Multimedia Tools and Applications
https://doi.org/10.1007/s11042-023-14959-0

Switching network for mixing experts with application to traffic sign recognition

Amir Ahangi[1] · Rico Möckel[1]

Received: 11 April 2022 / Revised: 28 September 2022 / Accepted: 22 February 2023

Abstract

The correct and robust recognition of traffic signs is indispensable to self-driving vehicles and driver-assistant systems. In this work, we propose and evaluate two network architectures for multi-expert decision systems that we test on a challenging Traffic Sign Recognition Benchmark dataset. The decision systems implement individual experts in the form of deep convolutional neural networks (CNNs). A gating network CNN acts as final decision unit and learns which individual expert CNNs are likely to contribute to an overall meaningful classification of a traffic sign. The gating network then selects the outputs of those individual expert CNNs to be fused to form the final decision. In this work we study the advantages and challenges of the proposed multi-expert architectures that in comparison to other network architectures allow for parallel training of individual experts with reduced datasets. Under the challenging conditions introduced by the benchmark dataset, the demonstrated multi-expert decision systems achieve a recognition performance that is superior to those of humans: with an accuracy of 99.10%, when training experts with the complete dataset and 98.94%, when individual experts are only trained with 36% of the training samples. Overall, our approach ranked fourth on the list of the applied approaches proposed for the German traffic sign Recognition Benchmark (GTSRB) dataset.

Keywords Traffic sign · Ensemble learning · Mixture of experts · Deep learning

1 Introduction

Road traffic accidents are a leading health hazard in transportation [10]. The cause of many road accidents is disobeying traffic signs such as speed limits. It is imperative to reduce road

✉ Amir Ahangi
amir.ahangi@maastrichtuniversity.nl

Rico Möckel
rico.mockel@maastrichtuniversity.nl

[1] Department of Advanced Computing Sciences (DACS), Maastricht University, Maastricht, The Netherlands

Published online: 25 April 2023

traffic accidents, for instance with the help of Driver Assistant Systems (DAS) [27] that e.g., make drivers aware of traffic signs they might have overlooked. To achieve the robust application of DAS, automatic recognition of traffic signs under realistic conditions must be achieved [30]. Unfortunately under realistic conditions a variety of factors can lead to poor image quality thus making the automatic recognition challenging. Such realistic factors include weather, intensity and direction of light, image angle and contrast [33]. DAS must provide reliable traffic sign recognition under those conditions if DAS should fulfill the promise to reduce road traffic accidents. Thus, to compare the performance of different traffic sign recognition approaches, the German Traffic Sign Recognition Benchmark (GTSRB) dataset was created [28]. The GTSRB dataset, that is described in more detail in Section III, contains more than 50,000 images of different quality, taken under various conditions. As it provides realistic and challenging conditions for image recognition, we use the GTSRB dataset to test the performance of our proposed multi-expert decision system architectures that implement individual experts in form of deep convolutional artificial neural networks.

Artificial neural networks (ANNs) play an essential role in modern machine learning and image recognition systems. Neural networks stem from our understanding of the nervous system's working and simulate neural models in artificial intelligence. These models are suitable to address primarily data-related issues in machine learning [23]. From the general class of ANNs, Deep Neural Networks (DNNs) have been shown to be particularly suitable for extracting features and classifying complex datasets [17].

This paper proposes an architecture where individual DNNs are trained and combined into a joint decision system. Single ANNs tend to learn from a single perspective only. However, multi-expert systems learn from different perspectives, which are various architectures, other initial parameters, and different types of neural networks. Joint decision systems are a subset of ensemble learning where the opinion of several experts is combined to form an accurate decision. The ensemble method [7, 15, 21, 25, 36] in our study is a model which recognizes traffic signs by mixing the opinion of all DNN experts [1]. We study how best to combine individual experts' output (DNNs) to achieve maximum accuracy in recognizing traffic signs. Like Ciresan et al. [6] we use Convolutional Neural Networks (CNNs), which use convolutional kernels mimicking the connectivity pattern of neurons in the animal visual cortex, to propose a multi-column DNN approach for the German traffic sign dataset. The basis of the proposed approach is Mixture of Experts (ME) [6]. ME [20] and Mixture of Active Experts (MAE) [2] are two ensemble methods that were applied to the German traffic sign dataset. ME is composed of many experts and one Gating Network (GN) that combines the outputs of individual experts (DNNs) into a single decision. By using GN, weights can be assigned to each expert. Thus through a GN the experts that are most trusted in recognizing a specific traffic sign or a traffic sign under specific conditions can receive the highest weights and thus have the highest impact on the final combined decision of all experts. After weighting individual expert outputs, the final result can be calculated e.g. by averaging the weighted outputs of all experts [2].

Rasti et al. [24] used CNNs in the mixture of experts algorithm for the first time to which the structure of our proposed approach is similar. They introduced an ME algorithm focused on breast cancer and found a way for distinguishing benign and malignant tumors using a mixture of CNNs. The mixture of CNNs is an ensemble method that is composed of a combination of CNNs and a GN to assign weights to experts. The CNN has three types of layers: Convolutional layers, sub-sampling (max-pooling) layers, and output layers (fully-connected layer or last layer). These layers organize a feed-forward structure where a max-pooling layer is always placed after a convolutional layer. Moreover, Rasti et al. [24] addressed ave-ensemble and soft-ensemble methods. Ave-ensemble is a method used to train many CNNs with ground-truth labels and soft-ensemble is a method to make many branches of CNNs and combines the output of branches with a fully-connected layer in addition to a softmax layer [24].

The proposed and studied approach in our work composes multi-expert decision systems out of three central units: a number of individual experts implemented in form of CNNs, a Switching System (SS), and a GN. Two types of GNs are being compared. SS and GN facilitate the implementation of the joint decision system by selecting influencers in the decision-making process. Influencers are those experts that impact the decision-making process. For these influencers high recognition accuracy is expected. Based on training samples the GN learns how individual experts' output should be combined by the SS depending on a received image. The SS implements the combination of experts by allowing experts to contribute to the joined decision.

The novelty of this work is in the exploration of the integration of Switching Systems with Gating Networks for mixing experts. Moreover, we explore specializing experts and reducing overall training time by training individual experts on subsets of the total training set only. Two architectures are proposed and evaluated: a Switching Network (SN) and a Switching ResNet Network (SRNN). Their recognition performance is evaluated on the German traffic sign dataset and compared to a single expert CNN as well as to a simple Averaging Network (AN) that averages the outputs of experts without a situation-depended weighting or selection scheme.

With this work we focus on the following challenges and innovations:

(1) We explore a new way of mixing expert opinions through the integration of Switching Systems with Gating Networks. Two architectures are proposed and evaluated to fuse the opinion of individual experts: a Switching Network (SN) and a Switching ResNet Network (SRNN). Their recognition performance is evaluated on the German traffic sign dataset and compared to a single expert CNN, to an Averaging Network (AN) that averages the outputs of experts without a situation-depended weighting or selection scheme as well as to other approaches that have been evaluated on the German traffic sign dataset in the literature.

(2) Furthermore with the newly proposed multi-expert architectures we explore how to train experts in parallel in such a way that each expert is only trained on a subset of the overall dataset while we maintain an overall high decision accuracy of the combined expert opinions. We explore if we can train individual experts on subsets of the overall dataset to reduce training load of individual experts and to train experts in parallel without the need of sharing all data among all experts. The latter is particularly interesting when training data is sensitive and should not be shared among all experts. We demonstrate that we can run the training of experts in parallel with reduced training sets where the accuracy still improved in comparison to other methods training CNNs with all available data.

The remainder of the paper is organized as follows: In Section 2, we review related works in ensemble methods. In Section 3, we describe the dataset and proposed methods. In Section 4, we demonstrate the experimental results and compare the proposed method with other methods. Section 5 concludes this contribution.

