IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 4 November 2023, accepted 21 November 2023, date of publication 12 December 2023,
date of current version 20 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3342109

RESEARCH ARTICLE

Spatially Recalibrated Convolutional Neural Network for Vehicle Type Recognition

SHI HAO TAN, JOON HUANG CHUAH, (Senior Member, IEEE), CHEE-ONN CHOW, (Senior Member, IEEE), AND JEEVAN KANESAN
Department of Electrical Engineering, Universiti Malaya, Kuala Lumpur 50603, Malaysia
Corresponding author: Joon Huang Chuah (jhchuah@um.edu.my)

ABSTRACT Vehicle Type Recognition (VTR) is a significant segment within the vehicle recognition field. It provides an alternative identification method aside from license plate recognition and vehicle make and model recognition. Most of the recent studies use Convolutional Neural Networks (CNNs) to perform VTR. However, the feature responses obtained from CNNs are not recalibrated based on saliency and this hinders the classification performance. In this study, we propose a Spatial Attention Module (SAM) that is compatible with the existing CNNs. We aim to exploit the spatial relationship between feature responses by scaling them according to their relative importance to increase classification accuracy. The results reveal the exceptional performance of SAM on Beijing Institute of Technology (BIT)-Vehicle, Stanford Cars and web-nature Comprehensive Cars (CompCarsWeb) with 96.92%, 84.48% and 95.96% accuracies, respectively. A qualitative inspection of the learned feature embedding suggests the high cohesivity of the features within the group. Furthermore, an ablation study is conducted to justify the hyperparameters of choice for SAM. SAM is also modular where it is highly compatible with other CNNs and it leads to considerable performance improvement. A comparison with existing attention modules suggests our proposal prevails in the VTR application. The inference times of 1 ms and 10 ms for CaffeNet-SAM and ResNet-SAM also make them suitable for real-time classification tasks.

INDEX TERMS Convolutional neural network, multi-head self-attention, spatial attention module, transformer, vehicle type recognition.

I. INTRODUCTION

Advancement in technology induces drastic changes to the human's way of life, especially for mobility. Traveling from one place to another in a vehicle has become a norm since several decades ago and this has translated to an enormous increase in vehicle volume. To safeguard the safety of all road users, traffic monitoring and regulation are now a necessity so that users can continue enjoying a seamless traveling experience. Vehicle Type Recognition (VTR) is considered one of the essential elements that makes traffic monitoring viable [1]. Knowing the types of vehicles helps the officials estimate the wear-out rate of tar roads and thereby schedule pavement maintenance work in time. Moreover, VTR can be implemented in toll collection booths to collect tolls automatically based on vehicle types [2], [3], [4]. Vehicle type information also serves as supplementary information in tracking down the identity of a vehicle, especially during criminal investigations where license plates are normally forged [5], [6], [7]. Despite its importance, deploying a human workforce to perform VTR is not sensible due to the massive traffic volume. As the task is repetitive, laborious and tedious, the probability of a human committing a mistake will increase over time due to fatigue [5]. In addition, VTR requires a certain level of expertise for accurate judgment since there are various vehicle types.

Sensing equipment and Computer Vision (CV) are among the techniques used for VTR. As compared to CV, sensors are less favorable due to stringent operating conditions. For instance, the performance of piezoelectric sensors is affected by the vehicle speed and road surface temperature whereas

The associate editor coordinating the review of this manuscript and approving it for publication was Abdullah Iliyasu.

LiDAR is sensitive to ambient conditions such as weather and lighting [1]. These disadvantages have driven the adoption of CV-based techniques which are more versatile and yet demonstrate stronger classification performance. The early CV-based VTR solutions exploit the raw vehicle features by examining the gradient orientation information of the vehicle [2], [8], [9], [10], [11], [12]. Since these raw features are less intolerant to environmental disturbances, their performances suffer when deployed to the actual environment. Deep learning algorithms, particularly Convolutional Neural Networks (CNNs), underwent rapid development a decade ago due to their astounding learning capability. They perform feature extraction hierarchically and eventually generate the global features that carry strong semantic information [13], [14], [15], [16], [17], [18], [19]. The advent of deep learning algorithms has lifted the experience-based feature engineering process practiced in the local feature-based techniques and consistently achieved state-of-the-art performance. Another line of work focuses on unsupervised filter learning techniques. Unsupervised filter learning serves two purposes, either to reduce training complexity [4], [20], [21] or to learn superior convolutional kernels [22]. Although these works bring competitive results, more studies need to be done to validate the generalization ability of the networks.

Despite the remarkable performances of the existing works, we realize that most of them do not implement differential learning which treats the feature responses according to information saliency. We reckon that every vehicle part does not assume the same importance level. They should be treated differently so that the discriminative parts are given higher attention. In this regard, many studies focus on exploiting the channel and spatial relationship to highlight the salient features [23], [24], [25], [26]. Inspired by their encouraging results, we propose a Spatial Attention Module (SAM) to enhance the high-level features deduced from the convolutional operation. SAM aims to quantify the feature relativeness by computing the attention maps. The deduced attention maps are then used to tune the top-level feature maps through element-wise multiplication and a softmax classifier is used to perform classification. The contributions of this work are stated as follows:

- Exhibited the ability of SAM to compute spatial relationships among global features and thus underscore the exclusive features
- Demonstrated SAM can be integrated into existing classification models to improve classification accuracy and it is trainable end-to-end
- Proved that SAM is better than existing attention mechanisms in terms of classification accuracy

The rest of this paper is organized as follows. We review several relevant works in Section II. In Section III, we unravel the architecture of the proposed SAM and how it is combined with available classification models. Details of the experiments and comprehensive analysis of our results are elaborated in Section IV. We conclude this work in Section V.

II. RELATED WORK

In this section, we present a brief review of existing works in the vehicle recognition domain. Based on the nature of the proposed methodologies, they can be broken down into local feature-based, deep learning, unsupervised filter learning and attention mechanism.

