Combining Deep and Handcrafted Image Features for Vehicle Classification in Drone Imagery

***Abstract*—using unmanned aerial vehicles (UAV) as devices for traffic data collection exhibits many advantages in collecting traffic information. This paper presents an efficient method based on the deep learning and handcrafted features to classify vehicles taken from drone imagery. Experimental results show that compared to classification algorithms based on pre-trained CNN or hand-crafted features, the proposed algorithm exhibits higher accuracy in vehicle recognition at different UAV altitudes with different view scopes, which can be used in future traffic monitoring and control in metropolitan areas.**

***Keywords*—deep feature; handcrafted features; classification**

I. INTRODUCTION

Although vehicle recognition has been an area of great recent interest in the machine-learning community [1], most recent works consider more lateral and frontal views, either from wide area monitoring imaging or from on-board-like camera views [2][3]. Little prior research study has used drone imagery to build an on-road vehicle recognition. This work is an extension of the previous research published in [4]. A small dataset, Vehicle Recognition in Drone Imagery (VRDI), designed to address the task of small vehicle detection and recognition in drone images, was introduced as part of our previous work. The handcrafted feature, histogram of oriented gradients (HoG) was used for Sequential Minimal Optimization (SMO) training and classification. The experimental results showed that the classification accuracy was close to 90%. However, the classification performance degrades quickly when the number of training images at different orientations increase. The problem is that HoG is very sensitive to image rotation and the capability of handling the robustness to rotation by SMO classifier is limited.

The recent deep convolution neural network (ConvNet), which are acknowledged as the most successful and widely used deep learning approach in most of recognition and detection tasks [5][6] seems to be a solution here. However, new problems arise in this proposal. At first, training a deep network from scratch is not a feasible option when solving a classification problem with a small number of labelled training samples. On the other hand, directly training on ConvNets in this case would result in overfitting and reduce classification accuracy. At present, many recent works [7][8] have demonstrated that the features learned with deep CNNs pre-trained on large datasets such as ImageNet [9] can be transferable to many other recognition tasks with limited training data. The subsequent problem is that the features learned with pre-trained deep CNNs models are based on the lateral and frontal views, which do not represent the top views of objects well in drone imagery.

Within this context, the motivation for this work is to take advantage of features extracted from deep learning and selected handcrafted features together to address the task of small vehicle recognition in drone images. Vehicles to be recognized have different orientations, at different scales, occluded or masked. An overview of the complete training framework can be seen in Fig. 1.

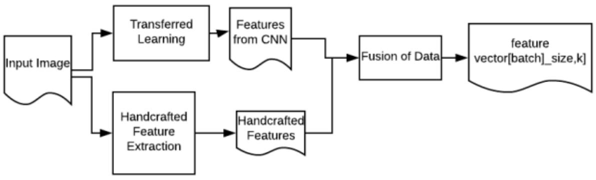


Fig. 1. The learning framework for vehicle recognition system

The remainder of this paper is structured as follows: it starts describing feature extraction via deep learning in section II. Section III details the various hand-crafted features and fine tunes parameters in each feature extraction process. Following that, the fusion of two kinds of features is presented in section IV. In section V, we thoroughly evaluate the possible combinations of handcrafted and deep-learned features that are needed to classify small sized vehicles in drone imagery. At last, we draw a conclusion in section VI.

II. FEATURE EXTRACTION FROM DEEP LEARNING

Most ConvNets contains millions of parameters. Directly learning so many parameters from only a few thousand training images is problematic. Since the target dataset, VRDI, is significantly smaller than the base dataset ImageNet in which most ConvNets are trained, the internal layers of the CNN can act as a generic extractor of mid-level image representation, then be re-used on further classification on a smaller dataset. The outcome is a classifier that fits the new dataset with significantly less work than retraining a new network.

In this work, Inception-ResNet-v2 (IRv2) [10] is preferred as it has been shown to achieve very good performance at relatively low computational cost [11]. The IRv2 was trained to recognize 1,000 object categories. The architecture of IRv2 is described in detail in [10]. Since IRv2 is already a trained model, we only fine-tune it to recognize types of vehicles in this work.

