Multimedia Tools and Applications (2022) 81:8429–8449
https://doi.org/10.1007/s11042-022-12177-8

Pakistani traffic-sign recognition using transfer learning

Zain Nadeem[1] · Zainullah Khan[1] · Usama Mir[2] · Umer Iftikhar Mir[1] · Shahnawaz Khan[1,3] · Hamza Nadeem[1] · Junaid Sultan[1]

Received: 28 June 2020 / Revised: 20 November 2021 / Accepted: 7 January 2022 /
Published online: 2 February 2022

Abstract

Initially, the traffic-sign recognition was done using the conventional image processing techniques which are sluggish and can cause fatal delays in real-world implementations. Majority of the state-of-the-art detectors are based on a Convolutional Neural Network (CNN) which is a de-facto leader in computer vision research over the past decade. Easy availability of datasets is the main reason for the interest of researchers in CNNs. These datasets are needed to be organized and maintained as the CNN requires colossal amounts of data to work well. Unfortunately, no traffic-sign dataset exists in Pakistan to enable any detection based on the CNN. Therefore, in our work, we have collected and annotated a dataset to help foray into this research area. We propose an approach revolving around the deep learning where a model is pre-trained on the German traffic-sign dataset. This model is then fine-tuned using the Pakistani dataset (of 359 different images) collected across Pakistan. Preprocessing and regularization are used to improve the overall performance of the model. Through results, we show that our fine-tuned model reaches to a training accuracy of nearly 55% outperforming the other related techniques. The results are encouraging as we have achieved a high accuracy keeping in mind the small size of the available Pakistani dataset.

Keywords Pakistani traffic-sign datasets · Machine learning · Deep learning · Convolutional neural networks

1 Introduction

In the past decade, leading up to 2019–20, thousands of deaths were recorded in Pakistan under the banner of "Road Accidents". A recent survey by the World Health Rankings conducted in 2019 ranks traffic collisions as one of the leading causes of deaths in Pakistan [45]. Approximately more than 37,000 people suffered from the havoc of road-side accidents

✉ Usama Mir
dr.usama.mir@gmail.com

Extended author information available on the last page of the article

across the country. Moving forward, the fatalities caused by accidents in the year 2020 and 2021, as enunciated by the Pakistan Bureau of Statistics, were recorded at 27,504 and 21,167, respectively [32]. Although, the death rate has decreased as the years have progressed, it does not diminish the significance of roadside safety and the dire need of an advanced system to curb this lethal concern.

The government of Pakistan, in collaboration with the motorway police, is making tremendous efforts to investigate this matter and deduce the major reasons behind the escalating road accidents. According to their report, the traffic authorities lack the proper mechanism to monitor the speed limits [9]. Moreover, the drivers often ignore the nature of the terrain they are driving on, which causes them to take no notice on the sharp curves ahead. Despite that, the cause that resulted in 67% of the total deaths is due to the ignoring of road safety rules as well as the improper driving.

This brings the role of traffic sign detection and recognition into greater focus, which has increased the interest of the researchers in the last few years. This modernized system holds quite a considerable significance in the on-going situation as it provides its solutions to a wide range of problems affecting traffic related injuries. In addition, a system with the capability of this sort may help us in curbing the on-going issue by providing:

1. Surveillance of the highway, traffic monitoring as well as maintaining the condition and presence of the road signs. Until now, the road signs had to be watched by human patrolling officers resulting in a high probability of human errors.
2. An ample sign inventory comprising of proper signs especially for towns and cities as their environment is more difficult than the highways.
3. Standard rules that are applicable for all drivers helping them to ensure the compliance of the traffic rules, even if one has a lapse in concentration.
4. Monitory alarms in case of unforeseen circumstances providing the user with a support system while driving somewhere that is alien to him/her.
5. Safety to the pedestrians and property as any unfortunate incident could happen, even to the best of us causing serious damage to the public assets and the lives of others.

Traffic sign recognition and self-driving highly automated vehicles are considered as some of the most researched areas nowadays [7, 13]. These two terms are buzzwords around the globe due to their tremendous significance and application for the future of automation in the world we live in. A great amount of emphasis has been given by the manufacturers in developing the safety systems that allow the reduction of the number of accidents due to the driver's negligence and various other aspects.

In this context, the traffic sign detection has played a vital role as it has given the world a vision of how the future of automation will look like. At first, it was only utilized as a speed limit assistance system, however, with the passage of time, it has improved tremendously in terms of accuracy, quality, and number of functions. These state-of-the-art features assist the drivers by allowing and prohibiting certain maneuvers, such as the speed limits, passing indicators, or approaching an obstacle [48]. The main purpose of this technology is to help improve the safety of the driver on the road and it would not be wrong to say that these systems are precursors of what might eventually become fully automated driving.

Considering the significance of automated vehicles and observing their applications in numerous countries, we have devised our research to develop a similar system for our country (Pakistan). Until today, the Pakistani traffic system does not even have the luxury of traffic sign recognition; therefore, the purpose of this research is to equip the country and the interested researchers with a framework on which they can build on. This could allow the researchers to work towards developing a better technique or technology, bringing the idea of self-driving cars to Pakistan, making them ready for the future.

The main contributions of this research are as follows:

1. Providing researchers with a labeled dataset comprising of traffic-sign images unique to Pakistan.
2. The images in the dataset are cropped, labelled, and resized to make them usable for the implementation purposes.
3. Training a model using the modified images to recognize Pakistani traffic-signs and be deployed in real-world applications.
4. Writing a review of traffic-sign datasets and their respective highest achieved accuracies for comparison with the future studies.
5. To compare our results with several machine and deep learning algorithms.
6. To create a benchmark for future research in Pakistani traffic-sign dataset by achieving higher accuracy than the compared models.

This paper is an extension of our previous work [29] where our main contribution was to provide a labeled dataset of the traffic-sign images unique to Pakistan and we also trained a CNN (Convolutional Neural Network) model to recognize these images. Our task was exceedingly difficult because the number of images collected was very small and the CNNs require a higher number of images to generalize. To overcome the difficulties, we used the transfer learning by fine-tuning a pre-trained model and achieved admirable results. In this article, we improve upon the CNN architecture that was used in our previous work and hence we also improve the classification accuracy on the same dataset.

The rest of the paper is organized as follows. Section 2 present the state-of-the-art and related work of this research. Section 3 provides an overview of Neural Networks in general which leads to Convolutional Neural Networks (CNNs). Our Methodology is discussed in Section 4. Results and Discussions are presented in Section 5. Finally, Section 6 concludes our paper.

2 Related work

Traffic-sign detection has been researched a lot by various renowned authors and scientists across the globe. According to [44], the first well-known research on traffic sign detection dates to 1984 which was conducted in Japan. Since then, multiple approaches have been developed based on the visual aspects of an image such as the shape and the color [1]. Image processing has also been used for traffic sign identification in [31] termed as the conventional image processing technique. Such image processing techniques are not really beneficial for identifying the traffic-signs since a delay of seconds can prove fatal. To make the detection process more robust, a better detection and recognition method is needed, one such method is the famous CNN [11]. CNN is a class of neural networks which is mainly used to analyze visual data such as images. CNN has become the mainstream for any image recognition and detection task [15, 43], due to its high accuracy during inference [14]. Some of the widespread applications of CNNs are facial recognition [36], disease detection [33, 47], and self-driving cars [42].