A. LOCAL FEATURE-BASED

Local feature-based methods look out for the texture, gradient orientation, or interest points information for feature extraction. These features are normally fed to machine learning algorithms for classification. A cascade two-stage classifier ensemble was proposed by Zhang [8] where Gabor Wavelet Transform and Pyramid Histogram of Oriented Gradient (PHOG) were utilized to characterize vehicles. The feature vectors were subsequently fed into an ensemble of 25 models to perform classification through majority voting. Despite achieving 98.65% accuracy for 21 vehicle models, having numerous models may impact the feasibility of real-time implementation. Peng et al. [11] applied a clustering technique for VTR. K-means clustering was performed on the features deduced from Principal Component Analysis (PCA) and 88.8% accuracy was reported. Sun et al. [2] derived global and local features from an improved canny edge detector and Gabor wavelet. A two-stage classification framework, namely the k-Nearest Neighbor Probability Classifier (KNNPC) and Discriminative Sparse Representation-Based Classifier (DSRC), was then used as the classifier. Their framework was tested on a limited number of images and reported an average accuracy of 93%. Derrouz et al.'s work [9] was based on stereo vision. Using the disparity map generated from stereo vehicle images, they derived actual vehicle dimensions. Next, HOG was applied to enrich the feature representation and the feature vector was downsized through PCA. Eventually, the feature vector together with vehicle dimensions served as input to SVM. They reported 95.2% on Beijing Institute of Technology (BIT)-Vehicle dataset [20]. Sathyanarayana and Anand () [10] described vehicles through Gabor filters, HOG and Local Optimal Oriented pattern [27]. Then, Ant Colony Optimization [28] was utilized to select the top 30% best features, thus reducing features from 12,260 to 3,676 before feeding into a deep neural network. Their framework recorded 97.88% accuracy on the MIO-TCD dataset [29] and outperformed other deep CNNs such as ResNet50 [30] (96.9%), DenseNet [31] (97.0%) and Xception [32] (97.6%). Wang et al. [12] improvised the Spatiotemporal Sample Consistency algorithm (STSC) to reduce lighting interference in the background subtraction technique during vehicle detection. The segmented vehicle was then fed into a cascade classifier to predict vehicle type based on the ratio of the license plate and vehicle dimensions, HOG features, passenger face as well as vehicle area. However, their network is not easily scalable as the inputs include the actual dimensions of license plates that can vary from country to country.

Although these works reported high accuracy, dependency on handcrafted features causes them to be less robust. The network performance can be highly swung by translation, rotation, scaling and change in light illumination [21], [33].

B. DEEP LEARNING

Deep learning has garnered attention in the image classification domain since AlexNet [34], a variant of CNN, reported astounding results on ImageNet. Convolutional architecture is used to deduce hierarchical features by generating global features from local features. CNN is robust [33] as compared to handcrafted features as it is relatively invariant against external disturbances such as geometric transformation and brightness variation. Jung et al. [13] studied an ensemble technique based on deep learning models. They proposed Joint Fine-tuning (JF) to train several CNNs through joint loss function. They also implemented DropCNN to randomly drop CNN from the logits averaging process to prevent overfitting. An ensemble of 8 ResNet18 reported 97.95% accuracy on the MIO-TCD dataset. Rachmadi et al. [17] modeled image classification as a time series problem by attending to different parts of vehicles sequentially. ResNet18 was used to describe every image partition as well as the original image before they were attended in turn by Long-Short Term Memory (LSTM). They achieved 97.98% accuracy on the MIO-TCD dataset. Despite the high recognition rate, frameworks by [13] and [17] are memory intensive and they are not fit for deployment on lightweight devices. Boonsirisumpun and Surinta [18] fine-tuned MobileNet [35] to differentiate 5 vehicle types and reported 93.40% accuracy. Arinaldi et al. [19] applied Faster Region-Based CNN which uses region proposal network, region of interest pooling and convolutional architecture to carry out detection and classification for 6 vehicle classes. They reported an accuracy of 69.4% based on the MIT Traffic dataset. Another technique by Li et al. [14] is the combination of the compressed sensing technique and ResNet [30]. Compressed sensing which has the advantage in terms of faster computational speed was used to generate a saliency map for vehicle detection and ResNet50 was used to carry out classification. Accuracies of 94.12% and 95.04% were reported for 3 vehicle classes based on MIT CBCL and Caltech Database, respectively. With inference time as the primary focus, Tajar et al. [15] performed pruning for YOLOv3-tiny to reduce the number of parameters. Zhao et al. [16] improvised YOLOv4 [36] by integrating the Convolutional Block Attention Module (CBAM) [26] and modifying Path Aggregation Network [37]. Tajar et al. [15] and Zhao et al. [16] attained mAP 95.05% and mAP 83.45% for 6 vehicle types, respectively.

C. UNSUPERVISED FILTER LEARNING

Instead of optimizing convolutional kernels through backpropagation, some studies suggested unsupervised techniques. Dong et al. [20] proposed a semi-supervised CNN that learns convolutional kernels through Sparse Laplacian Filter Learning (SLFL) and multitask learning. Their technique delivered 88.11% accuracy for 6 vehicle classes but was not discriminative enough between Sport Utility Vehicle (SUV) and sedan. Similarly, Huang et al. [21] made use of PCA to deduce convolutional kernels and the feature maps were used by SVM for classification. Their framework which delivered 99.07% accuracy on 10 vehicle makes has a longer inference time than CNN which uses backpropagation. The network proposed by Soon et al. [4] also adopted PCA filters to derive vehicle features and they reported 88.52% for 6 vehicle types. In addition, Local Tiled CNN was proposed by Gao and Lee [22] in which Topographic Independent Component Analysis was used to deduce the convolutional kernels and 98.5% accuracy was reported. Although the unsupervised filter learning technique shows promising results, it is disputed by [38] and [39] due to low robustness.

D. ATTENTION MECHANISM

Works categorized under the attention mechanism treat the feature maps according to information saliency. A recalibration operation is carried out to adjust the feature responses so that the distinctive information is given more focus whereas the inconsequential information is suppressed. Our work falls under this category. Ma and Boukerche [40] proposed a Lightweight Recurrent Attention Unit (LRAU) that successively refines the feature maps based on the attention state matrices deduced from image pyramids. Despite reporting 93.9% accuracy on Stanford Cars [41], the utilization of 1×1, stride 2 convolution to deduce attention state causes information loss. In SAM, we preserve the completeness of the learned contextual information by sending the top-level feature maps in full form into SAM to render more accurate classification results. In Attention Pyramid (AP) CNN [42], the Feature Pyramid Network (FPN) [43] is used to generate multi-scale features and they are further refined by the spatial and channel gates. APCNN achieves 95.3% accuracy on Stanford Cars but we conjecture that the learned features are suboptimal due to the limited receptive field of convolutional kernels. With Multi-Head Self Attention (MHSA), SAM is able to track long-range dependencies and generate more holistic features. Attentive Pairwise Interaction Network (APINet) [44] identifies the salient region of the image by comparing and contrasting an object pair. A careful design of image pair construction strategy is essential to ensure the convergence of loss function. On the contrary, the training pipeline of SAM is relatively simpler and it has higher generalization ability when being applied to different datasets.