By removing the last two layers, dropout and softmax, we use the weights from the average pooling layer (which contains 1792 neurons) as a global feature representation of the input image. Thus, in the case of IRv2, a features vector size is 1792.

As Inception-ResNet-v2 requires inputs image to have the minimum size of 139 by 139 pixels while the size of most of the vehicles in drone imagery are small, the original images are required to scale up. The reason of up-sampling is as ConvNets become increasingly deep, the size of feature maps can vanish and “wash out” by the convolution and max-pooling operation at multiple layers. Therefore, input image size must be reasonable large.

III. HANDCRAFTED FEATURE EXTRACTION

Several popular hand-crafted feature descriptors have emerged in the literature, Local binary patterns (LBP)-based [12], histogram of oriented gradients (HoG) [13], and Bag-of-Visual-Words (BoVW) [14].

1) *Feature Extraction with LBP*

The LBP operator was originally designed for texture description. The operator assigns a binary label to every pixel of an image by thresholding the P sampling points on a circle of radius of R with the center pixel value and considering the result as a binary number. Then the binary feature vectors obtained through the application of LBP are captured in a histogram as a feature descriptor. To obtain a rotation invariant descriptor [3] and reduce the dimension of the descriptor, P-1 bitwise shift operations on the binary pattern are performed, and the smallest value is selected. A pattern is defined ‘‘uniform’’ if the number of transitions between ‘0’ and ‘1’ of the sequence is less or equal to two, with the number of different types of uniform patterns that can occur being P + 1. To describe a given image, the histogram of dimension P + 2 is extracted. It contains the occurrence of the P + 1 types of uniform patterns, and the number of non-uniform patterns.

Due to size of vehicles images are less than 100x100 pixels and, in our experiment, the radius parameter R used is set to 2 (pixels) and the number of points to consider are set to 16 by empirical experiment. Therefore, the length of feature vector is 18.

2) *Feature Extraction with HoG*

HoG feature descriptor has good performance in characterizing object shape and appearance and it is considered one of the most accurate feature descriptors for visual classification problems. Hence, we include HoG features in this work as one of the selected hand-crafted features.

Considering most training samples are less than 100x100, all the images for HoG feature extraction are resized to fit into a 64x64 pixel detection window first. Next, we use 2x2 cells, which are grouped into a bigger unit called block first and normalized based on all histograms in the block. If the number of histogram bins is set to 9, concatenating the histograms of the four cells within the block during the normalization creates a vector with 36 components (4 histograms x 9 bins per histogram). Then each component is divided by the magnitude of the vector to reduce the light variations or shadowing problems. If we overlap 50% of the blocks in a 64x64 detection window, it will be divided into 7 blocks across and 7 blocks vertically, for a total of 49 blocks. Each block contains 4 cells with a 9-bin histogram for each cell, for a total of 36 values per block. This brings the final vector size to 49 blocks per detection window x 4 cells per block x 9-bins per histogram = 1764 values.

3) *Bag-of-Visual-Words*

BoVW is defined as a histogram of visual words included in an image. The basic procedure to extract BoVW from an image in this work is the following. 1) The first step is called sampling. In this work, due to the training vehicles are less than 100x100 pixels and resized into 64x64, Dense SIFT features instead of SIFT [15] are sampled every 8x8 pixels, leading to 64 SIFT feature vectors 2) The second step is called dictionary generation. In this step, each local descriptor is assigned to the nearest cluster centre or visual word. All the visual words are combined as a visual dictionary and the size of dictionary, k is determined empirically. As shown in Fig.2, the performance in terms of precision stabilizes at K=1000. Given the dictionary of visual words, then a histogram is computed by counting how many descriptors are assigned to each visual word. 3) The last step is called the normalization. Term frequency - inverse document frequency (TF-IDF) [16] is used to weight the visual words. This normalization approach discounts the visual words those occur in all the images and concentrates on the ones that occur less frequently.