2 Related work

The use of Gating Networks (GN) as part of a ME algorithm was first proposed by Jacobs et al. in 1991 [14]. Ahangi et al. [2] introduced a new version of ME, which is called MAE, that makes use of an active learning technique for training experts individually. This advanced approach was applied to the German traffic sign dataset in 2019 [2]. In our work we propose a method that is an adapted version of these two works, where we changed the structure of GN to add a SN module. In MAE, active experts were proposed to reduce the load of experts. We however use a data parallelism approach to distribute training samples to experts. In our proposed system, the load of experts (in terms of the number of training samples that an individual expert has to process during training) is reduced more than in the MAE approach while the accuracy of our proposed approach is higher than MAE. MAE used a GN module to assign weights to the Multi-Layer Perceptrons (MLPs). This module is similar to our GN module but with a considerable change. In our study, the functionality of the GN is changed from a weight assigner to a boss or an expert selector that chooses experts (or influencers) in which the selector is highly confident, instead of assigning weights. This new form of the GN helps to integrate the data parallelism during training in our proposed approach to reduce the training load of experts by training experts with reduced datasets.

From another view, the MAE algorithm is based on MLP. This reduces the number of required computations of the MAE in comparison to a CNN architecture when processing data and allows to reach a fast reaction in the decision-making process. However, the CNN that we employ provide higher data classification accuracy. The committee of CNNs [6], which is close to our proposed approach, is also CNN-based. Its accuracy has been reported to rank first in the German traffic sign competition [6]. (Later, after the competition, two other works have been published that reported even higher performance [11] and [3]). However, the high training time for the committee of CNNs is quite demanding because of it's structure with 90 million parameters and 25 experts. We used the committee of CNNs idea to integrate multi-CNNs into our switching network. In our proposed approach however, we only use five experts with a total number of parameters in our model of only 11.2 million compared to the proposed committee of CNNs. Therefore, our proposed approach is a solution to reach an acceptable accuracy similar to the committee of CNNs while it is a cost-effective model similar to MAE.

Our approach generates decisions by fusing the opinion of individual experts. Hence, we reviewed similar works to identify promising fusion strategies. Zhang et al. [35] e.g., proposed a deep ensemble method, named multicontext networks, to address monaural speech separation. Monaural speech separation aims at removing background noise from the target speaker's voice [35]. One proposed multicontext network by Zhang et al. averages the outputs of multiple DNNs to provide a combined output. Wang et al. applied an ensemble approach on a wind dataset. They as well used an average method to compute the output of the ensemble system [32] to anticipate probabilistic wind power. Maji et al. proposed an ensemble of DNNs for enhancing the accuracy in detecting retinal vessels. The ensemble outputs the average of 12 DNNs [19].

Not only CNNs, combining different types of experts has been used previously [5]. For instance, a handwritten character classification used CNNs as an expert in the ensemble system to classify different categories in the handwritten dataset [5]. Choosing a fusion approach is necessary to design a high-accuracy model. The averaging is a common fusion method in the multi-expert decision systems. Deng et al. [8] classified a speech recognition dataset with an ensemble learning approach. Cireşan et al. addressed the multi-column DNNs that output is the average of the DNN opinions, but the main difference is the way of

Table 1 The accuracy of single CNN and proposed approaches

Approaches	*Accuracy*	*SD*	*Experts*	*The number of samples*
SN	**99.10**	**0.05**	**5**	**39209**
SRNN	98.94	0.09	5	14186
SN	98.88	0.05	5	14186
SRNN	98.87	0.32	5	39209
AN	98.76	0.10	5	14186
Single CNN	98.63	0.39	1	39209

combining the experts [4]. Huang et al. addressed a different ensemble method, called convolutional deep belief networks [13]. Sun et al. suggested an ensemble method that deviates slightly from standard combination methods. Sun et al. [31] recommended, this difference is using DNNs in two levels by adding a Restricted Boltzmann Machine (RBN) model [31].

3 Material and methods

3.1 Dataset for traffic sign recognition

The German Traffic Sign Recognition Benchmark (GTSRB) [28] that was used to test our proposed approach is a traffic sign dataset provided by the Institut für Neuroinformatik of the Ruhr-University Bochum in Germany. The dataset was originally created as part of a public competition on traffic sign recognition. The first competition took place in 2010, but the dataset is still continuously used as a benchmark for newly developed algorithms. In this paper, we compare the performance of our approach with the finalists of the GTSRB competition (Table 1). The GTSRB dataset is a multi-class dataset with more than 50,000 images and 43 classes of traffic signs. The images have different qualities because of environmental impacts such as rainy or sunny weather and fog, light intensity, image resolution, and camera angle. The number of images is different per class and the images vary in size from 15*15 to 250*250 pixels. The number of training and test samples is 39,209 and 12,630, respectively. Tuning the parameters of experts (CNNs) and gating network is the challenge addressed in our approach. To set these parameters, we used 30% of the training set as the validation set to tune the dropout, regularization and the percentage of overlap parameters. We used a 10-fold cross-validation approach to find the number of epochs. In Fig. 1, a sample images of the GTSRB traffic sign dataset with all 43 classes are shown.

3.2 Proposed network architectures

Figure 2 provides an overview of the two network architectures newly proposed and studied in this paper: (1) a multi-expert Switching Network (SN, Fig. 2.c) and (2) a multi-expert Switching ResNet Network (SRNN, Fig. 2.d). Both network architectures are composed of a variable number of expert CNNs, a switching system, and a GN.

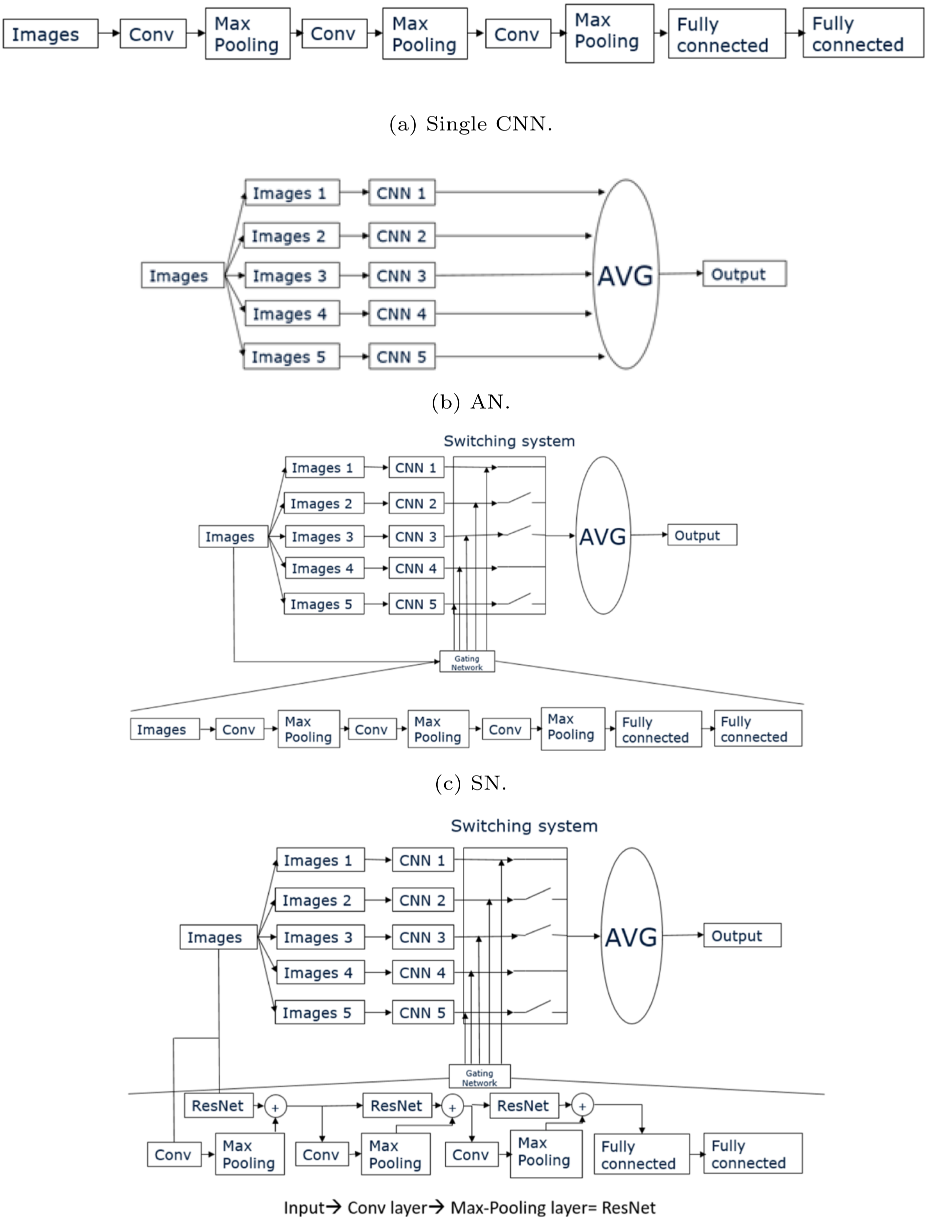
Network architectures with 5 expert CNNs have been tested in our study. An architectural overview of these individual expert CNNs is given in Fig. 2.a: three blocks of convolutionary and max-pooling layers are sequentially combined. The final network outputs are processed

Fig. 1 Traffic sign dataset

by two layers of fully interconnected neurons. The architecture of the experts and GN is shown in Tables 2 and 3. The total number of parameters in our proposed approach with five experts and one GN is above 11,270,000. This network topology (i.e. the type of layers, number of layers and neurons) is taken from D. Cireşan et al. [4] so that the accuracy of a single expert CNN can be compared to the accuracy of the recognition systems proposed by other authors that entered the German Traffic Sign Recognition competition.