III. PROPOSED FRAMEWORK

In this study, we carry out experiments on SAM using CaffeNet [45] due to its relatively shallow architecture as compared to VGG [46], GoogLeNet [47] and ResNet [30]. Shallow architecture results in lower floating-point operations (FLOPs) and hence shorter model training time.

CaffeNet is a CNN that is responsible for producing low-level and high-level features from raw image pixels through convolution operations. The high-level features are

subsequently enhanced by the proposed SAM upon considering the spatial relationship among feature responses.

It is worth noting that the proposed SAM can be embedded into any CNNs and thus it is handy to use. We demonstrate this in Section IV-D-4 and its benefit is validated through classification accuracy.

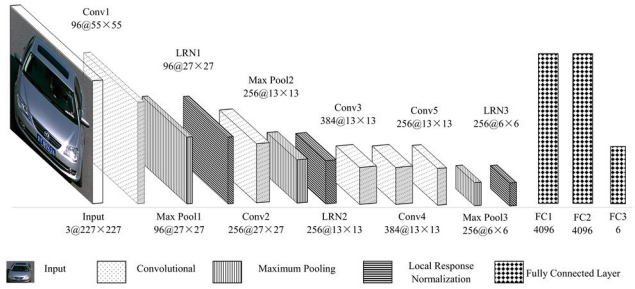


FIGURE 1. CaffeNet architecture. The value before and after @ indicates the number of channels and the size of feature maps, respectively.

A. CaffeNet

CaffeNet consists of 5 convolutional layers, 3 local response normalization (LRN) layers and 3 fully connected layers as shown in Fig. 1. Denoting an input to a layer as $\{\boldsymbol{I}_i^s\}_{i=1}^{N} \in \mathbb{R}^{\boldsymbol{H}\times\boldsymbol{W}\times\boldsymbol{C}}$, where $\boldsymbol{H}$ is height, $\boldsymbol{W}$ is width, $\boldsymbol{C}$ is the number of channels, $\boldsymbol{s}$ is layer index and $\boldsymbol{N}$ is the total number of images, convolution is performed on $\boldsymbol{I}_i^s$ using convolutional kernels. The convolutional kernels capture the local relationship within the local receptive field and subsequently aggregate the local features into global features. The convolutional kernels are shared across the entire feature maps and the number of trainable parameters is reduced as a result. A convolution operation is formulated as

$$I_i^{s+1} = I_i^s * W_{conv}^s \tag{1}$$

where $*$, W_{conv}^s and I_i^{s+1} are 2D-convolution, convolutional kernels and output at layer s, respectively. Group convolution is performed on 'Conv2', 'Conv4' and 'Conv5' in which I_i^s is first split into $I_{i,1}^s$ and $I_{i,2}^s$ along the channel axis. Convolution is then performed separately to get $I_{i,1}^{s+1}$ and $I_{i,2}^{s+1}$ and they are concatenated to form I_i^{s+1}. To introduce non-linearity to the network, the Rectified Linear Unit (ReLU) is opted as the activation function. It is a piecewise linear function given by $ReLU(x) = max(0, x)$ where x is a single feature response on the feature maps. Pooling operation is also implemented to provide translation and rotational invariance. Maximum pooling is chosen to retain the most salient feature within the pooling kernels. Besides, the pooling operation reduces the size of feature maps and hence results in a shorter model training time. Apart from the above, LRN provides lateral inhabitation. by performing normalization over neighboring feature maps. Denoting an activity of a neuron in feature map i as a_i, the response-normalized activity b_i is computed as

$$b_i = \frac{a_i}{\left(k + \alpha \sum_{j=max\left(0,1-\frac{r}{2}\right)}^{min\left(C-1,i+\frac{r}{2}\right)} a_j^2\right)^{\beta}} \tag{2}$$

where bias $k = 2$, alpha $\alpha = 10^{-4}$, beta $\beta = 0.75$ and radius $r = 5$ for all LRN layers. CaffeNet culminates with fully connected layers and a softmax classifier. The input to the fully connected layer is the flattened feature vector from the previous layer. It should be noted that ReLU is used as the activation function for the first two fully connected layers whereas softmax is used in the last fully connected layer to compute the class probability. The output of the last fully connected layer is represented as

$$P\left(Class = m \,|\, x; W^L; B^L\right) = \frac{exp\left(w_m^T x + b_m\right)}{\sum_{i=1}^{M} exp\left(w_i^T x + b_i\right)} \tag{3}$$

where $x \in \mathbb{R}^{K\times 1}$ is the K-dimensional feature vector and $K = 4,096$ for CaffeNet, M is the number of classes, $m \in \{1, 2, \ldots M\}$, $W^L = [w_1, w_2, \ldots, w_M] \in \mathbb{R}^{K\times M}$ and $B^L = [b_1, b_2, \ldots, b_M] \in \mathbb{R}^{1\times M}$ are neuron weights and biases.

To increase the sensitivity of CaffeNet towards spatial importance, SAM is embedded after the 'LRN3' since the feature map size at this stage is the smallest and richest in information. Using MHSA as the core building block, SAM allocates higher weightage to the spatial positions corresponding to crucial vehicle parts guided by the attention matrices. In particular, the scaled dot-product attention is employed to deduce the attention matrices that quantify the correlation of the spatial positions. To promote diversified learning, the attention matrices are computed in multiple feature spaces via different attention heads and they are eventually used to scale the feature responses appropriately. More deliberations about SAM are provided in Section III-B.

B. SPATIAL ATTENTION MODULE (SAM)

SAM is inspired by MHSA in the transformer architecture [48]. MHSA is first applied in Natural Language Processing (NLP) for machine translation tasks and its advent has since challenged the status quo of recurrent neural networks. It allows parallel computation and the modeling of long-range dependency. Motivated by the success of MHSA in NLP, various research issues are steered toward the application of MHSA in the image classification domain. Vision Transformer (ViT) [49] and Perceiver [50] were proposed recently and they rendered comparable classification results against CNNs. Nevertheless, the lack of inductive bias lands the transformer at a disadvantage, especially in the low data regime. In regard to this, we decide to build our work based on CNN and augment its understanding at the global level through the incorporation of SAM. SAM leverages MHSA to gain a global understanding of the vehicles. This eases the exploitation of spatial relationships within the high-level feature maps to further enrich the feature embeddings. Generally, SAM will be inserted after the last feature maps of the backbone CNN and the underlying operations are depicted in Fig. 2.

