VEHICLE CLASSIFICATION USING ALEXNET AND EnAET

***Abstract* - Image processing is a method to perform some operations on an image, in order to get an enhanced image or to extract some useful information from it. Vehicle classification has become important due to its several applications such as smart parking systems, fuel determination, traffic analysis and monitoring speed of vehicles. In this paper, the algorithms tested with the MIO-TCD dataset are AlexNET and EnAET. AlexNet is a CNN based architecture having several advantages such as speed and accuracy over traditional CNN architectures. In EnAET architecture several different transformations are used to improve its performance. The preprocessing of dataset has been done using image augmentation in this paper. AlexNet showed 78% accuracy on the training dataset while EnAET showed comparatively a very high accuracy of 98.3% while classifying vehicles using MIO-TCD dataset. EnAET proved to be a better model than AlexNet for vehicle classification.**

***Keywords*: - Image processing, vehicle classification, deep learning and computer vision.**

I. INTRODUCTION

Vehicle Classification can provide various advantages to traffic applications such as traffic analysis, speed monitoring, helping traffic police in managing and tracking traffic during busy days. Vehicle classification is used to classify vehicles into different categories. There are several vehicle classification-based methods which were made to improve the accuracy of classification of vehicles without using hardware components except digital cameras. Studies dealing with a comprehensive study of the effect of spatial resolution and color of digital images on vehicle classification has shown that using different techniques used in vehicle classification can produce different results in accuracy. Few studies compares different vision-based classification methods and deep learning models to classify vehicles into categories by comparing their accuracy when applied to BIT Vehicle and LabelMe Dataset [Hussain et. al.]. The paper by Khaled F. Hussain proves that spatial resolution and color of vehicles are not essential for vehicle classification.

The aim of this paper is to apply and compare between CNN based model AlexNet and an Autoencoder based algorithm EnAET (Self-Trained Ensemble AutoEncoding Transformations for Semi-Supervised Learning) for classification of the vehicles. The dataset used in this paper is the MIO-TCD classification dataset. Many classification techniques using feature-based methods such as used such Haar, Histogram of Oriented Gradients (HOG), Principal Component Analysis (PCA) etc. can be used to extract edge features from the image. Machine learning techniques such as Support Vector Machine (SVM), k- Nearest Neighbor (kNN) and deep learning techniques are then applied to these features to classify the vehicles and get highly accurate results.

AlexNet is a convolutional neural network (CNN) [Krizhevsky et. al.]. Its architecture consists of eight layers in which five layers are convolutional layers and the rest three are fully connected layers. AlexNet is special in its own way. It has various advantages over traditional models such as it uses ReLU function instead of tanh function which provides it faster training time. A CNN with ReLU function was six times faster than CNN using tanh function on CIFAR-10 dataset. Originally, AlexNet had 60 million parameters. To reduce overfitting the authors used two techniques. First one was data augmentation in which they increased the size of training dataset and also made it more varied using label preserving transformation. Second technique that was used to reduce overfitting problem was Dropout method. This technique consists of turning off neurons with predetermined probability. The reason to use AlexNet in this paper is because it has shown high accuracy on many challenging datasets previously.

The second method that we are going to use to classify vehicles in MIO-TCD dataset is EnAET [Wang, X. et. al.]. In the original paper, it showed 9.35% error rate on CIFAR-10 and 16.92% on SVHN dataset.

Preprocessing of the dataset has been done using image augmentation and then the models are applied using Alexnet and EnAET.

II. RELATED WORKS

There are several studies done for image classification ranging from the effects of color, spatial resolution, effect of different deep learning models, technologies and computer vision methods. Earlier, methods such as Virtual Detection Line(VDL) were proposed to count the number of vehicles and classify it into different categories based on vehicle size using the kNN classification algorithm.

