EDeN: Ensemble of Deep Networks for Vehicle Classification

Rajkumar Theagarajan, Federico Pala, Bir Bhanu
Center for Research in Intelligent Systems,
University of California, Riverside
{rthea001, fedpala}@ucr.edu, bhanu@cris.ucr.edu

Abstract

Traffic surveillance has always been a challenging task to automate. The main difficulties arise from the high variation of the vehicles appertaining to the same category, low resolution, changes in illumination and occlusions. Due to the lack of large labeled datasets, deep learning techniques still have not shown their full potential. In this paper, thanks to the MIOvision Traffic Camera Dataset (MIO-TCD), an Ensemble of Deep Networks (EDeN) is used to successfully classify surveillance images into eleven different classes of vehicles. The ensemble of deep networks consists of 2 individual networks that are trained independently. Extensive evaluations were carried out using individual networks and their ensemble, using the MIO-TCD dataset that consists of 786,702 diverse images resembling a real-world environment. Experimental results show that the ensemble of networks gives better performance compared to individual networks and it is robust to noise. The ensemble of networks achieves an accuracy of 97.80%, mean precision of 94.39%, mean recall of 91.90% and Cohen kappa of 96.58.

1. Introduction

Automatic vehicle classification plays a vital role for the safety and efficient traffic surveillance. To date, the majority of the traffic data acquisition and measurements are obtained using sensors such as radar, loop detectors and road tubes. A drawback of these sensors is the requirement of intrusive installations and calibration procedures. Instead, non-intrusive video-based traffic measurement systems are becoming popular for two main reasons. First, humans can more easily review the data collected from a video camera. Second, advanced computer vision and machine learning algorithms can be employed at different stages of the data acquisition pipeline. This is useful for extracting scalable information that can be used in designing efficient and intelligent transportation systems.

Many modern vehicle classification algorithms rely on machine learning to classify vehicles [16]. These algorithms are trained on small traffic datasets that do not contain sufficient diversity for training a real-world traffic monitoring system [17-19]. Furthermore, these datasets do not contain a sufficient variability in terms of weather conditions, camera perspectives, roadway conditions and roadway configurations.

In this paper, we present an Ensemble of Deep Networks (EDeN) for the classification of vehicles from traffic surveillance images using the MIOvision Traffic Camera Dataset (MIO-TCD). To date, MIO-TCD is the largest dataset collected so far for the task of vehicle surveillance. The dataset consists of 786,702 images taken from 8,000 different traffic surveillance cameras deployed all over the USA and Canada. These images are taken at different times of the day and different times of the year. Additionally, the images are taken from a different angle, scale and resolution. Fig.1 shows a few examples of images taken from the dataset.

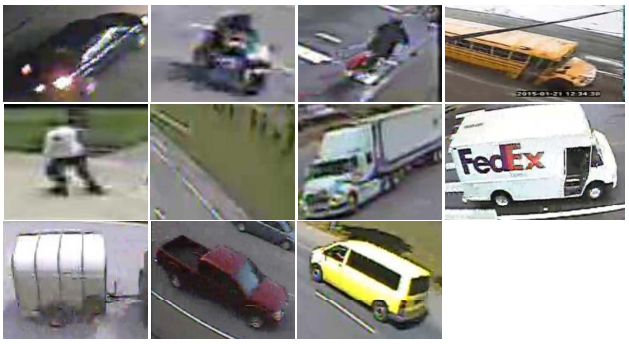

Fig. 1: Sample images from the MIO-TCD dataset.

In Fig. 1, the images in the first row are car, motorcycle, bicycle and bus. The images in the second row are pedestrian, background, articulated truck and single unit truck. The images in the last row are non-motorized vehicle, pickup truck and work van.

In our Ensemble of Deep Networks, we train 2 individual networks (Network A and Network B) independently. During testing, we use 3 networks (Network A, B and C) and get the final prediction by taking the weighted average of the predictions of the individual networks. Network C is a copy of Network B but with the inclusion of logical reasoning. We perform extensive experiments on each network individually and in ensemble for evaluating the validation and testing accuracy. Our experiments show that using the networks in ensemble achieved better results. We found that the ensemble of networks is more robust to noisy data.

