A Model for Classification of Traffic Signs Using Improved Convolutional Neural Network and Image Enhancement

Attoumane Loukmane
Laboratory of Signals, Distributed Systems and Artificial Intelligence (SSDIA)
ENSET Mohammedia, Hassan II University
Casablanca, Morocco
attoumaneloukmane@gmail.com

Manuel Graña
Computational Intelligence Group
Faculty of Informatics UPV/EHU
San Sebastian, Spain
ccpgrrom@gmail.com

Mohammed Mestari
Laboratory of Signals, Distributed Systems and Artificial Intelligence (SSDIA)
ENSET Mohammedia, Hassan II University
Casablanca, Morocco
mestari@enset-media.ac.ma

***Abstract*—In an advanced driver assistance system (ADAS), recognition of traffic signs is very important for safety driving. Recently, the convolutional neural networks (CNNs) have presented promising results. In this work, we propose a robust model based on VGG network by adding batch normalization operation. Dropout is also used to reduce the overfitting of the model. Due to the imbalance of the dataset, data augmentation is performed. Then, in order to enhance images, Contrast limited adaptive histogram equalization (CLAHE) and normalization are performed. The performance of the model is evaluated on German traffic sign recognition benchmark (GTSRB) dataset using different performance metrics namely confusion matrix, precision, recall. Experiments results show that, the proposed model reaches a state-of-art accuracy of 99.33 % and surpasses the best human performance of 98.84 %. This model can be used for real world system.**

***Index Terms*—Traffic signs, CNNs, GTSRB**

I. Introduction

Nowadays, advanced driver-assistance systems (ADAS) [1], are designed to help drivers and avoid accidents. Following by important challenges such as safety of pedestrian and drivers, these systems must be efficient and available in real time. Traffic sign recognition (TSR) is an important part of ADAS. Currently, a TSR system is divided in two stages: detection and classification. In this work, we focus on classification stage. However, despite the growing use of TSR systems, recognition of traffic signs encounters several challenges such as, distance, lighting and weather conditions, illumination change, partial occlusions, orientation, rotation, low quality, distortion.

978-1-7281-8084-7/20/$31.00 ©2020 IEEE

In the past, more works were based on traditional methods [2]–[9]. They used hand-crafted methods to extract features from an image. However, using hand-crafted features can generate a loss of information during extraction which results in low classification accuracy. Moreover, due to their high dimension, these methods are very difficult to design and need high number of calculations. These methods are not often robustness for real time systems.

Recently, with the power of graphics processing units (GPUs) and the availability of several public datasets such as German Traffic Sign Benchmark (GSTRB) dataset [10], deep learning based methods are used in more various works in the field of TSR. These methods use neural networks and are able to learn automatically distinct and complex features of an input image. These methods are also more robustness and achieve high classification accuracy compared to traditional methods. Indeed, there are different types of deep learning algorithms. Supervised learning algorithms such as Multilayer Perceptron Neural Network (MLPNN), Convolutional Neural Network(CNN), used in speech recognition, image classification or regression task with labelled data. Unsupervised learning algorithm such as Generative Adversarial Network (GAN), Deep Belief Network (DBN), used in speech processing, health diagnostics, image and face recognition, clustering task with unlabelled data. Reinforcement learning algorithms such as Q-learning used in exploration task without predefined data. In the last decade, several works [11]–[16] use one of deep learning methods called CNN based methods and present promising results in the field of traffic sign classification. These methods which are generally supervised, learn to extract features, from an input image and achieve state-of-

Authorized licensed use limited to: University of Gothenburg. Downloaded on December 20,2020 at 13:52:46 UTC from IEEE Xplore. Restrictions apply.

art accuracy on a labelled dataset such as GSTRB dataset [10].

The aim of this work is to present an efficient model using fewer parameters and able to classify traffic signs in real world with high accuracy. In order to build this model, a model based on VGG network [17] and improved by adding batch normalization is proposed. This model consists of 6 convolutional layers and 3 max pooling layers. Firstly, due to the imbalance from GTSRB [10] dataset, and variation in visual appearance, dataset is augmented, then also pre-processed using Contrast limited adaptive histogram equalization (CLAHE) [18] and normalization to enhance contrast on low quality images. Secondly, two first convolutional layers are used to extract key features from the input image. Max pooling layers are added to reduce parameters size and render the model invariant to small transformations. Dropout is also used in the first fully connected layer to prevent the overfitting. Finally, the proposed model is trained and evaluated on GTSRB [10] dataset using different performance metrics. Experimental results give a test accuracy of 99.33 % which achieves state-of-art performance on GTSRB [10] dataset, and surpasses the best human performance of 98.84%.

The rest of this paper is organized as follows: section II presents the related works on the traffic sign recognition field. The proposed method used to classify signs is described in section III. The experimental results are shown in section IV. Finally, in section V, conclusion is drawed.

II. Related works

Works in traffic signs classification can be classified into two categories, traditional methods and CNN-based methods. In this section, we first survey the traditional methods and then CNN based methods.