Like expert CNNs, the GN is trained on the images from the benchmark dataset. However, in contrast to the individual expert CNNs, the GN is not trained to recognize the correct traffic sign, but to select the outputs of those expert CNNs to pass through the switching network that are expected to be best suited to correctly recognize a certain traffic sign. In that respect the GN acts like a boss in a group meeting of experts, influencing the choice of the team of experts by choosing those experts to combine their opinion that the GN beliefs to be best suited given a certain situation, here an image of a traffic sign. The outputs of all selected expert CNNs are then averaged (AVG) to form a combined opinion. Part of the novelty in this paper lies in the exploration of the particular type of switching network. Two types of GNs have been studied. SN (Fig. 2.c) uses as GN a similar architecture as the individual expert CNNs (Fig. 2.a). SRRN (Fig. 2.d) uses a ResNet network architecture in which a ResNet layer [12] is added to each max-pooling layer. The plus sign in the figure indicates this procedure. Through the ResNet, the output of the previous layer is added to the output of the current layer to feed the next layer. ResNet layers usually work well when the number of layers is sufficiently high. Adding this layer to the shallow network is worth examining in order to evaluate the performance of our proposed approach. By adding the ResNet layers, we want to review the effect of those layers in the gating network.

To better study and compare the effect of the switching network, a multi-expert baseline model without switching system architectures has been introduced to which we compare the performance of the SN and SRNN: Fig. 2 shows the architecture of this baseline multi-expert system without switching system (AN, Fig. 2.b). Unlike the proposed SN and SRNN architectures, this AN averages the outputs of all expert CNNs without any situation-depended selection of weighting scheme. Furthermore, to demonstrate the effect of the multi-expert systems, we compare the SN and SRNN results to the performance of a single fully-trained expert CNN.



(a) Single CNN.

(b) AN.

(c) SN.

(d) SRNN.

Fig. 2 Structure of a single CNN and three proposed approaches: a) A single CNN composed of eight layers: three convolutional layers (named conv), three max pooling, and two fully connected layers. b) Averaging Network (AN) that averages the outputs of 5 CNNs. c) Switching Network (SN) that uses a switching system to allow only the outputs of certain selected CNNs to be averaged. The decision which CNNs are selected contribute to the final network output is made based on the preceived image by a Gating Network which has a structure similar to the CNN shown in a). d) Switching ResNet Network (SSRN) with a structure almost identical to the SN in c) except for the ResNet layers that are added to its gating network

Table 2 The architecture of experts

	Layer	*Output shape*	*Number of parameters*
1	Input	3*48*48	0
2	Conv2d	100*42*42	14,800
3	MaxPool2d	100*21*21	0
4	Conv2d	150*18*18	240,150
5	MaxPool2d	150*9*9	0
6	Conv2d	250*6*6	600,250
7	MaxPool2d	250*3*3	0
8	Fully connected	300	675,300
9	Fully connected	43	12,943

3.3 Parallel training of CNN experts and GN

In this work, we study the effect of parallel training of domain experts, represented by individual CNNs. Such parallel training is useful to distribute the training of experts over distributed computing facilities e.g. to reduce the overall training time. Parallel and independent training of domain experts is also useful for federated learning [18] when data cannot be shared but the individual experts must be trained by different end users.

As baseline to compare our results on parallel expert training in AN, SN, and SRNN architectures, we trained the single expert CNN (Fig. 2a) with all available 39209 data samples from the GTSRB traffic sign dataset.

The switching network architecture, that we propose, is a new ensemble method, which allows combining the opinion of several expert CNNs that can be trained in parallel. For parallel training, smaller training sets are generated for each expert by drawing samples randomly from the overall training set without repetition. Additionally, an overlap training set, that is shared by multiple but not by all experts, is added to each of these sets. For the parallel decentralized training these overlap training sets make an interconnection between the local training sets for individual experts. Adaboost and Bagging strategies [21] use a similar strategy. The Adaboost uses a strategy to find hard samples for each iteration and adds them to the training set of the next expert. As a result in contrast to our proposed

Table 3 The architecture of Gating Network(GN)

	Layer	*Output shape*	*Number of parameters*
1	Input	3*48*48	0
2	Conv2d	100*48*48	2,800
3	MaxPool2d	100*24*24	0
4	Conv2d	150*25*25	240,150
5	MaxPool2d	150*12*12	0
6	Conv2d	250*13*13	600,250
7	MaxPool2d	250*6*6	0
8	Fully connected	300	2,700,300
9	Fully connected	32	9,632

approach, the Adaboost method is not parallelizable. The bagging method uses only a fully random mechanism to make training sets. In contrast, our approach allows to and guarantees to have training samples in each training set which are not existing in the training sets of other experts while we can control which training samples are shared between experts.

For training the individual expert CNNs in the AN, SN, and SRNN multi-experts systems, the dataset was divided into 5 subsets, one for each of the 5 experts. Each individual expert CNN was trained with data from one of those subsets plus some random samples from the neighbor expert's subset.

The separation of training data reassembles the situation where individually trained domain expert cooperate to find a common conclusion. The separation of training data is in contrast to the approach proposed by D. Cireşan et al. [4], who trained all 25 experts in their work [4] with all training samples (39209 images) and scored second place in the GTSRB competition.

Training experts also with random samples from their neighbor expert's subset was done to achieve some overlap of recognition competence of expert CNNs within the group of experts. In Section 4, we show that this overlap of training helps to improve the recognition accuracy of the overall multi-expert system.

A challenge of the parallel training approach with subsets is the training of the GN that selects individual experts through the switching system. The GN module in our research is similar to the GN in the ME method [14] but with significant changes. The experts in ME are trained with all training samples in contrast to our proposed approach. In ME, the GN assigns weights to each expert to calculate the percentage of the contribution of experts. However, the GN in our proposed approach trains how to select the qualified experts for classifying each new test sample. The opinion of qualified experts is fused by an averaging method. In our proposed approach, the experts are trained as specialists or influencers, to which the GN recognizes them. However, in ME, the opinion of all experts is fused.

Diversity ensures that experts provide discriminative outputs. A similarity between our proposed approach and ME is that both try to increase the diversity of experts so that experts can make diverse outputs. Making diverse experts is the main goal of the mixture of experts algorithm to utilize the difference between experts' opinion to improve the effectiveness [16, 21].

To make the best selection of individual experts, ideally the GN is trained with the full dataset. However, if trained with the full dataset, the GN becomes a bottleneck in our parallel training system. The total training time would then be limited by the GN. Hence, we propose a different approach where we feed the GN with the same number of training samples such as individual expert CNNs: To generate the subset for training the GN, an algorithm randomly picks training samples from the datasets of all individual expert CNNs, thus ensuring that the GN has access to at least some data samples across all expert CNNs. The results prove that picking the samples randomly from the training set for feeding the GN is not working better than training the GN with all training samples. However, by training the GN with all training samples, we no longer gain from the parallel training of experts with reduced training sets as the training of the GN will limit the minimum time for training. To resolve this bottleneck, two inner and outer loops based on two epochs are designed to make parallel training possible between GN and experts. An inner loop is added to the algorithm for training experts and an outer loop for the GN epoch. The GN is trained with all training samples in its epoch loop. Meanwhile, the experts are trained in their epoch loop with their sub-training sets as shown in Fig. 3.