The use of CNN in self-driving cars includes (but is not limited to) lane detection, steering control, pedestrian detection, and traffic sign detection [18, 35]. Traffic sign detection and recognition through CNN have produced very promising results which are certainly better than the conventional methods. For example, the accuracy in detecting traffic signs can reach up to 95%. Such a high accuracy is only possible because of the large amount of data. In other words, the data accuracy increases as more and more data goes through the CNN. However, since the CNN needs a lot of data to be trained, therefore the chosen datasets should be reliable and well-maintained.

In relation to above, several datasets are available for researchers such as the famous German Traffic-sign Dataset (GTSRB) [39], the Belgian traffic-sign dataset (KULD) [24], and Laboratory for Intelligent and Safe Automobiles (LISA) Traffic-Sign Dataset formed by American Traffic-signs [26]. All these datasets have varying numbers of classes however, they all consist of a huge amount of data. The German dataset, for example, has almost 39,000 images distributed over more than 40 classes. These datasets are well maintained since they are regularly used for machine learning competitions. Unfortunately, no such dataset exists in Pakistan which significantly reduces the amount of research being done in the area of self-driving cars [48] in Pakistan. Despite some recent research efforts such as [20, 21] for the traffic-sign recognition the area still needs further attention especially with the inclusion of CNNs.

In addition to above, multiple traffic sign datasets can be used in machine learning competitions, benchmarking tests, and for research purposes. Some of these datasets along with their number of classes, number of images, image sizes, and highest achieve accuracies are shown in Table 1. The table includes the Sweden Traffic-Sign Dataset (STSD), the United Kingdom Online Dataset (UKOD), and the Russian Traffic-Sign Dataset (RTSD) along with the datasets we mentioned in the above discussion. Clearly. the greater the number of images in a dataset, the better the highest achieved accuracy. Another factor that influences the effectiveness of a dataset is the size of the images; if an image size is bigger, more features can be extracted from an image and later learned by the CNN.

The only dataset comparable with the Pakistani traffic-sign dataset (PTSD) in terms of the number of images is the Netherlands' Robot Unstructured Ground Driving (RUGD) dataset [5]. This dataset, although contains only 48 images, can achieve an accuracy of 54.64%, which is very high considering the use of only 48 images. One of the reasons for such a higher accuracy is the high resolution of used images resulting in more images being learned by the algorithm. The Pakistani dataset has a greater number of images than the RUGD dataset, but the size of those images is a lot

Table 1 Dataset size, classes and training, and testing accuracies

Dataset	Country	Classes	Images	Image size (px)	Highest achieved accuracy
GTSRB (2012–13) [38]	Germany	43	39,209 (train) 12,630 (test)	15×15 to 250×250	99.3% [40]
KULD (2009) [24]	Belgium	100+	13,444	1628×1236	99.72% [19]
STSD (2011) [12]	Sweden	7	3488	1280×960	82% [22]
RUGD (2003) [5]	Netherlands	3	48	360×270	54.64% [46]
Stereopolis (2010) [3]	France	10	251	1920×1080	N/A
LISAD (2012) [27]	US	49	7855	640×480 to 1024×52	97.31% [16]
UKOD (2012) [4]	UK	100+	1200 (synthetic)	648×480	94.3% [4]
RTSD (2013) [28]	Russia	140	80,000+ (synthetic)	1280×720	92.90% [34]
PTSD (2018) [29]	Pakistan	35	359	64×64	54.8% (our results)

smaller than the majority of the datasets. This small size does not allow enough image features to be learned but even with such limitation, the dataset still performs well and produces a respectable accuracy of 54.8% which will later be discussed in our forthcoming sections.

To conclude this section, we can clearly say that the research on traffic sign detection is Pakistan is very limited, therefore, this paper is an attempt to set a benchmark for testing the Pakistani traffic sign datasets. Furthermore, we also aim to carry out a comparative study between different models to single out a model that performs best on our dataset which will help the future researchers if they wish to use our dataset for analysis and testing.

3 Convolutional neural networks: an overview

Generally, a CNN is a class of Neural Networks that is used in analyzing visual data, such as images. CNNs are most commonly used in classification tasks where the model detects and differentiates between different types of objects present in an image. A general representation of a CNN is shown in Fig. 1.

CNNs take images as inputs and perform an operation called convolution on these images. Convolution extracts features from the images such as lines, edges, and corners. These features are extracted from multiple images of a single object giving the model a general understanding of what the objects look like. In other words, the model is being trained to recognize an object. Once trained, the model will be able to recognize the same object in images for which it has not yet been trained for.

As specified in the introduction section, the conventional methods used to detect traffic signs are very inaccurate, hence a better approach is needed. CNNs can perform very well after being trained on a set of images providing very high inference accuracy. All these factors make a CNN an optimal candidate for traffic sign detection.

The CNN architecture greatly affects the accuracy of the model. Therefore, numerous architectures have been researched over the years. One of the simplest (and yet the most effective) architectures is called the LeNet which was introduced in 1998 [14]. LeNet is very simple to implement because the number of layers in the network is very small, hence, the used parameters within the network also remain a few [30]. The simple design of LeNet makes it possible to train the model using just a CPU without using any hardware accelerators like the GPUs. The architecture of LeNet is discussed further is section 4.2.

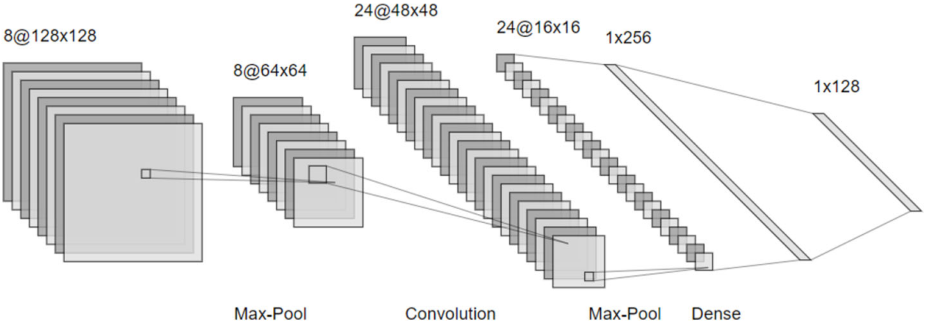

Fig. 1 General architecture of a CNN as presented in [26]. The model consists of several types of layers including Convolution Layers, Max-Pooling Layers, and Fully connected Layers also called Dense layers. The final layer has neurons equal to the number of classes

CNNs require a large amount of data for training, however, sometimes the dataset available is small and thus, cannot give very accurate results. Therefore, to achieve a higher accuracy, the model is first trained on a large dataset with the same classes and then further trained on the small dataset. This method of training is referred to as the transfer learning [41], which is a sub-category of supervised learning in the broader field of Deep Learning. Transfer learning is a method of training pre-trained CNNs on small datasets that are not sufficiently large enough to be used to train a CNN from scratch. The pre-trained model parameters have the ability to extract features from the images, however, in order to adapt to the new data, the model must be fine-tuned on it. Transfer learning has been used in traffic sign recognition on many occasions, such as in [17] authors used transfer learning to fine tune a model on the Belgian traffic sign dataset. Moreover, in [2] the authors used transfer learning to train a model to segment road and to detect traffic signs. Section 4.3 of this paper discusses the use of transfer learning for traffic sign recognition on the Pakistani dataset.