A. Traditional methods

The principle of model based on traditional algorithms is firstly to use hand-crafted methods such as histogram of oriented gradient (HOG), Localized binary model (LBP), to extract the color or shape features from an input image and finally to use a classifier in image classification. In fact, works which use traditional methods consists of two stages, detection performed by detectors such as HOG [4] and classification which use classifier such as support vector machine (SVM) [2], [3], [5], [8], [9], K-d tree [3] and random forest [3], [6], [7]. In [2], to classify the signs, a two hierarchies model based on HOG and SVMs was proposed. First, the input image was classified with HOG features and SVM. Finally, after performing perspective adjustment on the image, the classification was performed with HOG and SVM. The proposed model reached a classification accuracy of 99.52% on the GTSRB [10] dataset. Zaklouta et al. [3] proposed a three-stage traffic sign recognition. First, the authors improved color enhancement in RGB space for segmentation, moreover, they combined HOG and SVM for detecting circular and triangular signs. The traffic signs were classified using K-d trees, the Random Forests and the one vs-all SVM classifiers on HOG features. The Random Forest reached a classification accuracy of 97% on GTSRB [10] dataset. Hou et al. [5], propose a cognitively method focused on occlusion analysis. Samples occluded from real and synthetic traffic signs are classified using cascaded SVMs classifiers. These classifiers improve the classification of the occluded samples.

B. Convolutional neural network based methods

Recently, deep learning methods are much used in field of traffic signs classification. In comparison with the traditional methods based, they can learn to extract samples features automatically and achieve high accuracy. There are different types of deep learning such as GANs, Restricted Boltzmann Machine (RBM), DBNs, RNNs, CNNs, etc. Since the multilayer perceptron was not able to realize some tasks, CNNs were introduced. The CNNs are inspired by the visual cortex of mammals and widely used in the field of computer vision due their efficiency, powerful and their few parameters in training phase. Moreover, CNNs model are invariant to small translation, scaling, rotation [19] by using pooling layer and each output unit is connected only to few input unit not all, this function is known as sparsity of connections and helps to avoid overfitting in training phase and also to reduce the computational cost and the complexity of the model. The first CNN architecture is developed in 1998 to classify handwritten digits by LeCun et al. [20] and known as Lenet. In the last decade, different architectures of CNNs have been presented: AlexNet [21], VGG [17], DenseNet [22], etc.

Several works use CNN based methods in field of traffic signs recognition [11]–[16]. In [11], Ciresan et al. proposed a robust system to classify traffic sign. The images were cropped and processed using CLAHE [18] algorithm, then the images were augmented. A committee of CNNs and multi-layer perceptrons (MLP) were respectively used to train a raw image pixels and standard feature descriptors. This approach won the preliminary stage of the GTSRB Dataset. Finally, a recognition rate of 99.15% was reached by further training the nets. In [12], Sermanet and LeCun proposed a Multi-Scale Features CNN system. The traffic signs were firstly processed by resizing to 32 x 32 and converting from RGB image to YUV space. Secondly, they were augmented. Finally, the system was evaluated and reached a test accuracy of 99.17% after gray scaling. Ciresan et al. [13] presented a multi-column (MCDNN) system. The traffic signs were cropped, adjusted, equalized, normalized and augmented before classified. The model was trained with 25 columns on GPU during 37 hours and achieved an accuracy of 99.46 % on GTSRB dataset. In [14], Jin et al. proposed a hinge loss stochastic gradient descent (HLSGD) to train CNN. The method reduced the error rate, helped CNN to focus on misclassified training samples and improved training speed. After augmentation and pre-processing such as histogram equalized image, adjust image intensity values, and CLAHE [18] on images, the system was evaluated on GTSRB dataset and achieved an accuracy of 99.65%.

Authorized licensed use limited to: University of Gothenburg. Downloaded on December 20,2020 at 13:52:46 UTC from IEEE Xplore. Restrictions apply.