Another challenging part is how to provide correct feedback to the GN during training. In the training phase, each expert makes an individual decision based on their input. The

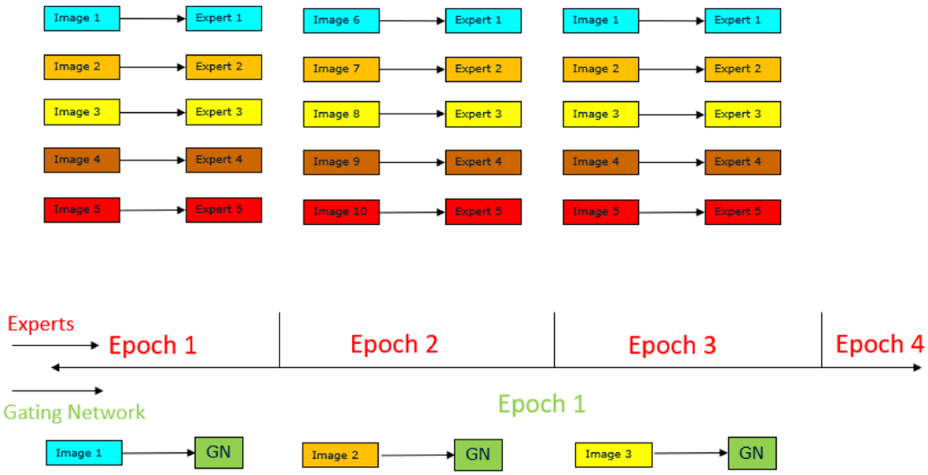


Fig. 3 How to train GN and experts parallelly

output of all experts helps our system to provide training feedback to the GN. To overcome this challenge we propose two approaches considering either the hard or the soft output of experts:

The hard output is a strict approach to just round the numbers to 0 or 1. In contrast, the soft output does not round the values to keep using the actual information provided by the experts. In the first approach, the hard output of experts is considered. The feedback F_{hard} to the GN is calculated as a weighted aggregation of expert outputs O_i: $F_{hard} = \sum_{i=0}^{i=n} O_i \cdot 2^i$. Thus, the feedback to the GN is the output of the first expert $\cdot 1+$ the output of second expert $\cdot 2+$ the output of the third expert $\cdot 4+$ the output of the fourth expert $\cdot 8+$ the output of the fifth expert $\cdot 16$. If the output of experts (experts 1, 2, 3, 4, and 5) is 0, 0, 1, 1, 0 respectively, the output of experts as feedback is $0 \cdot 1 + 0 \cdot 2 + 1 \cdot 4 + 1 \cdot 8 + 0 \cdot 16 = 12$ which means the mix of experts 3 and 5.

When the soft output of experts is considered, the same weighted aggregation function is used. If the soft output of experts is 0.2, 0.1, 0.9, 0.7, 0.3, the feedback of GN is $0.2 \cdot 1 + 0.1 \cdot 2 + 0.9 \cdot 4 + 0.7 \cdot 8 + 0.3 \cdot 16 = 14.4$ which means the mix of experts 2, 3 and 4. According to our experimental results, the second approach (using soft output) works better than the first approach (using hard output). In the test phase, the output of GN is used to switch experts on or off. If GN's output is 10, the mix of experts 2 and 4 is considered due to the weighted aggregation function ($10 = 0 \cdot 1 + 1 \cdot 2 + 0 \cdot 4 + 1 \cdot 8 + 0 \cdot 16$).

The pseudo-code for training and testing the AN, SN, and SRNN is shown in Tables 4 and 5, respectively. Both pseudo-codes (training and test) are slightly similar, but their functionality is different. Dividing the training sets into N subsets is the first step in the training procedure. Next, overlap training subsets are generated from the neighboring subsets and added to the training subsets. The GN trains with samples from all training sets. The switching system selects active experts based on GN's information. The GN uses the soft value of experts to adapt itself with the output of the experts. Therefore, the final output is the average of influencers' opinions.

Initially, the proposed method distributes samples (images) to N sets (line 1). Subsequently, it gives each set to one expert training a convolutional neural network and provides

Table 4 Training phase of the proposed method

```
1. Images1, images2,...,imagesN=distribution(images)
2. For i :1 to the number of samples
3. Output_gating_network=Train_gating_network(Images(i))
4. For j:1 To N
5. Output(j)=Train(Images(j,i))
6. Active_experts=Switching_system(Output, Output_gating_network)
7. For j:1 To N
8. Final_output= Final_output+ Active_experts(j)
```

all samples to the GN (from lines two to five). The switching system then follows the commands that are generated by the GN learned from the output of the experts (line 6). Finally, our proposed method averages the output of the influencers (high-accuracy experts) (lines 7 and 8).

4 Experimental results

We report the results of the SN, SRNN, AN, and single expert CNN on the GTSRB dataset. In the following section, we compare the traffic sign recognition accuracy of all four approaches with each other and also with other approaches in the competition.

To compare and optimize performance, we first tried to identify optimal parameters (regularization, overlap percentage, dropout, and the number of epochs) for the proposed approaches. Initially, the regularization, overlap percentage, and dropout were discovered by using a validation set. Afterwards, the number of epochs parameters for the experts, the regularization and dropout parameters of the GN were found by using a 10-fold cross-validation approach. Then, we compared the result of all approaches based on these parameters. The dropout is a parameter to reduce overfitting. It is tuned during the training phase. The best dropout parameter for the experts was found to be zero as shown in Fig. 4. If the dropout is increased for experts, the accuracy is decreasing. Thirty percent of data was labeled as the validation set and 70 percent as the training set for setting dropout and regularization parameters in our proposed approach.

Regularization is useful to reduce the generalization error in classification and recognition tasks. Figure 5 shows a regularization parameter of 0.001 to provide the best recognition accuracy.

Table 5 Test phase of the proposed method

```
1. For i: 1 to the number of samples
*all samples
2. Output_gating_network=Test_gating_network(Images(i))
3. For j:1 To N
4. Output(j)=Test(Images(j,i))
5. Active_experts=Switching_system(Output, Output_gating_network)
6. For j:1 To N
7. Final_output= Final_output+ Active_experts(j)
```

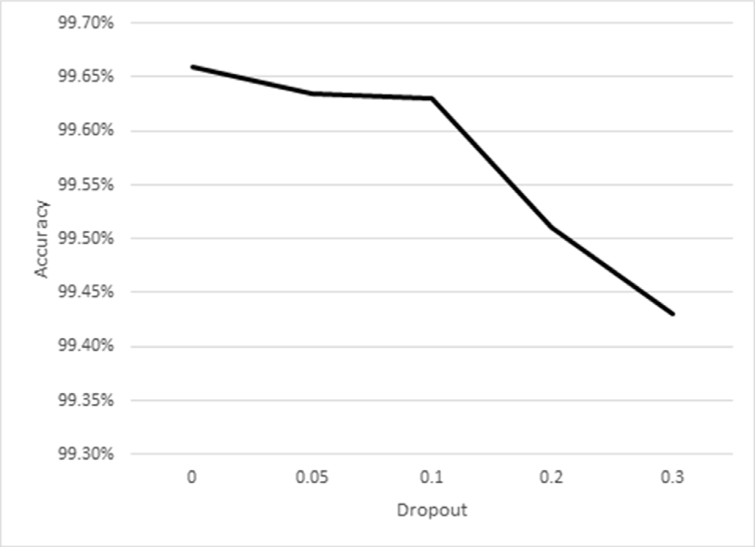


Fig. 4 Finding dropout parameter for experts

In our study, we aim at finding a valid trade-off between training individual expert CNNs with all training samples and a reduction in training time by reducing the dataset samples with which individual expert CNNs are trained. We propose to split the dataset equally among all experts and to allow for some overlap of expertise by training experts on a certain percentage of training samples of their neighbor expert. For finding the optimal overlap (=percentage of training samples from the neighbor expert to be added to the training set of the individual expert) we used 10- fold cross-validation. Figure 6 shows a rise when the overlap is 10% and a fall when it is 30%. The accuracy is increasing smoothly and is highest when the overlap is 80%. As a result, we added 80% of samples as an overlap training set