Accuracy of VDL methods with kNN are not affected by environmental factors but it doesn't show satisfactory results in traffic [Mithun et. al. 1]. Visual based dimension estimation methods can be used for obtaining vehicle length and using a simple 3D cuboid model but its accuracy is also affected by estimation of road to bumper height [Lai, A.H. et. al. 2]. To overcome this drawback, a generic 3D model with 12 parameters can be used for vehicle recognition [Zhang et. al.]. Feature based methods deal with extracting visual features of vehicles. Principal Component Analysis, Haar Classifier [Wen, X. et. al.] and HOG are few examples of feature based methods. Semi-Supervised convolutional network can be used to train data from vehicle's frontal view image[Dong, Z. et. al.].Automatic Vehicle Classification using range sensors and laser based approach gives high accuracy but it requires sensors to be placed at the right position and is ineffective during traffic[Hussain et. al.]. Pre-trained CNN such as AlexNet [Krizhevsky et. al.], [Molina-Cabello et. al.] uses ReLU function instead of tanh function to add non-linearity and it also increases the computation speed by 6 times. But the computational cost is expensive. Another deep learning technique is ResNET which is also pre-trained using large number of images and uses ResNetBlock to learn residual function [He, K. et. al.]. VGGNet is deeper more deeper and more accurate than AlexNet, but its computation is very costly [Chatfield, K. et. al.]. Similarly GoogleNet has less depth but its more wider due to decreased number of parameters and development of Inception Module and replacing the FC layer with Average Pooling layer [Szegedy et. al.]. VGGNet, GoogLeNet has high computation cost. A Sparse Stack encoder is a collection of sparse autoencoders which are typically a sigmoid function. It can solve the approximation problem of complex function [Liu, J.E et. al.]. The classification methods such as BoVW [Csurka, G. et. al.], VLAD [Jégou, H. et.

al.] and FV [Sánchez, J. et. al.], has also been used previously in many research papers. In Bag of visual words(BoVW) ,the visual features are extracted from an image to form a distribution and clusters are formed using K-Means clustering. But it shows very low accuracy on color pictures [Csurka, G. et. al.]. Vector of Locally Aggregated Descriptors(VLAD) is an extension of BoVW where each descriptor is quantized by a vector and clustered using K-Means clustering. But in case of color images, the accuracy is still less when compared to deep learning model. The FV offers more complete representation of the sample set, as it encodes not only the (probabilistic) count of occurrences but also higher order statistics related to its distribution with respect to the words in the vocabulary [Sánchez, J. et. al.]. Two types of data augmentation techniques are used in EnAET. Pyramid Net [Han, D. et. al.] is a powerful but expensive architecture that can improve the accuracy of EnAET and ShakeShake [Gastaldi, X. et. al.] is a powerful regularization method that can also improve the accuracy of EnAET.

In MixMatch [Berthelot et. al.] approach Random flip and crop and mosaic mask inspired by Cutout [DeVries, T. et. al.] to compute the corresponding SSL loss. For mosaic mask, we use the average pixel value of the masked area to fill the mask. Second method is to use spatial and non-spatial transformation on augmented data. Various attempts have been made to classify vehicle by using traffic signal videos. Deep learning models such as feed forward neural networks and feature extraction techniques were also used few years ago. But it had its own limitations in high traffic [Daigavane et. al.]. In video surveillance, kNN method can be used for vehicle classification and fuzzy C Means (FCN) for clustering in desired number of vehicle classes. It uses very less memory and search time but has relatively low accuracy [Mithun, N.C. et. al.]. In Binary code-based image classification, binary feature is computed and then nonlinear SVM is applied for classification. It requires less preprocessing and low memory storage [Peng, Y. et. al.]. The combination of neural networks and adaptive clustering can be used for high accuracy in vehicle classification [Lin, M. et. al.]. Another method is using Virtual Detection Zone (VDZ) to detect vehicle and then classifying it using different classification method [Seenouvong et. al.].

III. PROPOSED SYSTEM

The methodology proposed in this article includes: AlexNet CNN Network, EnAET (Self-Trained Ensemble Autoencoding Transformations for
Semi-Supervised Learning).