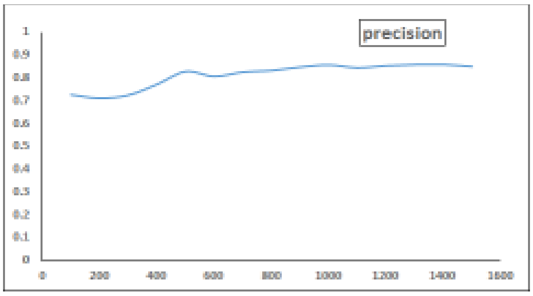


Fig. 2. Precision curve at different K

IV. FEATURE FUSION

After deep learning and handcrafted features extraction, a 1792-feature vector, denotes as, $\boldsymbol{X}$, extracted from IRv2 is concatenated with selected hand-crafted feature vectors, denotes as $\boldsymbol{Y}$, to form a better image representation.

Due to size of length of deep-learned feature vector, $\boldsymbol{X}$, is 1792 and small sized hand-crafted feature vector has little impact on the accuracy, only 1764-HoG and 1000-BoVW feature vectors are fused with $\boldsymbol{X}$ separately to evaluate the classification accuracy.

As the concatenated feature vectors come from different sources, two additional dense layers with ReLU activations are built to further reduce concatenated feature vectors into 1024 dimensions. There are two purposes of this dimension reduction. The first one is that the property of non-linearity in two ReLU layers allows the loss to be back-propagated and the weights to be accordingly updated. The second purpose is to reduce the processing time with lower dimensions. In such a way, feature vector comes from different sources can be fused together and characterise images after training. At first, the learning rate is set to 0.01, and then programmatically set to 1/1.01 if there is no improvement on loss function. At last, another dense layer and a soft-max activation layer are added for classification prediction. The detailed steps in fusion is illustrated in Fig. 3.

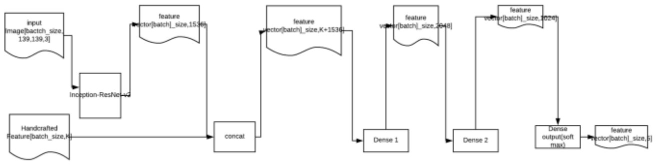


Fig. 3. Fusion of deep learned and handcrafted features, k=1764 for HoG and k=1000 for BoVW

V. EXPERIMENT AND DISCUSSION

In this section, experimental results of the proposed method on urban traffic videos are presented to analyze the performance of our approach. It starts with a brief introduction of the latest VRDI dataset. Then a list of performance metrics used in the following experiment are given. A novel method of using super-resolution on a sequence of low resolution testing images to improve classification accuracy is also analysed. Following that, results from our proposed method is compared with various standard classification methods.

A. VRDI dataset

The availability of vehicles recognition datasets from drone imagery is limited. Several datasets are available for the evaluation of vehicles detection tasks either frontal or side views of vehicles [17-18]. The current VRDI dataset contains 5508 images in four vehicle major classes and one background class, namely the ‘sedan’, ‘suv’, ‘truck’, ‘bus’, and the ‘background’ category. To exploit the spatial temporal relationship among training images during the tracking, vehicles are tracked and extracted as sequence of bounding boxes. Given above configurations, Fig. 4 snapshots a few images at different orientations per class in current data set.

