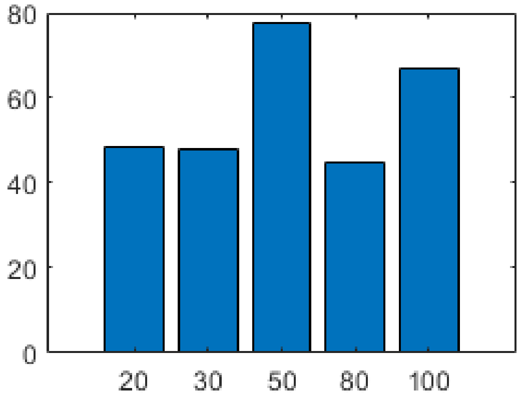
- After the image detection the image was cropped by using the rectangle function of cropimage and the rectangular bounding box was determined using the radius and the center values obtained by the imfindcircles function. The pixel size 70 was used for cropping the image.
- After cropping the final image looked like the following.



- Although most of the images were detected the images with disturbances or some part of the board hidden due to leaves etc. were not detected and very small images also were not detected , due to difference in position of sign and its distance from vehicle when image captured showed that the frames cropped on small images has 30 noise of the environment.
- The accuracy of detection can be improved by manually implementing the houghs and handling all types of images differently rather than same size bounding box.
- The cropped image was then passed to the classify where the trained model predicted the sign.

Dataset Classification

- For the purposes of speed limit classification, a deep learning approach was used and the model was trained accordingly on the dataset.
- The layers were specified for training the neural network, a series of convolutional2d layers along with batch normalisation and relu activation and then transformation with maxpooling was done.
- Lastly the fullyconected layer was used to drop down the final classification to 5 for 5 labels and a softmax layer follwed by final classification layer was added.
- The dataset was trained with stochastic gradient descent optimizer for 10 epochs with a initial learning rate of 0.0001 and drop factor of 0.1.
- After training the model was stored for further used and loaded to predict the label.
- For prediction the classify function was used and the labels achieved from classify were compared to actual dataset label to then find accuracy of the model trained on that specific dataset.



- After training, the model gave a accuracy of 52 % on the test data, and when the model was tested again on the whole dataset the accuracy achieved was 54 %.
- Apart from that the model was individually tested for accuracy on all the speed limits. For limit 20 a accuracy of 49% was achieved, for limit 30 accuracy of 48% was achieved, for limit 50 the accuracy was 76%, for limit 80 accuracy was 45% and for limit 100 the accuracy was 67%.
- From the observation we can conclude that for speed limit 50 the highest accuracy was achieved and the lowest one was achieved for speed limit 80. The accuracy for limits 20, 30 and 80 were almost similar.
- From the observations we can tell that the overall model performed greatly for all the categories considering the model was only trained on around 1200 images combined of each class and the quality and size of images was extremely poor. To improve the accuracy further with neural net might be tough but other algorithms can be implemented such as HOG etc. instead of NN that can give better results.

Conclusion

- Although based on the dataset and methods used the images gave good accuracy on individual label and also the overall accuracy for the images was good, but the model failed on the stress dataset as it was very hard for it to predict the correct label and detect.

- The reason for this might be that more than 50 % of the signs in dataset were of speed limit 40 but in the training set 40 was not present so the model was not trained on it properly and along with that in 3-4 images the signs were either far or blur that it was also not possible for the human eye to see them.
- Considering the dataset and the methods used for detection and classification the overall algorithm and flow of the project was good but can be surely improved by some other considerations and training model on more images would definitely help.
- Along with images the methods suggested in each part would also help improve the overall accuracy of the model and aid in more discrete image detection and classification.

References

- https://in.mathworks.com/help/matlab/ref/matlab.io.datastore.imagedatastore.spliteachlabel.html
- https://in.mathworks.com/help/deeplearning/ref/imagedataaugmenter.html?s_tid=doc_ta
- https://in.mathworks.com/help/deeplearning/ref/nnet.cnn.layer.layer.html?s_tid=doc_ta
- https://in.mathworks.com/help/deeplearning/index.html?searchHighlight=deep learning toolbox&s_tid=srchtitle
- https://in.mathworks.com/help/images/ref/imfindcircles.html?s_tid=doc_ta
- https://in.mathworks.com/help/deeplearning/ref/trainingoptions.html?s_tid=doc_ta
- https://in.mathworks.com/help/images/ref/viscircles.html?s_tid=doc_ta