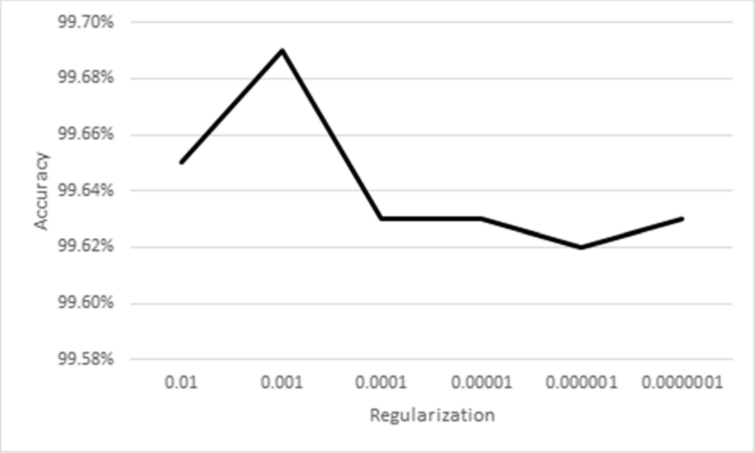


Fig. 5 Finding the regularization parameter

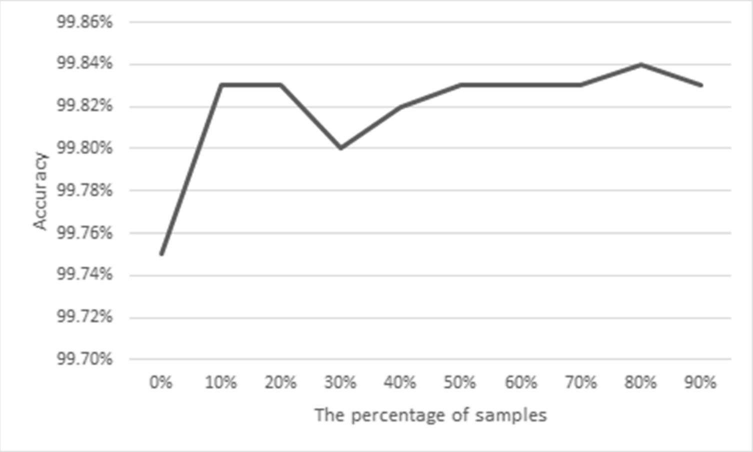


Fig. 6 Finding the best percentage of overlap

for each expert so as to make an overlap with its neighbor. For example, if we have five experts, expert 2 has its own training set plus 80% of the training set of expert 3.

The number of epochs is the last parameter that had to be evaluated. The SN was run with a different number of epochs and the average of accuracy and standard deviation for 10 runs was calculated. 10-fold cross-validation was used to find this parameter. In Fig. 7, the standard deviation and accuracies are shown. With regards to accuracy and standard deviation, the best epoch value was found to be 25.

In Figs. 8 and 9, the MSE error and the accuracy in the validation set for the different epochs is shown. As seen, the plots show that the MSE error and the accuracy of the proposed approach follow a stable trend.

In Fig. 10, the number of experts that are selected by a GN to contribute to a joint decision system (SN approach) is shown. Based on this figure, for more than 72.91% of dataset samples, only three experts are selected to fuse their outputs. Two experts are selected to contribute to the recognition process for only about 15% of the dataset samples. More than

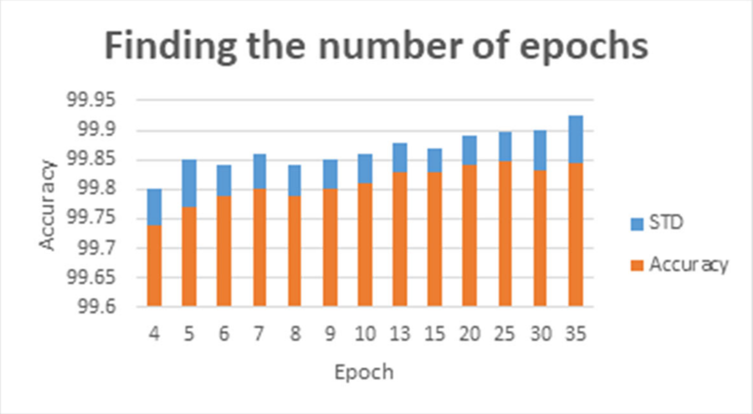


Fig. 7 Finding the number of epochs

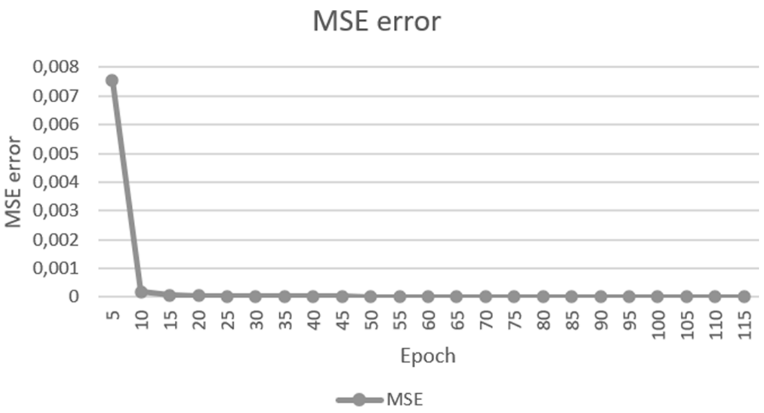


Fig. 8 MSE error over epoch

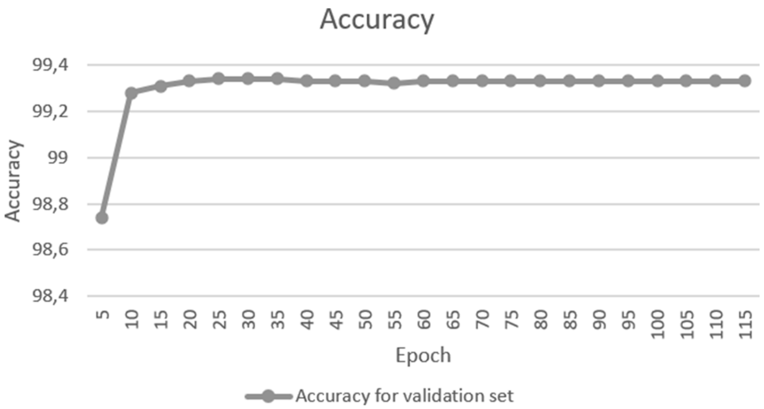
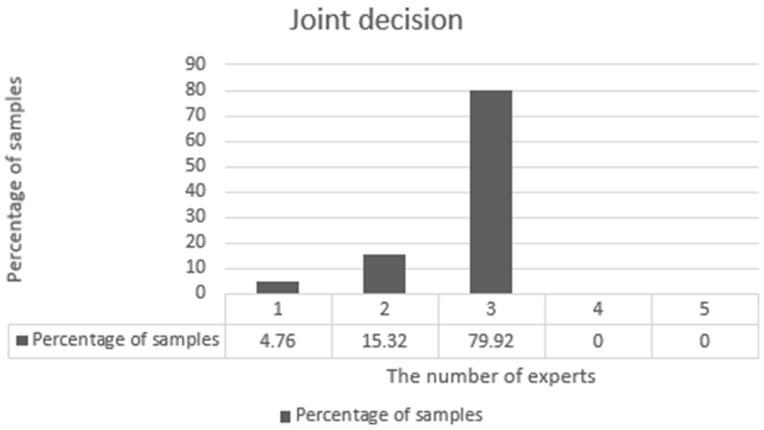


Fig. 9 Accuracy in the validation set over epoch



	1	2	3	4	5
■ Percentage of samples	4.76	15.32	79.92	0	0

Fig. 10 The number of experts which is involved in decision-making process

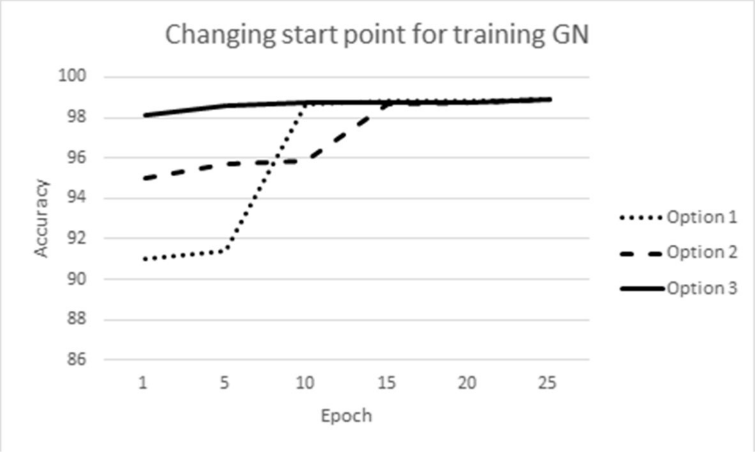


Fig. 11 Option 1: the start point for training GN is epoch 6. Option 2: The start point for training GN is epoch 11, Option 3: The GN training starts with experts simultaneously

4% of samples are classified by only one. However, no samples are classified by either four or all five experts.