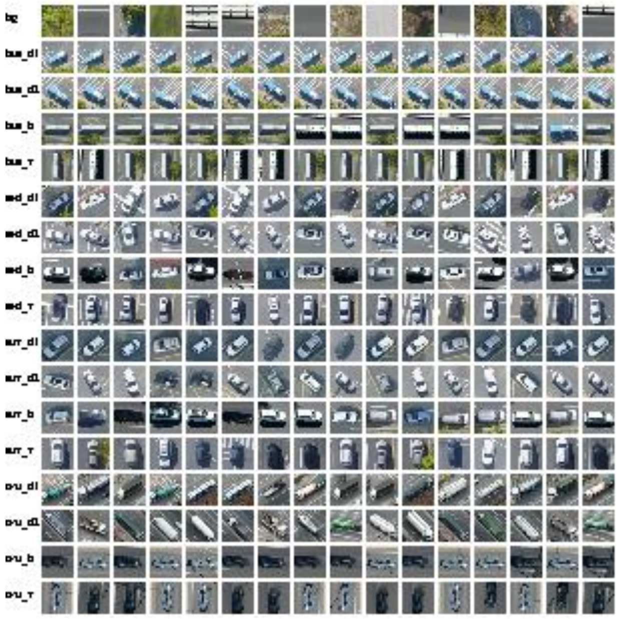


Fig. 4. Categorized sample images of the VRDI dataset

The process of acquiring training data is described as two steps in [4]. At the first step, vehicles images are manually cropped from the 1st frame of each video segment and the type, the frame ID in each video segment, the top left position of the cropped rectangle, the width, and the height of the rectangles are annotated. In the second step, object tracking on each annotated vehicle is performed to generate a sequence of images. Each image within the tracked sequence is fed into current classifier and can be re-trained to increase current training data set if it is classified with high error probability. In such way, the time-consuming work in labelling data during the acquisition of training data can be minimized.

B. Performance Metrics

To quantify the performance of classification, the following metrics: precision, recall, and F measure are used.

Given the definition of TP, FP, FN, and TN in Table I, we can then define the precision for each class as follows:

TABLE I. PERFORMANCE TABLE FOR INSTANCES LABELLED WITH A CLASS LABEL A.

	True label A	True not A
Predicted label A	True Positive (TP)	False positive (FP)
Predicted not A	False Positive (FN)	True negative (TN)

$$\mathrm{Precision} = \mathrm{TP}/(\mathrm{TP}+\mathrm{FP})\,, \qquad (\mathbf{1})$$

$$\mathrm{Recall} = \mathrm{TP}/(\mathrm{TP}+\mathrm{FN})\,, \qquad (\mathbf{2})$$

$$F_1 = 2(\mathrm{Precision} * \mathrm{Recall})/(\mathrm{Precision}+\mathrm{Recall})\,, \qquad (\mathbf{3})$$

C. Comparison Methods

We conducted a thorough experimentation on handcrafted feature vectors with SMO classifier, CNN-based classifier and handcrafted feature + CNN-based as shown in table 2, In the first three methods, handcrafted feature vectors were used for SMO training and classification.

In the fourth method, each 139*139*3 training image is feed-forwarding through IRv2 models pre-trained on ImageNet and a 1792 deep-learned feature vector is output first. Then the last dense layer and the final softmax layer are modified together to output 5 classes.

In the fifth and sixth methods, deep feature vector, $\boldsymbol{X}$, learned from IRv2 and handcrafted features, $\boldsymbol{Y}$, are concatenated together first, then sent to a shallow neutral network designed in Fig.3 for final classification prediction. The only difference between two different methods is a simple element-wise scaling operation $10 * \boldsymbol{Y}$ is applied on 1764-HoG feature vector $\boldsymbol{Y}$ before concatenation. The reason of this scaling up operation is each element in CNN feature vector ranges between 0 to 10 while each element in HOG feature vector ranges between 0 and 1.

TABLE II. CLASSIFICATION METHODS

Classification Methods
LBP+SMO
HoG+SMO
BoVW+SMO
CNN
CNN+HoG
CNN+BoVW

D. Super-Resolution to Improve Classification Accuracy

Although using high quality images for object classification would be ideal, this is not always possible in practice in drone imagery. In these cases, performing transformations to increase image quality at testing stage proves to be useful in the attempt to identify and classify relatively small objects. At pre-processing stage, small testing images are up-sampled to match the input image size of the deep learning network. Super-resolution in our experiment has been shown to outperform basic interpolation methods in terms of recognition accuracy.

Figure 5 below shows that super resolution is applied on sequences of low resolution tracked frames to generate higher resolution images.