1) FEATURE MAP PREPROCESSING

As MHSA works with one-dimensional input, we first reshape the feature maps I_i^s into $I_i^{patch} = \{I_{i,1}^{patch}, I_{i,2}^{patch}, \ldots,$

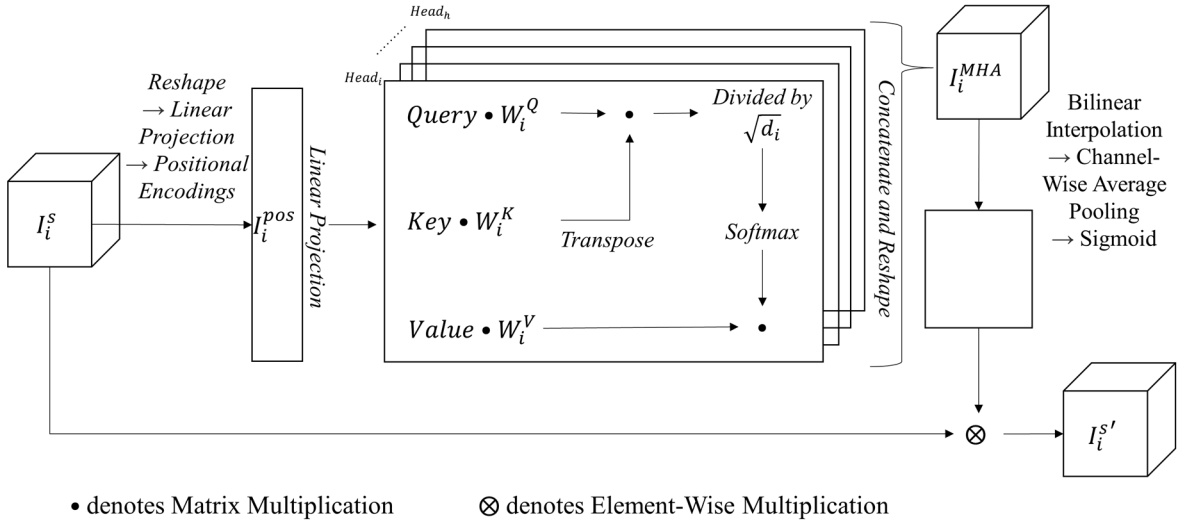


FIGURE 2. Spatial attention module.

$I_{i,P}^{patch}\} \in \mathbb{R}^{P\times(k_H\times k_W\times C)}$ where k_H is the height of the patch, k_W is the width of the patch and $P = (H \times W)/(k_H \times k_W)$ is the number of patches.

Subsequently, each image patch is linearly projected such that $I_i^{proj} = \{I_{i,1}^{proj}, I_{i,2}^{proj}, \ldots, I_{i,P}^{proj}\} \in \mathbb{R}^{P\times(k_H\times k_W\times C)}$ where $I_{i,p}^{proj} = I_{i,p}^{patch} W_p$, p is the patch index and $W_p \in \mathbb{R}^{(k_H\times k_W\times C)\times(k_H\times k_W\times C)}$.

2) INJECTING POSITIONAL INFORMATION

The position of image patches is important in computing the spatial relationship. However, MHSA does not account for positional differences as the attention operation is carried out in parallel. Being permutation invariant makes MHSA less competitive in modeling highly structured data like images [24] and injecting the positional information into the image patches can bring MHSA more clues regarding the object structures. In SAM, we choose positional encodings to incorporate positional information and we reason this in Section IV-D-I based on our dataset.

Positional encodings inject positional information into I_i^{proj} using sine and cosine functions. The benefit of positional encodings is it involves no training parameters. The operation to generate positional encodings $P_{enc} \in \mathbb{R}^{P\times(k_H\times k_W\times C)}$ is as follows:

$$P_{enc}(p, 2d) = sin\left(\frac{p}{10,000^{\frac{2d}{D}}}\right) \tag{4}$$

$$P_{enc}(p, 2d+1) = cos\left(\frac{p}{10,000^{\frac{2d}{D}}}\right) \tag{5}$$

where $d \in \{0, 1, \ldots, (k_H \times k_W \times C - 2)/2\}$ and $D = k_H \times k_W \times C$. P_{enc} is then added to I_i^{proj} such that $I_i^{pos} = I_i^{proj} + P_{enc}$.

3) MULTI-HEAD SELF-ATTENTION

MHSA computes the attention distribution among I_i^{pos} by using query Q, key K and value V where $Q, K, V \in \mathbb{R}^{P\times(k_H\times k_W\times C)}$. In self-attention, Q, K and V are essentially I_i^{pos}. As stated by Vaswani et al. [48], instead of using single-head, it is advantageous to adopt multi-head by linearly projecting Q, K and V for H number of times where H is the number of heads. MHSA performs the attention computation among Q, K, V matrices that have been linearly projected into different subspaces. In particular, scaled dot-product attention is calculated in which the output is the weighted sum of V and weight is the degree of compatibility between Q and K. Mathematically, a single-head self-attention operation for $Head_i$ is represented as follows:

$$Head_i = Softmax\left[\frac{QW_i^Q\left(KW_i^K\right)^T}{\sqrt{d_i}}\right]\left(VW_i^V\right) \in \mathbb{R}^{P\times d_i} \tag{6}$$

where $d_i = (k_H \times k_W \times C)/H$ is the dimension of linearly projected Q, K and V per head, $W_i^Q, W_i^K, W_i^V \in \mathbb{R}^{(k_H\times k_W\times C)\times d_i}$ are projection matrices for Q, K and V, respectively. Upon computing scaled dot-product attention, all $Head_i$ are concatenated and reshaped into a tensor of shape $(P, k_H \times k_W \times C)$ before being sent to the final projection layer.

$$I_i^{MHA} = Reshape\left(Cat\left(Head_1, Head_2, \ldots, Head_H\right)\right) W^O \tag{7}$$

where $Cat(\cdot)$ is concatenate operation, $W^O \in \mathbb{R}^{(k_H\times k_W\times C)\times(k_H\times k_W\times C)}$ and $I_i^{MHA} \in \mathbb{R}^{P\times(k_H\times k_W\times C)}$.

