Vehicle Type Classification Using Bagging and Convolutional Neural Network on Multi View Surveillance Image

Pyong-Kun Kim and Kil-Taek Lim
Electronics and Telecommunications Research Institute (ETRI)
Daegu, South Korea
{iros,ktl}@etri.re.kr

Abstract

This paper aims to introduce a new vehicle type classification scheme on the images from multi-view surveillance camera. We propose four concepts to increase the performance on the images which have various resolutions from multi-view point. The Deep Learning method is essential to multi-view point image, bagging method makes system robust, data augmentation help to grow the classification capability, and post-processing compensate for imbalanced data. We combine these schemes and build a novel vehicle type classification system. Our system shows 97.84% classification accuracy on the 103,833 images in classification challenge dataset.

1. Introduction

Recently, pattern recognition and computer vision technology have been widely applied for safety and convenience in our lives. The government has been building an information system that monitors traffic like cars, motorcycles, bicycles, and pedestrians. The cameras installed on urban roads improve crime prevention and transportation capacity. Accordingly, data analysis of traffic image is helpful for police crime prevention and transportation investment. Traditional traffic surveillance systems consist of manually inspecting video and attaching labels to the required frames, but recent research automates these. Detecting objects of interest on images and classifying the specific information like vehicle type, moving direction, and unusual action can be automatically processed. For these automatic processes, the vehicle type classification is important, but it is difficult to classify traffic objects because many of the images from camera have poor quality and various view points. Recent development in pattern recognition, like deep neural network has made successful progress in automatic image recognition, and it is widely used in the surveillance video system. But deep neural network needs much of image data, and it is not easy to acquire enough data to train deep neural network.

Fortunately, Traffic Surveillance Workshop and Challenge is offering large vehicle image dataset for surveillance. This image dataset, called "classification challenge dataset" [1] is divided into 11 categories. Each is articulated truck, bicycle, bus, car, motorcycle, non-motorized vehicle, pedestrian, pickup truck, single unit truck, work van, and background. This data is offered for the challenge and the goal of the classification challenge is to correctly label each image.

The classification challenge dataset consists of 648,959 images acquired at different times by thousands of traffic cameras deployed all over Canada and the United States. 519,164 images of them are for training and the others are for testing. The number of images in each category is very different and imbalanced. Table 1 shows the composition of the training dataset. It reveals that background and car classes are dominant. In contrast, bicycle, motorcycle, and non-motorized vehicle are too deficient. This imbalanced distribution makes it difficult to build a stable and robust classification system.

This paper proposes a new vehicle type classification scheme for the traffic surveillance system. The rest of this paper is organized as follows. The related works are given in Section 2. A new vehicle type classification method is represented in Section 3. The experimental results are conducted in Section 4. Finally, some conclusions will be shown in Section 5.

2. Related works

Wang *et al*. [10] use HOG Descriptor and SVM classifier for three-class vehicle type classification from surveillance videos. Llorca *et al*. [7] also use HOG Descriptor and SVM classifier for vehicle logo recognition in traffic images. The logo location is roughly assumed from the license plate location and sliding window method is employed. These studies tell us that HOG-SVM [3] is a good method for classifi-