4 Methodology

Our methodology is divided in following three main phases:

1. Resizing images
2. Pre-training and optimization of a CNN model on the German traffic-sign dataset
3. Fine-tuning the pre-trained model on Pakistani dataset, using transfer learning

4.1 Phase I

The images used in the original research article [29] are all 32 × 32 which reduces the training time significantly, however, the size reduction also causes the images to lose some of the features that might be crucial in detecting the traffic signs. In order to preserve some of the features, we try to resize the images to be 64 × 64 in both German and Pakistani datasets. The methodology for the rest of the paper remains the same except for the size of the convolution layers, which has been changed to match the size of the input images.

The images, in the German dataset, are all iterated through and resized to be 64 × 64. These datasets are then saved in a pickle file. The images in the Pakistani dataset have also been resized and saved in a new directory. The images are labeled in the same way as before. Image collection, resizing, and training are depicted in Fig. 2.

4.2 Phase II

The second phase is completed using the Keras machine learning framework with TensorFlow at the backend.

4.2.1 Pre-processing

Pre-processing makes it easier for a CNN model to converge. This experiment uses normalization to reduce training times and improve convergence, especially while training on the German Dataset. Generally, pixel values in an image vary greatly

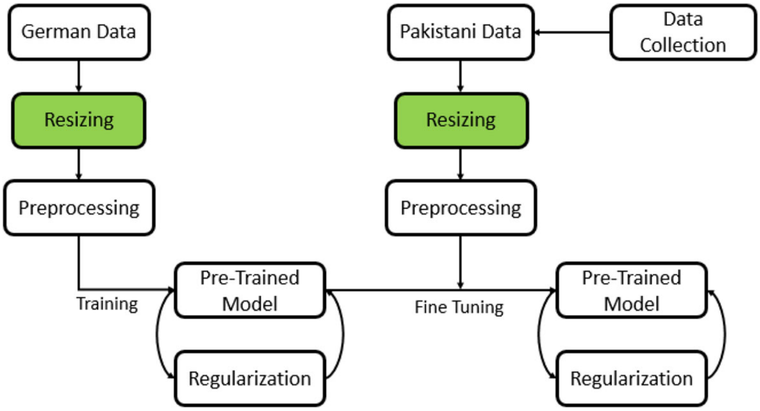

Fig. 2 Flow diagram of our methodology. It can be noted that a new block is added which resizes the images to 64 × 64 size to preserve some of the features in the images

over a big range, however, the normalization process reduces this range and puts the data on a smaller range with the same scale. Additionally, normalization balances the pixel intensities which helps in the optimization process by stopping the exploding and vanishing gradient problems. Our pre-processing also includes one-hot encoding, which is a representation of categorical data in a more expressive way w.r.t a machine-learning model. This is actually one of the requirements of the Keras function 'model.fit()' and the 'categorical cross-entropy' loss function [8].

4.2.2 Model architecture

The CNN architecture used in this experiment is the LeNet Architecture [14] which is most widely used in existing research and easy to comprehend. The LeNet architecture generally works with images and requires very little pre-processing. Figure 3 shows the main components of a LeNet CNN including the convolution layers which perform the convolution operation on all the images, dense layers which are basically fully connected neuron layers, max-pooling layers used for reducing an image's dimensions, dropout layers used for regularization, and finally a fully-connected layer needed for classification. Our CNN has 2

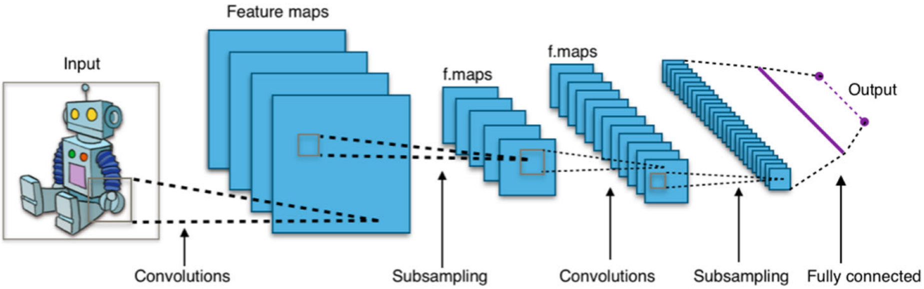

Fig. 3 LeNet-5 Architecture proposed in [31]. This architecture, as discussed above, has a certain configuration of how the layers are stacked. It is one of the most used architectures due to its simplicity of implementation and a relatively less number of parameters to handle

convolutional layers, each layer is followed by a max-pooling layer. The CNN model is coded using the Keras machine learning framework.

4.2.3 Training

The training is performed using different number of epochs (such as 5, 20, and 50) where the response of each case has been compared with other responses. Later, the CNN is optimized through the Adam optimizer [10] and the loss is calculated using categorical cross-entropy'. As mentioned before, in order to view the progress of the training, the Keras function 'model.fit' is used which returns the metrics such as the loss values for the test and validation datasets while preserving the accuracy of CNN.

4.2.4 Categorical cross-entropy

The categorical cross entropy loss criterion is used to calculate the loss of a model when it is trained on a dataset that consists of multiple classes. This entropy determines the disparity between the prediction and the ground truth label by only calculating how far away the prediction is from the ground truth class, however, it does not consider the wrong predictions made by the model for all the other classes. The formula for the categorical cross entropy loss is:

$$L = -\sum_{i=0}^{N} P_i \log\left(\hat{P}_i\right)$$

Where N denotes the total number of classes in the data, i denotes the *ith* class. P_i represents the ground truth probability for the *ith* class and $\hat{P}_i$ is the *ith* class probability predicted made by the model. When the model is evaluated using the loss function, it only calculates the value of the class which has a value of 1 in the ground truth, hence all the other dataset class losses are not calculated by this method.

4.2.5 Regularization

Since overfitting of a CNN is undesirable because it does not allow generalized solutions to be formed, a technique is needed to avoid overfitting of the network. The overfitting normally occurs when a model memorizes the data instead of learning it and hence the precision increases by decreasing the accuracy resulting in a reduced count of 'false positive' results. To overcome the aforementioned issue, we use a regularization technique called the dropout. The aim of using dropout is to make CNN learn features instead of memorizing them i.e., "learn less to learn better." Dropout reduces the value of certain weights and biases to 0, therefore the learned features take more than a single path to produce the correct output. This makes sure the entire network is generalized. Academic research such as the one in [37] clearly shows the dropout can produce better results as it is used quite frequently in deep learning implementations.

4.3 Phase III

The third phase is to fine-tune the pre-trained model through transfer learning. This phase helps the CNN to learn the features from the Pakistani dataset. As depicted in Table 1, our data collection process is carried out for several months from various cities of Pakistan (including but not limited to Quetta, Islamabad, and Karachi) where we collected details about different sign names with their corresponding number of images. It can be inferred from the table that the number of images is in a small quantity and the classes are also distributed unevenly, just like the German dataset [38], as can be seen from Fig. 4.

Out of a total of 359 images, most had an unwanted background which is manually cropped to only keep the region of interest. The images are then stored into different folders which are named according to the label of the images. The images are then renamed from 0 to 359 in an alphabetic order of their classes using a python script (e.g., the images from a "bridge ahead" are named from 0 to 12, and the images from a "zigzag road" are named from 346 to 359, respectively). Since all cropped images have non-uniform dimensions, we resize each such image to a shape of 64 × 64 to meet the requirements of the CNN. The preprocessed images are then flattened and stored in an array with their respective classes. These images are later passed through our model for training. The preprocessing step is similar to the German dataset.