2160-7516/17 $31.00 © 2017 IEEE
DOI 10.1109/CVPRW.2017.125

2. Related Work and Contributions

2.1. Related work

In the past, computer vision has mostly been used in combination with different handcrafted features and sensors [1-4]. Cho *et al.* [1] fused radar and laser systems together using a Kalman filter for object detection and classification. They used different motion models for tracking pedestrians, bicyclists and cars. Held *et al.* [2] used different road and deformable parts based model to detect generic moving objects on roads. A probabilistic model is used to combine multiple forms of evidence to locate cars in real-world scenarios. Carafii *et al.* [3] detected moving objects in real-time using a single car-mounted camera. The vehicles are tracked using a WaldBoost detector along with a Tracking-Learning-Detection (TLD) tracker. Jazayeri *et al.* [4] used temporal information of features of the detected object and a front-view motion model to reduce false positives. To separate the vehicles from the background, they used a hidden Markov model to characterize the continuous movement of features. Thakoor and Bhanu [23] used the rear view to classify vehicles on highways. They used the variation in the structural signature as a vehicle moved forward to classify them as sedan, pickup truck and SUV/minivan. They classified the vehicles using support vector machines (SVM). Another rear view based classification was done by Kafai and Bhanu [26] where they used the spatial information between landmarks of the vehicle (e.g. taillights and license plates) and a dynamic Bayesian network for vehicle classification. They had a drawback similar to as [23] where they could not differentiate between SUV and minivan because these vehicles look aesthetically similar from the rear view. Theagarajan *et al.* [14] classified vehicles in images using their rear view. They were able to distinguish between SUV and minivan from the rear view using the visual rear ground clearance. They classified the vehicles into high and low visual rear ground clearance. The visual rear ground clearance of each vehicle is estimated as a physical measurement using a multi-frame tracking approach.

Before the widespread adoption of Convolutional Neural Networks (CNNs) and deep learning within computer vision, one of the most successful methods for vehicle detection was the deformable parts based model [5]. After the Imagenet competition [27] entry of Krizhevsky *et al.* [6], state-of-the-art for feature extraction shifted towards CNNs [7-10].

Girshick *et al.* developed Regions and CNN features (R-CNN), a two-part system which used selective search [11] to propose regions and the architecture of [6] to classify them. Szegedy *et al.* [9] detected objects using a regression network that detected high resolution bounding boxes using a multi-scale inference procedure. Huval *et al.* [12] used The OverFeat [7] architecture along with a mask detector similar to Szegedy *et al.* [9] to detect highway lanes and vehicles in real-time. Wang *et al.* [13] used CNN along with Fisher feature encoding algorithms to classify the type of a vehicle. They used CNN to compensate the information loss that occurs by using handcrafted features. Chen *et al.* [15] used parallel deep neural networks (PNN) to localize vehicles from satellite images. The authors did not use direct connections between branches in order to maintain the structure and dimension of each branch, and in doing so, their architecture achieved 10X speed compared to a single Deep Neural Network (DNN).

However, the datasets used in the above mentioned state-of-the-art approaches did not contain a sufficient number of diverse examples that resemble real-world traffic surveillance images. The following works used real-world traffic surveillance videos/images. Salvi [20] used vehicle surveillance videos during the night time. They counted the number of vehicles on the highways by using morphology and image processing algorithms. Aslaine *et al.* [21] used optical flow combined with morphology to track and classify vehicles according to their size.

Although, DNN's have been successful in the task of vehicle classification, not much work has been done for real-world traffic surveillance applications using DNN.

2.2. Contributions of this paper

In view of state-of-the-art, the contributions of this paper are:

- An ensemble of deep networks for classifying vehicles from traffic surveillance images.
- Logical reasoning to solve dual class misclassifications (explained in Section IV. E).
- Extensive experimental evaluation of our model on a huge real-world traffic surveillance dataset.

3. Technical Approach

This section describes the framework and architecture of the individual models in our ensemble of deep networks (EDeN). Fig. 2 shows the overall architecture of our approach.

3.1. Framework of Ensemble of Deep Networks

As depicted in Fig. 2 the input of our network is a batch of images. From each image, we randomly crop a fixed size patch and pass the batch into 2 individual networks (Network A and Network B). Each network is trained independently and we denote the final predicted vector of each network as X'_i. The size of X'_i is 1xN, where N is the number of output classes to be predicted. During testing, we use 3 networks (Network A, Network B and Network C). Network C is a copy of Network B but with the inclusion of logical reasoning. The logical reasoning is added after the fully connected layer.

Each element of X'_i is multiplied by the corresponding element of a weight vector W_i, where:

$$W_i X'_i = [W_{i1} X'_{i1}, W_{i2} X'_{i2}, ..., W_{iN} X_{iN}]^T \quad (1)$$

The final prediction vector is the average of the weighted predictions of each network.