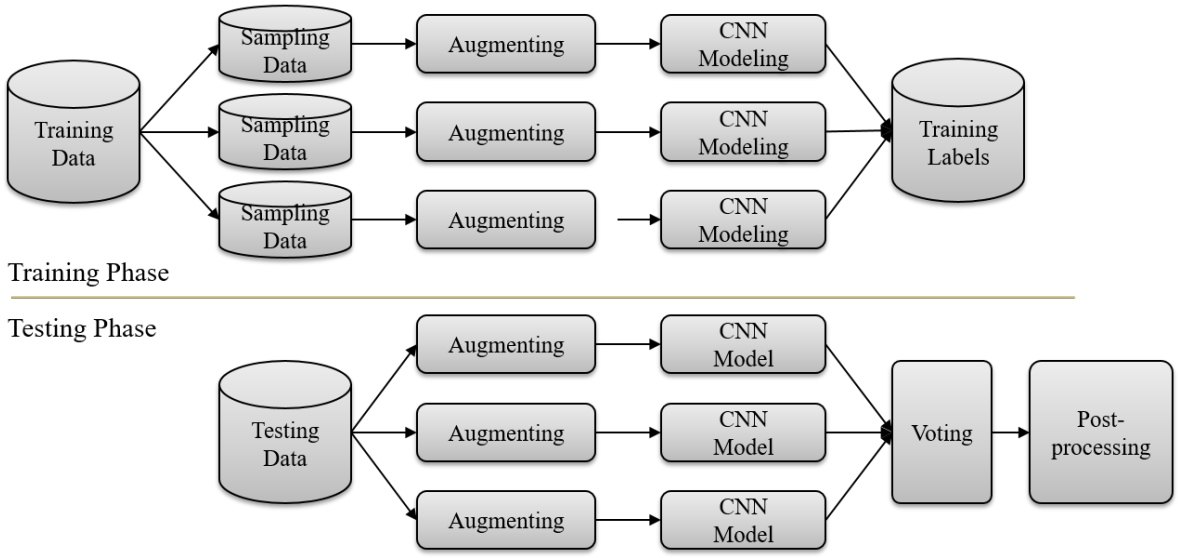

Figure 1. Overall architecture of the proposed approach.

Table 1. Composition of training dataset and ratio.

Category	Number	Ratio
Articulated truck	10,346	2.0%
Background	160,000	30.8%
Bicycle	2,284	0.4%
Bus	10,316	2.0%
Car	260,518	50.2%
Motorcycle	1,982	0.4%
Non-motorized vehicle	1,751	0.3%
Pedestrian	6,262	1.2%
Pickup truck	50,906	9.8%
Single unit truck	5,120	1.0%
Work van	9,679	1.9%
Total	519,164	100%

cation on fixed-view images of good quality .

Pearce and Pears [8] recognize the Make and Model from frontal images of cars. They use Harris corner strengths and naive Bayes classifier on recursive partitions and get 96.0% accuracy. In spite of their good result, the number of sample images just is as small as 262. Dong *et al*. [4] classify vehicle types using a Convolutional Neural Network on 9,850 high-resolution vehicle frontal-view images. Their performance is good, but the number of vehicle type is just six.

Yang *et al*. [11] collects a large-scale dataset "CompCars", that cover not only different car view, but also their different internal and external parts. Firstly, they employ the Overfeat model for find-grained model classification. They examine classifications from different views and find that rear side view is better than others. But the results from other views, like "front", "rear", "side", "front-side", are similar. The best performance is achieved by "All-View" model, although it does not leverage the information of viewpoints. This result is an example of the discriminative capability of CNN model from different views. Later they report updated results that the classification accuracy using GoogLeNet is 98.4%. Considering the number of output classes, this performance is excellent. But note that the quality of the input image is also good compared to classification challenge dataset.

3. Proposed method

To get a good vehicle type classification result from large dataset like classification challenge dataset, we propose a new system with four basic concepts, Deep Learning, Bagging [2], Data Augmentation, Post-processing. The processing diagram of the proposed approach is displayed in Figure 1.

3.1. Deep learning

Previous works show that the deep learning methods represent good results for the image recognition of the various viewpoints. Because classification challenge dataset is also acquired from the different view and its size is very large, it has a necessary condition to apply deep learning method. So we adopted deep learning as a basic method.

