IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 25, 2019, accepted May 20, 2019, date of publication May 27, 2019, date of current version June 20, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2919125

A Novel Method for Traffic Sign Recognition Based on DCGAN and MLP With PILAE Algorithm

MOHAMMED A. B. MAHMOUD [1] AND PING GUO [1,2], (Senior Member, IEEE)
[1]School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100091, China
[2]School of Systems Science, Beijing Normal University, Beijing 100875, China

Corresponding author: Ping Guo (pguo@ieee.org)

This work was supported in part by the National Natural Science Foundation of China under Grant 61375045, and in part by the Joint Research Fund in Astronomy through cooperative agreement between the National Natural Science Foundation of China (NSFC) and Chinese Academy of Sciences (CAS) under Grant U1531242.

ABSTRACT This paper centers on a novel method for traffic sign recognition (TSR). The method comprises of two major steps: 1) make strong representations for TSR images, by extraction deep features with the deep convolutional generative adversarial networks (DCGANs) and 2) classifier defined by multilayer perceptron (MLP) neural networks trained with a pseudoinverse learning autoencoder (PILAE) algorithm. The PILAE training process is considered efficient in which it does not require the number of hidden layers specified nor does it need the setting of the learning control parameters. This results in the PILAE classifier attaining a better performance in terms of both accuracy and efficiency. Empirical results from the German TSR (GTSRB) and Belgium traffic sign classification (BTSC) have proved that TSR achieves excellent results with other algorithms and reasonably low complexity.

INDEX TERMS Deep convolutional generative adversarial networks (DCGAN), feature extraction, pseudoinverse learning autoencoder (PILAE), traffic sign recognition (TSR).

I. INTRODUCTION

To ensure optimum road safety, it is imperative that traffic sign data is known [1], consequently this has led to traffic sign recognition receiving immense attention. More so, TSR is expedient in traffic sign oversight and upkeep. Over the past ten years, TSR has received immense academic attention in canny transportation networks as well as in the example pattern recognition community. The most important factors that have attracted the testing of TS include, variable perspective, incomplete impediment, movement, obscure, differentiate basement, shading bending among others.

Being a normal pattern recognition errand, TSR preciseness lies principally on a feature extractor and a classifier. Before the invention of TSR, strategies, mostly relied on a comparable plan which consisted ordinary classifier and handcrafted features. Although there has been multiple hand crafted features coordinated and made with classifiers like support vector machine(SVM) [2], [3] and random forest [4], it is not yet simple to manage the increasing assorted variety and fickleness of traffic signs. Nonetheless, owing to the fact of their hidden binary classification technique, the strategies need a way they can address the imbalance that results among the quantity of negative and positive training samples. Consequently, these strategies only accomplish local optimum or an overfitting solution. Therefore, in TSR the first and most important task is making a classifier with the ability to achieve a generalized and ideal solution. Various academicians, scientists and scholars have tried out this. Specifically, Vondrick *et al.* [5] illustrated that handcrafted features like histogram of oriented gradients (HOG) [6] were not satisfactorily discriminative. In the handcrafted features space, samples from several classes proved to be sufficiently comparative, resulting to representation of deep features for traffic signs becoming a second issue requiring attention.

With time and development of technology gigantic superior computer hardware such as GPUs and gigantic data sets have been developed and have continuously demonstrated their amazing feature and learning ability. Resultantly, strategies that are DNN [7]–[9] based have been accompanied by multiple pattern classification errands. To handle TSR assignments Convolution neural networks (CNN) have been used and positive results have emerged [10]–[13]. However, attaining computational efficiency bearing in mind the great

The associate editor coordinating the review of this manuscript and approving it for publication was Chang-Tsun Li.

recognition rates and the high computational cost it comes with prove to be colossal third challenges. To overcome these challenges and to achieve high efficiency rates, we suggest the DCGAN-PILAE method. This method utilizes the PILAE [14] classifier with deep features which are extracted from DCGAN [15]. As far as recognition accuracy is concerned TSR method surpasses almost all state-of-the-art techniques and achieves relatively high results with reasonably low computational costs.

Recently learning strategies utilizing deep learning features have been efficiently connected to resolve multiple computer vision issues [8], [16]–[20] that prove troublesome. The core idea of deep learning is using first stemming forthright representations to learn hierarchical feature representation and then proceeding to unremittingly coming up with more complex ones from the preceding level. Compared to the handcrafted models, deep learning design is capable of encoding data that is multilevel from their initial simple nature to a more intricate one. Therefore, for feature learning, this method is highly encouraging as: 1) it doesn't necessitate ground truth; 2) It induces intricate non-linear connections using deep architecture that is hierarchical in nature; 3) It doesn't depend on chosen handcrafted features but rather is completely driven by date, and 4) owing to the fact that it possesses trained hierarchical deep network, it is capable of effectively and rapidly figuring the feature representation of low level images. Zhang *et al.* [21] utilized DCGAN for feature extraction in retrieval task for Hyperspectral Images. In [22] the authors used DCGAN to extract features as an alternative to use hand-crafted features. After training the network, they utilize the discriminator's features from all layers as signature features and then train hybrid user-independent/ user-dependent classifier. Gu [23] proposed a brief overview of several important generative models including WINN, WGAN-GP, VAE, and DCGAN. He investigates that DCGAN supplies the carefully-designed structures to settle the training process and attains the competitive results at the same time. The DCGAN feature can thus handle the second issue mentioned above.

Pseudoinverse learning algorithm (PIL) [24], [25], it is a multilayer perceptron (MLP) learning algorithm composed of stacked generalization connected such that it dominates the Neural Networks (NNs) degradation predictive accuracy. It's structure possesses identical number of hidden neurons as the number of samples that are to be learned. PIL overcomes learning errors by performing addition of hidden layers. It had been fully automated, feed-forward and did not contain critical user subordinate parameters, for example, learning rate, the maximum epoch and momentum constants. PIL has been proven to be an efficient algorithm and by far much better than the standard back propagation (BP) and other algorithm of gradient descent. Wang *et al.* [14] asserted that PILAE was a fully and fast automated framework that uses deep neural networks to train stacked autoencoders [9]. PILAE trains the stacked autoencoder by embracing the PIL algorithm with the low rank approximation, PILAE isn't a gradient descent method and doesn't have the shortcoming of gradient vanishing. It also don't have the problem of saturation activation as a matrix that is multiplied by its pseudoinverse. For the above mentioned issues specifically the first and the third, the combine PILAE with extracted features from DCGAN and can be able to get the balance between computational complexity and recognition accuracy.