1. AlexNet
   The architecture consists of eight layers: five convolutional layers and three fully-connected layers.
   AlexNet had 60 million parameters, a major issue in terms of overfitting. Two methods were employed to reduce overfitting:
   a. Data Augmentation. The authors used label-preserving transformation to make their data more varied. Specifically, they generated image translations and horizontal reflections, which increased the training set by a factor of 2048. They also performed Principle Component Analysis (PCA) on the RGB pixel values to change the intensities of RGB channels, which reduced the top-1 error rate by more than 1%.

b. Dropout. This technique consists of “turning off” neurons with a predetermined probability (e.g. 50%). This means that every iteration uses a different sample of the model’s parameters, which forces each neuron to have more robust features that can be used with other random neurons. However, dropout also increases the training time needed for the model’s convergence.

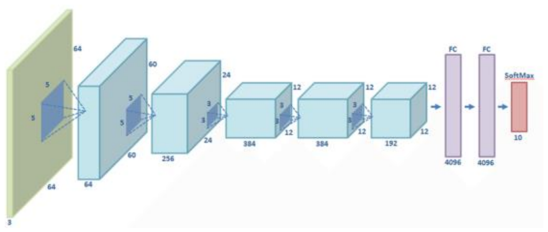

Fig. 1 AlexNet Arch. Used for Classification

2. EnAET

More recently, the AutoEncoding Transformations (AET) model has demonstrated the state-of-the-art performances in many unsupervised tasks. It aims to learn a good representation of visual structures that can decode the transformations from the learned representations of original and transformed images. We will adopt this self-supervised model to develop a self-trained model for semi-supervised tasks by exploring unlabeled data under a transformation ensemble. the difference between the features extracted from original and transformed images is caused by the applied transformations. Therefore, the transformation decoder can recover the transformations so long as the encoded features capture the necessary details of visual structures. AutoEncoding Transformation (AET) can self train a good feature representation upon which a competitive semi-supervised classifier can be developed to explore an ensemble of spatial and non-spatial transformations.

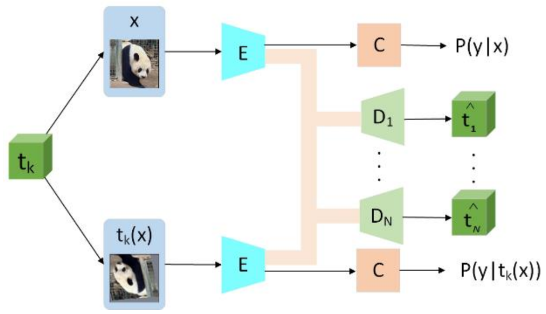

Fig. 2 illustrates the framework of the EnAET model.

Algorithm 1 Training Ensemble AutoEncoding Transformations.

Input: a batch of labelled data pair x, unlabelled data u

1. $x', u' = MixMaxtxh(x, u)$
2. $\mathcal{L}_{mix} = \mathcal{L}_{x'} + \lambda_{u'} * \mathcal{L}_{u'}$
3. **for** $k = 1$ *to* N **do**
4. $\boldsymbol{\mathcal{L}}_{AET} = \mathbb{E}_{x \in u, t_k} \parallel D\left[E(x), E\left(t_k(x)\right)\right] - t_k \parallel^2$
5. $\boldsymbol{\mathcal{L}}_{KL_k} = \mathbb{E}_{x \in u, t_k} \sum_y P(y|x) \log \frac{P(y|x)}{P_k(y|x)}$
6. **end for**
7. $\boldsymbol{\mathcal{L}} = \mathcal{L}_{mix} + \sum_{k=1}^{N} \lambda_k \boldsymbol{\mathcal{L}}_{AET_k} + \gamma \sum_{k=1}^{N} \boldsymbol{\mathcal{L}}_{KL_k}$
8. Apply $\boldsymbol{\mathcal{L}}$ to update model.
9. Update teacher model $\Theta'_\tau = \alpha\Theta'_{\tau-1} + (1 - \alpha)\Theta_\tau$

Output: Student model with Θ and teacher model with weight Θ'.

IV. MATHEMATICAL PROOF

AutoEncoding Transformation (AET) extracts the most representative features so that a transformation decoder can successfully recover parameterized transformations. In the SSL setting, instead of pretraining the model with the AET loss, we formulate AET as a regularizer along with the SSL loss to train classifiers.