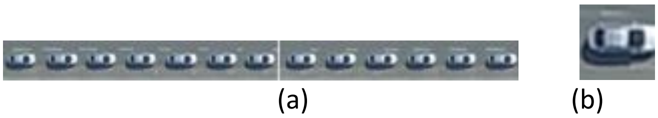


Fig. 5. Visual Appearance Improved Image (b) after SR on (a)

In table 3 and 4, 11 image testing sequences are selected against CNN+HOG, CNN+ BoVW respectively. The original dimension of each testing image varies from 25x25 to 48x48, as shown in column 3. As each testing image is up-sampled to 139x139 to match the input size in CNN, details are missing in high-resolution and accuracy stays poor in column 4. After applying SR at scale 2, missing details in high-resolution image based on similarities between the low-resolution images are recovered, resulting much higher precision as shown in column 6 of both tables.

TABLE III. COMPARISON OF THE ACCURACY BEFOR AND AFTER SR CNN+HOG.

image sequence	#images	dimension before SR	accuracy before SR	dimension after SR*2	accuracy after SR*2
sed_h_o85	35	26x26	69%	38x38	97%
sed_h_o87	59	28x28	62%	42x42	100%
sed_h_o88	12	28x28	54%	42x42	100%
sed_h_o89	32	29x29	58%	44x44	94%
sed_v_o81	28	26x26	52%	38x38	71%
suv_h_o71	54	48x48	44%	82x82	83%
suv_h_o73	44	40x40	44%	66x66	89%
sed_v_o94	19	27x27	37%	40x40	94%
sed_v_o96	38	25x25	63%	36x36	54%
sed_v_o98	46	28x28	72%	42x42	93%

TABLE IV. COMPARISON OF THE ACCURACY BEFOR AND AFTER SR CNN+BOVW.

image sequence	#images	dimension before SR	accuracy before SR	dimension after SR*2	accuracy after SR*2
sed_h_o85	35	26x26	72%	38x38	100%
sed_h_o87	59	28x28	82%	42x42	100%
sed_h_o88	12	28x28	57%	42x42	100%
sed_h_o89	32	29x29	75%	44x44	94%
sed_v_o81	28	26x26	63%	38x38	93%
suv_h_o71	54	48x48	78%	82x82	100%
suv_h_o73	44	40x40	56%	66x66	100%
sed_v_o94	19	27x27	21%	40x40	50%
sed_v_o96	38	25x25	27%	36x36	35%
sed_v_o98	46	28x28	37%	42x42	82%

E. Experimental Results

To evaluate the performance of those methods, table 5 lists the category types and number of images per category in the 1st two columns and the results measured in precision, recall and F-Measure per class in remaining columns.

Given the availability of each types of sequence in the current testing data set, the performance of deep-learned features mixed with traditional features are significant better comparing to deep learned features and handcrafted features alone. LBP presents the worst performance on average due to its poor representation of small objects. Similarly, CNN presents the second worst performance on average due to the fact extracted features are washed out when small input training sample size. HoG based method performs slightly better than CNN-based but fails to outperform remaining methods due to increasing number of rotated images. BoVW performs almost well as deep learned CNN+ HOG due to its rich representation, but it still cannot outperform CNN+ BoVW.

It is of note that the experiment result is very promising given that both the precision and recall rates are well above 80% when deep features are mixed with hand-crafted features. Some vehicle classification results from the real-time videos are shown in Fig. 6. As we can see, most of the vehicles have been correctly classified

VI. CONCLUSION

In this paper, we present an efficient method which exploits learned feature from pre-trained CNNs and mixes traditional hand-crafted features to improve classification accuracy for small-sized vehicle classification. The proposed method has been evaluated and compared with standard classification methods on real-world videos. The effectiveness of the proposed algorithm to robust vehicle classification is demonstrated for a variety of real environments given current dataset.