3.2. Network architecture

We employ the Residual Network architecture by [21], where identity mapping is used for increasing the depth of the network thereby avoiding the vanishing gradient problem. In EDeN we used the RESnet34 and RESnet50 architectures. RESnet34 has 34 convolutional layers, and RESnet50 has 50 convolutional layers. In both architectures, max pooling is replaced by convolutional down sampling.

The network takes 224x224 image patches, and batch normalization is performed to get a faster training convergence. Rectified linear units (ReLU) are used as non-linearities. In Fig. 2, network A is a RESnet34 architecture and network B is a RESnet50. Both networks A and B use a weighted cross-entropy loss function. Network C has the same architecture as network B but includes logical reasoning. Table I is a summary of the individual networks used in EDeN.

Table I: Summary of the individual networks

Network	Arcitecture	Loss function
Network A	Resnet34	Weighted cross entropy
Network B	Resnet50	Weighted cross entropy
Network C	Resnet50 with logic	Weighted cross entropy

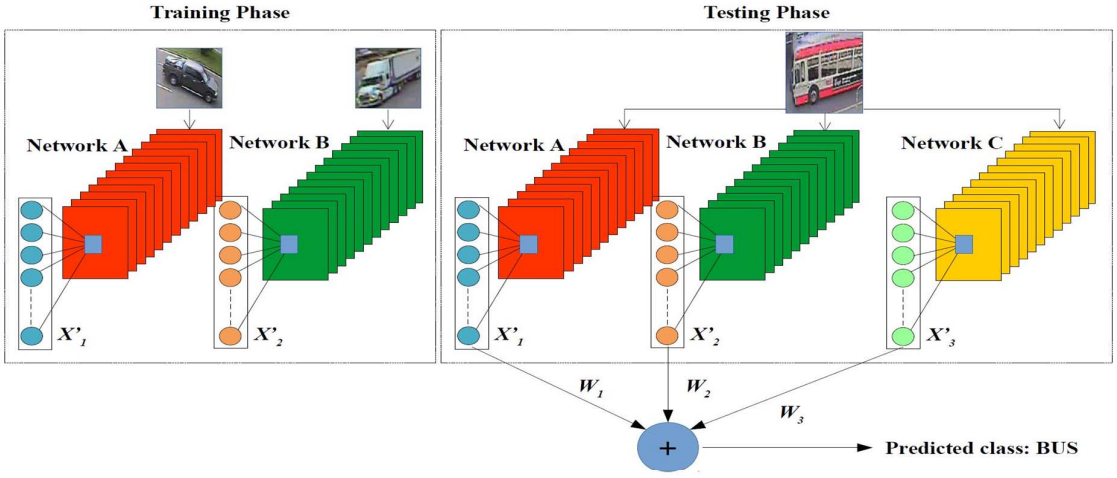

Fig. 2: Overall architecture of EDeN.

During training, Network A and B are trained independently. By doing so each network learns its own representation of the image. During testing, the image is passed through the individual networks, and the weighted average of the softmax output gives us the final predictions.

3.3. Weighted Cross-entropy loss function

Traffic surveillance involves classification of different vehicles as well as pedestrians and background. All classes do not have an equal amount of training data because some vehicles are rarely seen on the road compared to others. For example, surveillance cameras on highways are more likely to see trucks than bicycles or pedestrians, and vice versa for a college campus. In order to handle this unbalanced nature of surveillance data, we use a weighted cross-entropy loss function. It is desirable to give more weight to classes that have very few training data compared to classes that have more training data. In our network, we used the complementary *a-priori* probability of each class as weights. The complementary *a-priori* probability for class $C_i = 1 - X_{ci} / X$, where X_{ci} is the total number of images belonging to class C_i and X is the total number of images for all classes.

4. Experiments

We evaluated our proposed approach on the MIO-TCD dataset especially made for traffic surveillance tasks for the Traffic Surveillance Workshop and Challenge held in conjunction with the conference on Computer Vision and Pattern Recognition (CVPR), 2017.

The network architecture along with the overall framework have been implemented using the Pytorch computing

framework [22] on a NVIDIA DIGITS DevBox with four TITAN X GPU's with 7 TFlops of single precision, 336.5GB/s of memory bandwidth and 12 GB of RAM memory per board.