4) INJECTING SPATIAL RELATIONSHIP

After capturing the attention information, we embed this information into I_i^s. As the output of MHSA is one-dimensional, we first reshape it back into two dimensions

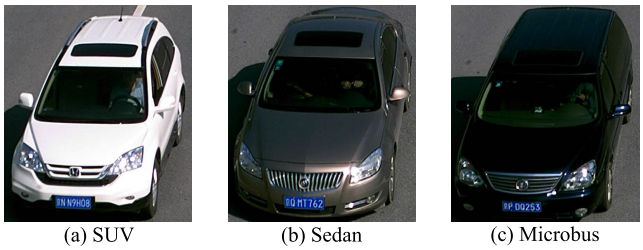
such that $I_i^{MHA'} = Reshape\left(I_i^{MHA}\right) \in \mathbb{R}^{\sqrt{P}\times\sqrt{P}\times(k_H\times k_W\times C)}$. When k_H or k_W is larger than 1, it causes $\sqrt{P}$ to be smaller than H and W and hence bilinear interpolation is carried out so that the resultant dimension matches that of I_i^s.

$$I_i^{MHA'} = BilinearInterpolation\left(Reshape\left(I_i^{MHA}\right)\right) \quad (8)$$

It is important to note that interpolation is an optional operation and it is needed only when $k_H \neq 1$ or $k_W \neq 1$. Subsequently, the channel information of $I_i^{MHA'}$ is aggregated using channel-wise mean operation and the feature responses are kept within 0 to 1 by applying the sigmoid function. Finally, the attention information which contains the spatial information is incorporated into I_i^s by

$$I_i^{s'} = I_i^s \otimes \sigma\left(Mean\left(I_i^{MHA'}\right)\right) \in \mathbb{R}^{H\times W\times C} \quad (9)$$

where $Mean\,(\bullet)$ is channel-wise mean, $\otimes$ is element-wise multiplication and σ is the sigmoid function.



(a) SUV (b) Sedan (c) Microbus (d) Minivan (e) Truck (f) Bus

FIGURE 3. Sample images from BIT-Vehicle dataset.

IV. EXPERIMENTS

A. DATASET

We carry out the experiment for VTR using the BIT-Vehicle dataset [20]. It consists of 9,850 images of different viewpoints captured by surveillance cameras and they are subject to external disturbances such as lighting variations, scaling and rotation, etc. This serves as a good reference point to examine the robustness of the proposed framework. Since some images contain more than one vehicle and to suit the vehicle classification task, we utilize the provided annotations to segment individual vehicles. The class distribution among SUV, sedan, microbus, minivan, truck and bus are 1,372, 5,776, 860, 467, 820 and 555, respectively. Fig. 3 shows some sample images from the BIT-Vehicle dataset.

Aside from using the full BIT-Vehicle dataset, we perform random sampling to sample 400 images from each class in which 200 are used for training and 200 for testing to produce a balanced dataset. In other words, this subset contains a total of 2,400 images and the ratio of training to testing images is 50:50. It is worth noting that we report the performance of SAM based on this subset unless stated otherwise.

To ensure SAM is highly generalizable, we validate further the framework on two publicly available datasets, namely Stanford Cars [41] and web-nature Comprehensive Cars (CompCarsWeb) [51]. The particulars of these datasets are shown in Table 1. The labels for Stanford Cars are pickup, convertible, sports car, hatchback, MPV, sedan, SUV, minibus and wagon whereas CompCarsWeb has fastback, hardtop and crossover as additional labels.

TABLE 1. Statistics of Stanford cars and CompCarsWeb.

Dataset	Train/ Test	Number of Classes
Stanford Cars [41]	8,144/ 8,041	9
CompCarsWeb [51]	36,456/ 15,627	12

B. TRAINING OF PROPOSED FRAMEWORK

Firstly, we resize the images to 224×224 before normalizing the pixel values based on the ImageNet dataset. We then initialize CaffeNet with weights pretrained on the ImageNet dataset. For SAM, we set $k_H = k_W = 1$ and $H = 8$. On the BIT-Vehicle dataset, CaffeNet-SAM is fine-tuned based on cross entropy loss using Adam [52] for 50 epochs. The learning rate α is set as 1e-4 for the first 25 epochs and it is decayed by factor of 10 at 26th epoch. For Stanford Cars and CompCarsWeb datasets, we train the network for 90 epochs since they are larger in quantity. Stochastic gradient descent is chosen as the optimizer with 0.9 momentum and 5e-4 weight decay. The learning rate is set as 0.01 and decays by a factor of 10 for every 40 epochs. We also perform random cropping to prevent overfitting during training.

The experiment is performed on a machine with the specification of Intel Core i7-9750H 2.6GHz, 32GB RAM and NVIDIA Quadro T1000 4GB video memory.

C. RESULTS AND ANALYSIS

Fig. 4 shows the training curve of CaffeNet-SAM. The training process for 50 epochs takes about 5 minutes on the GPU. Due to the fine-tuning strategy, our proposed framework records satisfactory accuracy during the first few epochs and it has a high convergence rate. The highest testing accuracy occurred at 6^{th} epoch, recording 94.17%. The accuracy is calculated using the following equation

$$Accuracy = \frac{\sum_m^M \frac{TP_m}{N_m}}{M} \times 100 \quad (10)$$

where TP_m and N_m are true prediction count and total image count in m^{th} class.

To examine the learned deep features, we apply t-distributed Stochastic Neighbor Embedding (TSNE) [53]

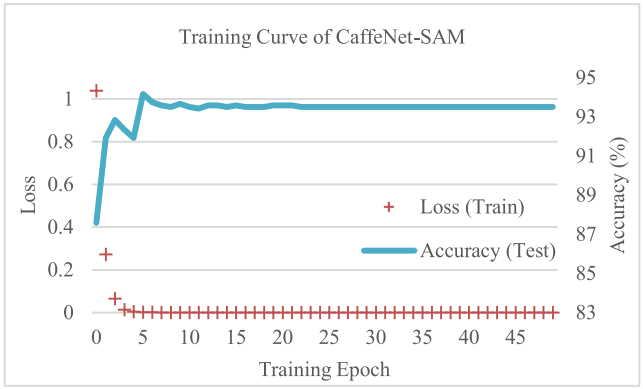


FIGURE 4. Training curve of CaffeNet-SAM. Maximum testing accuracy of 94.17% is recorded at the 6^{th} training epoch.

