Traffic Sign Classification using Convolution Neural Network

A. Introduction

Traffic Sign Classification deals with identification of traffic sign in the given image/frames of images. There are various important applications to it. One of its applications is to to solve the problem of accidental loss of life and property, wherein the aim is to increase the driver's focus by automatic detection and classification of these traffic signs for the driver[1]. It is also an important use case in autonomously driven vehicles which extensively rely on such systems to appropriately follow the traffic rules [2][3].

A convolution neural network is a class of deep learning networks, used to examine and check visual imagery. It is used to train the image classification and recognition model because of its high accuracy and precision.

Some of the major challenges associated with this application are the changes in illumination in the captured sign images, weather conditions, occlusion, damages to the signs, cascade of the traffic signs [4] and the fact that the standard priority signs adopted internationally differ in shape, color, and border [5]. We have tried to address these issues by choosing datasets having signs of different countries and using ColorJitter transform to artificially mimic these illumination changes.

While solving this problem, several challenges were faced along the way. Class imbalance in the datasets caused the model to favour classes with more samples. It was addressed by using weighted loss function. Availability of resources to train large networks like AlexNet is another challenge. Free resources available at Google Colab and Kaggle does help in the training, but the computational power provided in the free version is still limited.

The goal behind this application is to compare the performance of different CNN architectures in classification of different traffic signs against datasets of varying classes and samples, analyze the effects of different hyper-parameters and choose a final model as per these evaluations. We expect the datasets with more number of samples to give better trained models and deeper architectures to give the most performance on classification. Several models have been trained so far, results of which have been shown in following sections.

Final comparison will be made after doing hyper-parameter tuning. Results will be evaluated using the metrics F1 score, AUC(Area Under Curve), precision and recall.

B. Proposed Methodologies

The chosen datasets were taken from Kaggle.com and are accessible via the links specified in references. Architectures are trained on a subset of the original data-set, updated statistical details of which are as follows:

Dataset	Classes	Training Samples	Test Samples
Dataset 1 [6]	15	2617	710
Dataset 2 [7]	12	7956	1228
Dataset 3 [8]	8	13984	4590

Table 1. Dataset Statistics.

Subset of classes were chosen based on the max number of samples available per class. Train-set is split into train and validation with split size of 0.2. Whereas separate samples are available for testing in the datasets.

The samples in the dataset vary from each other in terms of their colour, shape, borders. In dataset 1 the maximum image size is 3192 x 1400 and minimum size is 13 x 11. In dataset 2 the maximum image size 4608 x 3456 and minimum size is 42 x 50. In dataset 3 the maximum size is 1496 x 974 and minimum size is 22 x 44. The samples also vary in terms of their complexity. Some samples include a lot of background information whereas others are more focused on the traffic signs itself as shown in the figure below:

Figure 1. Random Image Samples Grid

As briefed before the chosen datasets also have class imbalance. Average number of samples available per class in dataset 1 is 174 wherein the lowest number of samples is 101 and highest being 242. For dataset 2 average samples is 663 with lowest and highest being 399 and 1075. For dataset 3 average samples is 1748 with lowest and highest being 1403 and 2243 respectively.

We have chosen AlexNet, VGG-11 and ResNet-18 as our architectures. The rationale behind the choice of the architectures is that these are the runner-up/winners of the ILSVRC 2012, 2014 and 2015. As ILSVRC is a classification challenge of objects in images, so we expect them to perform well for our problem too where the objects now are traffic signs.

AlexNet has 5 convolution layers and 3 fully connected layers. VGG-11 has 8 convolution layers and 3 fully connected layers. ResNet-18 has 17 convolution layers and 1 fully connected layers. Computational complexities for these are shown in Figure 2.

So far we have trained these 3 architectures against each of the 3 chosen datasets without transfer learning to get a

	Learnable parameters (in millions)	GFlops	Training time (in seconds)
AlexNet	57.05±0.02	0.71	3002 (for dataset1) 3151 (for dataset2) 5761 (for dataset3)
VGG-11	128.82±0.02	7.63	3347 (for dataset1) 4070 (for dataset2) 7353 (for dataset3)
ResNet-18	11.18	1.83	3071 (for dataset1) 3300 (for dataset2) 6600 (for dataset3)

Figure 2. Architecture Comparison

total of 9 models. These all were trained with same fixed values for hyper-parameters with batch size as 32, loss function as weighted cross entropy and learning rate as 0.0001, input image size as 224x224 and epochs=10. Results of these models are specified in the following sections.

Next we will be doing hyper parameter tuning on at-least the learning rate within range of (0.01, 0.001, 0.0001) and evaluating them using the specified metrics. The rationale behind the chosen range is that we are expecting a increase in model performance with a slightly higher learning rate because of our choice of using Adam as the optimizer [9].