4.1. Datasets

We performed our experiments on the MIO-TCD dataset where the images are obtained from real-world traffic surveillance cameras deployed all over the USA and Canada. The total number of training images in the dataset for the classification task is 521,451 with 11 different classes namely: articulated truck (1.98%), background (30.68%), bus (1.98%), bicycle (0.44%), car (49.96%), motorcycle (0.38%), non-motorized vehicle (0.34%), pedestrian (1.2%), pickup truck (9.76%), single unit truck (0.98%) and work van (1.86%). The percentage above indicates the data distribution of each class in the dataset.

Clearly, it can be noticed that the dataset is unbalanced with data distribution for cars nearly 50%. According to the Bureau of Transportation Statistics for 2012 [28], 63.96% of all registered vehicles in the USA are light duty vehicles that include cars and pickup trucks, which is close to the data distribution of cars and pickup trucks (59.72%) in the dataset. Additionally, 3.22% of the registered vehicles in 2012 were 2 axles with 6 or more tires which include articulated trucks and single unit trucks from our dataset which corresponds to a data distribution of (2.96%). This makes the data distribution in the MIO-TCD dataset close to real-world vehicle data distribution. More details are provided on the workshop's website [23].

4.2. Data Augmentation

From the data distribution, it can be noticed that 8 out of 11 classes have less than 5% of the total images. In order to add more diversity for those classes, we added more images from the Imagenet dataset [27] and training images from the localization dataset of MIO-TCD. The total number of images added was 2,247 from Imagenet and 101,234 from the training images of the localization dataset of MIO-TCD. Additionally, we added 18,000 more images for the pedestrian class from the benchmark pedestrian re-identification database PETA [24].

4.3. Preprocessing

All the images in our dataset were resized to maintain aspect ratio such that the shorter side has length of 256 pixels and the longer side has the corresponding length to maintain the aspect ratio. For example, for a given M x N image where M is the row dimension, and N is the column dimension, the aspect ratio is given by $AR = N/M$ and the resized image has the following dimensions:

$$\begin{cases} if\ M < N; M = 256, N = AR * M \\ if\ M > N; N = 256, M = N / AR \end{cases} \quad (2)$$

4.4 Experimental setup

For all the experiments we evaluate performance in terms of accuracy, mean precision, mean recall and Cohen kappa score. These measures are used to evaluate entries in the Traffic Surveillance Challenge. Since a validation set is not provided, we split the training set (before data augmentation) into 75% data for training and 25% data for validation.

We set the mini-batch size as 128, and during each epoch, the training data is randomly shuffled, and we take a randomly cropped 224x224 patch from each input image. We used the stochastic gradient descent algorithm to minimize the weighted cross-entropy loss function.

We did random parameter selection on the validation set for individual networks to obtain the best learning rate, momentum and weight decay for each network.

The best parameters for Network A were found to be, learning rate = $6x10^{-3}$, momentum = 0.9 and weight decay = 10^{-4}.

The best parameters for Network B were found to be, learning rate = $6.5x10^{-3}$, momentum = 0.9 and weight decay = $4x10^{-4}$.

For both networks, the learning rate is reduced when the training loss has not decreased after 3 consecutive epochs.

The learning rate is reduced by a factor of 5 after the 15^{th} and 25^{th} epoch for Network A.

For Network B the learning rate is reduced by a factor of 5 after the 12^{th} and 20^{th} epoch.

After every epoch, we check the validation error and if it is decreasing, we save the model. If the validation error has not decreased after 5 consecutive epochs we employ early stopping and stop the training.

4.5. Experimental results

• **Network A:** We trained Network A, which is a RESnet34 architecture pre-trained on the Imagenet dataset, with the parameters as explained in the experimental setup. The evaluated performance of the network on the validation set achieved an accuracy of 97.12%, mean recall of 90.12% and mean precision of 91.35%. Fig. 3 shows the performance evaluation plot for network A on the validation set.

• **Network B:** We trained Network B, which is a RESnet50 architecture pre-trained on the Imagenet dataset, with the parameters as explained in the experimental setup. The evaluated performance of the network on the validation set achieved an accuracy of 97.51%, mean recall of 90.78% and mean precision of 92.23%. Fig.4 shows the performance evaluation plot for network B on the validation set.