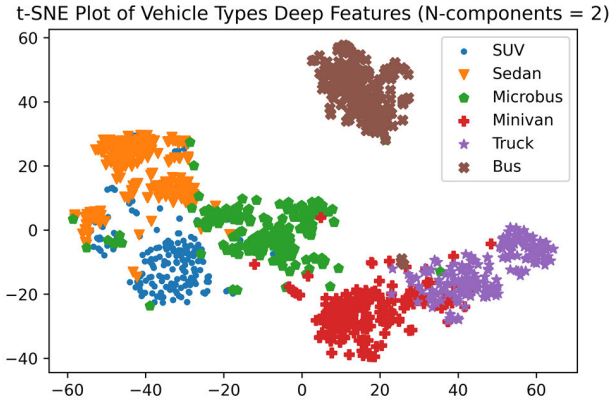


FIGURE 5. TSNE plot of deep features from CaffeNet-SAM.

to project the 4,096-dimensional features extracted from the penultimate fully connected layer into two dimensions for visual inspection. The TSNE plot of the deep features is shown in Fig. 5 and each point is labeled with the actual class label.

We observe that the bus is distinctive from the rest of the classes as it forms an independent cluster far away from others. We reckon that this is due to its prominent large vehicle size and the rigid rectangular shape which present a large visual difference from the rest. As for SUV, sedan and microbus, separation is less clear but the three clusters are still clearly visible. Besides, we notice that minivan and truck share a higher inter-class similarity as compared to other classes due to similar vehicle fronts.

We tabulate the confusion matrix in Fig. 6 to show the breakdown analysis of the prediction. The confusion matrix shares similar findings with our deep features interpretation based on the TSNE plot. The accuracy of SUV, sedan and microbus are 89%, 97% and 94.5%, respectively. Most of the wrong predictions for SUV fall into sedan and vice versa. There are 7 samples of microbus being misclassified into SUV and they account for 3.5%. None of the minivans and trucks are misclassified into small-size vehicle categories (SUV and sedan) and this suggests that the framework is

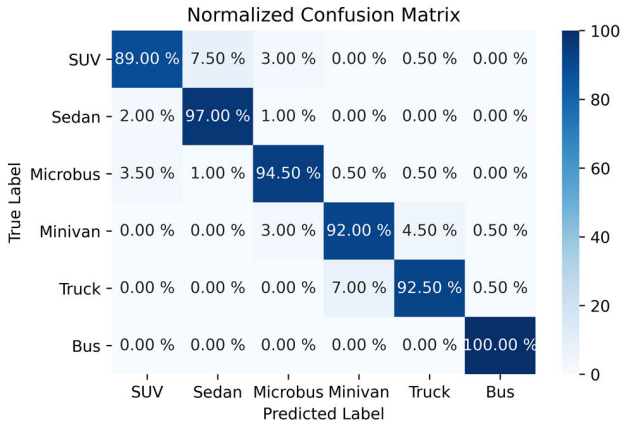


FIGURE 6. Normalized confusion matrix.

distinctive at least between different sizes of vehicles. Similar to the observation from the TSNE plot, the bus is a distinctive category and 100% accuracy is reported.

D. DISCUSSION

1) EFFECT OF POSITIONAL INFORMATION ON SAM

By nature, MHSA takes no notice of the order of image patches [48]. However, the sequence information is important as it avoids permutation equivariance and reinforces the contextual understanding of an object. In our proposal, we explicitly insert the positional information using positional encodings. We also experiment with alternative ways to inject the positional information. Specifically, one-dimensional learnable positional embeddings [49] is adopted and it is initialized using uniform distribution $U(-0.05, 0.05)$.

Table 2 shows the performance of SAM using different approaches to inject the positional information. It is observed that positional information is important in SAM and positional encodings performs better than positional embeddings by 0.42% on our dataset. Comparing positional encodings and embeddings, we conjecture that the sinusoidal waveform utilized by positional encodings retains the inter-patch relativity and thus allows the spatial structure of the vehicle to enrich the feature representation. On the contrary, positional embeddings encodes only the absolute positional information and it fails to model the patch-to-patch relationship. Another additional benefit of positional encodings is it reduces the trainable parameters of the network since the positional information is calculated explicitly rather than being learned during the training process.

TABLE 2. Effect of positional information on SAM.

Method to Inject Positional Information	Accuracy
W/O Injecting Positional Information	93.50%
Positional Embeddings [54]	93.75%
Positional Encodings [48]	**94.17%**

2) EFFECT OF PATCH SIZE ON SAM

In this section, we are interested to find out the optimal value for k_H and k_W to restructure the feature maps into patches. Based on our implementation, since CaffeNet performs convolution to achieve spatial reduction, we set k_H, k_W to be significantly smaller than ViT [49] i.e. 16 to prevent overboard generalization over a large number of feature responses. It is worth noting that we adopt the same values for both k_H and k_W to produce square-size patches.

Table 3 presents the classification results for different values of k_H and k_W. Our results suggest that setting k_H, $k_W = 1$ reaches the highest classification accuracy and the performance declines with increasing patch size. Increasing patch size from 1 to 3 brings close to 1% reduction in accuracy. This is because each feature response on the feature maps produced by the last convolutional layers corresponds to large receptive fields. Setting $k_H, k_W > 1$ is detrimental to the feature distinctiveness as the over-integration of vehicle parts results in the loss of fine-grained vehicle cues.

TABLE 3. Effect of patch size on CaffeNet-SAM.

Patch Size	Accuracy
1	**94.17%**
2	93.67%
3	93.25%

3) EFFECT OF NUMBER OF HEADS ON SAM

In SAM, we choose multi-head over single-head self-attention. MHSA provides the flexibility of attending to different subspace representations. Nevertheless, there are no standard methods to determine the optimum number of heads. Hence, we determine it empirically and the results are presented in Fig. 7.

Conforming to our expectation, MHSA indeed performs better than single-head self-attention. MHSA promotes diversified learning where each attention head models different intricate vehicle parts to improve the feature expressiveness. Furthermore, computing the attention in different subspaces provides better generalization ability and eventually reduces the chances of overfitting. The optimum number of heads for CaffeNet-SAM is 8 in this study. Setting H as 4 and 16 delivers the same performance.