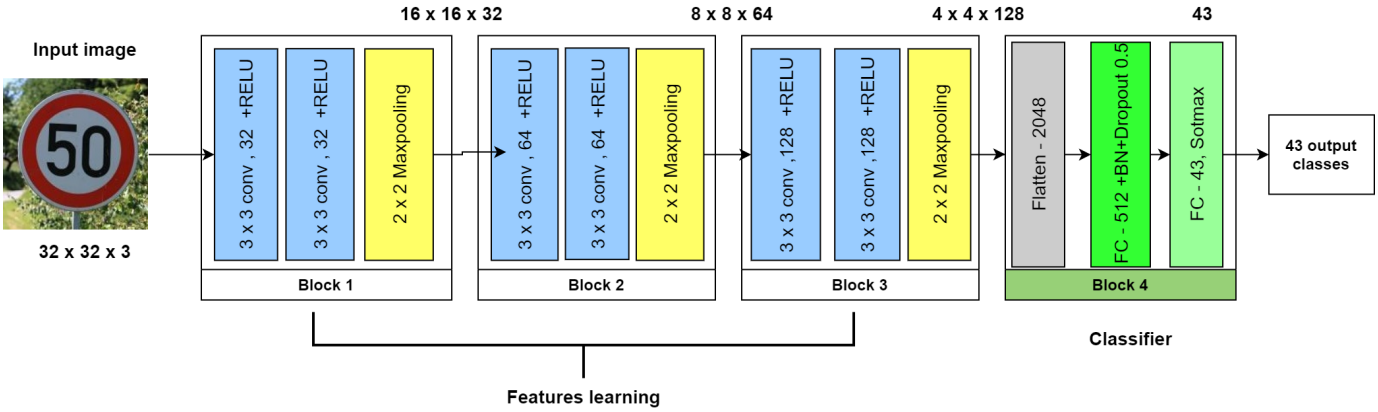


Fig. 1: Architecture of the proposed model. The first three blocks learn features and the last block used for classification.

III. Method

In this section, firstly we present a brief overview of CNN, secondly we introduce our proposed architecture and finally the images pre-treatment steps are presented.

A. Convolutional neural networks

Since the CNNs are multi-layer neural networks, they are usually composed of two parts: Feature learning part which consists of convolutional layer, pooling layer and classification part which consists of fully connected layers and classification layer. Each layer contains neurons with 3D volumes, width, height and depth.

1) Convolutional layer: Used for extracting local features from input data, the convolutional layer is the essential layer of a CNN. It is usually the first of a CNNs and uses a convolution operation, for the efficiency of the algorithm. Convolution is performed at the input image using a filter/kernel. The depth of kernel is the same of the input. The output is called feature map. A convolutional layer is characterized by a number and size of maps, a kernel size.

2) Non linearity: After a convolutional layer (or fully-connected), an activation function is applied. Rectified line unit denotes RELU is used in the output of convolutional layer for convert all the negative values to zero and improves the training [21].

3) Pooling layer: Also known as sub-sampling layer, it always follows a convolutional layer, it is used to reduce the noise and the spatial size of the convolved features without losing information. Pooling layer makes the outputs invariant to small translation [19] and decreases the number of parameters. There are generally two types of pooling layer: Max pooling, and Average Pooling. After pooling, the depth of feature map remains unchanged.

4) Fully connected layer (FC): It comes after a succession of convolution and pooling layers. It consists of a one dimensional features vector which takes the result of convolution or pooling layer after being flattened and reaches a classification decision. The flatten layer convert a 3-d volume into one dimensional, the result is fed into the first fully connected layer. Every input neuron is connected to all neurons in the output. At the end, a softmax activation is applied on a fully connected layer, and used to predicted the output using probability. The softmax layer contains the same number as the number classes in the dataset.

B. Architecture of the proposed model

Due to the efficiency and the power of CNNs in the image classification task, we propose a network architecture based on VGG [17] with fewer parameters. Our proposed model is shown in Fig. 1 and consists of several layers, there are input layer, 6 convolutional layers, 3 maxpooling layers, 1 fully connected layer with dropout and batch normalization and finally one softmax layer as the output. In order to reduce parameters, our model is fed by a 32 x 32 RGB image in input when the input of original VGG [17] is 224 x 224. Each output features maps feeds the input maps of the next layer. As shown in Fig. 1, the model is divided by 4 blocks. The first three block consist of 2 convolutional layers, 1 max pooling layer. Each convolutional layer has a kernel of 3 x 3 with a stride 1 for extracting features use RELU as activation function.

Firstly, we apply the two first convolutional layers on the input image with a filter of 3 x 3, we obtain twice 32 x 32 x 32 feature maps. The maxpooling layer reduces the feature maps to 16 x 16 x 32 using a kernel size of 2 x 2 with stride 2. Then we apply two convolutional layers with a filter of 3 x 3, we obtain twice 16 x 16 x 64 feature maps. The second max pooling layer reduces the feature maps to 8 x 8 x 64 using a kernel size of 2 x 2 with stride 2. Finally we apply two convolutional layers with a filter of 3 x 3, we obtain twice 8 x 8 x 128 feature maps. The third max pooling layer reduces the feature maps to 4 x 4 x 128. After the first three blocks, a flatten layer is added to convert three-dimensional maps to vector, his output size is 2048 neurons. After flattening the output features maps, a fully connected layer with 512 output neurons is added. In this fully connected layer, batch normalization [23] is added to increase accuracy and dropout with a 0.5 probability to avoid overfitting in training phase.

Authorized licensed use limited to: University of Gothenburg. Downloaded on December 20,2020 at 13:52:46 UTC from IEEE Xplore. Restrictions apply.

Finally, we apply a softmax on the last fully connected layer, which contains 43 classes.

Our proposed model uses 2 fully connected layers and has just around 1.36 million parameters to train when VGG net [17] uses 3 fully connected layers and has more than 138 million parameters. In comparison with VGG net [17], our model has fewer parameters, which allows to converge faster

TABLE I: The details of the proposed model.