4.3.1 Fine tuning on the Pakistani dataset

The architecture for fine-tuning is similar to the one used for pre-training. The layer weights in the pre-training stage have been frozen so that they would not be updated in the fine-tuning process—the complete weight matrix is imported from the pre-training stage and all individual weights remain unchanged. However, the configuration of the fully connected layer has been changed to a total of 35 classes, which corresponds to the total number of classes in the collected dataset. This new fully connected layer is the only layer that is trained in the fine-tuning stage to get better results (Table 2).

The model is trained on the German dataset and then tested on the validation images from that dataset. After this, the transfer learning is implemented by freezing the layers of the CNN and adding a new fully connected layer. This new layer helps in making the correct predictions from the network since all the lower-level features (in the form of human unreadable extracted feature maps) have already been learned by the convolutional layers. The fully connected layer

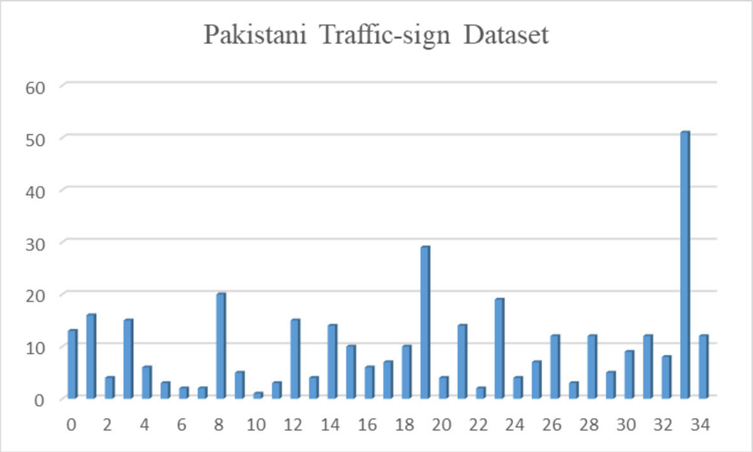

Fig. 4 Image distribution for Pakistani dataset. The figure shows the number of images for each class of traffic signs collected

Table 2 Details of Pakistani traffic-sign dataset

#	Sign name	No. of images
1	Bridge Ahead	13
2	Cross Roads	16
3	Give way	04
4	Left Bend	15
5	No Horns	06
6	No left turn	03
7	No Mobile allowed	02
8	No Overtaking	02
9	No Parking	20
10	No right turn	05
11	No U-Turn	01
12	Parking	03
13	Pedestrians	15
14	Railway Crossing	04
15	Right Bend	14
16	Road Divides	10
17	Roundabout Ahead	06
18	Sharp Right Turn	07
19	Slow	10
20	Speed Breaker Ahead	29
21	Speed Limit (20 kmph)	04
22	Speed Limit (25 kmph)	14
23	Speed Limit (30 kmph)	02
24	Speed Limit (40 kmph)	19
25	Speed Limit (45 kmph)	04
26	Speed Limit (50 kmph)	07
27	Speed Limit (60 kmph)	12
28	Speed Limit (65 kmph)	03
29	Speed Limit (70 kmph)	12
30	Speed Limit (80 kmph)	05
31	Steep Descent	09
32	Stop 1	12
33	Stop 2	08
34	U-Turn	51
35	Zigzag Road Ahead	12

is then fine-tuned to the Pakistani dataset as there are not enough pictures in the Pakistani dataset to train our model from scratch.

5 Results and discussions

5.1 Simulation setup

The training process is performed on an Nvidia 860 M GPU. The architecture of CNN is coded in Keras with a TensorFlow backend. Categorical cross-entropy is chosen as a criterion for the loss function and a batch size of 150 has been used. Batch size is basically the number of images passed on the CNN at a given interval and is changed depending on the processing power available, as a higher number mean more consecutive operations being performed. Meanwhile, an epoch is the number of times a complete dataset is passed through a model to train its weights. The training for CNN has been performed for 20 and 50 epochs and the results are compared. The training time for 20 epochs is approximately 40 min while the training time for 50 epochs is 2 h, respectively.

The LeNet model we are using has 2 convolutional layers and each layer has 32 filters, hence, each layer will produce 32 feature maps. Each convolutional layer is followed by a ReLu activation layer and a max pooling layer. Finally, the output layer is present with a Softmax activation. The models are trained for 50 epochs on a dataset consisting of grayscale images of 64 × 64 resolution. The learning rate is the factor by which the change is introduced in the weights of the network/model and is set at 0.001 while categorical cross-entropy is used as the loss function.

The hyper parameters such as the learning rate, batch size, and number of epochs have been selected after running several models with different values for each of these hyper parameters, whereas, only the best performing have been adopted in our work.

5.2 Pre-training results

Our research does not need any data collection for the pre-training phase since the German traffic sign dataset is an extensive dataset with multiple classes and it is very well maintained. Moreover, this dataset is regularly used in competitions in which maximum test/validation accuracy is the goal [39]. However, for the fine-tuning phase, a large amount of data collection is required which is quite a hectic task. The fine-tuning phase is also a major part of this research along with the novelty in data collection. During data collection, we incurred the problem of variation in traffic sign standards since some of the signs in the Pakistani dataset are not the same as those in the German dataset such as the 'No-horn' sign does not exist in the German dataset but is present in the Pakistani dataset. Likewise, there is no 'Roadwork' sign in the Pakistani dataset, but it does exist in the German counterpart and so forth. The pre-training on the German dataset produces high accuracies despite the small number of epochs as seen in Figs. 5 and 6. The accuracy of 99.3% surpasses the accuracy of an average human which normally falls somewhere between 95% to 98% [39]. Figures 7 and 8 show the training and validation losses as the model converges towards the minimum.

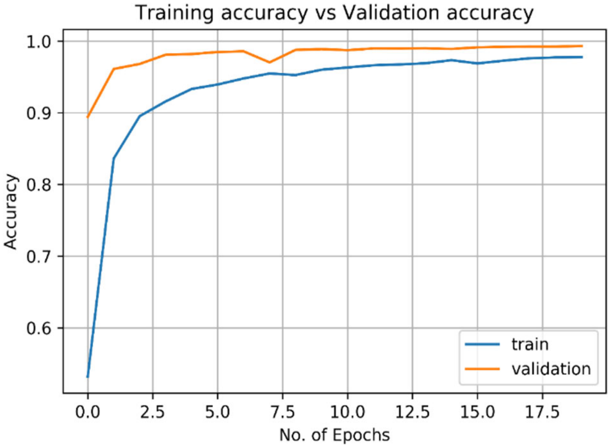

Fig. 5 Training/validation accuracy for pre-training model (20 Epochs)

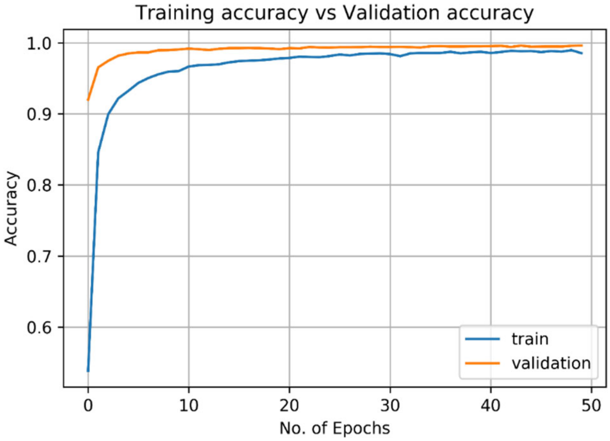

Fig. 6 Training/validation accuracy for pre-training model (50 Epochs)

5.3 Fine-tuning

It is known through experiments and research that CNNs require a huge amount of data to generalize well enough otherwise, the accuracy of the CNN will make it less reliable and hence it will not be suitable to be used for real-world applications [15]. The German dataset had a huge number of images (ca. 39,000 [40]) but even this amount was not enough to generalize the model completely, therefore, the dataset was extended using data augmentation techniques. The Pakistani dataset on its own is too small and it is not enough to train a model from scratch. Therefore, we applied transfer learning to compensate for the lack of data.