We illustrate the architecture of the proposed EnAET. For each image x, we apply five different transformations: t1(Projective), t2(Affine), t3(Similarity), t4(Euclidean), t5(CCBS). After that, the network is split into three parts: an representation encoder E, a classifier C, and a set of decoders Dk each for a type of transformation tk. The original input x and all its transformed counterparts tk(x) are fed through the network. The original and transformed images have a Siamese encoder E and classifier C with shared weights.

The AET loss can be written as

$$\mathcal{L}_{AET} = \mathbb{E}_{x\in u,t_k} \parallel D\left[E(x), E\left(t_k(x)\right)\right] - t_k \parallel^2 \quad (1)$$

where D denotes the transformation decoder, E represents the encoder, and t_k is the sampled transformation of type k. The AET loss computes the Mean-Squared Error (MSE) between the predicted transformation and the sampled transformation.

We will show that the self-trained AET regularization can help EnAET set a new record in all SSL tasks under

an ensemble of spatial and non-spatial transformations. Spatial Transformations As in [Tarvainen, A. et. al.], for any 2D spatial transformation, we can represent it with a matrix below

$$\begin{bmatrix} a_1 & a_2 & b_1 \\ a_3 & a_4 & b_2 \\ c_1 & c_2 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \quad (2)$$

The representations of the original and transformed images will be concatenated to predict the parameters of each transformation t_k by the corresponding decoder D_k. The classifier C is built upon the encoded representation to output the label predictions $P(y|x)$ and $P(y|t(x))$ for both the original and transformed images, respectively. The label prediction of original image needs to be “sharpened” to reach a high degree of prediction confidence by minimizing the prediction entropy.

V. EXPERIMENTAL SETUP

The experiment was done on MIO vision Traffic Dataset. It contains more than half million images captured by traffic cameras. To apply models, first, data preprocessing was used. Image Augmentation was used to preprocess the data.

rotation_range=40,

width_shift_range=0.2,

height_shift_range=0.2,

shear_range=0.2,

zoom_range=0.2,

horizontal_flip=True,

fill_mode='nearest'

The hyperparameters were set according to the following

Table 1: Hyperparameters use for Training the EnAET

Batch Size	128
No. Of workers	4
Learning rate	0.1
Lambda	10
Max Lambda	1
Portion	0.005
Mix Mode	1
Mixmatch Warm	50

Implementation of EnAET was done using PyTorch. The dataset was then trained with Alex NET and EnAET. Experiment was conducted using NVidia 1650 (CUDA enabled) 4GB GPU with 8GB of RAM.

For AlexNet same GPU configuration were used and TensorFlow library was used to build the model.

Table 2: Model Summary of the AlexNet Model

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 35, 35, 96)	34944
batch_normalization (Batch Norm)	(None, 35, 35, 96)	384
max_pooling2d (MaxPooling2D) (None, 17, 17, 96)	0	
conv2d_1 (Conv2D)	(None, 17, 17, 256)	614656
batch_normalization_1 (Batch Norm)	(None, 17, 17, 256)	1024
max_pooling2d_1 (MaxPooling2)	(None, 8, 8, 256)	0
conv2d_2 (Conv2D)	(None, 8, 8, 384)	885120
batch_normalization_2 (Batch Norm)	(None, 8, 8, 384)	1536
conv2d_3 (Conv2D)	(None, 8, 8, 384)	147840
batch_normalization_3 (Batch Norm)	(None, 8, 8, 384)	1536
conv2d_4 (Conv2D)	(None, 8, 8, 256)	98560
batch_normalization_4 (Batch Norm)	(None, 8, 8, 256)	1024
max_pooling2d_2 (MaxPooling2	(None, 4, 4, 256)	0
flatten (Flatten)	(None, 4096)	0
dense (Dense)	(None, 4096)	16781312
dropout (Dropout)	(None, 4096)	0
dense_1 (Dense)	(None, 4096)	16781312
dropout_1 (Dropout)	(None, 4096)	0
dense_2 (Dense)	(None, 11)	45067

Total params: 35,394,315

Trainable params: 35,391,563

Non-trainable params: 2,752

VI. RESULTS AND DISCUSSION

The accuracy obtained after training the dataset using AlexNET was only 78% while in the case of EnAET , the accuracy increased up to 98.7%.

