INT-248 DEEP LEARNING PROJECT

VEHICLE DETECTION USING CUSTOMIZED CNN TECHNIQUE

Name:-B. Om Sudha

Reg.no:-11703536

Mail:- omsudha100@gmail.com

Mobile.No:-8374655702

Abstract:- Most object detection models cannot achieve satisfactory performance under nighttime and other insufficient illumination conditions, which may be due to the collection of data sets and typical labeling conventions. Public data sets collected for object detection are usually photographed with sufficient ambient lighting. However, their labeling conventions typically focus on clear objects and ignore blurry and occluded objects. Consequently, the detection performance levels of traditional vehicle detection techniques are limited in nighttime environments without sufficient illumination. When objects occupy a small number of pixels and the existence of crucial features is infrequent, traditional convolutional neural networks (CNNs) may suffer from serious information loss due to the fixed number of convolutional operations. This study presents solutions for data collection and the labeling convention of nighttime data to handle various types of situations, including in-vehicle detection. Moreover, the study proposes a specifically optimized system based on the Faster region-based CNN model. The system has a processing speed of 16 frames per second for 500 × 375-pixel images, and it achieved a mean average precision (mAP) of 0.8497 in our validation segment involving urban nighttime and extremely inadequate lighting conditions. The experimental results demonstrated that our proposed methods can achieve high detection performance in various nighttime environments, such as urban nighttime conditions with insufficient illumination, and extremely dark conditions with nearly no lighting.

1. **INTRODUCTION:-** In these, first we are loading dataset and training and doing validation of dataset. After that, plotting a summary for dataset then model is processed and done augmentation , main model work is also then after main model building coding in compiler and in model fit run the model and

finally after all the process plotting a graph of validation and training and accuracy, loss of train and validation and Actual labels are also printed.

2. PROPOSED WORK:

2.1.DATASET: Vehicle detection dataset is downloaded from kaggle

Link:- https://www.kaggle.com/rishabkoul1/vechicle-

Dataset is divided into 32 batch size, seeding to 42 , image size of 160, 160 pixels and now dataset is processed of images and in train set 20% is sent for validation purpose and shuffle is true because printing of random images are done. Now in train data we are giving class name for every vitamin is printed after that we are plotting images using subplot (6 sections in one row) and giving title to every image and train_ds.take(1) it means batch of one is used and now as image maybe in different different pixels from 0 to 255 so it takes time every image for loading we are converting every image/255 so that it gives value from 0 to 1 it will be easy now to load the data and no wasting of time is done.

2.2.PREPROCESSING: As we know if images is zoom the images we loose the quality of an image for that we have two types of process for preprocessing one is rescale and the other is resize. In reshape we can use our choice of pixels and convert an image. In rescaling same 1/255 so that every image display in same pixels.

2.3.MODEL AUGMENTATION: A image can be flipped, rotated, zoomed displayed depend upon your choice for that we are using RandomFlip, RandomRotate, Randomzoom etc. random images are selected and augmentation is done randomly.

2.4.MAIN MODEL: In these we are listing input images, model augmentation, model processed, maxpool2d, flatten, conv2d, dropout, dense, input image shape is given of our choice of image in pixels, Max pooling operation for 2D spatial data. Down samples the input representation by taking the maximum value over the window defined by pool size for each dimension along the features axis. The window is shifted by strides in each dimension. Con2d is to train 2d image. Flatten is converted to one vector form and dropout we are reducing so that our model is not overfitted.

2.5.REFERENCES:

1. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Proceedings of the International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015. Appl. Sci. 2019, 9, 4769 27 of 27

2. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016.

3. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 24–27 June 2014.

4. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 13–16 December 2015.

5. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Trans. Pattern Anal. Mach. Intell. 2017, 39, 1137–1149. [CrossRef] [PubMed]

6. . Chen, Y.; Wu, B.; Huang, H.; Fan, C. A Real-Time Vision System for Nighttime Vehicle Detection and Traffic Surveillance. IEEE Trans. Ind. Electron. 2011, 58, 2030–2044. [CrossRef]

7. Chen, Y.; Chiang, H.; Chiang, C.; Liu, C.; Yuan, S.; Wang, J. A Vision-Based Driver Nighttime Assistance and Surveillance System Based on Intelligent Image Sensing Techniques and a Heterogamous Dual-Core Embedded System Architecture. Sensors 2012, 12, 2373–2399. [CrossRef] [PubMed]

2.6.COMPILER: Here code is built using model.compiler and finding accuracy and loss