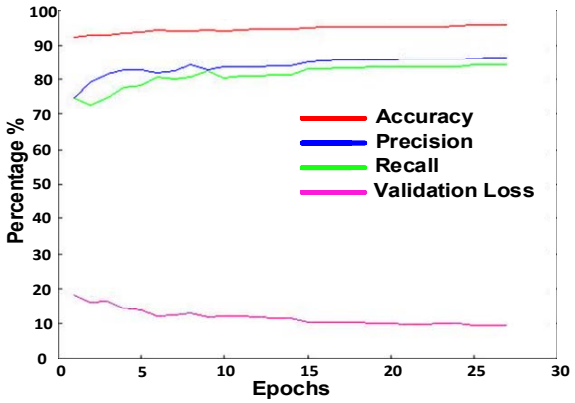

Fig. 3: Performance evaluation plot of Network A on the validation set.

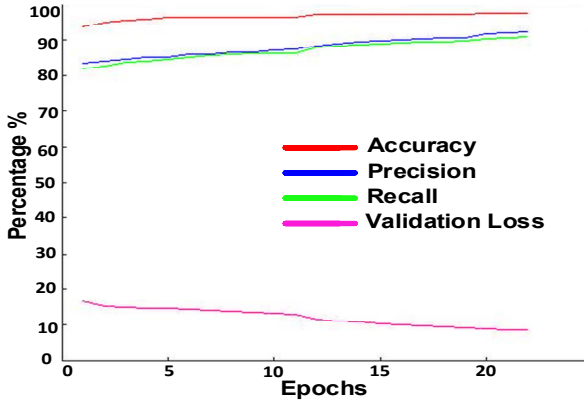

Fig. 4: Performance evaluation plot of network B on the validation set.

We individually tested Network A and Network B on the testing set and the results are shown in Table II. For testing, we employed the five patch testing method as in [6], where each image is split into 5 overlapping patches (4 patches from the corner and 1 patch from the center) and passed through the network. The average prediction of the 5 patches is taken as the final prediction of the image. Network A achieved a classification accuracy of 97.25%, mean recall of 89.28%, mean precision of 93.60% and Cohen kappa score of 95.73. Network B achieved an accuracy of 97.43%, mean recall of 91.61%, mean precision of 93.52% and Cohen kappa score of 96.02 on the testing set.

According to the competition rules the final score for each model is the average of the accuracy, mean precision, mean recall and Cohen kappa score. Based on this rule, network A scored 93.96 and network B scored 94.64.

In Table II, the classes are denoted as **AT**: Articulated truck, **BI**: Bicycle, **Bus:** Bus, **Car:** Car, **MO**: Motorcycle, **NMV**: Non-motorized vehicle, **PE**: Pedestrian, **PT**: pickup truck, **SUT**: Single unit truck, **WV**: Work van, **BG**: Background.

Table II: Evaluation of Network A and Network B on testing set

Network/Classes	AT	BI	Bus	Car	MO	NMV	PE	PT	SUT	WV	BG
Network A (precision)	0.9308	0.9128	0.9909	0.9831	0.9678	0.8529	0.9637	0.8821	0.8681	0.9494	0.9945
Network A (recall)	0.9354	0.8984	0.9709	0.9757	0.9111	0.5959	0.9323	0.9525	0.8070	0.8443	0.9968
Network B (precision)	0.9457	0.9042	0.9886	0.9876	0.9787	0.8238	0.9686	0.8803	0.8628	0.9527	0.9939
Network B (recall)	0.9362	0.9089	0.9794	0.9735	0.9273	0.7260	0.9259	0.9583	0.8547	0.8889	0.9981

• **Problems encountered during the training of Network A and Network B:** During the training of network A and network B, we noticed persistent misclassification between bicycles and pedestrians, non-motorized vehicles and articulated truck/ single unit truck, and pickup trucks and cars. Fig. 5 shows some of these examples. In Fig. 5 the top row corresponds to bicycles that were misclassified as pedestrians, the middle row corresponds to non-motorized vehicle misclassified as articulated truck and the bottom row corresponds to pickup truck misclassified as car. The reason for this is that an image that belongs to the class bicycle has both bicycle and pedestrian in it, while an image in the class pedestrian has only a pedestrian. Similarly, an image from the class non-motorized vehicle has both non-motorized vehicle and articulated truck/ single unit truck in it. So, as the network becomes deeper, it learns stronger features for pedestrian, articulated truck and single unit truck compared to bicycle and non-motorized vehicle.

Fig. 5: Misclassified images from Network A and B.

Hence when it sees a pedestrian in an image that belongs to bicycle, the network is more likely to overpower the class bicycle and predicts it as pedestrian. The same holds for articulated truck, single unit truck and non-motorized vehicle.