The next consideration is which model is best for the vehicle type problem. There are many deep learning models in the image classification field and each has its own strength. We tested some famous models and a few simple

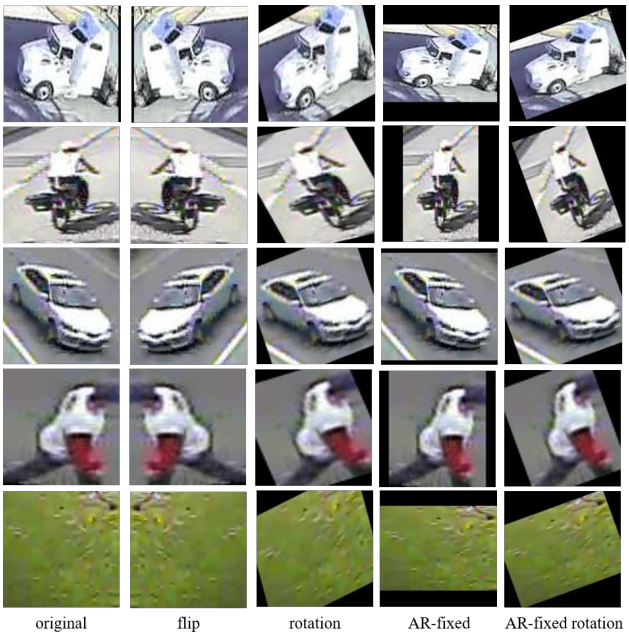

Figure 2. Examples of original and augmented images

CNN (Convolutional Neural Network) models, and adopted a middle size CNN model as a basic model.

3.2. Bagging

Classification challenge dataset has many images, but class distribution is not fair. Small size class can be overwhelmed by large class. To get a generalization capability, we decided to utilize randomness in bagging. We randomly sampled a half number from the training dataset and created a new training dataset. We repeated the sampling process and created many sampled training datasets with the same number of different images. These datasets were used for training each deep learning model, so we got many different models with the same function.

3.3. Data augmentation

It is well known that data augmentation helps to increase performance of machine learning model [9]. It is more helpful when training data size is not large. In our case the data size gets to be a half of the origin by bagging, so data augmentation would be a good helper.

There are many methods to create augmented image from an original image. To make a process simple, we considered only flip and rotation images. We believe that other type augmentation would be helpful, but performance increase would be saturated according to augmented data size.

In relation to this subject, we considered to keep an aspect ratio of an image fixed without stretching the image. When a rectangle image is rotated, the resulting image has a black region in the corners because of the empty information in the original image. Refer to third column in Figure 2. This might cause the performance to decrease, an examination was implemented. It was expected that the black regions from the augmented rotation image would not be a problem if the original image also had black regions. We trained each model with the stretched images and aspect ratio keeping images separately, compared each result, and found that there is no significant performance difference between two models. Figure 2 shows the examples of original and augmented images. Rows are specific classes and column are original, flip, rotation, aspect ratio fixed and rotation of aspect ratio fixed images each. The left three columns were used in our systems.

3.4. Post-processing

The imbalanced data make it difficult that the class with small numbers is classified rightly, and make the class with large numbers dominant. We tried to dissolve this problem by imposing weights to some rare classes. Our bagging system has many models and consequently outputs many results for a single input image. The voting process is necessary and it is natural to have case with same votes. In this case, class priority is necessary. We tested three priorities and found that it was better to classify it to be a rare class when the same votes.

Moreover, even if the voting number was different, we found that we got a better result when we imposed weight to rare classes. We expected that the rare class would be less selected by the trained model because of the lack of training data. Error distribution on the verification data shows that the error rate of rare class is apt to be more than that of frequent class. So we examined the error rate of each class and added weights to the voting system in proportion to the error rate and the number of basic models. The detailed equation about weights is like this.

$$w_i = c\frac{e_i}{\sum e_i}N_m, \tag{1}$$

where w_i means weight of i-class, N_m is the number of basic model, e_i is an error rate of each class, and c is a constant multiplier defined heuristically. Each weight is added during voting.