The major contributions of this paper are outlined as pursues:

1) Differently, to prior work of TSR, deep features are extracted to signify the traffic sign images utilizing the DCGAN model, which can learn powerful feature representation in an unsupervised way. Therefore, we first attempt to extract the deep features with the DCGAN model for traffic sign images.
2) The known normal practice is random initialization of weight variables in BP algorithm and creation of delta learning rule to update the weight matrix. While in PILAE algorithm, weight parameters are calculated with the pseudoinverse solution and don't have to be modified further, that have a important effectiveness on the implementation of the model.
3) Investigate the performance of the introduced DCGAN-PILAE model on TSR standard data sets (i.e., the GTSRB dataset, and the BTSC dataset) leading to comparison of the recognition rate of peer opponents and analysis the evolution performance of the proposed model.

The paper proceeds to evaluate TSR related issued in section II; Section III provides a detailed description of DCGAN; Section IV present details of PILAE based classification; Section V presents the framework of the proposed method; Implementation details, and experimental results will be provided in Section VI; Section VII provides the discussion and finally the conclusion in Section VIII.

II. RELATED WORK

In TSR work, majority of state-of-the-art methods depends on handcrafted features, e.g. SIFT [26], [27], HOG [4], [28]–[33], color global and local-oriented edge magnitude pattern [34], Gabor [35] or an amalgamation of these features included by coding procedure [36]. In any of the cases, the recognition accuracy of these techniques isn't gratifying enough as there is discrimination of the existing handcrafted features. It is also highly time consuming to outline discriminative and robust features making it impractical in real world scenarios. Recently the most utilized TSR techniques largely fall back on CNN.

Several CNN models sets have been suggested for TSR. Wang *et al.* [14] applied both the low-level and high-level in which diverse convolution layers of CNN have been utilized to extract them. Dan *et al.* [11] on other hand used an image enhancement algorithm (CLAHE) to enhance the execution of CNN with MLPs trained with HOG features. Multiple CNNs [12] were utilized to enhance its operation. Jin *et al.* [13] suggested (HLSGD) which is a method to

train 20 CNNs, they got the best result over an identical outfit methodology in [12]. However, such a tremendous accomplishment is founded on CNN gathering. The fact that there is the necessity of preparing many CNNs means a heavy dependence on elite parallel computing and a vast computational weight.

Recently, there has been an increase of evidences that point out the generalization potential of CNN fully connected layers and the utilization of CNN-learnt features to forthright robust classifiers will have an advantage. Reference [37] and [38] included the CNN-learnt representations to a linear support vector machine (SVM) classifier. Ensuing evaluations on multiple characteristic datasets [39]–[42] showed better execution. Although, cross-validation is usually certain while used to choose the suitable parameters for SVM training, both computational cost and time will be problematic as the dataset develops. More so, the generalization of its performance isn't assured to be the best. To ensure a balance between the computational cost and accuracy, it was suggested that small-scale CNN be utilized in [43]. This small-scale CNN utilized bootstrapping training to improve its capability, to be discriminative [44] and to improve the GTSRB dataset performance with the mixture of the expertise from the inception module [45] and the spatial transformer networks [46].

PILAE achieves a higher training efficacy as well as a convergent optimal generalization ability. A novel TSR architecture with ability of combining deep extracted features by DCGAN with greater generalization of PILAE classifier is chosen for this paper. At first traffic sign images are used to train DCGAN, ensuing in an extraction feature which has the ability to learn deep features. PILAE should be applied at a particular instance to perform classification as well as extract features.

III. DEEP CONVOLUTION GENERATIVE ADVERSARIAL NETWORK (DCGAN)

DCGAN [15] is an image generation model that is composed of two neural networks, a discriminative network D and a generative network G. The G is a transposed convolutional neural network that doesn't possess fully-connected and pooling layers which yields images from d-dimensional vectors utilizing transposed convolutional layers. On the other hand, a D acquires identical composition as G but in opposite direction that discriminates if the input is a genuine image from a predefined dataset or whether it's an image made by G.

DCGAN learning process is done by repeating these steps: 1) Images from the dataset x are used to train the D network; 2) the generative network outputs generated insight images $G(z)$ from an arbitrary vector z (see Figures 1 and 2). Ultimately, the discriminative D network is updated from the generated image. This process aim is the point that over repeated iterations, the G training produces images are gradually more imperceptible from the images from the dataset.

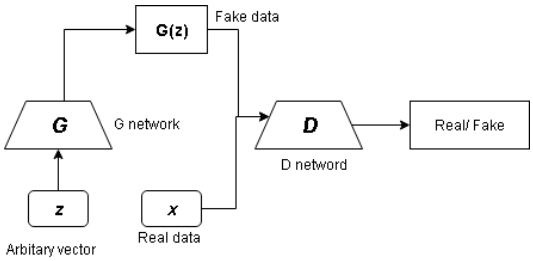

FIGURE 1. DCGAN architecture.

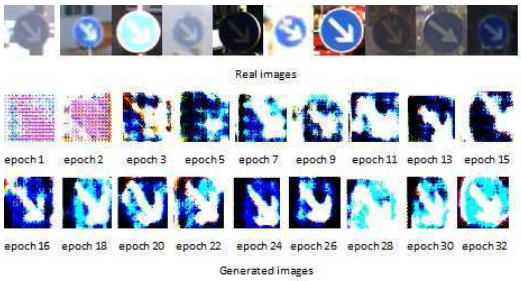

FIGURE 2. DCGAN real and generated images for class 38 of GTSRB dataset.