• **Network C:** To solve this problem of *dual class misclassification* we introduced network C as depicted in Fig. 2 and section III.B. Network C is the same as Network B, but with the addition of logical reasoning and is used only in the testing phase. Fig. 6 shows the functioning of the logical reasoning of network C. We employed the five patch testing method to evaluate network C. When the network sees an image that belongs to bicycle, if there is at least one patch out of the five predicted as bicycle, then the final prediction is bicycle. Similarly, when the network sees an image that belongs to non-motorized vehicle, if there is at least one patch that belongs to non-motorized vehicle, then the final prediction is non-motorized vehicle. The same is done for articulated truck/ single unit truck and pickup truck/ car. We individually evaluated Network C on the testing set of MIO-TCD with the same experimental protocols explained in section IV.D and the results are shown in Table III. Network C achieved an accuracy of 97.59%, mean recall of 91.85%, mean precision of 94.26% and Cohen kappa score of 96.26%. The final score for Network C is 94.99.

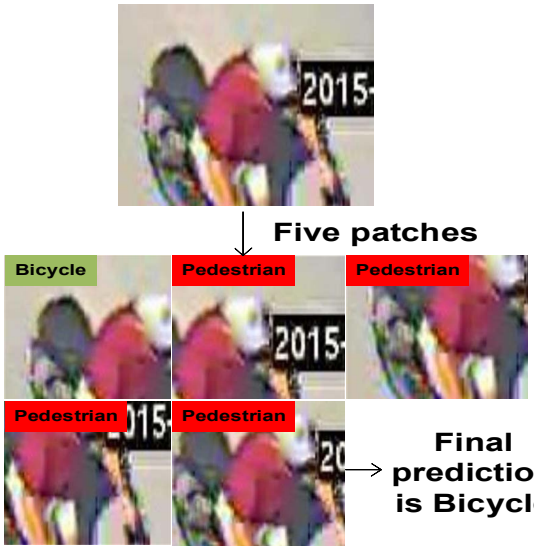

Fig. 6: Network C correctly classifies bicycle that was misclassified as pedestrian by Network B.

Table III: Evaluation of Network C on the testing set

Network/Classes	AT	BI	Bus	Car	MO	NMV	PE	PT	SUT	WV	BG
Network C (precision)	0.9383	0.9318	0.9913	0.9873	0.9748	0.8647	0.9662	0.8879	0.8763	0.9541	0.9954
Network C (recall)	0.9405	0.9089	0.9763	0.9764	0.9374	0.7443	0.9323	0.9574	0.8305	0.9009	0.9981

• **Ensemble of Deep Networks (EDeN):** In this experiment, we combine Networks A, B and C using weighted prediction vectors. In our experiments, we chose the weights W_{in} to be the average of the precision and recall for each individual class of that network.

$$W_{in} = average(Pre_{in}, Rec_{in}) \quad (3)$$

where $i=1,2,3$ refers to Network A, Network B and Network C, respectively. $n= 1$ *to* N refers to the class index.

$$Pre_{in} = \frac{TP_{in}}{TP_{in} + FP_{in}} \quad (4)$$

$$Rec_{in} = \frac{TP_{in}}{TP_{in} + FN_{in}} \quad (5)$$

The final prediction is the average of $W_1X'_1$, $W_2X'_2$ and $W_3X'_3$, where $W_1X'_1$, $W_2X'_2$ and $W_3X'_3$ are the weighted predictions of Network A, Network B and Network C respectively.

The weights for each network are obtained by evaluating the network on the validation set. The model achieved an accuracy of 97.80%, mean recall of 91.90%, mean precision of 94.39% and Cohen kappa score of 96.58 resulting in a final score of 95.17. Table IV shows the confusion matrix of EDeN and Table V shows the evaluation of EDeN on the test set.

Table IV: Confusion matrix of EDeN

Classes	AT	BI	Bus	Car	MO	NMV	PE	PI	SUT	WV	BG
AT	94.51	0.00	0.15	0.19	0.00	1.04	0.00	0.08	3.52	0.15	0.35
BI	0.00	89.84	0.18	0.18	1.40	0.00	7.18	0.00	0.00	0.00	1.23
Bus	0.31	0.00	97.94	0.89	0.00	0.00	0.00	0.31	0.08	0.16	0.23
Car	0.01	0.00	0.01	97.90	0.00	0.00	0.00	1.93	0.00	0.09	0.06
MO	0.00	1.41	0.00	1.82	93.74	0.00	0.40	0.00	0.00	0.00	2.63
NMV	6.39	0.23	0.46	1.37	0.23	72.37	0.46	3.88	5.25	1.83	7.53
PE	0.00	1.73	0.00	0.13	0.45	0.06	93.48	0.06	0.06	0.00	4.03
PI	0.02	0.00	0.01	3.56	0.00	0.02	0.00	96.24	0.09	0.04	0.02
SUT	8.83	0.00	0.16	0.70	0.00	0.23	0.00	3.75	84.45	1.17	0.70
WV	0.08	0.00	0.21	6.77	0.00	0.12	0.00	1.24	0.33	90.59	0.66
BG	0.01	0.00	0.01	0.12	0.00	0.02	0.02	0.01	0.00	0.01	99.80