Layer	Type	Kernel size	stride	Output size
0	Input			32 x 32 x 3
1	Convolutional	3 x 3	1	32 x 32 x 32
2	Convolutional	3 x 3	1	32 x 32 x 32
3	Max-pooling	2 × 2	2	16 x 16 x 32
5	Convolutional	3 x 3	1	16 x 16 x 64
6	Convolutional	3 x 3	1	16 x 16 x 64
7	Max-pooling	2 × 2	2	8 x 8 x 64
8	Convolutional	3 x 3	1	8 x 8 x 128
9	Convolutional	3 x 3	1	8 x 8 x 128
10	Max-pooling	2 × 2	2	4 x 4 x 128
11	Flatten			2048
12	Fully connected			512
13	Fully connected (softmax)			43

during training phase on a small dataset. The details of the proposed model are presented in Table. I.

C. Data augmentation and image pre-processing

In real world, traffic sign encounter several issues such as imbalance,weather conditions, lighting change, occlusions, etc. To improve the robustness of our model, we perform some operations.

1) Data augmentation: In order to improve the robustness of the model in case of partial occlusion, transformation of images and imbalance, we perform some augmentation techniques.

The data augmentation techniques consists of :

- Flipping : Some images can be flipped if they have an axis of symmetry such as horizontal or vertical line.
- Geometric transformation : After flipping, we perturb the images by some transformations to improve the robustness of model in case of transformed images. The images are rotated by a random degree in (-15, 15), scaled by a factor of 0.8 and 1.2 and translated by a random number of pixels in (-5, 5). After these transformations, the model is more robust to small transformations and occlusion. The training samples are increased.

2) Image pre-processing: In real world, traffic signs vary a lot in illumination, some images are almost dark and others are really bright. To render the proposed model more robustness to noise produced by illumination or light change, images must be enhanced. Firstly, all images are resized to 32 x 32. Then, some pre-processing techniques are performed. The dataset pre-processing consisted of :

- Contrast limited adaptive histogram equalization (CLAHE):Due to the illumination change from images, we apply CLAHE algorithm on the images to improve the local contrast [18]. Instead of standard histogram equalization, the images is divided by regions called tiles, then histogram is applied on each tile, CLAHE performs a good distribution of intensities over the images. CLAHE enhances low contrast images [18].
- Normalization: After the contrast enhancement, normalization is performed on images. Normalization is essentially use on images with poor contrast to help faster convergence. Normalization converts each pixel in [0,255] to float value in [-0.5,0.5] to prevent scale invariance. However, grayscaling is not applied on images because, color contains most important information for the training stage.

IV. Experimental Results

In this section we train and evaluate the proposed model using GTSRB dataset [10]. Finally a comparison with the state-of-art methods is presented.

A. The GTSRB dataset

The proposed CNN is trained and evaluated on the GTSRB dataset [10]. GTSRB dataset is a multi-class,image classification challenge held at the International Joint Conference on Neural Networks(IJCNN) in 2011. It contains 51839 images of 43 different classes. It is divided into a training set of 34799 images(67 %), a validation set of 4410 images (8.5%), and a testing set of 12630 images(24.5%). The dimension of each image vary between 15 x 15 to 250 x 250 pixels. All images are captured in real world in Germany. In this study, all images are resized to 32 × 32 for processing.Since the dataset is captured in real world, as shown in Fig. 2, it presents some issues such as weather conditions, illumination changes, partial occlusions, blurry, distortion, rotations, [10]. Then as shown in Fig. 3, the training dataset is very imbalanced. To solve these issues, data augmentation is performed to increase the dataset and also enhance the model robustness

Fig. 2: Some samples from the GSTRB dataset.

Authorized licensed use limited to: University of Gothenburg. Downloaded on December 20,2020 at 13:52:46 UTC from IEEE Xplore. Restrictions apply.

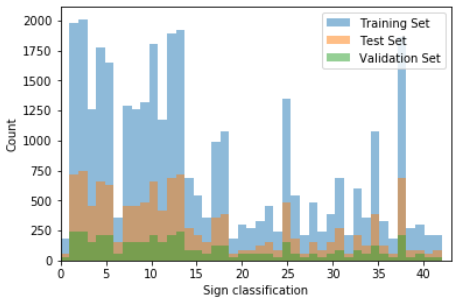


Fig. 3: Distribution of GTSRB dataset.

in case of occlusion and transformation, by flipping, rotation, scaling, translation and then, the dataset is preprocessed using image enhancement (CLAHE) and normalization to remove the noise and enhance images. The details of these methods are presented in section III-C. Fig. 4 shows some examples generated after augmentation when Fig. 5 shows results after the preprocessing phase.

B. Setup

The proposed model is trained on GTSRB dataset. All experiments are performed in jupyter notebook from google colaboratory with a nvidia Tesla K80 12 GB GPU, Intel(R) Xeon (R) CPU @2.30 Ghz and a RAM of 12 GB. The entire program is implemented using the tensorflow framework.