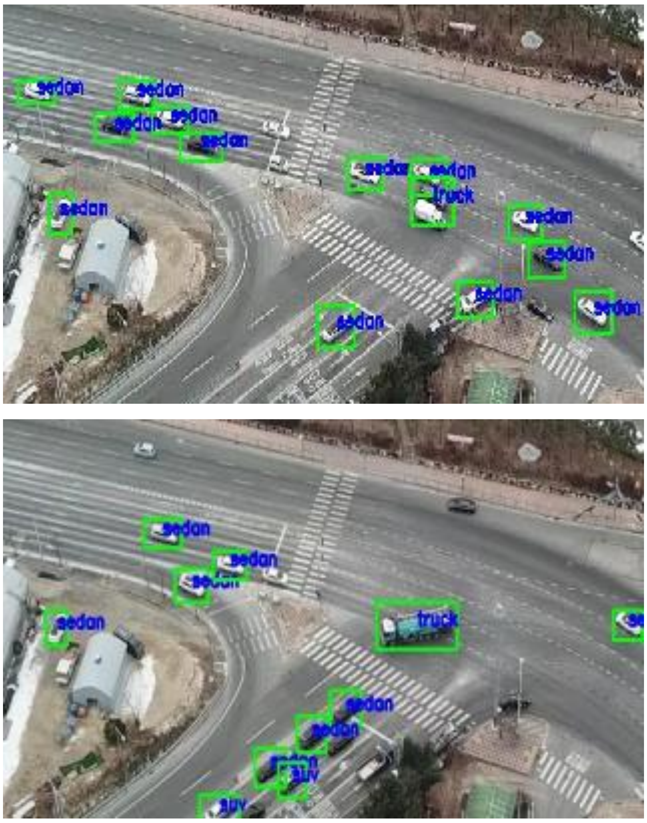
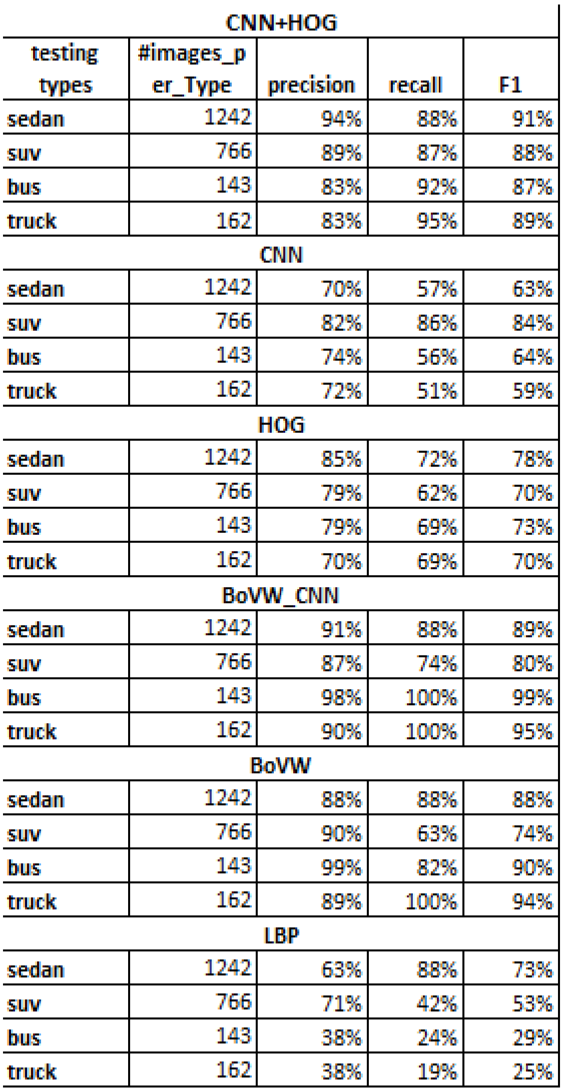


Fig. 6. Sample Vehicle Classification Results

TABLE V. CLASSIFICATION PERFORMANCE.



testing types	#images_per_Type	precision	recall	F1
CNN+HOG				
sedan	1242	94%	88%	91%
suv	766	89%	87%	88%
bus	143	83%	92%	87%
truck	162	83%	95%	89%
CNN				
sedan	1242	70%	57%	63%
suv	766	82%	86%	84%
bus	143	74%	56%	64%
truck	162	72%	51%	59%
HOG				
sedan	1242	85%	72%	78%
suv	766	79%	62%	70%
bus	143	79%	69%	73%
truck	162	70%	69%	70%
BoVW_CNN				
sedan	1242	91%	88%	89%
suv	766	87%	74%	80%
bus	143	98%	100%	99%
truck	162	90%	100%	95%
BoVW				
sedan	1242	88%	88%	88%
suv	766	90%	63%	74%
bus	143	99%	82%	90%
truck	162	89%	100%	94%
LBP				
sedan	1242	63%	88%	73%
suv	766	71%	42%	53%
bus	143	38%	24%	29%
truck	162	38%	19%	25%