The DCGAN training is expressed as follows [47]:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data^{(x)}}}[logD(x)] + \mathbb{E}_{z\sim p_{z(z)}}[log(1-D(G(z)))], \quad (1)$$

where x represents the original image, z is a d-dimensional vector made up of random numbers, and $p_{data^{(x)}}$ and $p_{z^{(Z)}}$ will be the likelihood distributions of x and z respectively. $D(x)$ represents the inputs likelihood of being an image truly produced from $p_{data^{(x)}}$. $1 - D(G(z))$ is the likelihood that it can be produced from $p_{z^{(Z)}}$. G is trained on such manner that it reduces $log(1 - D(G(z)))$ so that it can deceive D. On other hand D is trained so that it can increase the rate of right answer.

After unsupervised representation is learned by using DCGAN, we then use part of the convolution layers of the discriminator as feature extraction. The extracted features are gained by transferring data items through the discriminator until certain layer f. The features matrices f are flatten and afterward trained by a classifier which uses the extracted features as input.

IV. PSEUDOINVERSE LEARNING AUTOENCODER (PILAE)

It is popular the fact that a single layer feedforward network (SLFN) gets the capability of accomplishing universal approximation. This advantage has built SLFNs highly adjust to various environments where in fact the processes could have uncertainties and complicated dynamics, and the collected data could be degenerated by disruptions and noises. On the other hand, many present training techniques designed for the SLFN derive from conventional BP. Many of these BP-based techniques using the recursive feature have

drawbacks of the longer training period, stopping at local minima in training and slow convergence.

PIL [24], [48] is a supervised learning algorithm for feed-forward neural network, it is dependent on generalized linear algebra and was proposed by Guo et al.. PIL is different from algorithms that are gradient based including BP in that, PIL is not necessarily required to adjust affined parameters such as stride length, momentum and learning epoch instead these parameters are normally hard selected by the users.

The training set with N arbitrary multiple samples for a supervised learning problem are labeled $D = \{x^i, o^i\}_{i=1}^N$, where $x^i = (x_1, x_2, \ldots, X_d) \in R^d$ is the i-th input vector, $o^i = (o_1, o_2, \ldots, o_m) \in R^m$ is the congruent target output vector. Cogitate, SLFN posseting p hidden neurons that are totally linked with d inputs neurons and m output neurons, the supervised learning function implies attempting to realize the weight matrix that lessen the following function of sum-square-error:

$$E = \frac{1}{2N} \sum_{i=1}^{N} \sum_{j=1}^{m} \|g_j(x^i, \theta) - o_j^i\|^2, \tag{2}$$

where θ is network parameter which have connection weight W, bias parameter and $g_j(x^i, \theta)$ is a function that maps input vectors to the j-th output neuron of the output values. The mapping function could be determined as

$$g_j(x, \theta) = \sum_{i=1}^{p} w_{i,j}^1 \sigma(\sum_{k=1}^{d} w_{k,i}^o x_k + \theta), \tag{3}$$

where θ represents the input layer bias parameter.

To achieve simplification, the propagation of the SLFN can be determined in the matrix form:

$$H = \sigma(XW_0 + \theta), \quad X \in R^{Nxd}, \ W_0 \in R^{dxp}, \tag{4}$$

where H represents the hidden layer output and X comprises of N input rows vectors and d columns and is the input matrix. The expressions $W_0 = [w_1^0, w_2^0, \ldots, w_p^0]$ gives the input weight matrix. An arbitrary column of W_0, $w_i^0 = [w_{i,1}^0, w_{2,i}^0, \ldots, w_{d,i}^0]^T$, represent the bond weight amongst the i-th hidden neuron and whole input neurons. $\sigma(.)$ represents the activation function such as the sigmoidal, hyperbolic and rectifier functions.

The SLFN output should be

$$G = HW_1, \quad H \in R^{Nxp}, \ W_1 \in R^{pxm}, \tag{5}$$

where $W_1 = [w_1^1, w_2^1, \ldots, w_m^1]$ is the output weight matrix and W_1, $w_i^1 = [w_{1,i}^1, w_{2,i}^1, \ldots, w_{p,i}^1]^T$, is the i-th column of W_1 and represent the connection weight among entire hidden neurons and also the neuron that is the i-th output. Subsequently by reformulation of the formulas we can design the supervised learning problem as

$$\min_{W_1} : \|HW_1 - O\|^2, \tag{6}$$

where $O \in R^{Nxm}$ represents the target label matrix that comprising of m columns and N rows label vectors.

Guo and Lyu [24] solved the optimization problem in Eq.(6), using pseudoinverse as

$$W_1 = H^+ O \tag{7}$$

where H^+ identify the pseudoinverse of output matrix H of the hidden layer. This result is a far much better approximation for $HW_1 = O$ using the linear algebra theorem [24], [48], [49]. An autoencoder [7], [9], [50] can be observed as a specific kind of SLFN which models output to be equal to the inputs. PIL is utilized to train stacked autoencoder (SAE) with layer-wise learning procedure that is greedy based, PILAE is utilized as SAE building block. The number of hidden neuron is defined by equation in [14]. To enable the extraction of data feature, the number of hidden neuron is a bit greater to the rank of the input network and in general fewer than the input vectors dimension. The rank of input matrix is computed by the singular value decomposition method (SVD). Truncated SVD is specifically used to compute the input matrix pseudoinverse, which is then utilized as the encoder weight network. PIL is on the other hand used to compute the decoder weigh matrix. Additionally, so as to reduce the degree of independence parameter, the decoder and encoder weigh are tied up, allowing weight of encoder network equal to the decoder weight that is transposed. The autoencoder is capable of mapping the data to high or low rank approximation is used to map the data to low rank dimensions.

The following are the steps of the PILAE algorithm:

1) compute the input matrix X pseudoinverse with *SVD* technique. The *SVD* of X is identified as

$$X = U \Sigma V^T \tag{8}$$

From the *SVD* output, X pseudoinverse is

$$X^+ = V \acute{\Sigma} U^T \tag{9}$$

$\acute{\Sigma}$ represents the diagonal matrix that is transposed that consisted of the mutual of nonzero elements in matrix Σ. The number of hidden neurons are assigned as

$$p = \beta Dim(x), \quad \beta \in (0, 1]. \tag{10}$$

Dim is function that gives the input matrix dimension, on other hand β can be an empirical parameter that is based upon the degree type of the dimension to be reduced. This will ensure that dimension reduction is enhanced.