C. Training on dataset

In order to set our hyper-parameters, we perform several experimentations. Back-propagation [24] is used for training. Firstly we apply pre-processing such as normalization, CLAHE [18] on dataset and train the model using 34799

images from original dataset without augmentation during 20 epochs with a batch size of 128. Dropout of 50 % is also added after the fully connected layer. Learning rate is set to 0.0001. The accuracy of model is under 95 % and we observe it can be increased. In order to prevent overfitting, data augmentation is performed using the techniques presented in section III-C1, moreover a batch normalization is added. Finally, the training is performed using the following settings. To minimize the model error we use Adam optimization algorithm due his faster convergence, L2 regularization is also used to reduce the large weights of model. The learning rate is set to 0.00025. The batch normalization decay set to 0.9. A dropout of 0.5 is used in the fully connected layer to prevent the overfitting. To compute the loss rate, we use cross entropy. We also use a batch size of 128. The model is trained in 50 epochs during 32 minutes on our GPU. According to Fig. 6, the accuracy is increased when the loss is decreased, the model is able to learn faster.

D. Performance evaluation

In this section, we use some metrics to evaluate the performance of the proposed model. Confusion matrix is used to determine the classification accuracy of each 43 classes.

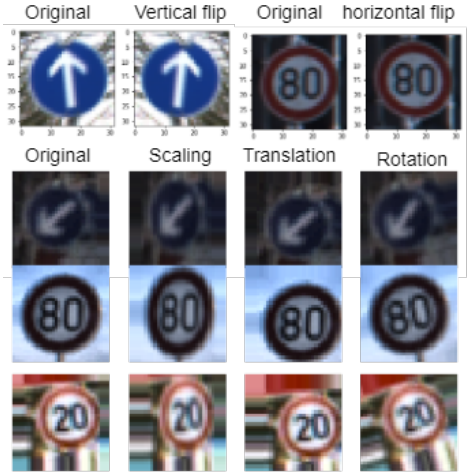


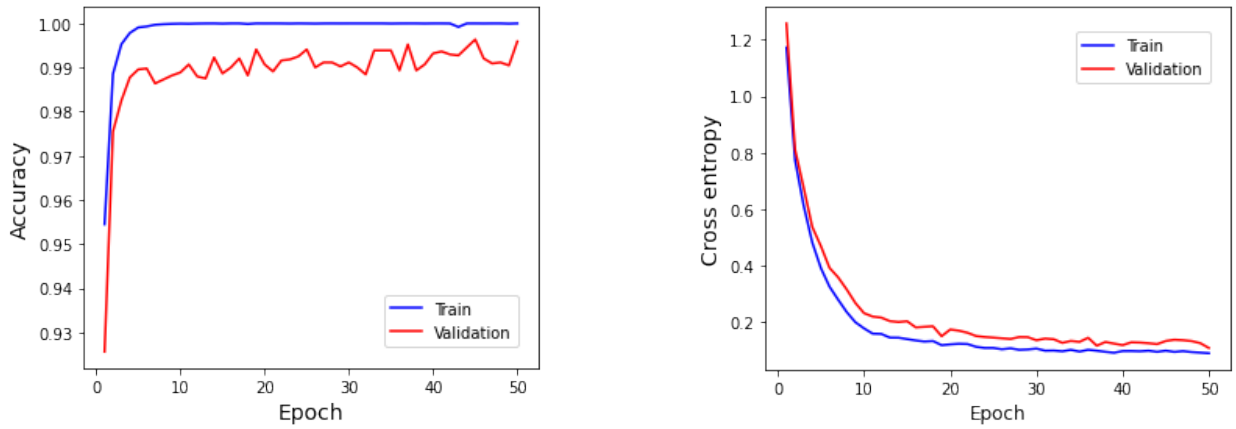
Fig. 4: Samples generated by flipping, scaling, translation and rotation.

Fig. 7 shows the confusion matrix, each row represents the actual label and each column represents the predicted label. Confusion matrix has known values : True positive (TP), True negative (TN), False positive (FP) and false negative (FN) .TP is found when the model correctly classifies the true traffic sign, TN is found when the model correctly classifies the negative traffic sign, FP is found when the model misclassifies the positive traffic sign and FN found when the model misclassified the negative traffic sign [25]. In the confusion matrix TP and TN are the values in diagonal. From the confusion matrix we can determinate precision, recall, and Overall accuracy metrics to evaluate the model performance . Precision is the ratio of true positive class to all positive class of traffic sign, it is computed by (1) as follow :

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

Fig. 5: Dataset preprocessing: CLAHE [18] and normalization reduce the intensity of images in first, third and fourth row. CLAHE enhances the contrast of the image in second row.

Authorized licensed use limited to: University of Gothenburg. Downloaded on December 20,2020 at 13:52:46 UTC from IEEE Xplore. Restrictions apply.



(a) Accuracy of the model

(b) Loss of the model

Fig. 6: Comparison between the accuracy and loss of the model during training. Curve validation accuracy remains under the train accuracy during the training phase. The accuracy of model increases when the epoch increases and the loss decreases.