Fine-tuning produces results that are, although unsatisfactory, but encouraging since the dataset is only limited to a mere 359 images. It should be noted since the transfer learning-based approach has not ever been used in traffic sign recognition; therefore, it is not possible to compare the accuracy of our model with other models directly. The final accuracy of 54.8% is encouraging considering the small number of images present in the dataset as can be seen from Figs. 9 and 10.

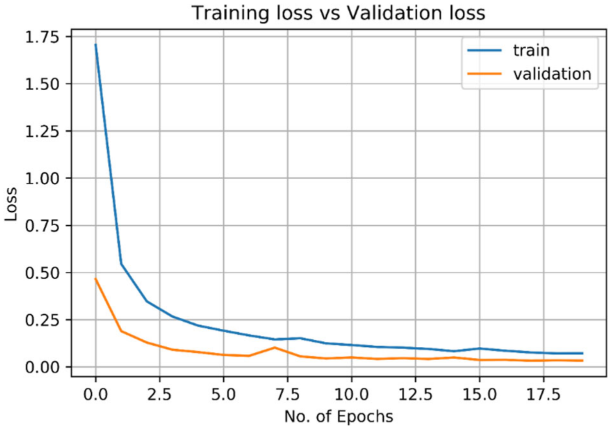

Fig. 7 Training/validation loss for pre-training model (20 Epochs)

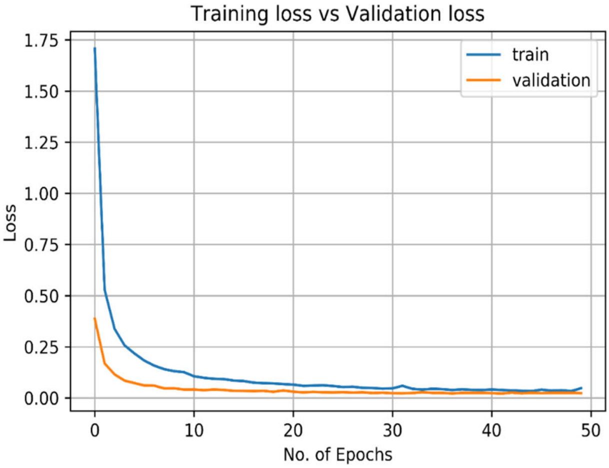

Fig. 8 Training/validation loss for pre-training model (50 Epochs)

The decrease in the training and validation losses for the fine-tuning stage is shown in Figs. 11 and 12, respectively. Compared to the pre-training model, the fine-tuning loss does not descend to much lower values. Therefore, it is evident the model is neither producing high final training accuracies nor is it generalized very well, due to the lack of data.

5.4 Comparison with machine learning and deep learning architectures

To validate our selection of LeNet and consolidate the claim of improved accuracy, we have implemented several machine and deep learning models. The obtained results are detailed as follows.

Table 3 shows the results for AlexNet [25] and Visual Geometry Group-19 (VGG-19) [23] models. The AlexNet model has 5 convolutional layers each followed by a ReLu activation layer and a max-pooling layer. The 5th max-pooling layer is followed with a Softmax activation. The VGG-19 model consists of 13 convolutional layers and 4 max-pooling layers,

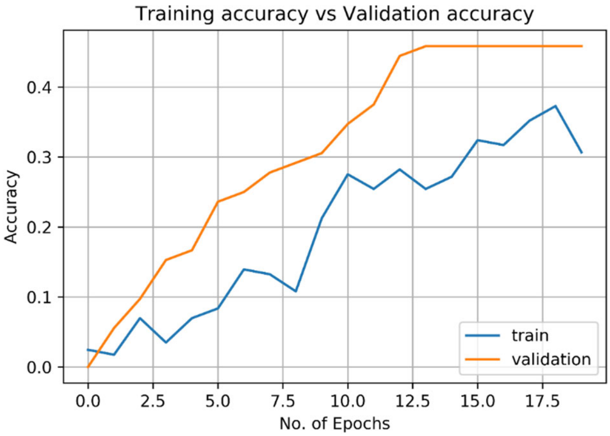

Fig. 9 Training/validation accuracy for fine-tuning model (20 Epochs)

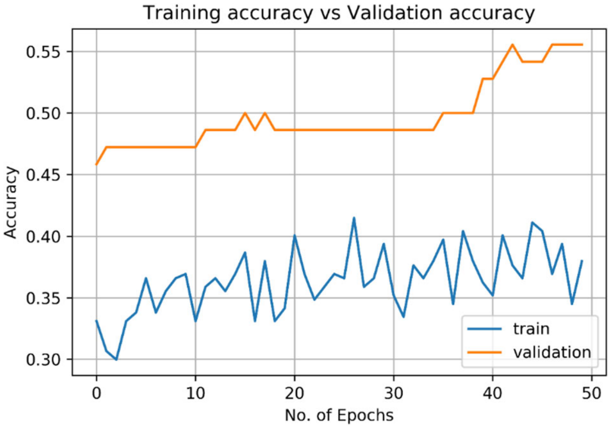

Fig. 10 Training/validation accuracy for fine-tuning model (50 Epochs)

followed by 2 fully connected layers and an output layer with a Softmax activation whereas all the other model layers have the ReLu activation.

The models are trained for 50 epochs on a dataset consisting of grayscale images of 64 × 64 resolution. The learning rate for AlexNet is 0.0001, while it stands at 0.001 for VGG-19 and ResNet, respectively. Once pre-trained, the models are fine-tuned on the Pakistani dataset for 50 epochs with the same learning rates.

When training the AlexNet model on the German dataset, the training accuracy reaches 98.2% and the validation accuracy climbs to 97.7%. This is because the German dataset is very extensive with thousands of images per class. When the convolutional layers are frozen and the model is trained on the Pakistani dataset, the training accuracy reaches 97.5% while the validation accuracy stays at a lower value of 44.4%, respectively.