2) calculating low rank approximation of matrix X^+ as $\hat{X}^+$:

$$\hat{X}^+ = \hat{V} \acute{\Sigma} U^T, \tag{11}$$

where $\hat{V}$ is made up of the initial p rows of singular matrix V. According to PIL algorithm the encoder weight is equal to truncated pseudoinverse matrix $W_e = \hat{X}^+$, the matrix X is projected into the p dimensional hidden feature space with W_e as

$$H = f(W_e X) \tag{12}$$

where $f(.)$ is the activation function.

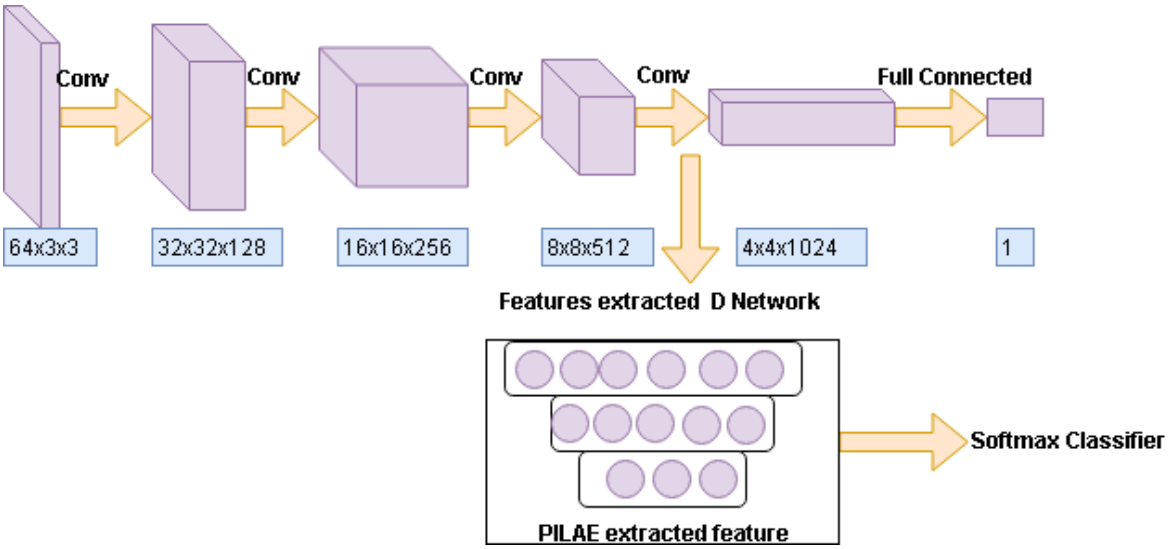

FIGURE 3. DCGAN-PILAE proposed method.

3) The decoder weight, $W_d H = H$ has the optimal pseudoinverse solution $W_d = XH^+$ and H^+ is calcuated by

$$H^+ = H^T(HH^T + kI)^{-1} \tag{13}$$

where $k > 0$ is regularization restriction which the user specifies. It could be approximated with the formulation produced in [51], thus the decoder weight is given by following formula

$$W_D = XH^T(HH^T + kI)^{-1} \tag{14}$$

the decoder and encoder weights will be tied together, namely, $W_e = (W_d)^T$.

4) If the condition $\|H^+H - I\|_p^2 < \varepsilon$ is satisfied, the process of training terminated, with all trained autoencoder (without decoders) stacked into a deep neural network. More so, the matrix H should be taken as input to train yet another autoencoder by iterating the training procedure mentioned above.

The trained stacked autoencoders are increasingly being utilized because of the feature learning design. We utilize the softmax framework as the classifier. The softmax classifier will gather the output from the stacked autoencoders as the input features. The softmax method estimates the probability of the class label adopting all the K potential rates. The class label is then divided by selecting the label with the maximum probability. For confirmed sample x^i, the softmax classifier result quotes the class label probabilities as

$$p(y^i = k|x^i; \theta) = \frac{exp(\theta_k^T x^i)}{\sum_{j=1}^{K} exp(\theta_k^T x^i)} \tag{15}$$

where y is the class label and $k \in \{1, 2, \ldots K\}$; θ is the parameter set of the classifier. Given a training set, the parameter θ is trained to minimize the following loss function:

$$J(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{K} 1\{y^i = j\} \log p(y^i = k|x^i; \theta), \tag{16}$$

where $1\{.\}$ returns 0 if its input argument is false, and 1 otherwise. The benefit of utilizing softmax is the output probabilities scope. The scope will be from 0 to 1, and the sum of the all of the probabilities will be equal to one. In case the softmax function utilized for multi-classification model like our framework it returns the probabilities of each class and the prospective class could have a high probability.

V. PROPOSED ARCHITECTURE

Currently most of the algorithms in use utilize CNNs to perform both feature extraction and classification. Although in such instances impressive results are obtained, there's always the disadvantage of exceptionally complicated and large network or ensemble learning, as well as large amounts of data. With the aim of fully utilizing both PILAE and DCGAN benefits, a novel TSR structure is proposed (see Figure 3). PILAE has the advantage of incorporating a prevailing classification performance, as conflicting to the SVM or CNN which back propagation is used in their training, consequently DCGAN is treated as the feature extractor, implying that only after the entire DCGAN training is completed only convolutional layers are maintained. There are several reason which make it preferable to choose a DCGAN network: 1) The fact that it possess a swift ability to converge; 2) The generator model is able to efficiently fill as the intensity model of the training data; and 3) It achieved sampling in a proficient and straight forward way. The Stochastic Adam Optimizer has been utilized in training DCGAN as it aims at propping and over fitting convergence freely. Moreover computationally it is proficient and is invariant to diagonal rescaling of the slopes,

has little memory necessities, and is suitable for problems that are substantial as far as data and parameters are concerned.

PILAE is given with features that are extracted by DCGAN and trained also as feature extraction model then use softmax classifier. The PILAE output is given to softmax classifier as input features. In this paper's approach, the trained discriminator network D is used as a feature extractor for TSR. We use the same structure proposed in [15]. Data augmentation isn't applied, that denotes deformation techniques (flip, rotation, scale, crop, etc.) aren't utilized in the training set. The D network fourth layer is booked out as an extractor feature as the DCGAN isn't used to perform classification but instead used to extract deep features. A 8,192 dimensional vector space for each sample is then formed by concatenating and flattening these features.