Recall is a ratio of true positive class to the sum of true positives and false negatives in the classifier, it is computed by (2) as follow :

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

The overall accuracy measure the performance of the model and it is given by (4) as follow :

$$Overall\ Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{3}$$

In the confusion matrix shown in Fig. 7, classification rate of each class is focus along the diagonal.

Using the confusion matrix, precision and recall are computed, and presented in Table. II. Precision, and recall average are 0.99 , it means the model is able to classify each class with high rate . In the next, we represent the experimental results and comparison with the state-of-the-art.

E. Results on dataset and comparision with state-of-arts

Our proposed model is evaluated on the GTSRB dataset presented in section IV-A. After augmented and preprocessed the training images, as shown in Fig. 6, the model achieves a state-of-art validation accuracy of 99.59% and training accuracy of 100 % after 50 epochs during 32 minutes. Each epoch takes between 36 and 44 seconds. To compute the loss we are used a cross entropy, we reach an average loss of 0.0885 on training set and 0.1070 on valid set. Confusion matrix in Fig. 7 shows high classification accuracy for each class along his diagonal, Table. II shows high average recall and precision rate around 99 %, the model achieves a state-of-art overall accuracy of 99.33 % on test data which is better than the human performance (98.84 %) [10].

The comparison between our proposed model and the state of art is shown in Table. III. The methods [2], [7] use traditional algorithms. In [2], the classification is performed with HOG and SVM. The model reaches a classification accuracy

TABLE II: Classification performance of the model using confusion matrix. Supp is number of traffic signs in test set

Class	Precision	Recall	Sup	Class	Precision	Recall	Support
0	1.00	1.00	60	23	0.96	1.00	150
1	1.00	1.00	720	24	0.98	0.99	90
2	1.00	1.00	750	25	1.00	1.00	480
3	1.00	0.98	450	26	0.95	1.00	180
4	1.00	1.00	660	27	0.89	0.98	60
5	0.98	1.00	630	28	0.99	0.99	150
6	1.00	0.98	150	29	0.99	1.00	90
7	1.00	1.00	450	30	0.99	0.95	150
8	1.00	1.00	450	31	1.00	1.00	270
9	1.00	1.00	480	32	0.97	1.00	60
10	1.00	1.00	660	33	1.00	1.00	210
11	1.00	0.99	420	34	1.00	1.00	120
12	1.00	0.97	690	35	1.00	1.00	390
13	1.00	1.00	720	36	1.00	1.00	120
14	1.00	1.00	270	37	1.00	1.00	60
15	0.93	1.00	210	38	1.00	1.00	690
16	1.00	1.00	150	39	0.99	1.00	90
17	1.00	1.00	360	40	0.99	0.98	90
18	1.00	0.94	390	41	1.00	1.00	60
19	1.00	1.00	60	42	0.98	1.00	90
20	0.99	1.00	90				
21	0.90	1.00	90				
22	1.00	1.00	120	Average	0.99	0.99	12630

of 99.52% on the GTSRB [10] dataset, after perspective adjustment on the image, and outperforms several CNN based methods. In [7], CLAHE is used to improve local contrast images [18], LBP improves the classification which take 211 ms. The methods [12]–[14] use a CNN. The methods

referenced in [13], [14] reach a better accuracy than ours. In [13], authors use Image adjustment to enhance images. In [14], data are augmented and hinge loss stochastic gradient descent (HLSGD) reduce the error rate. Images preprocessing such as histogram equalized image,etc are applied to enhance contrast images. In comparison, in [12] the authors apply multi scale CNN on gray images and increase the accuracy

Authorized licensed use limited to: University of Gothenburg. Downloaded on December 20,2020 at 13:52:46 UTC from IEEE Xplore. Restrictions apply.