REFERENCES

[1] S. Sivaraman and M. M. Trivedi., "A general active-learning framework for on-road vehicle recognition and tracking," A general active-learning framework for on-road vehicle recognition and tracking, vol. 11, no. 2, pp. 267-276, 2010.

[2] G. Ballesteros and . L. Salgado, "Optimized HOG for on-road video based vehicle verification," in IEEE 22nd European Signal Processing Conference (EUSIPCO), 2014.

[3] S. Agarwal, A. Awan and D. Roth, "Learning to detect objects in images via a sparse, part-based representation," IEEE transactions on pattern analysis and machine intelligence, vol. 26, no. 11, pp. 1475-1490., 2004.

[4] X. Le, J. Jo, S. Youngbo and D. Stantic, "Detection and Classification of Vehicle Types from Moving Backgrounds," in The 5th International Conference on Robot Intelligence Technology and Applications, Daejeon, KOREA, 2017.

[5] Z. Dong , M. Pei , Y. He , T. Liu , Y. Dong and Y. Jia , "Vehicle type classification using unsupervised convolutional neural network," in IEEE 22nd International Conference in Pattern Recognition (ICPR), 2014.

[6] P. K. Kim and K. T. Lim, "Vehicle Type Classification Using Bagging and Convolutional Neural Network on Multi View Surveillance Image.," in IEEE Conference In Computer Vision and Pattern Recognition Workshops (CVPRW), 2017.

[7] S. Wang, Z. Li, , H. Zhang, Y. Ji and Y. Li, "Classifying vehicles with convolutional neural network and feature encoding.," in IEEE 14th International Conference In Industrial Informatics, 2016.

[8] M. Oquab, L. Bottou, I. Laptev and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks. In Proceedings of the (pp. 1717-1724).," in 2014, IEEE conference on computer vision and pattern recognition.

[9] O. J. D. H. S. J. K. S. S. S. M. Z. H. e. a. Russakovsky, "Imagenet large scale visual recognition challenge," International Journal of Computer Vision , vol. 115, no. 3, pp. 211-252, 2015.

[10] C. Szegedy, S. Ioffe, V. Vanhoucke and A. Alemi, "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning.," AAAI , pp. 4278-4284, 2017 .

[11] F. Chollet, "Keras Applications," [Online]. Available:https://keras.io/applications/#inceptionresnetv2. [Accessed 22 01 2018].

[12] Ojala, Timo, Matti Pietikainen, and Topi Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns" IEEE Transactions on pattern analysis and machine intelligence 24.7 (2002): 971-987.

[13] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection.," in IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, 2005.

[14] Iscen, A., Tolias, G., Gosselin, P.H. and Jégou, H., A comparison of dense region detectors for image search and fine-grained classification. IEEE Transactions on Image Processing, 24(8), (2015): pp.2369-2381.

[15] Lowe, David G. "Distinctive Image Features from Scale-Invariant Keypoints". International Journal of Computer Vision. 60 (2),(2004) : 91–110

[16] Yang, J., Jiang, Y.G., Hauptmann, A.G. and Ngo, C.W., September. Evaluating bag-of-visual-words representations in scene classification. In Proceedings of the international workshop on Workshop on multimedia information retrieval, 2007, (pp. 197-206). ACM.

[17] M. Everingham, L. Van~Gool and C. K. Williams, "The PASCAL Visual Object Classes Challenge 2012 Results," 2012. [Online]. Available: http://www.pascal-network.org/challenges/VOC/voc2012/ workshop/index.html.

[18] Z. Dong, M. Pei, Y. He, T. Liu and Y. Jia, "Vehicle type classification using unsupervised convolutional neural network," in IEEE 22nd International Conference on Pattern Recognition, 2014.