In Fig. 11, the start point of GN training is changed to see which rise and fall of accuracy can occur in the training process. Three different options were compared: In options 1 and 2, the start points for GN training are epoch 6 and epoch 11 respectively. In option 3, the GN was trained from epoch 1 as usual. According to Fig. 11, the final result is not changed, but the learning process depends on GN training improved after adding GN in the training process. In option 1, the accuracy was not highly increased before epoch 5 and in option 2, the accuracy was raised immediately after epoch 11.

As seen in Fig. 12, the notable contribution to the decision-making process is for expert 4 with more than 99.5 %. Experts 4, 1, and 2 make a substantial contribution to the SN. On the contrary, the minor contribution belongs to the expert 5 with less than 0.5%. Experts 5

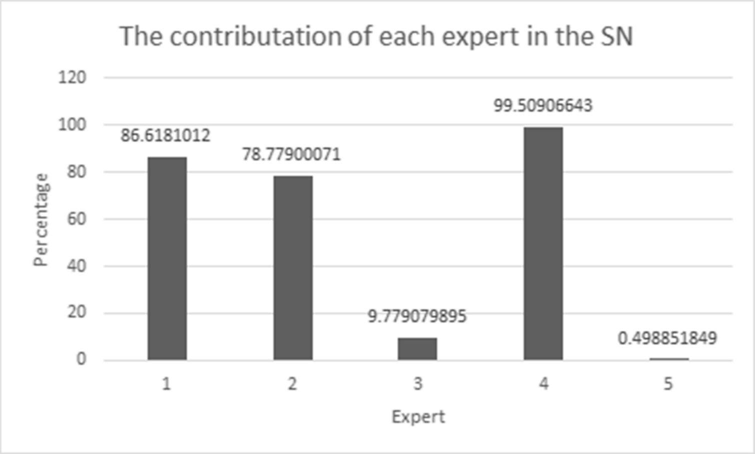


Fig. 12 The percentage of contribution per expert

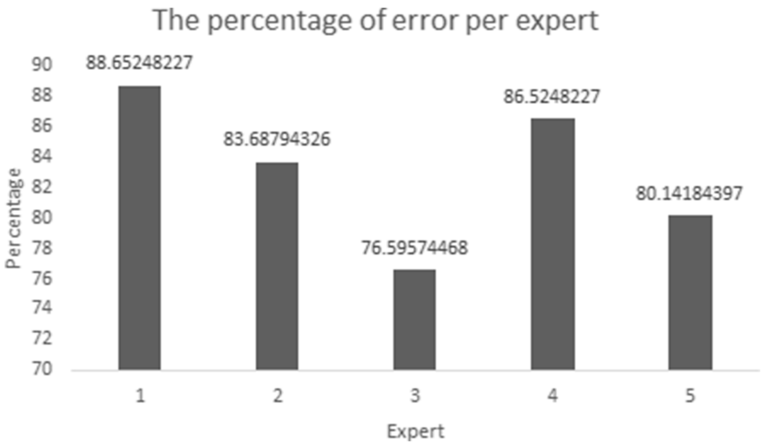


Fig. 13 The percentage of error per expert

and 3 contributed less than others. Due to this figure, the contribution rate of experts is not balanced.

The percentage of error per expert is shown in Fig. 13. The error rate of experts 1, 4, and 2 is higher than that of experts 5 and 3 as the contribution rate of experts 1, 4, and 2 is higher.

In 52.48 % of the errors (74 out of 141 samples), no expert generated the correct output. Thus also the SN could not generate a correct output by selecting experts. In 47.52 % of the errors (67 out of 141 samples), the wrong image class was predicted despite at least on expert generating the correct output. Thus here the GN caused the SN to select the wrong experts for making contributions to the overall output.

Two misclassified samples are demonstrated in Fig. 14. In the left picture, all experts failed in recognizing the correct image. So, the GN could not help the model to choose

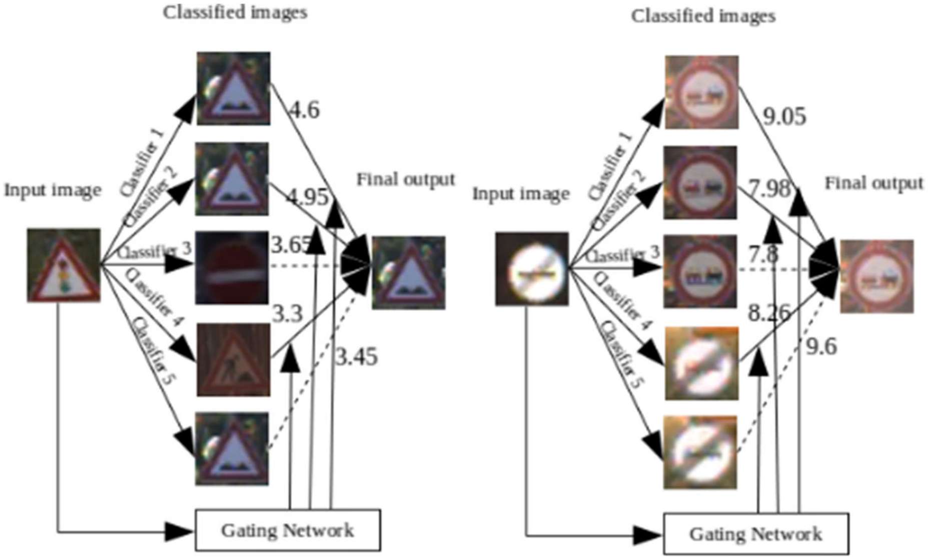


Fig. 14 Two misclassified samples (the left picture is from class 42 and the right picture is from class 27)

the correct class. In the right picture, experts 4 and 5 recognized the correct traffic sign, but experts 1, 2, and 4 were chosen by the GN to put into effect in the decision-making process. In this example, the GN failed to select the correct experts. The output numbers of each expert show the confidence level of each expert (i.e., experts 5, 1 and 4 show higher confidence rather than the other two experts in the right picture).

The overall recognition accuracy of the different architectures studied in this work is shown in Table 1 together with the number of training samples provided to individual experts. The best recognition accuracy of 99.10% has been achieved with the SN architecture using 100% of training samples closely followed by the SRNN architecture with an accuracy of 98.94% using 36% of training samples. Thus the effect of the ResNet layer in the GN architecture used by the SRNN architecture can be concluded to be existent but limited. The AN architecture and single expert CNN that were evaluated to form a baseline provided an accuracy of 98.76% and 98.63%, respectively. The accuracy of all approaches is the average of 10 runs. The results are in accordance with those shown in Fig. 10. It can be concluded that it indeed makes sense to block some experts from contributing to a final decision. Thus the contribution of the GN in selecting those experts that can contribute meaningfully to increase recognition accuracy is valuable for the overall recognition accuracy. It is interesting that the GN can perform a meaningful selection, given that GNs had only access to a subset of the total dataset.