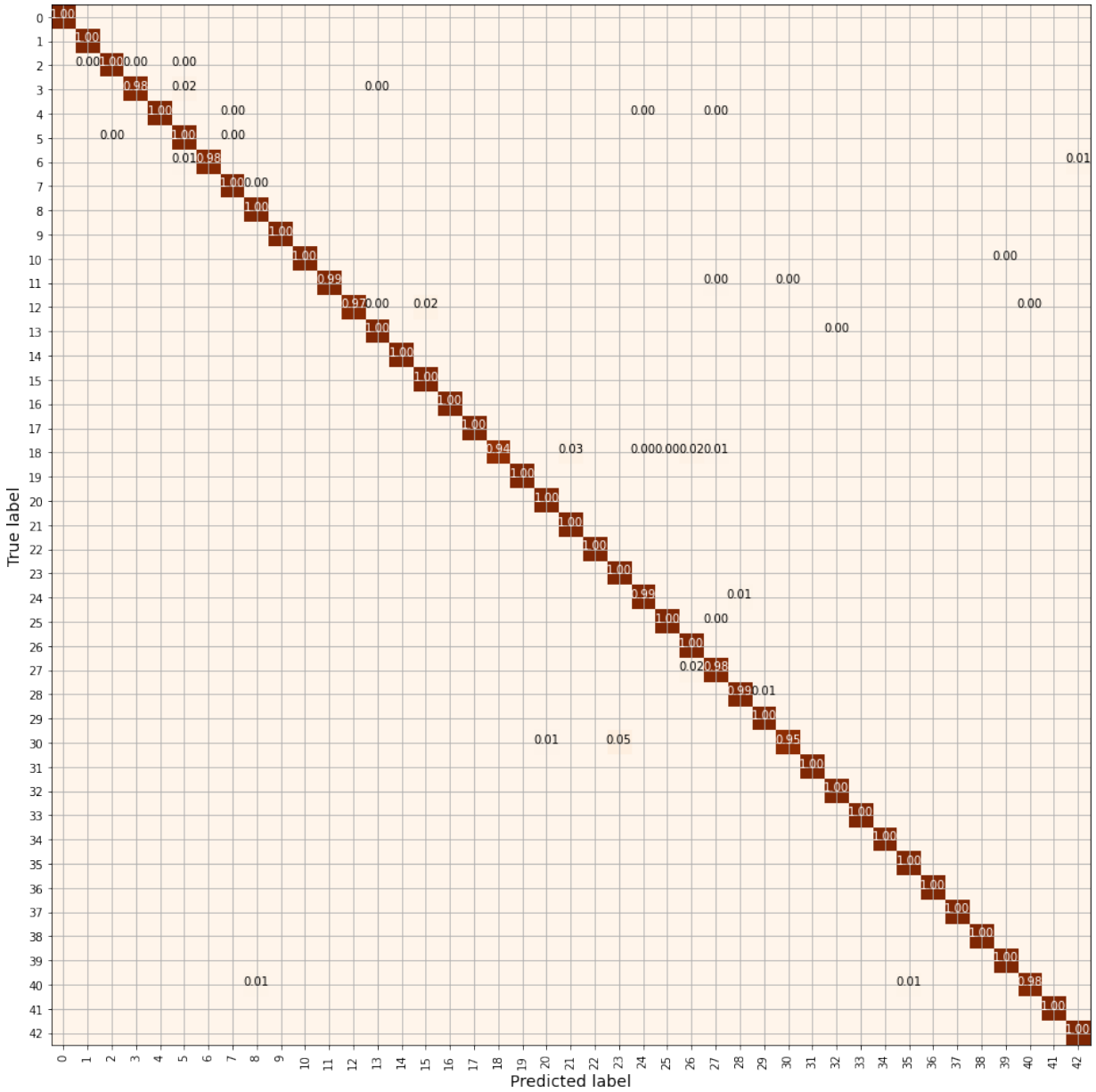


Fig. 7: Normalized confusion Matrix of model. True labels are on y-axis and predicted on the x-axis. Blank cases have zero as value. The model is able to classify each class according the values in diagonal.

to 99.17 %. In [13], the model uses around 90 Million parameters and takes 37 hours to train with 25 columns on four GPUs. In comparison, our model has 1.36 Million parameters and takes 32 minutes to train on one GPU.

F. Error analysis

The model achieves a test accuracy of 99.33 %, it means there are 85 misclassified signs by the model. According to the Table. II, the class 30 "Beware of ice/snow" and the class 18 "General caution" are the most misclassified due their low recall rate. Fig. 7 shows than, class 30 is often confused with the class 23 "Slippery road" due their similarity in shape. According to Fig. 8, traffics signs are misclassified due their low quality, illumination, occlusion, blur, similarities in shape. CLAHE algorithm has enhanced the images [18] and data augmentation has really improved the model which is able to classify each class, indeed some speed limits signs which

are very few and blur in the original dataset and not recognizable by human eye are correctly classified by the model.

V. Conclusion

In this paper, a traffic sign classification model based on VGG [17] with batch normalization is proposed. Augmentation and image enhancement are performed in order to enhance the image contrast and increase the classification

Authorized licensed use limited to: University of Gothenburg. Downloaded on December 20,2020 at 13:52:46 UTC from IEEE Xplore. Restrictions apply.

TABLE III: Comparison of accuracy of proposed model and the state-of-art methods on GSTRB dataset

Method	Parameter	Accuracy	Configuration
HLSGD [14]	1.16 M	99.65 %	CPU:I7-3960X GPU:2xTesla C2075
Hierarchical SVM [2]	N/A	99.52 %	CPU:I3
MCDNN [13]	90 M	99.46 %	CPU:I7-950 GPU:4xGTX 580
Multi-Scale CNNs [12]	1.4 M	99.17 %	N/A
Random forests [7]	N/A	97.16 %	N/A
Ours	1.36 M	99.33 %	CPU:Intel(R)Xeon(R) GPU:Tesla K80

rate. The model performance is evaluated using confusion matrix, precision, recall, on the GTSRB [10] dataset. The model achieves a state-of-art accuracy of 99.33 % on test set which surpasses the best human performance of 98.84%. Experiments show that our model is efficient to classify traffic sign in real world. Error analysis shows that, the model is able to classify some partially distorted and occluded traffic signs and some images which are not recognizable by human eye. In order to render the proposed model more efficient, and robust to occluded and transformed images, some pre-processing techniques can be performed to increase the classification rate in the future.

