Traffic Sign Classification Using Convolution Neural Networks

Abstract

Traffic Signs are easy to identify for humans. However for computer systems, its a challenging problem. Traffic Sign Classification is the process of automatically recognizing traffic signs along the road. In this report, the performance of different CNN architectures for Traffic sign classification is analyzed against datasets of varying classes and samples. Effects of different hyperparameters on model performance and its convergence is also studied. The trained models are evaluated using different performance metrics and a comprehensive comparison report is being generated. Careful consideration was given on the choice of datasets for the study to make sure they can represent signs of varied variety and real-world distortions and weather conditions. All the models performed the best for dataset 2 among the chosen 3 datasets. Moreover out of all the models best performance has been shown by ResNet-18 with 96.42% accuracy. Second and third best performing models being VGG-11 and ResNet-18 with 95.11% and 91.78% respectively.

A. Introduction

A.1. Problem Statement

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].

Many of the existing solutions have come up with ways to tackle various challenges faced by the problem. For instance, some proposed a database generation method to avoid reliance on real target-domain images for training because of the difficulties in acquiring real-world data and its annotation. [4]. Some used Pyramid Transformer which is able to learn local feature representation of objects at various scales, thereby enhancing the network robustness against the size discrepancy of traffic signs. [5]

But some of the major challenges faced by the existing solutions and the problem in general are the changes in illumination in the captured sign images, weather conditions, occlusion, damages to the signs, cascade of the traffic signs [6] and the fact that the standard priority signs adopted internationally differ in shape, color, and border [7].

We have tried to address these issues by meticulously choosing datasets having signs of different countries and using ColorJitter transform to artificially mimic these illumination changes. Moreover one of the datasets also tried to take the real-world weather conditions and distortions into account by using augmentation techniques.[15].

A convolutional 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.

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 hyperparameters and finally give a comprehensive comparative analysis of the results obtained.

Three CNN architectures of varying computational complexities and three datasets were chosen. These architectures trained were trained against each of the datasets to get different models and compared/evaluated based on different evaluation metrics.

Models trained using the hyperparameters selected from the hyperparatmer tuning phase helped increase the performance of almost all the models. Specifically, the models performed the best for dataset 2 among the chosen 3 datasets. Models trained using transfer learning also showed improvement in performance.

A.2. Related Works

Several of the related works for the problem were viewed and some of them have been viewed as follows.

A comprehensive approach was proposed to recognize traffic signs from video input [8]. It firstly uses a cascade of trained classifiers to locate regions of interest by scanning the background. Upon it Hough transform is applied for detecting shape. These extracted features are being fed to a neural network for classification of the target traffic signs. Evaluation of this method is based on an image database including 135 traffic signs. Average recognition rate of 94.90 % was achieved by this method.

A rich feature hierarchical structure was proposed by Girshick et al. [9] for precise target detection and semantic segmentation. This method uses R-CNN for region proposals rather than the traditional mechanisms. The target regions extracted using R-CNN are fed to the deep convolutional network for classification. More than 2000 target candidate regions got extracted. Major drawback with this approach was the detection speed because the method involve running classification on each of the target regions.

Douville, et al. [10] work focused on classifying traffic signs in background clutter under rotation, scale and translation invariant conditions. His research demonstrated the possibility of a robust traffic sign classification model in

real-time using a single Fast Fourier Transform (FFT), a bank of filters and a trained neural network.

Meanwhile some researches started using different features to classify sign boards such as Haar-like features and histogram of oriented gradients (HOG). They use a permutation-based image feature to describe traffic sign, which takes advantage of illumination variance and fast implementation. Gradients in the colour image were also taken into consideration. However problem in this method occurs when there is a large monochrome in the center or by classification by colour beforehand [11].

B. Methodology

The chosen datasets contain images of traffic signs from different countries. They contain signs from India, Persia and others. They were taken from kaggle.com and are accessible via the links specified in references. Architectures are trained on a subset of the original dataset, updated statistical details of which are as follows:

Dataset	Classes	Training Samples	Test Samples
Dataset 1 [12]	15	2617	710
Dataset 2 [13]	12	7956	1228
Dataset 3 [14]	8	13984	4590

Table 1. Dataset Statistics.

Classes were filtered based on the max number of samples available per class. Trainset is split into train and validation with split size of 0.2. Whereas separate samples are available for testing in the datasets.

The datasets suffer from class imbalance as 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.

The samples in the datasets 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:

Image samples are first pre-processed to 224x224 image size and applied with transforms of ColorJitter with brightness = (0.5,1.2) , RandomHorizontalFlip, RandomAdjustSharpness and finally the image is normalized. Instead of doing weighted sampling of the images[15] to handle the class imbalance, a weighted cost function [16] is being used.