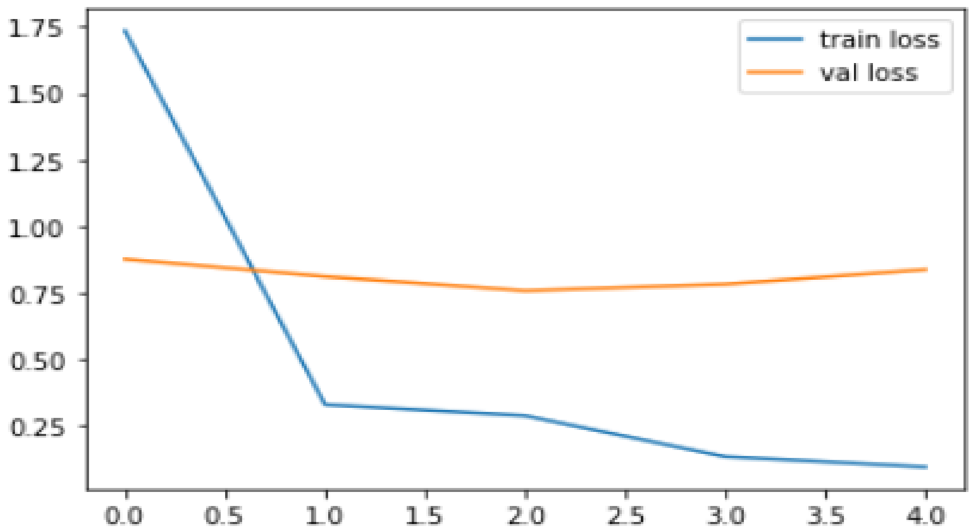
2.7.RUN MODEL: By, using Epoch run the data how many times you want (initialized in epoch for suppose epoch=10 then from 1 to 10 loss and accuracy is printed .

2.8.PLOT A GRAPH:- A graph is plotted between test and validation loss. And another graph for test and validation accuracy.

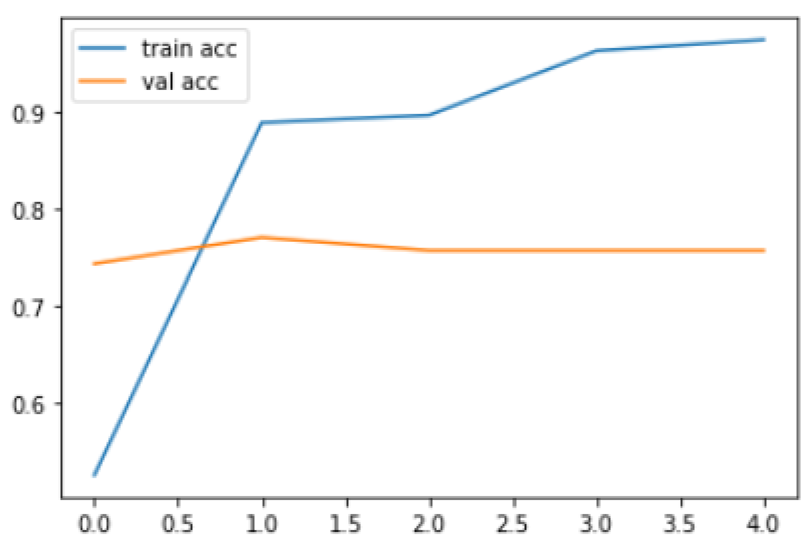
3. RESULT AND DICUSSION:

3.1.ACCURACY: Test accuracy of validation for top 3 i.e train ,truck and bus is been considered,

```
# loss
plt.plot(r.history['loss'], label='train loss')
plt.plot(r.history['val_loss'], label='val loss')
plt.legend()
plt.show()
plt.savefig('LossVal_loss')
```




```
plt.plot(r.history['accuracy'], label='train acc')
plt.plot(r.history['val_accuracy'], label='val acc')
plt.legend()
plt.show()
plt.savefig('AccVal_acc')
```



4. **CONCLUSION:-** CNN is inspired from biological process. It is popular techniques very dependent on size and quality of training data. Over fitting is reduced by using CNN model by using dropout. A loss function estimates how far off a model’s predictions are from the truth. After every batch of data gets run through the model, the optimizer tries to reduce the loss by adjusting the weights. The amount it changes the weights is determined by the learning rate. If you set the learning rate too high, then the optimizer may not be able to find optimum values, but if you set it too low, then it will take a long time for the model to find optimum values. By using a optimizer such as the Adam Optimizer.

5. **PLAGIARISM REPORT:-**