Green line is the training accuracy on the train set, blue is the validation set (which is not actually used for validation, it's actually the whole unlabeled data + labeled data in training set), grey is the testing set performance, yellow is the student model's performance on testing set. More details related to loss please check in "Records" directory.

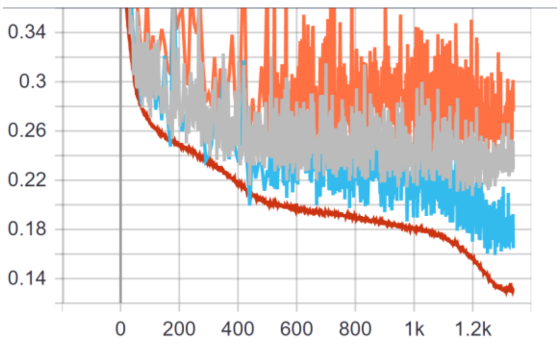

Figure 1 Training, Validation and Testing Loss in EnAET

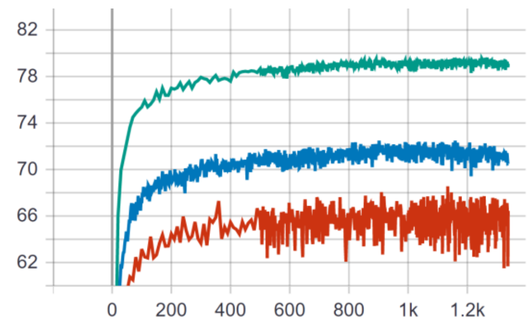

Figure 2 Training, Validation and Test Accuracy for AlexNet

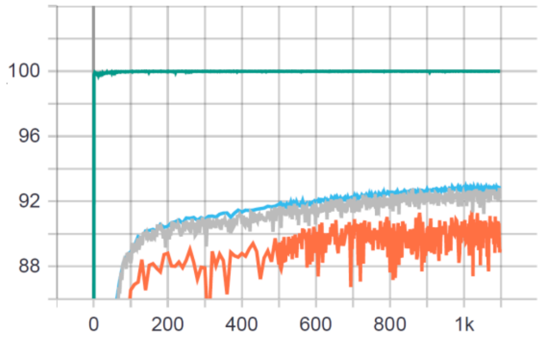

Figure 3 Training, Validation and Test Accuracy for EnAET

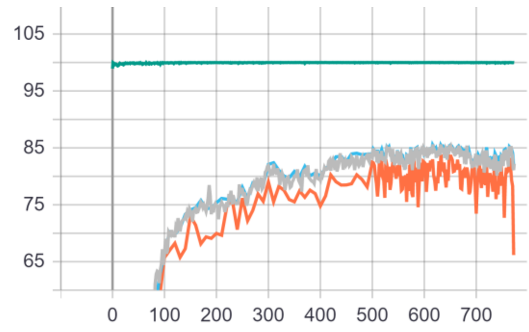

Figure 4 Accuracy measure for 700 labels

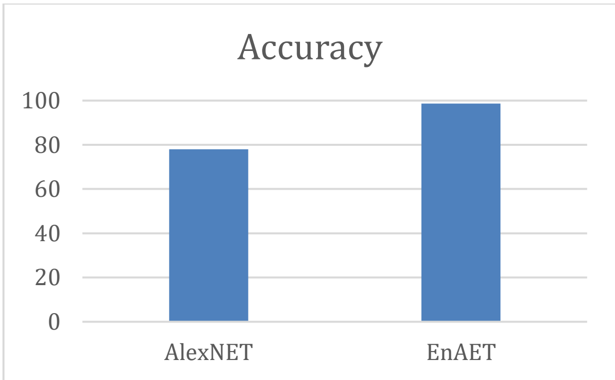

Figure 5 AlexNet vs EnAET: Training Accuracy

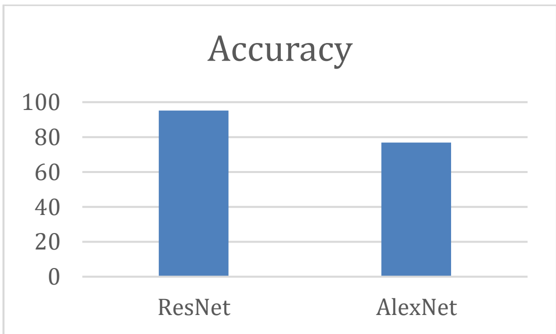

Figure 6 ResNet vs AlexNet

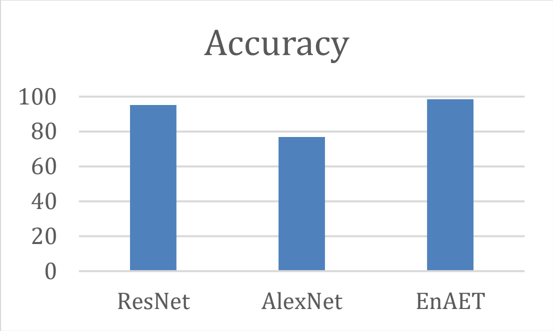

Figure 7 ResNet vs AlecNet vs EnAET

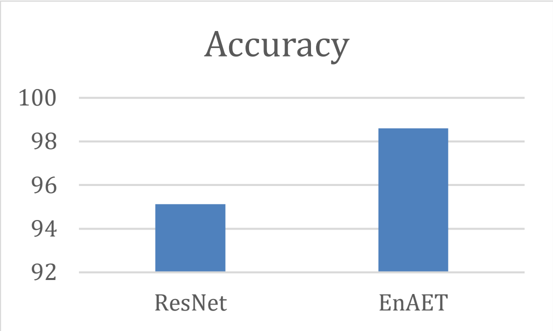

Figure 8 ResNet vs EnAET

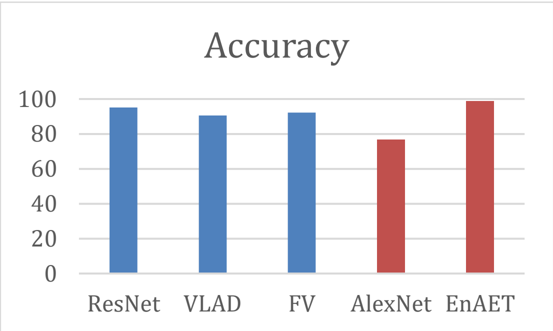

Figure 9 ResNet vs VLAD vs FV vs AlexNet vs EnAET

VII. CONCLUSION AND FUTURE WORK

EnAET proved to be a better model than AlexNET, as it produced better accuracy while classifying images in dataset.

The overall framework of EnAET.For each image x, we apply five different transformations: Projective, Affine, Similarity, Euclidean, CCBS (Color+Contrast+Brightness+Sharpness).

The network is split into three parts: an representation encoder E, a classifier C, and a set of decoders D_k each for a type of transformation t_k. The original input x and all its transformed counterparts t_{k}(x) are fed through the network. The original and transformed images have a Siamese encoder E and classifier C with shared weights.

VIII. REFERENCES

[1] Mithun, N.C., Rashid, N.U. and Rahman, S.M., 2012. Detection and classification of vehicles from video using multiple time-spatial images. *IEEE Transactions on Intelligent Transportation Systems*, *13*(3), pp.1215-1225.

[2] Lai, A.H., Fung, G.S. and Yung, N.H., 2001, August. Vehicle type classification from visual-based dimension estimation. In *ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No. 01TH8585)* (pp. 201-206). IEEE.

[3] Zhang, Z., Tan, T., Huang, K. and Wang, Y., 2011. Three-dimensional deformable-model-based localization and recognition of road vehicles. *IEEE transactions on image processing*, *21*(1), pp.1-13.