Figure 1. Random Image Samples Grid

B.1. CNN Models

AlexNet, VGG-11 and ResNet-18 have been chosen as the architectures. The rationale behind the choice of the architectures is that these are the runner-up/winners of the ILSVRC 2012, 2014 and 2015 and also the fact that these architectures casued the transition from traditional computer vision to deep learning based. 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 convolutional layers and 3 fully connected layers. VGG-11 has 8 convolutional layers and 3 fully connected layers. ResNet-18 has 17 convolutional layers and 1 fully connected layers. Computational compelexities for these are shown in figure 2.

	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

Estimated computational complexity, top 1 % accuracy, top 5 % accuracy of the other available architectures along with the chosen architectures trained mostly on ILSVRC 2012 [17] is as follows[18]:

As observed in the last 3 entries in the above table there are other architectures available with much higher computational complexity. They also have considerably high accuracy in most of the cases. Also many smaller computational footprint architectures were designed in recent years as seen in the top 3 entries above and surprisingly they still give competitive results as compared to the larger ones.

B.2. Optimization Algorithm

Adam has been used as an optimization algorithm to train the CNN models. After every 10 batches the accuracy

Model	GFLOPS	Top 1% Accuracy	Top 5% Accuracy
ShuffleNetV2 1.0x	0.149	69.55%	88.92%
MobileNet V2	0.319	71.86%	90.42%
ReXNet_1.0	0.4	77.90%	93.90%
AlexNet	0.727	63.30%	84.60%
ResNet-18	1.82	70.07%	89.44%
VGG-11	7.63	68.75%	88.87%
RegNetX-12GF	12.15	79.67%	95.03%
VGG-19	19.67	72.41%	90.80%
ResNeXt-101 32x32d	174	85.10%	97.50%

Figure 3. Computational Complexity and Top 1% and Top 5 % Accuracy for other available CNN Models

of the model is computed to monitor the performance of the optimization algorithm. Also at the same time the accuracy is also computed on the validation set. Both accuracies can then be compared to evaluate the generalization power of the model for new data.

Adam optimizer leverages the power of adaptive learning rates methods to find individual learning rates for each parameter[19]. It gives the advantages of both Adagrad and RMSprop. Adagrad is a stochastic optimization method that adapts the learning rate to the parameters. It performs smaller updates for parameters associated with frequently occurring features, and larger updates for parameters associated with infrequently occurring features [20]. RMRprop is an extension of gradient descent and the AdaGrad version of gradient descent that uses a decaying average of partial gradients in the adaptation of the step size for each parameter. The use of a decaying moving average allows the algorithm to forget early gradients and focus on the most recently observed partial gradients seen during the progress of the search, overcoming the limitation of AdaGrad [21].

Hyperparameters chosen for Adam in our training are betas as 0.9 and 0.999, epsilon for denominator as 1e-8 and weight decay as 0.

C. Results

C.1. Experiment Setup

As a first step, the 3 chosen architectures were trained against each of the 3 chosen datasets without transfer learning to get a total of 9 models. On top of these 9 models 2 additional models were trained with transfer learning. Architectures picked for transfer learning were AlexNet and ResNet-18 on dataset 1. All of these models were trained the same and fixed hyperparameters with batch size of 32, loss function as weighted cross entropy [16] and learning rate of 0.0001, input image size as 224x224 and epochs=10. Optimization algorithm and validation steps have been discussed in detail in section 3.3. These models were then evaluated on the test set with metrics: Accuracy, F1-score. Results of these models are specified Fig. 4 , Fig. 5 and Fig. 6.

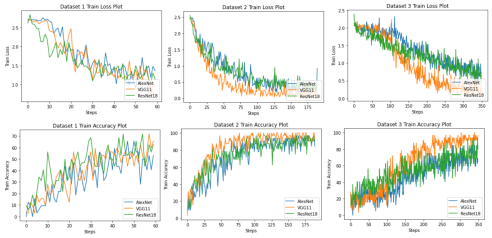

Figure 4. Accuracy and Loss Plot on 9 Models before Hyper Parameter Tuning

Figure 5. Accuracy and Loss plots for 2 Transfer Learning models before Hyper Parameter Tuning

		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 6. Performance Metrics for 9 models before Hyper-Parameter Tuning

Out of the 9 models the best performing model was then picked and hyperparameter tuning was performed on the learning rate within range of (0.01, 0.001, 0.0001) and evaluating them using mean loss and accuracy over the trainset set batches. 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 [22]. Results of the hyperparameter tuning phase are shown in section 4.3

The learning rate which gave the best performance in the hyperparameter tuning phase is then picked and used to retrain all the 11 models. These models are then evaluated using Accuracy, F1-score, precision, recall and AUC score metrics. Also TSNE has been used to visualize 4 models to compare data separability. Models used for TSNE are AlexNet with/without transfer learning on dataset 1,

ResNet-18 without transfer learning on dataset 1 and 3. Results of this phase are shown in section 4.2 along with the detailed comparison on the performance of these models against each other.