4. Experimental results

As the traditional way, we splited classification challenge dataset into two parts for training and verifying with a ratio of 4 vs 1. Splinting was implemented periodically so every fifth images were regarded as verification images and others as training images. Although we know that Nth folding way is recommended, we just examined one training dataset because it takes a long time to train a deep learning

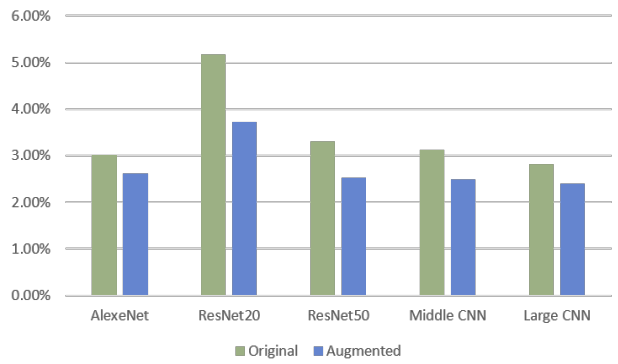

Figure 3. Error rates of various deep learning models on verification dataset

model. We believe that large data size and bagging system would reduce the problem from this.

First, we examined some famous deep learning models and our own CNN models. To reduce the examining time, it was implemented on not sampling dataset, but a total training dataset. We used CNTK of Microsoft [12] for the deep learning library. We tested AlexNet [6], ResNet20 and ResNet50 [5] in CNTK Examples [12] by adapting some parameters if necessary. A middle size CNN and a large size CNN were created by stacking some convolution layers and max pooling layer repeatedly by ours. Figure 3 shows the results of basic models examinations. From Figure 3, we have known that there is a little difference in performance between deep learning models, and the difference decrease if augmented data is added. Despite a large size CNN had the best result, we chose a middle size CNN model as a basic model because bagging system needed many models and the training time was critical to build the system. In our experiment, a large size CNN system takes 11 times more than a middle CNN system to be trained. In addition, the performance is similar on augmented dataset.

Next, we considered how to sample data from training data. Because the distribution of each class was imbalanced, we expected that if the sampling data was more balanced than original data by appropriate manipulation during sampling, it would give rise to good effect to the final result. To check this, we sampled data with random, with the same number of class, and in proportion to the number of class. As our experiment, the performance of bagging system had little relation to sampling methods and strong relation to the sampling number on the dataset, so we adopt random sampling method. Table 2 shows the structure of a middle size CNN, which has 3,816,299 parameters. A large size CNN has the same structure, but four times filter size.

We sampled a half number from training set randomly and created sampling datasets repeatedly to 21 sets. We added the augmented data like flip, clockwise, and anti-clockwise rotation to each one and trained 21 middle size CNNs on these augmented sampling datasets. Table 3 shows the training and verification errors of some basic CNN models on sampling datasets which are augmented with flip and rotations. The performances are similar to that of full training dataset, and the difference in verification error rate is just about 0.3%. In other words, thanks to the augmentation, the performance decreases just 0.3% even if a half of data is not used.

We combined the basic models results by maximum vot-

Table 2. Structure of a middle size CNN. It takes $128 \times 128 \times 3$ images as input.

Type	Filters	Size/Stride	Output
Convolutional	16	3×3	128×128
Maxpool		$2 \times 2/2$	64×64
Convolutional	32	3×3	64×64
Convolutional	32	3×3	64×64
Maxpool		$2 \times 2/2$	32×32
Convolutional	64	3×3	32×32
Convolutional	64	3×3	32×32
Convolutional	64	3×3	32×32
Maxpool		$2 \times 2/2$	16×16
Convolutional	128	3×3	16×16
Convolutional	128	3×3	16×16
Convolutional	128	3×3	16×16
Convolutional	128	3×3	16×16
Maxpool		$2 \times 2/2$	8×8
Convolutional	256	3×3	8×8
Convolutional	256	3×3	8×8
Convolutional	256	3×3	8×8
Convolutional	256	3×3	8×8
Convolutional	256	3×3	8×8
Maxpool		$2 \times 2/2$	64×64
Dense(Dropout)		Global	128
Dense(Dropout)		Global	64
Linear		Global	11