Table V: Evaluation of EDeN on the testing set

Network/Classes	AT	BI	Bus	Car	MO	NMV	PE	PT	SUT	WV	BG
EDeN (precision)	0.9368	0.9361	0.9910	0.9889	0.9587	0.8661	0.9650	0.8997	0.8868	0.9585	0.9951
EDeN (recall)	0.9451	0.8984	0.9794	0.9790	0.9374	0.7237	0.9348	0.9624	0.8445	0.9059	0.9980

4.6. Discussion of results

Table VI shows the summary of the results of the individual networks and EDeN. In Table VI, Acc, Prec, Rec, and CK refer to accuracy, precision, recall and Cohen Kappa score respectively.

Table VI: Summary of the results

Network	Acc.	Prec.	Rec.	CK	Avg. score
Network A	97.25	93.60	89.28	95.73	93.96
Network B	97.43	93.52	91.61	96.02	94.64
Network C	*97.59*	*94.26*	*91.85*	*96.26*	*94.99*
EDeN	**97.80**	**94.39**	**91.90**	**96.58**	**95.17**

In Table VI, Acc, Prec, Rec, CK refer to accuracy, precision, recall and Cohen kappa score. From Table VI, it can be observed that EDeN performs better than the individual networks. Furthermore, Network C, had better performance compared to Network A and Network B. This corroborates the fact that, the logical reasoning of Network C is able to solve the *dual class misclassification* problem.

On comparing EDeN with individual networks, EDeN had better accuracy, precision, recall and cohen kappa score. The reason for this is that, although there could be a possibility that Network C predicted some images wrong due to noise in some patches of the images, the weighted predictions of Network A and Network B for those patches were higher and hence overriding Network C's prediction. On the other hand, Network C was able to dominate whenever there was a genuine *dual class misclassification* problem.

We also evaluated EDeN on the validation set and reviewed the classified images. Fig. 6 shows some of the results.

Fig. 6: Classified images from the validation dataset using EDeN. The red text is the ground-truth and the green text is the predicted class. The images in the green frame are correct classification, the images in the red frame are wrong classifications and the images in the yellow frame are noisy data with wrong ground-truth but EDeN predicted them correctly.

In Fig. 6, the red color text is the ground-truth and the green color text is the predicted results. The images with the green box indicate correct classification and the images with the red box indicate incorrect classification. The images with the yellow box indicate that the ground-truth for those images was mislabeled and the predicted results were correct indicating that our model is robust to noisy data.

Thus, from our experiments we can conclude that using an ensemble of deep networks helped to improve performance, mitigate *dual class misclassification problem* and it is also robust to noisy labels.

5. Conclusions and Future work

In this paper, we introduced an ensemble of deep networks for the classification of vehicle surveillance images and performed extensive evaluation of our model on the MIO-TCD dataset which is a real-world traffic surveillance dataset. The results obtained from our evaluation showed that our ensemble of networks performed better than individual networks and it was robust to noisy labels. Future work will involve using our Ensemble of Deep Networks to perform localization and evaluating our approach on other datasets that are acquired under different time periods and environments.

Acknowledgment

This work was supported in part by NSF grant 1330110 and ONR grant N00014-12-1-1026. The contents of the information do not reflect the position or policy of US Government.

References

[1] H. Cho, Y.W. Seo, B.V. Kumar and R.R. Rajkumar. "A multi-sensor fusion system for moving object detection and tracking in urban driving environments." ICRA, 2014, pp. 1836-1843.

[2] D. Held, L. Jesse and T. Sebastian. A probabilistic framework for car detection in images using context and scale. ICRA, 2012.

[3] C. Caraffi, V. Tom, T. Ji, S. Jan and M. Ji. "A system for real-time detection and tracking of vehicles from a single car-mounted camera." ITSC, 2012, pp. 975-982.

[4] A. Jazayeri, H. Cai, J.Y. Zheng and M. Tuceryan. "Vehicle detection and tracking in car video based on motion model." TITS, 2011, 12(2), pp. 583-595.