4) EFFICACY OF SAM ON EXISTING CNNS

Fig. 8 shows the accuracy of different CNNs tested on our experimental dataset. These CNNs are implemented using the open-source PyTorch framework and they are trained in the same fashion as described in Section IV-B.

Based on the results, it is observed that before incorporating SAM, the lowest accuracy is recorded by the shallowest CNNs, namely AlexNet [55] and CaffeNet [45]. This is followed by GoogLeNet (93.67%) [47] which adopts Inception Module. VGG16 [46] which uses fixed-size 3×3 convolutional kernels reports 94.50% accuracy. ResNet50 [30] which implements a skip connection strategy has the largest number of layers among all. It shares the same accuracy with VGG16.

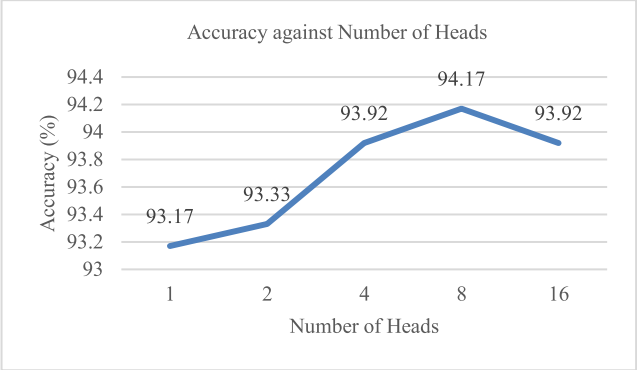


FIGURE 7. Effect of number of heads on CaffeNet-SAM.

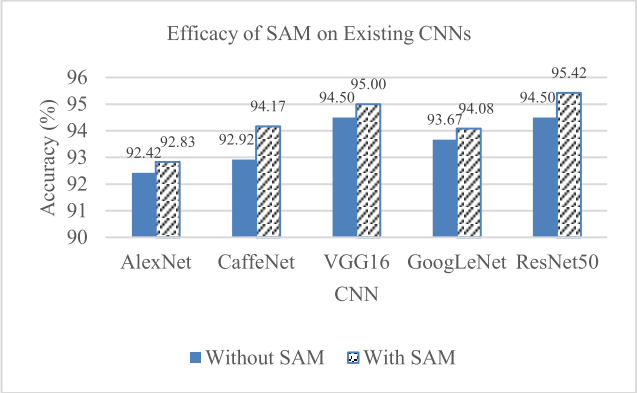


FIGURE 8. Efficacy of SAM on existing CNNs.

To test the efficacy of SAM, we incorporate it into AlexNet, VGG16, GoogLeNet and ResNet50. Specifically, we insert SAM into AlexNet and VGG16 before the first fully connected layer. As for GoogLeNet and ResNet50, the SAM is positioned before the global average pooling *GAP* layer. Upon incorporating SAM, all networks show improvement by an average of 0.7%. CaffeNet records the largest leap in accuracy, which is by 1.25% followed by 0.92% of ResNet50. The results indicate that due to the limited receptive field of convolutional kernels, the feature maps fail to gain a holistic understanding of the vehicles and hence the learned embeddings are still weak semantically. With SAM, the inter-spatial relationship is computed by MHSA which exerts a global receptive field. The thorough propagation of all spatial information enables the distinctive features to be pinpointed and recalibrated to elevate the classification performance.

5) PERFORMANCE OF SAM AGAINST EXISTING ATTENTION MODULES

As SAM employs the concept of attention, we benchmark its performance against the existing attention modules. Hu et al. [23] introduced a Squeeze-and-Excitation (SE)

block to exploit the inter-channel relationship. The squeezing operation first encapsulates each feature map using *GAP*. Subsequently, the excitation operation models the nonlinear inter-dependency between the channels and the feature maps are recalibrated according to channel importance. The operations of the SE block are represented as follows

$$I_i^{s'} = I_i^s \otimes \sigma\left(W_2 ReLU\left(W_1 GAP\left(I_i^s\right)\right)\right) \tag{11}$$

where $GAP\left(I_i^s\right) \in \mathbb{R}^{C\times 1}$, $W_1 \in \mathbb{R}^{\frac{C}{r}\times C}$, $W_2 \in \mathbb{R}^{C\times\frac{C}{r}}$ and r is the reduction ratio. In this study, we use $r = 16$ as suggested in [23].

Bottleneck Attention Module (BAM) [25] is introduced to reweight the feature responses by examining both spatial and channel relationships. The channel and spatial attention are carried out independently and they are combined eventually to jointly adjust the feature responses. Similar to the SE block, the channel attention branch within BAM models the inter-channel relationship. As for the spatial attention branch, it makes use of 1×1 convolution and dilated convolution to aggregate the contextual information, respectively. Mathematically, the operations carried out by BAM are represented as

$$I_i^{s'} = I_i^s + I_i^s \otimes \sigma\left(M_c\left(I_i^s\right) + M_s\left(I_i^s\right)\right) \tag{12}$$

where $M_c\left(I_i^s\right) \in \mathbb{R}^{1\times 1\times C}$ is the channel attention branch and $M_s\left(I_i^s\right) \in \mathbb{R}^{H\times W\times 1}$ is the spatial attention branch. We set the reduction ratio as 16 and the dilation value as 4, which are the optimal values found empirically in [25].

CBAM [26] is an improved version of BAM. The channel attention and spatial attention operations are carried out sequentially. On top of *GAP*, it adopts global maximum pooling in the channel attention branch to preserve the salient object features. In the spatial attention branch, the inter-spatial relationship is computed using channel-wise average pooling and channel-wise maximum pooling followed by a convolution operation. CBAM operation is denoted as

$$I_i^{s'} = \left(I_i^s \otimes M_c\left(I_i^s\right)\right) \otimes M_s\left(I_i^s \otimes M_c\left(I_i^s\right)\right) \tag{13}$$

The reduction ratio and convolutional kernel size are set as 16 and 7, respectively as suggested in [26].

Table 4 compares SE, BAM, CBAM and SAM when integrated with CaffeNet in terms of classification accuracy. We insert SE, BAM and CBAM after every LRN layer of CaffeNet. In other words, there are a total of three attention blocks being added. As SE, BAM and CBAM are originally proposed based on ResNet, we also integrate them into ResNet50 following the practices in [23], [25], and [26].