C.2. Main Results

Results of the Hyperparameter tuning phase are shown in Fig. 7

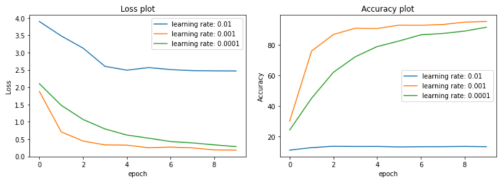

Figure 7. Accuracy and Loss Plot for Hyperparameter Tuning

As observed in the loss and accuracy plot above, learning rate of 0.001 resulted in more accuracy and faster convergence for the best performing model AlexNet on dataset 2.

This learning rate was then used to retrain all the 11 models and evaluated against the specified metrics. Performance of the trained models on test set is shown below:

	On Test	AlexNet	VGG-11	ResNet-18
Dataset 1	Acc.:	55.21%	11.4%	62.25%
	F1:	0.55	0.11	0.62
	Precision:	0.63	0.007	0.66
	Recall:	0.59	0.07	0.65
	AUC:	0.07	0.5	0.10
Dataset 2	Acc.:	91.78%	95.11%	96.42%
	F1:	0.91	0.95	0.96
	Precision:	0.90	0.94	0.95
	Recall:	0.93	0.95	0.97
	AUC:	0.08	0.06	0.09
Dataset 3	Acc.:	80.13%	91.98%	95.77%
	F1:	0.80	0.91	0.95
	Precision:	0.79	0.93	0.96
	Recall:	0.79	0.92	0.95
	AUC:	0.28	0.21	0.19

Figure 8. Performance Metrics for 9 models without Transfer Learning

	On Test	AlexNet	ResNet-18
Dataset 1	Acc.:	75.77%	70%
	F1:	0.76	0.7
	Precision:	0.85	0.74
	Recall:	0.79	0.71
	AUC:	0.05	0.19

Figure 9. Performance Metrics for 2 models with Transfer Learning

As observed in the table for models without transfer learning above, among the 3 datasets all the 3 models performed the best with dataset 2 with test accuracies of 91.78 %, 95.11%, 96.42% for AlexNet, VGG-11 and ResNet-18 respectively. There are significant differences in performance for the models between dataset 1 and dataset 2. Although its not that significant in case of dataset 2 and 3.

Training the models with transfer learning for models Alexnet and ResNet-18 improved the performance of models in both cases as seen in Fig. 9. AlexNet's performance improved by 20% whereas 8% improvement is seen for ResNet-18

Out of all the models best performance has been shown by ResNet-18 on dataset 2 with 96.42 % accuracy, 0.96 F1 score, 0.95 Precision, 0.97 Recall, 0.09 AUC on the test set. Second and third best performing models being VGG-11 and ResNet-18 respectively. Plots for the model performance during the training phase has been shown below:

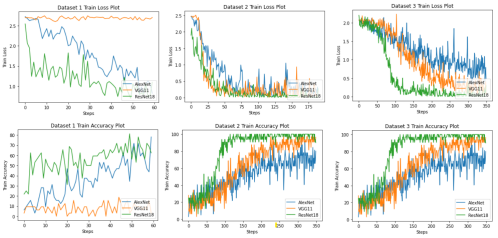

Figure 10. Accuracy and Loss Plots for 9 models without Transfer Learning

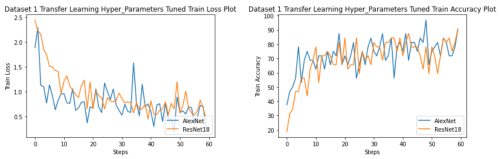

Figure 11. Accuracy and Loss Plots for 2 models with Transfer Learning

T-SNE has been used to visualize 4 models: AlexNet with/without transfer learning on dataset 1, ResNet-18 without transfer learning on dataset 1 and 3. Results of these are as follows

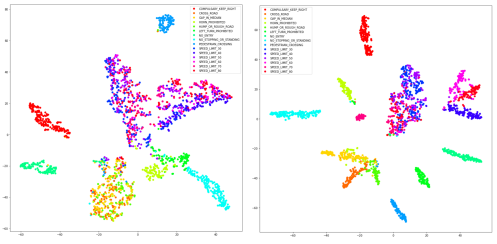

Figure 12. TSNE for Alexnet on dataset with (right)/without (left) Transfer Learning