Table 3. Performances of basic CNN models trained on the different sampling dataset. Each dataset is augmented with flip and rotations

Sampling Models	Training Error	Verification Error
Model 1	0.13%	2.82%
Model 2	0.13%	2.78%
Model 3	0.13%	2.80%
Model 4	0.12%	2.77%
Model 5	0.15%	2.82%
Model 6	0.13%	2.83%
Model 7	0.12%	2.85%

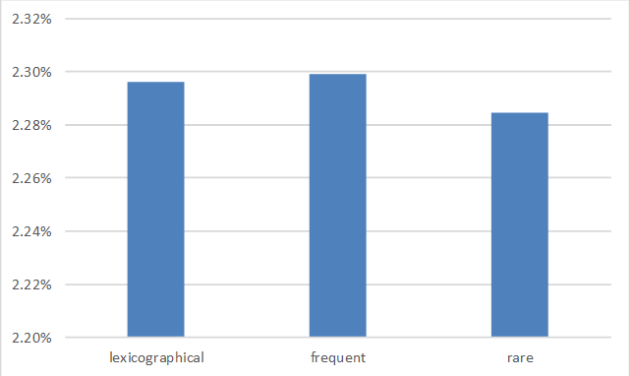

Figure 4. Error rates according to voting policy

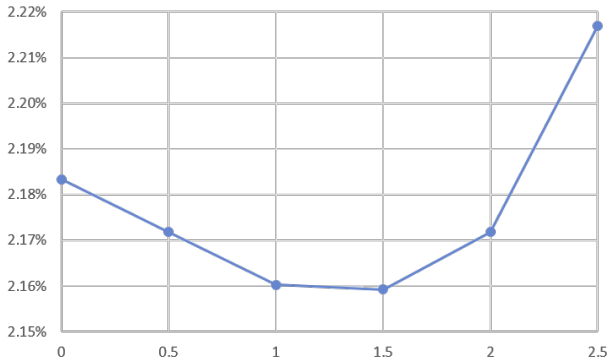

Figure 6. Error rate according to post-processing

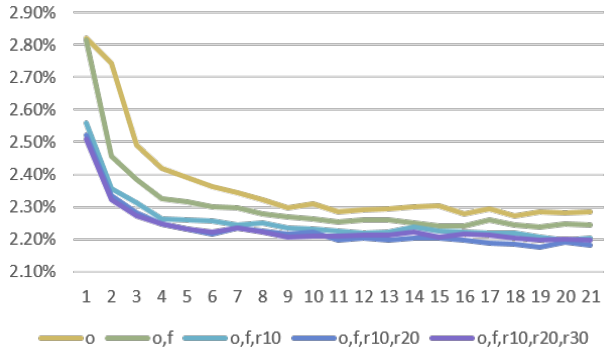

Figure 5. Error rates according to the number of basic models.

Table 4. Classification performances of four deep learning models on the verification dataset. Each model is trained on the augmented dataset.

Models	Accuracy	Mean Recall	Mean Precision	Cohen Cappa
AlexNet	0.9731	0.8983	0.9167	0.9581
ResNet50	0.9752	0.9057	0.9245	0.9612
Large CNN	0.9760	0.9093	0.9341	0.9625
Proposed	0.9784	0.9135	0.9384	0.9663

ing policy and inspected the performance according to the number of results. There are a few things to consider when voting. First, which class should be selected when the voting results are same. Second, how many basic models are necessary to get to best performance. Third, is it necessary to use the augmented image when test phase. Finally, is post-processing useful.