ACKNOWLEDGMENT

This work was supported by the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement No 777720.

REFERENCES

[1] M. Lu, K. Wevers, and R. Van Der Heijden, "Technical feasibility of advanced driver assistance systems (adas) for road traffic safety," *Transportation Planning and Technology*, vol. 28, no. 3, pp. 167–187, 2005.

[2] G. Wang, G. Ren, Z. Wu, Y. Zhao, and L. Jiang, "A hierarchical method for traffic sign classification with support vector machines," in *The 2013 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2013, pp. 1–6.

[3] F. Zaklouta and B. Stanciulescu, "Real-time traffic sign recognition in three stages," *Robotics and autonomous systems*, vol. 62, no. 1, pp. 16–24, 2014.

[4] P. H. Kassani and A. B. J. Teoh, "A new sparse model for traffic sign classification using soft histogram of oriented gradients," *Applied Soft Computing*, vol. 52, pp. 231–246, 2017.

[5] Y.-L. Hou, X. Hao, and H. Chen, "A cognitively motivated method for classification of occluded traffic signs," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 2, pp. 255–262, 2016.

[6] A. Ellahyani, M. El Ansari, and I. El Jaafari, "Traffic sign detection and recognition based on random forests," *Applied Soft Computing*, vol. 46, pp. 805–815, 2016.

[7] X. Kuang, W. Fu, and L. Yang, "Real-time detection and recognition of road traffic signs using mser and random forests," *International Journal of Online and Biomedical Engineering (iJOE)*, vol. 14, no. 03, pp. 34–51, 2018.

[8] Y. Chen, J. Xiong, W. Xu, and J. Zuo, "A novel online incremental and decremental learning algorithm based on variable support vector machine," *Cluster Computing*, vol. 22, no. 3, pp. 7435–7445, 2019.

[9] Y. Chen, W. Xu, J. Zuo, and K. Yang, "The fire recognition algorithm using dynamic feature fusion and iv-svm classifier," *Cluster Computing*, vol. 22, no. 3, pp. 7665–7675, 2019.

[10] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The german traffic sign recognition benchmark: a multi-class classification competition," in *The 2011 international joint conference on neural networks*. IEEE, 2011, pp. 1453–1460.

[11] D. Cireşan, U. Meier, J. Masci, and J. Schmidhuber, "A committee of neural networks for traffic sign classification," in *The 2011 international joint conference on neural networks*. IEEE, 2011, pp. 1918–1921.

[12] P. Sermanet and Y. LeCun, "Traffic sign recognition with multi-scale convolutional networks," in *The 2011 International Joint Conference on Neural Networks*. IEEE, 2011, pp. 2809–2813.

[13] D. CireşAn, U. Meier, J. Masci, and J. Schmidhuber, "Multi-column deep neural network for traffic sign classification," *Neural networks*, vol. 32, pp. 333–338, 2012.

[14] J. Jin, K. Fu, and C. Zhang, "Traffic sign recognition with hinge loss trained convolutional neural networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 1991–2000, 2014.

[15] J. Zhang, W. Wang, C. Lu, J. Wang, and A. K. Sangaiah, "Lightweight deep network for traffic sign classification," *Annals of Telecommunications*, pp. 1–11, 2019.

[16] S. K. Satti, K. S. Devi, P. Dhar, and P. Srinivasan, "Enhancing and classifying traffic signs using computer vision and deep convolutional neural network," in *International Conference on Machine Learning, Image Processing, Network Security and Data Sciences*. Springer, 2020, pp. 243–253.

[17] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[18] G. Yadav, S. Maheshwari, and A. Agarwal, "Contrast limited adaptive histogram equalization based enhancement for real time video system," in *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, 2014, pp. 2392–2397.

[19] Y.-L. Boureau, J. Ponce, and Y. LeCun, "A theoretical analysis of feature pooling in visual recognition," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 111–118.

[20] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[22] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[23] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *International Conference on Machine Learning*, 2015.

[24] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel, "Handwritten digit recognition with a back-propagation network," in *Advances in neural information processing systems*, 1990, pp. 396–404.

[25] J. Zhang, C. Lu, X. Li, H.-J. Kim, and J. Wang, "A full convolutional network based on densenet for remote sensing scene classification," *Math. Biosci. Eng*, vol. 16, no. 5, pp. 3345–3367, 2019.

Fig. 8: Some misclassified images produced by the model in GSTRB dataset. The images are dark, blur, occluded, illuminated. The number represents the class of each image.

Authorized licensed use limited to: University of Gothenburg. Downloaded on December 20,2020 at 13:52:46 UTC from IEEE Xplore. Restrictions apply.