When trained on the German dataset, the ResNet model achieves a training accuracy of 99.1% and a validation accuracy of 96.2%, respectively. When fine-tuned on the Pakistani

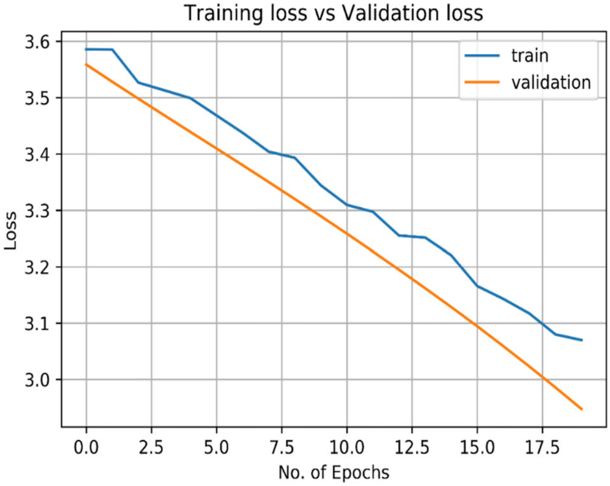

Fig. 11 Training/validation loss for fine-tuning model (20 Epochs)

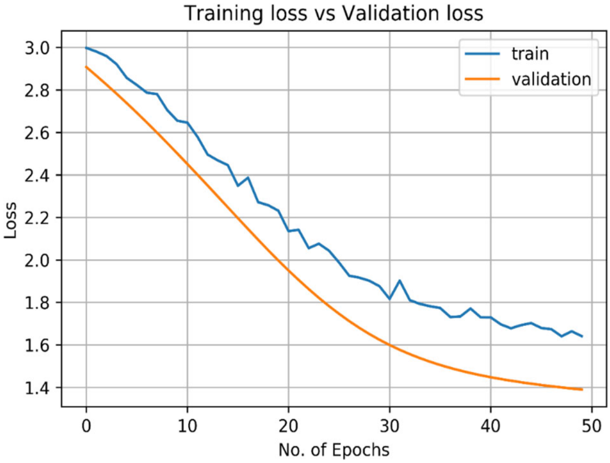

Fig. 12 Training/validation loss for fine-tuning model (50 Epochs)

dataset, the training accuracy peaks to 95.2% while the maximum testing accuracy slides down to 45.1%.

The VGG-19 model produces accuracies of 99.5% for both training and validation considering the German dataset while the accuracies for the Pakistani dataset are 95.6% for training and 40.2% for the validation/testing sets, respectively. Moreover, our LeNet model achieves the train accuracy value closed to the VGG-19 model and a middle test accuracy value of 98.8% considering the German dataset. However, LeNet outperforms AlexNet, ResNet, and VGG-19 models by achieving a test accuracy value of almost 55% on Pakistani dataset. This high accuracy is mainly because LeNet has lesser number of layers than AlexNet, ResNet, and VGG-19, therefore, the number of optimizable parameters in LeNet also remains fewer. On a small dataset, like ours, shallow networks perform well because, it does not take them long to optimize all their parameters and converge to a solution. On the other hand, relatively deeper networks require longer periods of time to converge on small datasets.

To justify the use of deep learning over machine learning techniques, different machine learning models are trained on the Pakistani dataset. Some of these models as given in Table 4 are, the Support Vector Machine (SVM), the one vs. all classification, and the random forest. The SVM model is trained using the Radial Bias Function (RBF) kernel to account for the non-linearity in the dataset. The one-

Table 3 Training and Testing accuracy comparison with Deep Learning models

Model name	Dataset type	Train accuracy	Test accuracy
AlexNet [25]	German Dataset	98.2%	97.7%
	Pakistani Dataset	97.5%	44.4%
VGG-19 [47]	German Dataset	99.5%	99.5%.
	Pakistani Dataset	95.6%	40.2%
ResNet [6]	German Dataset	99.1%	96.2%.
	Pakistani Dataset	95.2%	45.1%
LeNet (our results)	German Dataset	99.3%	98.8%.
	Pakistani Dataset	96.3%	54.8%

Table 4 Testing accuracy comparison with Machine Learning models

Model name	Accuracy
Support Vector Machine	36.1%
One Vs. All	43.0%
Random Forest	44.4%

vs-all classifier trains a classifier for each label in the dataset and then combines these classifiers together to create a single classifier for all the categories in the dataset. Finally, the random forest classifier uses multiple decision trees to classify between different classes, in this case, 100 decision trees are used to classify between the dataset categories.

Further, the results in Table 4 produce lower accuracies than the deep learning models given in Table 3 because the machine learning techniques can take an image as a whole, do not extract any features from it, and directly train on the pixel values of the image, thus, resulting in a lower performance accuracy. Machine learning methods on the other hand take an image as an input and then extract the features from it which are helpful in classifying the image accurately, resulting in higher accuracies.

In addition to above, we have also compared the training and validation losses and accuracies for the techniques mentioned in Table 3 as depicted in Figs. 13, 14, 15, and 16 considering the German (or GTSRB) and the Pakistani (or PTSD) datasets, respectively. Clearly, in Fig. 13, the loss for each model reduces over the number of epochs as expected since each model makes fewer misclassifications on the dataset as the time passes by. This reduction also means that the model is predicting the right class with more confidence. Moreover, after 10 epochs, the loss adapts a constant behavior indicating the generalization of each model on the dataset.

Figure 14 shows the training and validation accuracies of the four models. It is evident that the accuracies grow exponentially in the start and then slow down to a constant after the initial burst. Since the German dataset contains thousands of pictures, it helps the model generalize very well on the training data which results in a very high accuracy on the validation data for all three of the models.

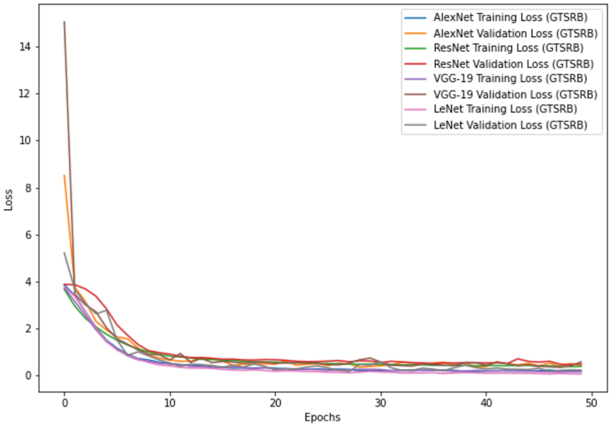

Fig. 13 The training and validation loss of AlexNet, VGG-19, ResNet, and LeNet on the German dataset (GTSRB)

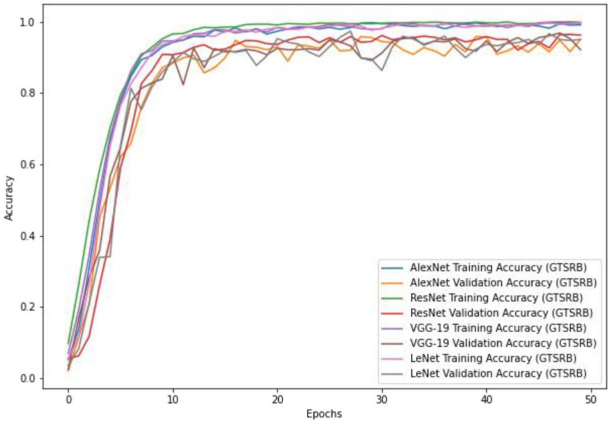

Fig. 14 The training and validation accuracy of AlexNet, VGG-19, ResNet, and LeNet on the German dataset (GTSRB)

Figure 15 shows the training and validation losses for all four modes on the Pakistani dataset. The validation loss starts at a very high value and then it drops down pretty quickly due to the small size of dataset. In the first few epochs, the training loss is very low while the validation loss remains high because the models overfit on the training data and do not perform well on the validation data. After a few epochs, the models generalize, and the validation losses are reduced indicating that the models have started to generalize on the data.

Figure 16 shows the training and validation accuracies on the Pakistani dataset. The accuracy on the training dataset for all the models is very high since the parameters for each model have been trained on this data resulting in higher values. For the validation accuracy, our model (LeNet) outperforms the other models because LeNet has lesser number of layers than AlexNet, ResNet, and VGG-19 as mentioned before.