C. Attempts at solving a problem

Image samples are first pre-processed to the mentioned image size and applied with transforms of ColorJitter with brightness = (0.5,1.2) , RandomHorizontalFlip, RandomAdjustSharpness and finally the image is normalized.

As discussed earlier, while solving the problem several challenges were encountered and some of them took more than one attempt to solve. Environment changes in causes various the inconsistencies in the real-time images. Various configurations of transforms were used to mimic these. Some of these transformations led to extremely distorted images which led to very low performing models. Effects of each transformation was individually analysed to get an appropriate configuration of transforms.

To address the class imbalance in the datasets, initially WeightedRandomSampler was used[10]. This sampler did not integrate well with our setting as we required train data to be split into train and validation sets and appropriate documentation were not available to handle this. Finally it was addressed using Weighted Cross Entropy Loss [11].

After addressing the issues above, the following results were obtained as per the fixed hyper-parameters discussed in preceding section Figure 3:

Upon analysis, with the current results VGG-11 seem to performing well on all the 3 datasets. The second best performance is observed for ResNet-18 and the lowest performance is observed for AlexNet. Possible reasons for this behavior could be because VGG-11 is the deepest and has the most number of learnable parameters than the others. More detailed comparison will be done in the final report.

		AlexNet	VGG-11	ResNet-18
Dataset 1	Train Acc.: Test Acc.: Test F1:	53.12% 46.90% 0.46	65.62% 57.32% 0.57	62.50% 53.09% 0.53
Dataset 2	Train Acc.: Test Acc.: Test F1:	84.38% 86.72% 0.86	93.75% 85.26% 0.85	90.62% 79.64% 0.79
Dataset 3	Train Acc.: Test Acc.: Test F1:	65% 75.14% 0.75	84.3% 79.23% 0.79	80% 61.80% 0.61

Figure 3. Model Training Results

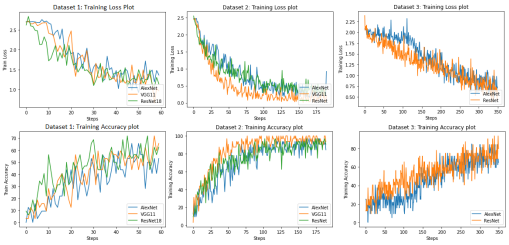

Figure 4. Plot for Comparisons on Training Results

D. Future Improvements

As a next step, we would be doing hyper-parameter tuning as per the range discussed in preceding sections to get a better set of hyper-parameters. Then the same models will be trained on the new hyper-parameters along with 2 additional models with transfer learning. Finally a comparative study would be performed on the results obtained.

E. References

[1] Seyedjamal, Zabihi. Detection and Recognition of Traffic Signs Inside the Attentional Visual Field of Drivers. The University of Western Ontario, 3 Oct. 2017, Thesis Paper.

[2] “Traffic Sign Recognition (TSR).” Car Rental Gateway, Research Gate

[3] Junzhou, Chen, et al. Research Gate

[4] Štefan , Toth. Difficulties of Traffic Sign Recognition. ITMS 26220120050 supported by the Research and Develop-ment Operational Programme funded by the ERDF, 2012, Research Gate.

[5] Seyedjamal, Zabihi. Detection and Recognition of Traffic Signs Inside the Attentional Visual Field of Drivers. The University of Western Ontario, 3 Oct. 2017, Thesis Paper.

[6] DILIP JODH, SARANG. Indian Traffic Signs Prediction (85 Classes), DILIP JODH,

SARANG. Indian Traffic Signs Prediction (85 Classes), https://www.kaggle.com/datasets/sarangdilipjodh/indian-traffic-signs-prediction85-classes.

[7] PARSASERESHT, SARA. Persian Traffic Sign Dataset (PTSD), https://www.kaggle.com/datasets/saraparsaseresht/persian-traffic-sign-dataset-ptsd.

[8] DELTSOV, DANIIL. Traffic Signs (GTSRB plus 162 Custom Classes) - Dataset 1. https://www.kaggle.com/datasets/daniildeltsov/traffic-signs-gtsrb-plus-162-custom-classes.

[9] Akshay L, Chandra. "Learning Parameters, Part 5: AdaGrad, RMSProp, and Adam." Learning Parameters, Part 5: AdaGrad, RMSProp, and Adam, Akshay L Chandra, Sept. 2019, https://towardsdatascience.com/learning-parameters-part-5-65a2f3583f7d.

[10] PyTorch.org. "WeightedRandomSampler." Torch.Utils.Data.WeightedRandomSampler, The Linux Foundation, PyTorch Documentation.

[11] PyTorch.ORG. "CrossEntropyLoss." Torch.Nn. CrossEntropyLoss, The Linux Foundation,PyTorch Document