The feature extractor and a classifier determine the TSR accuracy. The higher the distinct features are, the better the classifier is and is also directly proportional to the recognition rate. To begin with, BP possess the quality of being delicate to the training error surface minima. The other SLFN, can be over trained and thus achieve a generalized performance that is non ideal when BP learning is used. The PILAE output classifier has a better generalized performance, when SLFN is used to train PILAE.

VI. EXPERIMENTS AND ANALYSIS

Two datasets have been used to evaluate the proposed method, these are, (GTSRB) [52],[1] and (BTSC) [53].[2] The table 1 details the two datasets. They include their specific division of training and testing data. The GTSRB dataset contains more images than BTSC, it also possess traffic sign classes with small range and a small amount of traffic signs that are physically distinctive held by these images. The signs have been arranged in 43 classes and signs numbers amid classes that aren't balanced, the traffic signs can thus be divided into 6 groups as Figure 4 shows, including 8 speed limit signs, 4 other prohibitory signs, 8 mandatory signs, 4 derestriction signs, 15 danger signs, and 4 unique signs. The images size are divergent from 15x15 to 250x250 pixels. The BTSC dataset consists of cropped signs of annotations for 62 diverse traffic signs classes as [53]. In the two datasets the image sizes are altered to 64x64 pixels through bilinear interpolation before DCGAN training thus it has the same input size in [15]. All DCGAN training hyperparameters are set to a certain framework [15], after preliminary experiments, we found that it was the best to set the learning rate to 4e-3 for GTSRB dataset. For the BTSC dataset 3e-3 yields the best performance. All experiments are done on a computer with the Intel core i7-6800k CPU, one GPU GTX1080. We performed our framework in MATLAB, utilizing the MatConvNet library [54], [55] for our execution of DCGAN-PILAE model. The suggested method is weighed against released methods whose computation settings is similar or more than ours.

[1] http://benchmark.ini.rub.de/

[2] http://homes.esat.kuleuven.be/~rtimofte

FIGURE 4. Categories of GTSRB dataset.

TABLE 1. The three datasets details used for evaluate proposed method.

Dataset	Training	Testing	Classes
GTSRB	39209	12630	43
BTSC	4591	2534	62

A. ANALYSIS ON GTSRB DATASET

The best achieved results for the GTSRB data set are gotten from 11 diverse algorithms that incorporat.

1) Committee of CNNs [12]: It is used 25 distinctive deep neural networks with data augmentation methods for TSR.
2) Single CNN with 3 STNs [56]: it is based on a CNN that contains Spatial Transformer Networks (STN).
3) K-d Trees [4]: it uses HOG extracted features with K-d trees as classifier.
4) Multiscale CNNs [10]: It is depended on features learned from a group of multiscale CNNs.
5) mIncept [44]: It is based on the mixture of the expertise from the inception module [45] and the spatia transformer networks [46].
6) Random Forests [4]: It uses HOG features and Random Forests as a classifier.
7) LDA [52]: It is based on HOG features followed by linear discriminant analysis (LDA) framework to make the classifier.
8) Improved VGG [57]: it is based on modified model of VGG-16 which contains 9 layers. This method provides Batch Normalization (BN) and dropout operations after every fully-connected (FC) layer, to help expand speed up the model convergence, and then it can progress classification effect.
9) HLSGD [13]: It suggests a modified version of cross entropy loss which is used in training CNN. The features are learned utilizing these CNN model.
10) Residual convolutional blocks [58]: It is based on residual convolutional blocks and hierarchic enlarged skip connections joined in steps.

TABLE 2. Result comparisons for gtsrb dataset with best methods.

Method	Recognition rate	Training time	Configuration
Committee of CNNs	99.64	37h/dataset	CPU: I7-950 GPU: 4 x GTX580
Single CNN with 3 STNs	99.71	2h/dataset	CPU: i7-6700k GPU: 1 x GTX1070
K-d Trees	92.70	N/A	N/A
Multiscale CNNs	98.84	N/A	N/A
mIncept	99.81	N/A	GPU: 2 x NVIDIA Tesla K40c
Random Forests	96.14	N/A	N/A
LDA	95.68	N/A	N/A
Improved VGG	99.00	23h/dataset	CPU: I7 GPU: 1 x Quadro K2200 GPU
HLSGD	99.65	>7h/dataset	CPU: I7-3960X GPU: 2 x Tesla C2075
Residual convolutional blocks	99.33	2.03h/dataset	GPU: 8 x GTX 1070
BAGAN	96.75	N/A	N/A
DCGAN-PILAE	99.80	2.2h/dataset (12m for PILAE)	CPU: i7-6800k GPU: 1 x GTX1080

11) BAGAN [59]: It based on using samples generated by GAN model for minority class to overcome the problem of imbalanced datasets.

Table 2 provides an accurate recognition of the proposed methods. In essence, table 2 shows that the second position is achieved due to high recognition rate of the mIncept framework. This occurs since the mIncept framework has a recognition rate that is 0.01% higher than proposed method. This leads to a configuration that is higher but does not provide information about the training time. In this regard, the Committee of CNNs and improved VGG methods uses configuration that is lower that the GPU's RAM. Furthermore, the configuration helps to achieve the training time using the ratio between the proposed method and the training time of the committee of CNNs and the improved VGG. The training time for the Committee of CNNs and the Improved VGG was 1:16 and 1:10 respectively. The GPU used in the proposed method has 2GB more than the 4 GPUs used in the Committee of CNNs method and 4GB to the one used in the Improved VGG framework and does not degrade their training time. The Single CNN with 3 STN technique had 20 minutes lower than proposed method and used a GPU that has similar RAM size with the one we used. The main limitation of this method is that the best architecture is determined after performing many experiments with different values of the parameters. The HLSGD method configuration used 2 GPUs with 12 GB RAM. The results show that its performance was below the proposed one since it had lower recognition accuracy and training time. Hence, this shows that the main limitation of the method is that it uses pixels or structures of the traffic sign. The traffic sign is based on the large amount of traffic samples leaving to high training and classification time that cannot meet the requests of intelligent driving systems. The residual convolutional blocks method has a training time that is 17 minutes lower than the proposed method. Moreover, it has the highest configuration between all the reported methods that use 8 GPUs. The others methods used include K-d trees, Multiscale CNN, Random forest, LDA and BAGAN. These methods do not have information about their configuration and hence it is difficult to compare them in terms of training time. In this regard, one expects that the methods use handcrafted features such as LDA, K-d tree, and Random Forest. This helps to get training time that is less than the proposed method of running the configuration but less than the recognition rate. The more GPUs means that less training time is achieved [60]. However, this does not affect the recognition accuracy since it is based on the model's design.