In Table 6, the confusion matrix of the SN approach is shown. AC and PC stem from Actual Class and Predicted Class respectively. In other words, each column shows the actual label of samples and each row shows the predicted class. It is a visualization of the performance and errors of our proposed approach [29]. For example, 12 samples were misclassified by our proposed approach in row four and column six. The actual class is four, but our proposed approach classified them as class six. In Table 7, the Recall, Precision and F1-score are calculated for each class to see how the performance of our proposed method is per class. Moreover, the Marco-averaged and weighted approaches are used to see the performance of our model [22]. As shown, our proposed method is not working very well in predicting the classes 42, 27, and 7.

In Table 8, the recognition accuracy of the SRNN and SN architectures are compared to the competition results of other teams. The proposed approach that is fed by full training samples per expert achieved the fourth rank. This is still remarkable given that only 5 experts have been used and that individual experts had only been trained with a subset of 14,186 samples from the total dataset. Other competitors like the IDSIA team used a combination of 25 experts and parallelized their code in CUDA so that the training took one-day [4]. The accuracy of a person who recognized signs (human performance) [28] is 98.84% which shows our proposed approach works better than human performance. The top rank in the competition so far is a deep inception method based on CNN [11] with 99.81% accuracy and second rank is a CNN with three special transformers approach [3] which achieved 99.71%. The architecture of both teams is a single CNN with a big network. The number of parameters of both teams is 10.5 and 14.5 million respectively. Team 5 used a color-blob-based Cosfire filters [9]. Team 9 utilized a Multi-scale CNN [26] and team 10 proposed an MAE by designing an active expert module with a two-step training phase. This two-step training is active learning and Multi-Layer Perceptron (MLP). The number of active experts (the combination of active learning and MLP) is five. The application of the proposed method is for online situations or self-driving cars [2]. Team 11 applied Random Forests to the GTSRB dataset [34]. Teams 12, 13, and 14 used the LDA algorithm with one of HOG1, HOG2, and HOG3 features. These features are accessible publicly on the GTSRB website. Therefore,

Table 6 Confusion matrix of SN with 36% of training samples per expert

AC PC	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43
1	59	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	716	1	0	2	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
3	0	2	749	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	437	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	656	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	12	1	629	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	131	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	448	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	1	445	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	480	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	0
11	0	0	0	0	0	0	0	0	0	0	657	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	416	0	0	0	0	0	0	0	0	0	0	0	0	0	3	1	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	684	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	1	0	0	0	0	1	0	0	0	0	718	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	1	0	0	0	0	0	270	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	210	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	149	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	359	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	387	0	0	0	1	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	90	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	90	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	119	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6 (continued)

24	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	150	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	89	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	1	0	0	0	2	0	0	0	0	0	0	1	0	0	0	0	0	0	477	3	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	154	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	57	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	149	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	90	0	0	0	0	0	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	142	0	0	0	0	0	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	267	0	0	0	0	0	0	0	0	0	0	0
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	60	0	0	0	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	208	0	0	0	0	0	0	0	0	0
35	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	119	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	389	0	0	0	0	1	0	0
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	1	0	0	114	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0
39	0	1	0	0	0	0	7	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0	0	0	6	0	688	0	0	0	0
40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	90	0	0	0
41	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	88	0	0
42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	48	0
43	0	0	0	0	0	0	4	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	90

Table 7 Multi-class metrics of SN with 36% of training samples per expert

Class	Recall	Precision	F1-score
1	100.00	100.00	100.00
2	99.44	98.76	99.10
3	99.87	99.73	99.80
4	97.11	99.32	98.20
5	99.39	100.00	99.70
6	99.84	96.92	98.36
7	87.33	100.00	93.24
8	99.56	100.00	99.78
9	98.89	99.78	99.33
10	100.00	97.36	98.66
11	99.55	100.00	99.77
12	99.05	98.58	98.81
13	99.13	100.00	99.56
14	99.72	99.31	99.51
15	100.00	98.90	99.45
16	100.00	97.22	98.59
17	99.33	100.00	99.67
18	99.72	100.00	99.86
19	99.23	98.98	99.10
20	100.00	100.00	100.00
21	100.00	97.83	98.90
22	100.00	98.90	99.45
23	99.17	92.97	95.97
24	100.00	98.04	99.01
25	98.89	100.00	99.44
26	99.38	98.35	98.86
27	85.56	100.00	92.22
28	95.00	100.00	97.44
29	99.33	100.00	99.67
30	100.00	98.90	99.45
31	94.67	99.30	96.93
32	98.89	99.63	99.26
33	100.00	98.36	99.17
34	99.05	100.00	99.52
35	99.17	97.54	98.35
36	99.74	99.49	99.62
37	95.00	97.44	96.20
38	100.00	100.00	100.00
39	99.71	96.49	98.08
40	100.00	97.83	98.90
41	97.78	97.78	97.78
42	80.00	100.00	88.89
43	100.00	94.74	97.30
Macro-averaged approach	98.10	98.80	98.39
Weighted approach	98.88	98.91	98.87

Table 8 The results of the competition

	Papers	*Method*	*Accuracy*
1	Haloi M [11]	Deep Inception Based CNN	99.81
2	Deep Knowledge Seville [3]	CNN with 3 Spatial Transformers	99.71
3	IDSIA [4]	Committee of CNNs (25 CNNs)	99.46
4	**Proposed method**	**SN (100% of training samples)**	**99.10**
5	COSFIRE [9]	Color-blob-based COSFIRE filters	98.97
6	**Proposed method**	**SRNN (36% of training samples)**	**98.94**
7	**Proposed method**	**SN (36% of training samples)**	**98.88**
8	**Proposed method**	**SRNN (100% of training samples)**	**98.87**
9	INI-RTCV [28]	Human Performance	98.84
8	sermanet [26]	Multi-Scale CNNs	98.31
10	Ahangi [2]	MAE	96.69
11	CAOR [34]	Random Forests	96.14
12	INI-RTCV [28]	LDA on HOG 2	95.68
13	INI-RTCV [28]	LDA on HOG 1	93.18
14	INI-RTCV [28]	LDA on HOG 3	92.34

each team's result is equal to the accuracy of LDA applied to each feature set that order of each feature set for teams 11, 12, and 13 is HOG2, HOG1, and HOG3 respectively [28].

As shown in Table 9, each expert is fed with 36% of training samples (methods 5 and 6). Hence, five experts are trained with 70,930 training samples. On the contrary, the committee of CNNs are trained with all training samples per expert which are 980, 225. Therefore, the load of experts in our proposed approach (methods 5 and 6) is 13.8 times less than the committee of CNNs. The load of each expert in our proposed approach (methods 5 and 6) is 2.76 times less than others except Mixture of Active experts team. The parameters of our proposed approach are 11.2 million which is about 8 times less than the committee of CNNs approach, which is the closest method to our proposed approach. The number of parameters per expert in our proposed approach is lower than the method 1 (one-seventh of 10.5 million), method 2 (about one-tenth of 14.5 million) and method 3 (less than half of 3.6 million).

5 Conclusion

We propose and evaluate two new network architectures based on two different types of gating networks and a switching system for multi-expert recognition and classification systems. On the German Traffic Sign Recognition Benchmark, the proposed network architectures have been shown to create a recognition accuracy that outperforms human observations and that rank our contributions to the forth rank among thirteen other competing teams. The proposed training approach for the multi-expert recognition and classification systems has been developed to allow for simultaneous, parallel training of individual experts while limiting the number of training samples that must be provided to individual experts to a subset of the available dataset. We demonstrate that by training individual experts with different subsets that comprise only 36% of total training data we can still reach decision accuracies of the

Table 9 Complexity comparison of all CNN-based approaches

	Method	The number of training samples per expert	The number of experts	The number of parameters	The number of parameters per expert	Accuracy
1	Deep Inception Based CNN [11]	39,209 (100 %)	1	10.5 Million	10.5 Million	99.81
2	CNN with 3 Spatial Transformers [3]	39,209 (100 %)	1	14.5 million	14.5 Million	99.71
3	Committee of CNNs [4]	39,209 (100 %)	25 CNNs	90 million	3.6 Million	99.46
4	***Proposed approach (SN)***	***39,209 (100 %)***	***5 CNNs***	***11.2 million***	***1.5 Million***	***99.10***
5	***Proposed approach (SRNN)***	***14,186 (36 %)***	***5 CNNs***	***11.3 million***	***1.5 Million***	***98.94***
6	***Proposed approach (SN)***	***14,186 (36 %)***	***5 CNNs***	***11.2 million***	***1.5 Million***	***98.88***
7	***Proposed approach (SRNN)***	***39,209 (100 %)***	***5 CNNs***	***11.3 million***	***1.5 Million***	***98.87***
8	Multi-Scale CNNs [26]	39,209 (100 %)	1	N	N	98.31
9	Mixture of Active Experts [2]	23,521 (60 %)	5 MLPs	0.5 million	0.1 Million	96.69

combined experts comparable to other state-of-the-art approaches training experts with all data.