We examined all, and got the results like Figure 4, Figure 5 and Figure 6. Figure 4 shows that it is better to classify as an rare class when voting results are same. In Figure 5, x-axis represents the number of basic models and y-axis represents error rate. Each line represents the result where the different testing dataset is used. 'o', 'f', 'r10', 'r20', and 'r30' means original, flip, 10° rotation, 20° rotation, and 30° rotation dataset each. The graph shows that the performance increase by adding results from different basic models, but saturate slowly after 10. Also the performance increase by adding the augmented testing data to rotation 20°, and decrease to rotation 30°. In Figure 6, x-axis represents the constant multiplier c in (1), and y-axis represents error rate. We added weights from (1) to voting results to compensate the model for imbalanced data. In Figure 6, zero-value in x-axis means no weight is added to voting results. It shows that performance can be increased if proper weight is reflected during voting.

From these we can know that 1) rare class would be better when the voting values are same, 2) more than 10 basic models are recommanded, 3) it is better to use the augmented testing image to some extent, and 4) post-processing can be useful.

Finally, we got 97.84% classification accuracy, 91.35% mean recall, 93.84% mean precision, and 96.63% Cohen Kappa Score on verification data. The performance comparison with other deep learning models are shown in Table 4. Although all models achieve very high performances for this task, the proposed system is the highest. Failed images to classify are shown in Figure 7. The images in each row belong to the same ground truth, but are predicted as other classes in our system. It reveals that some classes are hard to classify. Articulated trucks are similar to single unit trucks. Backgrounds contain a part of car. Some single unit trucks are confused with pickup trucks. Specially images in the pedestrian class, classified as bicycles by our system, are almost similar to that of bicycle class.

5. Conclusions

A novel vehicle type classification scheme for multi view surveillance image has been proposed in this paper. We propose four schemes to get high performance in the ve-

Figure 7. Images that fail to classify on the verification dataset

hicle classification. Deep learning is indispensable in the case of the multi view point, but powerful deep learning model is not absolutely necessary. Bagging system helps to be a robust system. Data augmentation increases the performance of the basic model in the bagging system. And post-processing can compensate for imbalanced data distribution. A vehicle type classification system created by combining these schemes has shown good performance for classification challenge dataset. In the future, more intelligent policy on sampling data and selecting the basic model in the bagging system will be developed.

References

[1] The miovision traffic camera dataset (mio-tcd). `http://podoce.dinf.usherbrooke.ca/challenge/dataset/`. Accessed: 2017-04-28.

[2] L. Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[4] Z. Dong, Y. Wu, M. Pei, and Y. Jia. Vehicle type classification using a semisupervised convolutional neural network. *IEEE Transactions on Intelligent Transportation Systems*, 16(4):2247–2256, 2015.

[5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

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

[7] D. Llorca, R. Arroyo, and M. Sotelo. Vehicle logo recognition in traffic images using hog features and svm. In *Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on*, pages 2229–2234. IEEE, 2013.

[8] G. Pearce and N. Pears. Automatic make and model recognition from frontal images of cars. In *Advanced Video and Signal-Based Surveillance (AVSS), 2011 8th IEEE International Conference on*, pages 373–378. IEEE, 2011.

[9] P. Y. Simard, D. Steinkraus, J. C. Platt, et al. Best practices for convolutional neural networks applied to visual document analysis. In *ICDAR*, volume 3, pages 958–962. Citeseer, 2003.

[10] Y. C. Wang, C. C. Han, C. T. Hsieh, and K. C. Fan. Vehicle type classification from surveillance videos on urban roads. In *Ubi-Media Computing and Workshops (UMEDIA), 2014 7th International Conference on*, pages 266–270. IEEE, 2014.

[11] L. Yang, P. Luo, C. Change Loy, and X. Tang. A large-scale car dataset for fine-grained categorization and verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3973–3981, 2015.

[12] D. Yu, A. Eversole, M. Seltzer, K. Yao, Z. Huang, B. Guenter, O. Kuchaiev, Y. Zhang, F. Seide, H. Wang, et al. An introduction to computational networks and the computational network toolkit. *Microsoft Technical Report MSR-TR-2014–112*, 2014.