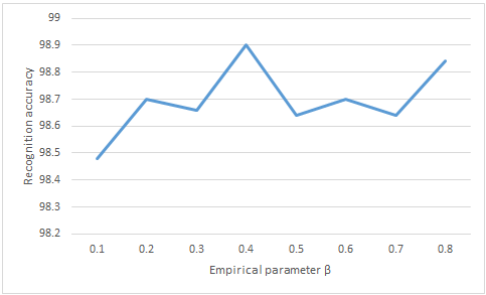

FIGURE 5. Recognition accuracy curve with regard to β in PILAE classifier on GTSRB.

In the PILAE classifier figure 5 the effect of β values concerning the accuracy of recognition on GTSEB dataset given $\beta = .4$. By examining the accuracy curve, It is evident that there are small changes as the parameters of β vary. This implies that, the empirical parameter β necessitates little influence on recognition performance. We compute the ratio between rank and the input data dimension in each layer in figure 6, where the error curve on dataset illustrated in figure 7 which normalized for the intent of presentation. From figure 7 it can be identified that the test error reduces while the network goes deeper. Finally, we can see that the proposed model DCGAN-PILAE achieves the balance between recognition rate and computation complexity.

B. ANALYSIS ON BTSC DATASET

On BTSC dataset there are 3 methods include Residual convolutional blocks [58], Single CNN with 3 STNs [56], and VGG-16 [61] which based on transfer learning using VGG-16 model where Genetic Algorithm (GA) is used for

TABLE 3. BTSC dataset comparison of proposed architecture and other reported results.

Method	Recognition rate	Training time	Configuration
Residual convolutional blocks	99.17	23m/dataset	GPU: 8 x GTX 1070
Single CNN with 3 STNs	98.87	1.29h/dataset	CPU: i7-6700k GPU: 1 x GTX1070
VGG-16	99.16	19.5h/dataset	CPU: i7-6850K GPU: 1 x GTX 1080 Ti
DCGAN-PILAE	99.72	33m/dataset (3m for PILAE)	CPU: i7-6800k GPU: 1 x GTX1080

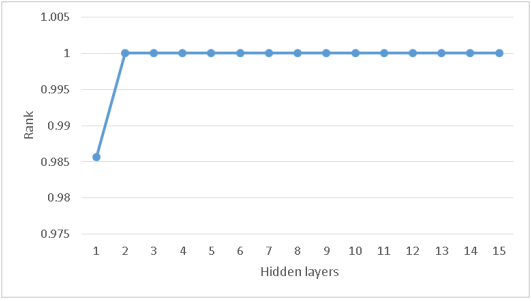

FIGURE 6. The rank ratio as the model layers increased.

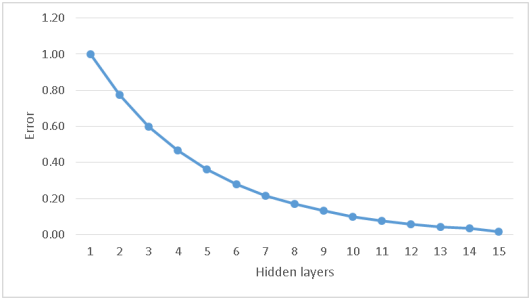

FIGURE 7. The error as the model layers increased.

discovering the optimal parameters of epochs number and the learning rate value. In table 3 recognition accuracy evaluation between reported results and the proposed method are illustrated. It is seen that the proposed method achieves the first position between the all the other 3 methods in conditions of training time and recognition accuracy with take into account the configuration of compared methods.

VII. DISCUSSION

From the above results over the two datasets GTSRB and BTSR we can investigate that the proposed method gets the balance between computation complexity and recognition rate. The DCGAN model are used for extract features from D network. We use the fourth layer from D network which is the top layer before output. The intuition is that these features are linearly separable because of the top layer is just a logistic regression. In general, the highest-level features are extracted in last convolution layer [62], [63].

The above experiments have proved that the suggested combination features leaned by DCGAN and PILAE supplemented by the softmax classifier method attains an excellent performance of recognition as compared to handcrafted features and other methods that are DNN based. However the handcrafted image feature extraction methods have shown to be sufficient recognition systems, a significant disadvantage is they are domain specific. For every kind of source data, there have to be an alternative group of features defined, proving it's important to possess a little knowledge of the data characteristics. More so, there's the disadvantage of not always having clarity of the features that will work best. Feature selection is usually made via empirical analysis of different combinations of features or with aid from feature selection algorithms, in our case however DCGAN is utilized as it could extract the relevant features in an unsupervised way and also without needing an expert analysis of the learning process.

The following reasons make PILAE possess a fast training time: 1) it doesn't require fine-tuning; 2) PILAE weights can be analytically identified, unlike in traditional autoencoders where iterative algorithms are essentially required; and 3) it learns to signify features through singular values unlike autoencoders where the representation of data is learned. Also the high classification ratios in suitable processing times, enforced by the fact that our model has been executed effectively using GPUs, the utilization of GPUs allow us to realize the full probable of DCGAN approach for feature extraction followed by PILAE structure. The proposed method thus gets the balance better between computational performance and recognition ability as compared to other classical methods.

VIII. CONCLUSION

A TSR innovative method has been proposed in the paper and one in which DCGAN functions as an extraction feature. A PILAE classifier is then trained on the extracted features of DCGAN. The deep features can therefore be fully tuned with the PILAE classifier generalization performance, occasioning in a recognition accuracy that is satisfying without the need of extra and complicated DCGAN framework. More so with other techniques, the DCGAN-PILAE method can get perfect results with a forthright structure which alleviates training process which is usually time consuming. Future work should be including PILAE development to solve the GTSRB dataset imbalance problem without needing any additional methods and extending our method to traffic sign detection.