[5] P.F. Felzenszwalb, R.B. Girshick, D. McAllester and D. Ramanan. "Object detection with discriminatively trained part-based models." TPAMI, 2010, 32(9), pp. 1627-1645.

[6] A. Krizhevsky, I. Sutskever and G.E. Hinton. "Imagenet classification with deep convolutional neural networks. Advances in NIPS.", 2012.

[7] S. Pierre, D. Eigen, X. Zhang, M. Mathieu, R. Fergus and Y. LeCun. "Overfeat: Integrated recognition, localization and detection using convolutional networks." arXiv preprint arXiv:1312.6229, 2013.

[8] C. Szegedy, S. Reed, D. Erhan, D. Anguelov and S. Ioffe. "Scalable, high-quality object detection.", 2014, arXiv preprint arXiv:1412.1441.

[9] C. Szegedy, T. Alexander, and E. Dumitru. "Deep neural networks for object detection." Advances in NIPS, 2013.

[10] R. Girshick, J. Donahue, T. Darrell and J. Malik. "Rich feature hierarchies for accurate object detection and semantic segmentation." CVPR, 2014, pp. 580-587.

[11] J.R. Uijlings, K.E. Van De Sande, T. Gevers and A.W. Smeulders. "Selective search for object recognition." IJCV, 2013, 104(2), 154-171.

[12] B. Huval, T. Wang, S. Tandon, J. Kiske, W. Song, J. Pazhayampallil, M. Andriluka, P. Rajpurkar, T. Migimatsu, R. Cheng-Yue and F. Mujica. "An empirical evaluation of deep learning on highway driving.", 2015, arXiv preprint arXiv:1504.01716.

[13] S. Wang, Z. Li, H. Zhang, Y. Ji and Y. Li. "Classifying vehicles with convolutional neural network and feature encoding." INDIN, 2016 pp. 784-787.

[14] R. Theagarajan, N. S. Thakoor and B. Bhanu. "Robust visual rear ground clearance estimation and classification of a passenger vehicle." ITSC, 2016.

[15] X. Chen, S. Xiang, L.C. Liu and C.H. Pan. "Vehicle detection in satellite images by parallel deep convolutional neural networks." ACPR, 2013, pp. 181-185.

[16] S. Sivaraman, and M.M. Trivedi. "Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis." TITS, 2013, 14(4), pp. 1773-1795.

[17] N. Saunier, H. Ard, J.P. Jodoin, A. Laureshyn, M. Nilsson, A. Svensson, L.M. Moreno, G.A. Bilodeau, and K. Strm. "A public video dataset for road transportation applications." Transportation Research Board Annual Meeting Compendium of Papers, 2014, pp. 14-2379.

[18] C. Papageorgiou and T. Poggio, CBCL car database, Center for Biological Computational Learning, 2000.

[19] H.M Dee, and S.A. Velastin. "How close are we to solving the problem of automated visual surveillance?." MVA, 2008, 19(5-6), pp.329-343.

[20] G. Salvi. "An automated nighttime vehicle counting and detection system for traffic surveillance." CSCI, 2014, 1.

[21] K. He, X. Zhang, S. Ren and J. Sun. "Deep residual learning for image recognition." CVPR, 2016 pp. 770-778.

[22] Pytorch: Tensors and Dynamic neural networks in Python with strong GPU acceleration https://github.com/pytorch/pytorch

[23] Traffic Surveillance Workshop and Challenge, Conference on Computer Vision and Pattern Recognition, 2017: http://podoce.dinf.usherbrooke.ca/challenge/tswc2017/

[24] Y. Deng, P. Luo, C.C. Loy and X. Tang. "Pedestrian attribute recognition at far distance." ACME, 2014, pp. 789-792.

[25] N.S. Thakoor and B. Bhanu, "Structural signatures for passenger vehicle classification in Video," TITS, 2013, 14(4), pp. 1796-1805.

[26] M. Kafai and B. Bhanu, "Dynamic Bayesian networks for vehicle classification in video," TII, 2012, 8(1), pp 100-109.

[27] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma and A.C. Berg. "Imagenet large scale visual recognition challenge.", 2015, IJCV, 115(3), pp. 211-252.

[28] "RITA BTS Table 1-11". *US Bureau of Transportation Statistics*. Retrieved 2015-02-19. *http://www.rita.dot.gov/bts/sites/rita.dot.gov.bts/files/table_01_11_1.xlsx*