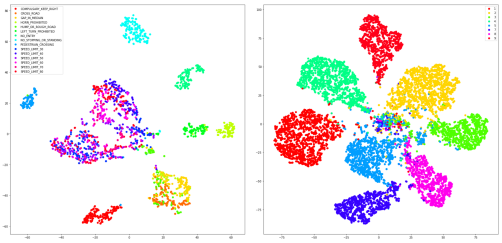

Figure 13. T-SNE for Resnet-18 for Dataset 1(left) and Dataset 3 (right)

As per the observed increased in distance among classes in Fig. 12 , we can say that transfer learning on AlexNet with dataset 1 helped the model to predict classes with more certainity.Although the model still suffered to differentiate between a few classes mainly the speed limit traffic signs.

In Fig. 13 , the bigger dataset 3 seemed to have helped the ResNet-18 model to better differentiate between the classes as prominent by the big nicely separated clusters of same classes on the right.

C.3. Ablative Study

In this section we are studying the effects of tweaking different components such as number of classes, image samples per class, choice of architectures and the hyperparameters tuned in the hyperparameter tuning phase etc.

Number of classes available for the 3 datasets were 15, 12 and 8 for Dataset 1, 2 and 3 respectively. Although the number of classes in dataset 1 are the highest but the available training samples are the lowest i.e. 2617. Because of this all the models were not able to train well for dataset 1 and thus poor performance has been observed.

Learning rate used for the initial training phase was 0.001 but as observed by the results shown in Fig. 8 and Fig. 9 , a higher learning rate of 0.01 helped the models to converge faster in all the cases except for VGG-11 on dataset 1 wherein the model failed to converge.

Among the chosen architectures, the computational complexties for AlexNet, VGG-11 and ResNet-18 are 0.71, 7.63 and 1.83 respectively as shown in Fig. 2. This is also reflected in the amount of training time required for each of these where VGG-11 has taken the most amount of time for training and AlexNet the lowest. Also it can be observed that higher computational complexity of a model doesn't always correspond to higher performance. This is because ResNet-18 having lower computational complexity than VGG-11 have performed better for all the 3 datasets as can be seen in Fig. 8.

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] Tabelini, Lucas, et al. Deep Traffic Sign Detection and Recognition Without Target Domain Real Images. 1, arXiv, 30 July 2020. arXiv.org, Paper

[5] Papers with Code - Pyramid Transformer for Traffic Sign Detection. Paper

[6] Stefan , Toth. Difficulties of Traffic Sign Recog- nition. ITMS 26220120050 supported by the Research and Development Operational Programme funded by the ERDF, 2012, Research Gate.

[7] 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.

[8] Ruta, Andrzej, et al. "Detection, Tracking and Recognition of Traffic Signs from Video Input." 2008 11th International IEEE Conference on Intelligent Transportation Systems, 2008, pp. 55–60. IEEE Xplore, Paper.

[9] Girshick, Ross, et al. "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation." 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–87. IEEE Xplore, Paper

[10] Douville, Phil. "Real-Time Classification of Traffic Signs." Real-Time Imaging, vol. 6, no. 3, June 2000, pp. 185–93. ScienceDirect, Paper

[11] Tian T. Sethi I. Patel N.: 'Traffic sign recognition using a novel permutation-based local image feature'. 2014 Int. Joint Conf. on Neural Networks (IJCNN), Beijing, China, July 2014, pp. 947–954

[12] DILIP JODH, SARANG. Indian Traf- fic Signs Prediction (85 Classes), DILIP JODH, SARANG. Indian Traffic Signs Prediction (85 Classes), Dataset 1

[13] PARSASERESHT,SARA.Per- sian Traffic Sign Dataset (PTSD), Dataset 2

[14] DELTSOV, DANIIL. Traffic Signs (GT- SRB plus 162 Custom Classes) - Dataset 1. Dataset 3

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

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

[17] ImageNet. ImageNet

[18] Model Zoo — MMClassification 0.25.0 Documentation. Accessed 8 Dec. 2022

[19] Kingma, Diederik P., and Jimmy Ba. Adam: A Method for Stochastic Optimization. arXiv, 29 Jan. 2017. arXiv.org, Paper

[20] Papers with Code - AdaGrad Explained. Paper Accessed 8 Dec. 2022

[21] Brownlee , Jason. Gradient Descent With RMSProp from Scratch. 24 May 2021, RMSProp

[22] Akshay L, Chandra. "Learning Parameters, Part 5: AdaGrad, RMSProp, and Adam." Learning Parameters, Part 5: AdaGrad, RMSProp, and Adam, Akshay L Chandra, Sept. 2019, Learning Parameters