REFERENCES

[1] A. Møgelmose, M. M. Trivedi, and T. B. Moeslund, "Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 4, pp. 1484–1497, Dec. 2012.

[2] S. Maldonado-Bascón, S. Lafuente-Arroyo, P. Gil-Jiménez, H. Gómez-Moreno, and F. López-Ferreras, "Road-sign detection and recognition based on support vector machines," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 2, pp. 264–278, Jun. 2007.
[3] A. Ruta, Y. Li, and X. Liu, "Robust class similarity measure for traffic sign recognition," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 4, pp. 846–855, Dec. 2010.
[4] F. Zaklouta, B. Stanciulescu, and O. Hamdoun, "Traffic sign classification using K-d trees and random forests," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, 2011, pp. 2151–2155.
[5] C. Vondrick, A. Khosla, T. Malisiewicz, and A. Torralba, " HOGgles: Visualizing object detection features," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1–8.
[6] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, vol. 1, no. 1, pp. 886–893.
[7] Y. Bengio, "Learning deep architectures for AI," *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, 2009.
[8] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.
[9] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
[10] P. Sermanet and Y. LeCun, "Traffic sign recognition with multi-scale convolutional networks," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, 2011, pp. 2809–2813.
[11] C. Dan, U. Meier, J. Masci, and J. Schmidhuber, "A committee of neural networks for traffic sign classification," in *Proc. Int. Joint Conf. Neural Netw.*, 2011, pp. 1918–1921.
[12] D. Ciresan, U. Meier, J. Masci, and J. Schmidhuber, "Multi-column deep neural network for traffic sign classification," *Neural Netw.*, vol. 32, pp. 333–338, Aug. 2012.
[13] J. Jin, K. Fu, and C. Zhang, "Traffic sign recognition with hinge loss trained convolutional neural networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 5, pp. 1991–2000, Oct. 2014.
[14] K. Wang, P. Guo, X. Xin, and Z. Ye, "Autoencoder, low rank approximation and pseudoinverse learning algorithm," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2017, pp. 948–953.
[15] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*. [Online]. Available: https://arxiv.org/abs/1511.06434
[16] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng, "Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 3361–3368.
[17] H. Lee, C. Ekanadham, and A. Y. Ng, "Sparse deep belief net model for visual area V2," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 873–880.
[18] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 609–616.
[19] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Unsupervised learning of hierarchical representations with convolutional deep belief networks," *Commun. ACM*, vol. 54, no. 10, pp. 95–103, Oct. 2011.
[20] H. Larochelle, D. Erhan, A. Courville, J. Bergstra, and Y. Bengio, "An empirical evaluation of deep architectures on problems with many factors of variation," in *Proc. 24th Int. Conf. Mach. Learn.*, 2007, pp. 473–480.
[21] J. Zhang, L. Chen, L. Zhuo, X. Liang, and J. Li, "An efficient hyperspectral image retrieval method: Deep spectral-spatial feature extraction with DCGAN and dimensionality reduction using t-SNE-based NM hashing," *Remote Sens.*, vol. 10, no. 2, p. 271, 2018.
[22] Z. Zhang, X. Liu, and Y. Cui, "Multi-phase offline signature verification system using deep convolutional generative adversarial networks," in *Proc. 9th Int. Symp. Comput. Intell. Design (ISCID)*, vol. 2, 2016, pp. 103–107.
[23] Z. Gu, "An overview and comparative analysis on major generative models," 2018. [Online]. Available: http://acsweb.ucsd.edu/~zig021/overview-comparative-analysis.pdf
[24] P. Guo and M. R. Lyu, "A pseudoinverse learning algorithm for feedforward neural networks with stacked generalization applications to software reliability growth data," *Neurocomputing*, vol. 56, pp. 101–121, Jan. 2004.
[25] J. Wang, P. Guo, and X. Xin, "Review of pseudoinverse learning algorithm for multilayer neural networks and applications," in *Proc. Int. Symp. Neural Netw.* Cham, Switzerland: Springer, 2018, pp. 99–106.
[26] M. Takaki and H. Fujiyoshi, "Traffic sign recognition using SIFT features," *IEEJ Trans. Electron. Inf. Syst.*, vol. 129, no. 5, pp. 824–831, 2009.
[27] A. Ihara, H. Fujiyoshi, M. Takaki, H. Kumon, and Y. Tamatsu, "Improvement in the accuracy of matching by different feature subspaces in traffic sign recognition," *IEEJ Trans. Electron. Inf. Syst.*, vol. 129, no. 5, pp. 893–900, 2009.
[28] Z.-L. Sun, H. Wang, W.-S. Lau, G. Seet, and D. Wang, "Application of BW-ELM model on traffic sign recognition," *Neurocomputing*, vol. 128, pp. 153–159, Mar. 2014.
[29] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The German traffic sign recognition benchmark: A multi-class classification competition," in *Proc. Int. Joint Conf. Neural Netw.*, 2011, pp. 1453–1460.
[30] G. Wang, G. Ren, Z. Wu, Y. Zhao, and L. Jiang, "A hierarchical method for traffic sign classification with support vector machines," in *Proc. Int. Joint Conf. Neural Netw.*, 2013, pp. 1–6.
[31] J. Greenhalgh and M. Mirmehdi, "Real-time detection and recognition of road traffic signs," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 4, pp. 1498–1506, Dec. 2012.
[32] K. Lu, Z. Ding, and S. Ge, "Sparse-representation-based graph embedding for traffic sign recognition," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 4, pp. 1515–1524, Dec. 2012.
[33] M. Mathias, R. Timofte, R. Benenson, and L. Van Gool, "Traffic sign recognition—How far are we from the solution?" in *Proc. Int. Joint Conf. Neural Netw.*, 2013, pp. 1–8.
[34] X. Yuan, X. Hao, H. Chen, and X. Wei, "Robust traffic sign recognition based on color global and local oriented edge magnitude patterns," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 4, pp. 1466–1477, Apr. 2014.
[35] F. Mariut, C. Fosalau, M. Avila, and D. Petrisor, "Detection and recognition of traffic signs using Gabor filters," in *Proc. Int. Conf. Telecommun. Signal Process.*, 2011, pp. 554–558.
[36] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 3360–3367.
[37] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: An astounding baseline for recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR) Workshops*, Jun. 2014, pp. 512–519.
[38] H. Azizpour, A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson, "From generic to specific deep representations for visual recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2015, pp. 36–45.
[39] M.-E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *Proc. 6th Indian Conf. Comput. Vis., Graph. Image Process.*, 2008, pp. 722–729.
[40] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-UCSD birds-200-2011 dataset," California Inst. Technol., 2011.
[41] A. Quattoni and A. Torralba, "Recognizing indoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 413–420.
[42] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. (2012). *The Pascal Visual Object Classes Challenge 2012 (VOC2012)*. [Online]. Available: http://www. pascal-network.org/challenges and http://VOC/voc2012/workshop/index.html
[43] Y. Zhu, C. Zhang, D. Zhou, X. Wang, X. Bai, and W. Liu, "Traffic sign detection and recognition using fully convolutional network guided proposals," *Neurocomputing*, vol. 214, pp. 758–766, Nov. 2016.
[44] M. Haloi, "Traffic sign classification using deep inception based convolutional networks," 2015, *arXiv:1511.02992*. [Online]. Available: https://arxiv.org/abs/1511.02992
[45] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. pattern Recognit.*, Jun. 2015, pp. 1–9.
[46] M. Jaderberg, K. Simonyan, A. Zisserman, and K. kavukcuoglu, "Spatial transformer networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2017–2025.
[47] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
[48] P. Guo, M. R. Lyu, and N. E. Mastorakis, "Pseudoinverse learning algorithm for feedforward neural networks," in *Proc. Adv. Neural Netw. Appl.*, Puerto De La Cruz, Spain, Feb. 2001, pp. 321–326.