6 Future work

Future work will focus on reaching a further increase of the prediction accuracy of the model. Here we target two main strategies: (1) Approaches that go beyond pure averaging in fusing opinions of individual experts have potential in leading to a further increase of overall model accuracy. (2) We believe that a diversification of experts could contribute to a further increase of prediction accuracy. Future work will also be targeted to further decrease training time through an even stronger parallelisation of training. We intent to explore a new set-up to train the gating network and experts at the same time with reduced data sets.

Funding We thank the Dutch National Science Foundation NWO for financial support through the project SWARMPORT (NWO project number 439.16.108).

Declarations

Competing interests The authors declare no conflict of interest.

References

1. Ahangi A, Karamnejad M, Mohammadi N, Ebrahimpour R, Bagheri N (2013) Multiple classifier system for eeg signal classification with application to brain–computer interfaces. Neural Comput Appl 23(5):1319–1327. https://doi.org/10.1007/s00521-012-1074-3
2. Ahangi A, Langroudi AF, Yazdanpanah F, Mirroshandel SA (2019) A novel fusion mixture of active experts algorithm for traffic signs recognition. Multimed Tools Appl 78(14):20217–20237. https://doi.org/10.1007/s11042-019-7391-0
3. Arcos-García Á, Álvarez-García JA, Soria-Morillo LM (2018) Deep neural network for traffic sign recognition systems: an analysis of spatial transformers and stochastic optimisation methods. Neural Netw 99:158–165. https://doi.org/10.1016/j.neunet.2018.01.005
4. Ciregan D, Meier U, Schmidhuber J (2012) Multi-column deep neural networks for image classification. In: 2012 IEEE Conference on computer vision and pattern recognition, pp 3642–3649. https://doi.org/10.1109/CVPR.2012.6248110
5. Ciresan DC, Meier U, Gambardella LM, Schmidhuber J (2011) Convolutional neural network committees for handwritten character classification. In: 2011 International conference on document analysis and recognition, pp 1135–1139. https://doi.org/10.1109/ICDAR.2011.229
6. Cireşan D, Meier U, Masci J, Schmidhuber J (2012) Multi-column deep neural network for traffic sign classification. Neural Netw 32:333–338. https://doi.org/10.1016/j.neunet.2012.02.023. Selected papers from IJCNN 2011
7. Combining pattern classifiers (2014) Methods and algorithms, 2nd edn Wiley Publishing
8. Deng L, Platt JC (2014) Ensemble deep learning for speech recognition. In: Interspeech
9. Gecer B, Azzopardi G, Petkov N (2017) Color-blob-based cosfire filters for object recognition. Image Vis Comput 57:165–174. https://doi.org/10.1016/j.imavis.2016.10.006

10. Gopalakrishnan S (2012) A public health perspective of road traffi c accidents. J Family Med Primary Care July 2012 1:144
11. Haloi M (2015) Traffic sign classification using deep inception based convolutional networks. CoRR arXiv:1511.02992
12. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: The IEEE conference on computer vision and pattern recognition (CVPR)
13. Huang GB, Lee H, Learned-Miller E (2012) Learning hierarchical representations for face verification with convolutional deep belief networks. In: 2012 IEEE conference on computer vision and pattern recognition, pp 2518–2525. https://doi.org/10.1109/CVPR.2012.6247968
14. Jacobs RA, Jordan MI, Nowlan SJ, Hinton GE (1991) Adaptive mixtures of local experts. Neural Computat 3(1):79–87. https://doi.org/10.1162/neco.1991.3.1.79. PMID: 31141872
15. Kuncheva LI (2004) Combining pattern classifiers: methods and algorithms
16. Kuncheva LI, Whitaker CJ (2003) Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy
17. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature Cell Biol 521(7553):436–444. https://doi.org/10.1038/nature14539
18. Li T, Sahu AK, Talwalkar A, Smith V (2020) Federated learning: challenges, methods, and future directions. IEEE Signal Process Mag 37(3):50–60. https://doi.org/10.1109/MSP.2020.2975749
19. Maji D, Santara A, Mitra P, Sheet D (2016) Ensemble of deep convolutional neural networks for learning to detect retinal vessels in fundus images
20. Masoudnia S, Ebrahimpour R (2014) Mixture of experts: a literature survey. Artif Intell Rev 42(2): 275–293. https://doi.org/10.1007/s10462-012-9338-y
21. Polikar R (2006) Ensemble based systems in decision making. IEEE Circuits Syst Mag 6(3):21–45. https://doi.org/10.1109/MCAS.2006.1688199
22. Powers DMW (2020) Evaluation: from precision, recall and F-measure to ROC, informedness markedness and correlation
23. Prieto A, Prieto B, Ortigosa EM, Ros E, Pelayo F, Ortega J, Rojas I (2016) Neural networks: an overview of early research, current frameworks and new challenges. Neurocomputing 214:242–268. https://doi.org/10.1016/j.neucom.2016.06.014
24. Rasti R, Teshnehlab M, Phung SL (2017) Breast cancer diagnosis in dce-mri using mixture ensemble of convolutional neural networks. Pattern Recogn 72:381–390. https://doi.org/10.1016/j.patcog.2017.08.004
25. Schmidhuber J (2015) Deep learning in neural networks: an overview. Neural Netw 61:85–117. https://doi.org/10.1016/j.neunet.2014.09.003
26. Sermanet P, LeCun Y (2011) Traffic sign recognition with multi-scale convolutional networks. In: The 2011 international joint conference on neural networks, pp 2809–2813. https://doi.org/10.1109/IJCNN.2011.6033589
27. Shopa P, Sumitha N, Patra PSK (2015) Traffic sign detection and recognition using opencv. In: 2014 International conference on information communication and embedded systems, ICICES 2014. https://doi.org/10.1109/ICICES.2014.7033810
28. Stallkamp J, Schlipsing M, Salmen J, Igel C (2012) Man vs. computer: benchmarking machine learning algorithms for traffic sign recognition. Neural Netw 32:323–32. https://doi.org/10.1016/j.neunet.2012.02.016. Epub 2012 Feb 20. PMID:22394690
29. Stehman SV (1997) Selecting and interpreting measures of thematic classification accuracy. Remote Sens Environ 62(1):77–89. https://doi.org/10.1016/S0034-4257(97)00083-7
30. Sun Z-L, Wang H, Lau W-S, Seet G, Wang D (2013) Application of bw-elm model on traffic sign recognition. Neurocomputing, vol 128. https://doi.org/10.1016/j.neucom.2012.11.057
31. Sun Y, Wang X, Tang X (2016) Hybrid deep learning for face verification. IEEE Trans Pattern Anal Mach Intell 38(10):1997–2009. https://doi.org/10.1109/TPAMI.2015.2505293
32. Wang H-Z, Li G-Q, Wang G-B, Peng J-C, Jiang H, Liu Y-T (2017) Deep learning based ensemble approach for probabilistic wind power forecasting. Appl Energy 188:56–70. https://doi.org/10.1016/j.apenergy.2016.11.111
33. Zaklouta F, Stanciulescu B (2014) Real-time traffic sign recognition in three stages. Robot Auton Syst 62:16–24. https://doi.org/10.1016/j.robot.2012.07.019
34. Zaklouta F, Stanciulescu B, Hamdoun O (2011) Traffic sign classification using k-d trees and random forests. In: The 2011 international joint conference on neural networks, pp 2151–2155. https://doi.org/10.1109/IJCNN.2011.6033494
35. Zhang X, Wang D (2016) A deep ensemble learning method for monaural speech separation. IEEE/ACM Trans Audio Speech Lang Process 24(5):967–977. https://doi.org/10.1109/TASLP.2016.2536478
36. Zhou Z-H (2012) Ensemble methods: foundations and algorithms, 1st edn Chapman & hall/CRC

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.