[4] Dong, Z., Wu, Y., Pei, M. and Jia, Y., 2015. Vehicle type classification using a semisupervised convolutional neural network. *IEEE transactions on intelligent transportation systems*, *16*(4), pp.2247-2256.

[5] Hussain, K.F. and Moussa, G.S., 2005, April. Automatic vehicle classification system using range sensor. In *International Conference on Information Technology: Coding and Computing (ITCC'05)-Volume II* (Vol. 2, pp. 107-112). IEEE.

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

[7] He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

[8] Wen, X., Shao, L., Fang, W. and Xue, Y., 2014. Efficient feature selection and classification for vehicle detection. *IEEE Transactions on Circuits and Systems for Video Technology*, *25*(3), pp.508-517.

[9] Liu, J.E. and An, F.P., 2020. Image Classification Algorithm Based on Deep Learning-Kernel Function. *Scientific Programming, 2020*.

[10] Molina-Cabello, M.A., Luque-Baena, R.M., López-Rubio, E. and Thurnhofer-Hemsi, K., 2017, June. Vehicle type detection by convolutional neural networks. In *International Work-Conference on the Interplay Between Natural and Artificial Computation* (pp. 268-278). Springer, Cham.

[11] Csurka, G., Dance, C., Fan, L., Willamowski, J. and Bray, C., 2004, May. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV* (Vol. 1, No. 1-22, pp. 1-2).

[12] Jégou, H., Douze, M., Schmid, C. and Pérez, P., 2010, June. Aggregating local descriptors into a compact image representation. In *2010 IEEE computer society conference on computer vision and pattern recognition* (pp. 3304-3311). IEEE.

[13] Sánchez, J., Perronnin, F., Mensink, T. and Verbeek, J., 2013. Image classification with the fisher vector: Theory and practice. *International journal of computer vision*, *105*(3), pp.222-245.

[14] Chatfield, K., Simonyan, K., Vedaldi, A. and Zisserman, A., 2014. Return of the devil in the details: Delving deep into convolutional nets. *arXiv preprint arXiv:1405.3531*.

[15] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A., 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).

[16] Hussain, K.F., Afifi, M. and Moussa, G., 2018. A comprehensive study of the effect of spatial resolution and color of digital images on vehicle classification. *IEEE Transactions on Intelligent Transportation Systems*, *20*(3), pp.1181-1190.

[17] Wang, X., Kihara, D., Luo, J. and Qi, G.J., 2019. Enaet: Self-trained ensemble autoencoding transformations for semi-supervised learning. *arXiv preprint arXiv:1911.09265*.

[18] Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A. and Raffel, C.A., 2019. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems* (pp. 5049-5059).

[19] DeVries, T. and Taylor, G.W., 2017. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*.

[20] Tarvainen, A. and Valpola, H., 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems* (pp. 1195-1204).

[21] Daigavane, P.M., Bajaj, P.R. and Daigavane, M.B., 2011, October. Vehicle detection and neural network application for vehicle classification. In *2011 International Conference on Computational Intelligence and Communication Networks* (pp. 758-762). IEEE.

[22] Mithun, N.C., Rashid, N.U. and Rahman, S.M., 2012. Detection and classification of vehicles from video using multiple time-spatial images. *IEEE Transactions on Intelligent Transportation Systems*, *13*(3), pp.1215-1225.

[23] Peng, Y., Yan, Y., Zhu, W. and Zhao, J., 2014, October. Binary coding-based vehicle image classification. In *2014 12th International Conference on Signal Processing (ICSP)* (pp. 918-921). IEEE.

[24] Lin, M. and Zhao, X., 2019, January. Application research of neural network in vehicle target recognition and classification. In *2019 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS)* (pp. 5-8). IEEE

[25] Seenouvong, N., Watchareeruetai, U., Nuthong, C., Khongsomboon, K. and Ohnishi, N., 2016, July. Vehicle detection and classification system based on virtual detection zone. In *2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)* (pp. 1-5). IEEE.

[26] Han, D., Kim, J. and Kim, J., 2017. Deep pyramidal residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5927-5935).

[27] Gastaldi, X., 2017. Shake-shake regularization. *arXiv preprint arXiv:1705.07485*