[49] T. L. Boullion and P. L. Odell, *Generalized Inverse Matrices*. Hoboken, NJ, USA: Wiley, 1971.
[50] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. 25th Int. Conf. Mach. Learn.*, 2008, pp. 1096–1103.
[51] P. Guo, M. R. Lyu, and C. L. P. Chen, "Regularization parameter estimation for feedforward neural networks," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 33, no. 1, pp. 35–44, Feb. 2003.
[52] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "Man vs. Computer: Benchmarking machine learning algorithms for traffic sign recognition," *Neural Netw.*, vol. 32, pp. 323–332, Aug. 2012.
[53] R. Timofte and L. Van Gool, "Sparse representation based projections," in *Proc. 22nd Brit. Mach. Vis. Conf. (BMVC)*, 2011, pp. 1–12.
[54] A. Vedaldi and K. Lenc, "MatConvNet: Convolutional neural networks for MATLAB," in *Proc. 23rd ACM Int. Conf. Multimedia*, 2015, pp. 689–692.
[55] B. SungHo. *Deep Convolutional Generative Adversarial Network (DCGAN) Implementation on MatConvNet*. Accessed: Feb. 19, 2019. [Online]. Available: https://github.com/sunghbae/dcgan-matconvnet
[56] Á. Arcos-García, J. A. Álvarez-García, and L. M. Soria-Morillo, "Deep neural network for traffic sign recognition systems: An analysis of spatial transformers and stochastic optimisation methods," *Neural Netw.*, vol. 99, pp. 158–165, Mar. 2018.
[57] S. Zhou, W. Liang, J. Li, and J. U. Kim, "Improved VGG model for road traffic sign recognition," *Comput., Mater. Continua*, vol. 57, no. 1, pp. 11–24, 2018.
[58] S. Saha, S. A. Kamran, and A. S. Sabbir, "Total recall: Understanding traffic signs using deep hierarchical convolutional neural networks," 2018, *arXiv:1808.10524*. [Online]. Available: https://arxiv.org/abs/1808.10524
[59] G. Mariani, F. Scheidegger, R. Istrate, C. Bekas, and C. Malossi, "BAGAN: Data augmentation with balancing GAN," 2018, *arXiv:1803.09655*. [Online]. Available: https://arxiv.org/abs/1803.09655
[60] H. Mikami, H. Suganuma, P. U-Chupala, Y. Tanaka, and Y. Kageyama, "Massively distributed SGD: ImageNet/ResNet-50 training in a flash," 2018, *arXiv:1811.05233*. [Online]. Available: https://arxiv.org/abs/1811.05233
[61] A. Jain, A. Mishra, A. Shukla, and R. Tiwari, "A novel genetically optimized convolutional neural network for traffic sign recognition: A new benchmark on Belgium and Chinese traffic sign datasets," *Neural Process. Lett.*, pp. 1–25, Feb. 2019.
[62] J. S. J. Ren, W. Wang, J. Wang, and S. Liao, "An unsupervised feature learning approach to improve automatic incident detection," in *Proc. 15th Int. IEEE Conf. Intell. Transp. Syst.*, Sep. 2012, pp. 172–177.
[63] B. Athiwaratkun and K. Kang, "Feature representation in convolutional neural networks," 2015, *arXiv:1507.02313*. [Online]. Available: https://arxiv.org/abs/1507.02313

MOHAMMED A. B. MAHMOUD received the B.S. and M.S. degrees in computer science from the Department of Mathematics, Faculty of Science, Assiut University, Egypt, in 2007 and 2014, respectively. He is currently pursuing the Ph.D. degree in pattern recognition and deep learning with the School of Computer Science and Technology, Beijing Institute of Technology, China. His research interests include pattern recognition, deep learning, and computational intelligence.

PING GUO (SM'05) received the B.S. and M.S. degrees from Peking University, Beijing, China, in 1980 and 1983, respectively, and the Ph.D. degree from The Chinese University of Hong Kong, Hong Kong, in 2002.

He is currently with the School of System Science, Beijing Normal University, Beijing, where he is the Founding Director of the Image Processing and Pattern Recognition Laboratory. He is also an Adjunct Professor with the School of Computer Science, Beijing Institute of Technology, Beijing. His research interests include computational intelligence, image processing, pattern recognition, and astronomy big data analysis systems and applications.

Dr. Guo was a recipient of the Science and Technology (Third Rank) Award of 2012 Beijing Peoples Government for his contributions to studies of regularization method and their applications.