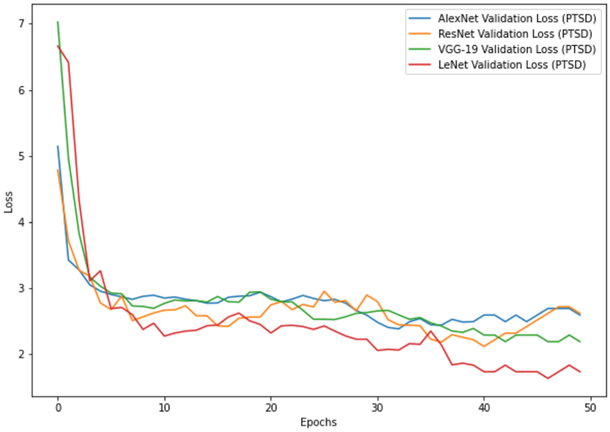

Fig. 15 The training and validation loss of the AlexNet, VGG-19, ResNet, and LeNet on the Pakistani dataset (PTSD)

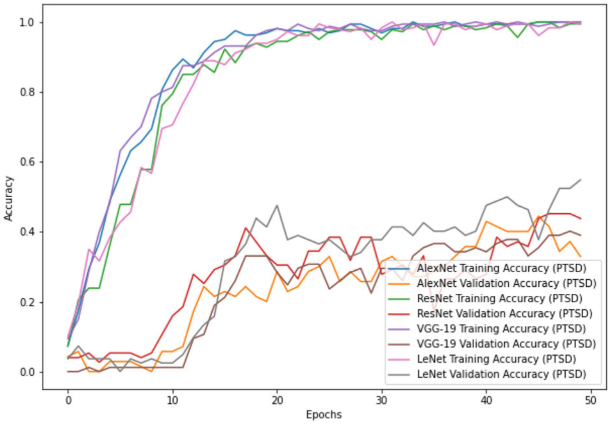

Fig. 16 The training and validation accuracy of the AlexNet, VGG-19, ResNet, and LeNet on the Pakistani dataset (PTSD)

6 Conclusion & future work

The goal of this research is to train a CNN model for Pakistani traffic-sign recognition. CNNs require a huge amount of input data to produce worthwhile accuracies, therefore, we have used transfer learning to compensate for the lack of data. The CNN model (pre-trained on the German dataset) has been fine-tuned using the Pakistani dataset and through experimental results, we have been able to achieve a final training accuracy of 54.8% over 50 epochs. Despite the time and money constraints, achieving such an accuracy is quite overwhelming, however, the accuracy can further be improved in the future. Normally, the accuracies are directly proportional to the number of dataset images and therefore an increment in images would ideally result in better accuracies. In our work, we used 35 classes of signs which will be increased in the future.

The novelty of this research lies in the fact that a new dataset was formed, and it was used to train a model to detect the Pakistani traffic-signs, however, our research is only a gateway to producing systems which are able to work in Pakistan to recognize traffic-signs. There is a need to collect more images to enhance the accuracy and ultimately the feasibility of implementation in real-world scenarios.

Declarations The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Alam A, Jaffery ZA (2020) Indian traffic sign detection and recognition. Int J Intell Transp Syst Res 18:98–112. https://doi.org/10.1007/s13177-019-00178-1
2. Bayoudh K, Hamdaoui F, Mtibaa A (2021) Transfer learning based hybrid 2D-3D CNN for traffic sign recognition and semantic road detection applied in advanced driver assistance systems. Appl Intell 51:124–142. https://doi.org/10.1007/s10489-020-01801-5
3. Belaroussi R, Foucher P, Tarel J-P, Soheilian B, Charbonnier P, Paparoditis N (2010) Road sign detection in images: a case study. In: Pattern Recognit. (ICPR), 2010 20th Int. Conf, pp 484–488. https://doi.org/10.1109/ICPR.2010.1125

4. Greenhalgh J, Mirmehdi M (2012) Real-Time Detection and Recognition of Road Traffic Signs. https://doi.org/10.1109/tits.2012.2208909
5. Grigorescu C, Petkov N (2003) Distance sets for shape filters and shape recognition. IEEE Trans Image Process 12:1274–1286. https://doi.org/10.1109/TIP.2003.816010
6. He K, Zhang X, Ren S, Sun J (2015) Deep residual learning for image recognition. In: Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. 2016-December, pp 770–778
7. Jochem T, Pomerleau D, Armstrong J, Kumar B (2014) PANS : A portable navigation platform. https://doi.org/10.1109/IVS.1995.528266
8. keras-team/keras (n.d.) Deep Learning for humans. https://github.com/keras-team/keras, last accessed 2020/06/16.
9. Khurshid A, Sohail A, Khurshid M, Shah MU, Jaffry AA (2021) Analysis of road traffic accident fatalities in Karachi. Pakistan: An Autopsy-Based Study Cureus 13. https://doi.org/10.7759/CUREUS.14459
10. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization:1–13. https://doi.org/10.1145/1830483.1830503
11. Kiranyaz S, Ince T, Abdeljaber O, Avci O, Gabbouj M (2019) 1-D Convolutional Neural Networks for Signal Processing Applications. In: ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings. pp. 8360–8364. Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ICASSP.2019.8682194
12. Larsson F, Felsberg M (2011) Using Fourier descriptors and spatial models for traffic sign recognition. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), pp 238–249. https://doi.org/10.1007/978-3-642-21227-7_23
13. Lawler R Riding shotgun in Tesla's fastest car ever, https://www.engadget.com/2014-10-09-tesla-d-awd-driver-assist.html
14. Lecun Y, Bottou L, Bengio Y, Ha P (1998) Gradient-Based Learning Applied to Document Recognition. https://doi.org/10.1109/5.726791
15. LeCun Y, Kavukcuoglu K, Farabet C (2010) Convolutional networks and applications in vision. IEEE Int. Symp. Circuits Syst:253–256. https://doi.org/10.1109/ISCAS.2010.5537907
16. Li Y, Mogelmose A, Trivedi MM (2016) Pushing the "speed limit": high-accuracy US traffic sign recognition with convolutional neural networks. IEEE Trans Intell Veh 1:167–176. https://doi.org/10.1109/tiv.2016.2615523
17. Lin C, Li L, Luo W, Wang KCP, Guo J (2019) Transfer learning based traffic sign recognition using inception-v3 model. Period Polytech Transp Eng 47:242–250. https://doi.org/10.3311/PPtr.11480
18. Lodhi A, Singhal S, Massoudi M (2021) Car Traffic Sign Recognizer Using Convolutional Neural Network CNN. In: Proceedings of the 6th International Conference on Inventive Computation Technologies, ICICT 2021. Institute of Electrical and Electronics Engineers Inc, pp 577–582. https://doi.org/10.1109/ICICT50816.2021.9358594
19. Mahmoud MAB, Guo P (2019) A novel method for traffic sign recognition based on DCGAN and MLP with PILAE algorithm. IEEE Access 7:74602–74611. https://doi.org/10.1109/ACCESS.2019.2919125
20. Malik Z, Siddiqi I (2014) Detection and recognition of traffic signs from road scene images. In: Proc. - 12th Int. Conf. Front. Inf. Technol. FIT 2014, pp 330–335. https://doi.org/10.1109/FIT.2014.68
21. Malik R, Khurshid J, Ahmad SN (2007) Road sign detection and recognition using colour segmentation, shape analysis and template matching. In: 2007 Int. Conf. Mach. Learn. Cybern, vol 6, pp 19–22. https://doi.org/10.1109/ICMLC.2007.4370763
22. Manikandan R (2018) Sign recognition - how well does single shot multibox detector sum up? A Quantitative Study. In: Proceedings - Applied Imagery Pattern Recognition Workshop Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/AIPR.2018.8707409
23. Mateen M, Wen J, Nasrullah, Song S, Huang Z (2019) Fundus image classification using VGG-19 architecture with PCA and SVD. Symmetry (Basel) 11:1. https://doi.org/10.3390/sym11010001
24. Mathias M, Timofte R, Benenson R, Van Gool L (2013) Traffic sign recognition—How far are we from the solution? Neural Networks (IJCNN), 2013. In: Int. Jt. Conf, pp 1–8. https://doi.org/10.1109/IJCNN.2013.6707049
25. Minhas RA, Javed A, Irtaza A, Mahmood MT, Joo YB (2019) Shot classification of field sports videos using AlexNet convolutional neural network. Appl Sci 9:483. https://doi.org/10.3390/app9030483
26. Møgelmose A, Trivedi MM, Moeslund TB (2012) Vision-based traffic sign detection and analysis for intelligent driver assistance systems: perspectives and survey. IEEE Trans Intell Transp Syst 13:1484–1497. https://doi.org/10.1109/TITS.2012.2209421
27. Møgelmose A, Liu D, Trivedi MM Detection of US Traffic Signs. IEEE Trans Intell Transp Syst 1
28. Moiseev B, Konev A, Chigorin A, Konushin A (2013) Evaluation of traffic sign recognition methods trained on synthetically generated data. In: Lecture Notes in Computer Science (including subseries Lecture

Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Springer Verlag, pp 576–583. https://doi.org/10.1007/978-3-319-02895-8_52

29. Nadeem Z, Samad A, Abbas Z, Massod J (2018) A transfer learning based approach for pakistani traffic-sign recognition using ConvNets. In: 2018 International Conference on Computing, Electronic and Electrical Engineering, ICE Cube 2018. IEEE, Quetta, pp 1–6. https://doi.org/10.1109/ICECUBE.2018.8610979
30. NN SVG (n.d.) https://alexlenail.me/NN-SVG/LeNet.html, last accessed 2020/06/12
31. Obulesh A, Sri Sahithi P, Deepesh Kumar M, Pavitra M (n.d.) Traffic-Sign Classification Using Machine Learning Concepts | Tathapi with ISSN 2320–0693 is an UGC CARE Journal, http://tathapi.com/index.php/2320-0693/article/view/317, last accessed 2020/06/12
32. Pakistan Bureau of Statistics: Annual Traffic Accidents in Pakistan, https://www.pbs.gov.pk/sites/default/files//tables/rename-as-per-table-type/tarffic_accidents_annaul__09_02_2021.pdf, last accessed 2021/10/13
33. Sedik A, Hammad M, Abd El-Samie FE, Gupta BB, Abd El-Latif AA (2021) Efficient deep learning approach for augmented detection of Coronavirus disease. Neural Comput Appl:1–18. https://doi.org/10.1007/s00521-020-05410-8
34. Shakhuro VI (2016) A.S.K.: Russian traffic sign images dataset. Comput Opt 40:294–300. https://doi.org/10.18287/2412-6179-2016-40-2-294-300
35. Shustanov A, Yakimov P (2017) CNN design for real-time traffic sign recognition. Procedia Eng 201:718–725. https://doi.org/10.1016/j.proeng.2017.09.594
36. Singh NS, Hariharan S, Gupta M (2020) Facial recognition using deep learning. In: Lecture Notes in Electrical Engineering. Springer, pp 375–382. https://doi.org/10.1007/978-981-15-0372-6_30
37. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15:1929–1958. https://doi.org/10.1214/12-AOS1000
38. Stallkamp J, Schlipsing M, Salmen J (n.d.) German Dataset, http://benchmark.ini.rub.de/?section=gtsrb&subsection=dataset.
39. Stallkamp J, Schlipsing M, Salmen J, Igel C (2011) The German Traffic Sign Recognition Benchmark: A multi-class classification competition. In: Proc. Int. Jt. Conf. Neural Networks, pp 1453–1460. https://doi.org/10.1109/IJCNN.2011.6033395
40. Stallkamp J, Schlipsing M, Salmen J, Igel C (2012) Man vs computer: Benchmarking machine learning algorithms for traffic sign recognition. https://doi.org/10.1016/j.neunet.2012.02.016
41. Tan C, Sun F, Kong T, Zhang W, Yang C, Liu C (2018) A survey on deep transfer learning. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Springer Verlag, pp 270–279. https://doi.org/10.1007/978-3-030-01424-7_27
42. Valiente R, Zaman M, Ozer S, Fallah YP (2019) Controlling steering angle for cooperative self-driving vehicles utilizing CNN and LSTM-based deep networks. In: IEEE Intelligent Vehicles Symposium, Proceedings. Institute of Electrical and Electronics Engineers Inc, pp 2423–2428. https://doi.org/10.1109/IVS.2019.8814260
43. Vidushi AM (2021) A study on image analysis and recognition using learning methods: CNN as the best image learner. In: Lecture Notes on Data Engineering and Communications Technologies. Springer Science and Business Media Deutschland GmbH, pp 23–30. https://doi.org/10.1007/978-981-15-8335-3_3
44. Wali SB, Hannan MA, Hussain A, Samad SA (2015) An automatic traffic sign detection and recognition system based on colour segmentation, shape matching, and SVM. Math Probl Eng 2015:1–11. https://doi.org/10.1155/2015/250461
45. WHO: Road Traffic Accidents in Pakistan, https://www.worldlifeexpectancy.com/pakistan-road-traffic-accidents, last accessed 2021/10/13
46. Wigness, M., Eum, S., Rogers Iii, J.G., Han, D., Kwon, H. (n.d.) A RUGD Dataset for Autonomous Navigation and Visual Perception in Unstructured Outdoor Environments.
47. Xia Z, Yue G, Xu Y, Feng C, Yang M, Wang T, Lei B (2020) A Novel End-to-End Hybrid Network for Alzheimer's Disease Detection Using 3D CNN and 3D CLSTM. In: Proceedings - International Symposium on Biomedical Imaging. IEEE Computer Society, pp 416–419. https://doi.org/10.1109/ISBI45749.2020.9098621
48. Zhao J, Liang B, Chen Q (2018) The key technology toward the self-driving car. Int J Intell Unmanned Syst 6:2–20. https://doi.org/10.1108/IJIUS-08-2017-0008

Affiliations

Zain Nadeem[1] · **Zainullah Khan**[1] · **Usama Mir**[2] · **Umer Iftikhar Mir**[1] · **Shahnawaz Khan**[1,3] · **Hamza Nadeem**[1] · **Junaid Sultan**[1]

Zain Nadeem
nadeem.zain@outlook.com

Zainullah Khan
zain.9496@gmail.com

Umer Iftikhar Mir
umar.i.mir@gmail.com

Shahnawaz Khan
skhan@ucb.edu.bh

Hamza Nadeem
contact.hamzasheikh@gmail.com

Junaid Sultan
mjunaidsultan30@gmail.com

[1] Engineering and Management Sciences, Balochistan University of Information Technology, Quetta, Pakistan

[2] University of Windsor, Windsor, Canada

[3] University College of Bahrain, Saar, Bahrain