The results show that although SE outperforms BAM and CBAM, its performance falls behind SAM by an average of 0.46%. Similar to the SE, SAM exploits the inter-channel dependency through linear projection operation. Nevertheless, we reckon that channel recalibration alone leads to less conclusive results since the spatial context information is not considered. Therefore, following channel recalibration, SAM leverages the scaled dot-product attention to compute the correlation of the spatial positions before scaling the feature responses based on their relative importance. Our results unravel that spatial feature refinement based on MHSA is pivotal to discriminating against different types of vehicles.

TABLE 4. Performances of various attention modules.

Attention Module	Model	
	CaffeNet [45]	ResNet50 [30]
SE [23]	93.67%	95.00%
BAM [25]	91.08%	94.92%
CBAM [26]	90.83%	94.67%
SAM	**94.17%**	**95.42%**

TABLE 5. Comparison with state-of-the-art on BIT-Vehicle dataset (subset).

Work	Train/ Test	Accuracy/ Overall Accuracy
2D Deep Boltzmann Machine [56]	1,200/ 1,200	80.62%
Haar Cascade Classifier [57]	-	81.83%
SF [20]	1,200/ 1,200	86.82%
SLFL [20]	1,200/ 1,200	88.11%
PCN-Softmax [4]	1,200/ 1,200	88.52%
KNNPC + DSRC [2]	1,200/ 1,200	90.10%
DPM + SVM [58]	1,200/ 1,200	91.08%
CaffeNet-SAM	**1,200/ 1,200**	**94.17%**
ViT-B [49]	1,200/ 1,200	94.50%
APINet [44]	1,200/ 1,200	94.83%
SwinV2-T [59]	1,200/ 1,200	94.92%
LRAU [60]	1,200/ 1,200	95.08%
APCNN [42]	1,200/ 1,200	95.25%
HERBS [61]	1,200/ 1,200	95.25%
CMAL [62]	1,200/ 1,200	95.42%
ResNet50-SAM	**1,200/ 1,200**	**95.42%**

6) COMPARISON WITH STATE-OF-THE-ART

We benchmark our proposed framework against the existing works on the BIT-Vehicle dataset [20] in Table 5. CaffeNet-SAM reports 94.17% accuracy whereas ResNet50-SAM tops the ranking with 95.42% accuracy. Santos et al.'s work [56] which fed the images projected by 2D Linear Discriminant Analysis into the Boltzmann Machine reported 80.62% accuracy. Baser and Altun [57] used the Haar Cascade Classifier to both detect and classify vehicle types and achieved 81.83% accuracy. Dong et al. [20] used a semi-supervised method to learn the convolutional kernels of CNN. Specifically, they experimented with Sparse Filtering (SF) [63] to optimize the convolutional kernels for sparsity. SLFL method is an improvisation over SF by taking reconstruction error, sparsity and manifold assumption into account [20].

Accuracies of 86.82% and 88.11% were reported by SF and SLFL, respectively. Soon et al. [4] used PCA to learn the convolutional filters and they reported 88.52% accuracy. Although both Dong et al. [20] and Soon et al. [4] achieved remarkable accuracy, their frameworks are not trainable end-to-end as the convolutional kernels need to be optimized beforehand separately. Work by Sun et al. [2] delivered 90.10% accuracy. Their network was trained to first classify the vehicles into heavy or light before recognizing the types and hence additional labels are required. Bai et al.'s work [58] with 91.08% accuracy is required to produce deformable part models (DPM) for each vehicle type and this may impact the requirement for real-time inferencing when the number of vehicle types increases.

Additionally, we include more works, especially those from the attention mechanism domain, for comparison purposes. Since these works did not report the performances on the BIT-Vehicle dataset originally, we perform the training as elucidated in Section IV-B and report the results in Table 5. ViT-B [49] which has around 86M parameters delivers 94.50% accuracy. It is criticized for the inability to encapsulate information from all image patches into the class embedding token [64], [65], [66]. APINet [44] which leverages pairwise contrastive clues achieves 94.83% accuracy. We hypothesize that better performance can be achieved by customizing a dataset-specific pair construction strategy for training. SwinV2-T [59] is a transformer network that employs a shifting windowing scheme for MHSA and it reports 94.92% accuracy. Although the devised MHSA has linear complexity, the capability to model global dependency is compromised.

LRAU [60] outperforms our shallow architecture i.e. CaffeNet-SAM but its accuracy is 0.34% lower than ResNet50-SAM. APCNN [42] reports 95.25% accuracy based on multi-level classification heads. It is reckoned that attaching a classification head at early convolution layers leads to contradiction in feature learning as the low-level feature maps have to learn both high-level semantic information and low-level fine-grained information at the same time. ResNet50-based High-temperaturE Refinement and Background Suppression (HERBS) [61] is as competitive as APCNN. With the help of the selector module, it identifies the salient feature responses from various pyramid levels and channels them into a Graph Convolutional Network-based combiner module for cross-granularity information exchange. For Cross-Layer Mutual Attention Learning (CMAL) [62], it trains multiple classification experts that first segment the vehicle from the image in a weakly supervised manner through feature maps binarization before performing the classification. Building upon TResNet-L [67], CMAL has parameters as many as 63.2M. Although ResNet50-SAM is smaller than CMAL by close to 20M, it renders the same performance level.

We also examine the proposed framework on a larger image pool. To compare with the works that do not report the performance based on (10), we attach the overall accuracy figure calculated using (14) in Table 6

TABLE 6. Comparison with state-of-the-art on BIT-Vehicle dataset (full).

Work	Train/ Test	Accuracy	Overall Accuracy
Inception-v3 [68]	7,880/ 1,970	Unreported	97.10%
Stereo-Vision Based Model [9]	7,895/ 1,955	Unreported	95.20%
CNN [69]	29,760/ 852	93.94%	93.90%
Super Leaner Ensemble [70]	8,039/ 2,014	96.77%	97.62%
CaffeNet-SAM	**7,881/ 1,969**	**95.44%**	**97.41%**
ViT-B [49]	7,881/ 1,969	95.92%	97.66%
APINet [44]	7,881/ 1,969	96.01%	97.56%
SwinV2-T [59]	7,881/ 1,969	96.08%	97.56%
HERBS [61]	7,881/ 1,969	95.80%	97.61%
LRAU [60]	7,881/ 1,969	96.15%	97.71%
APCNN [42]	7,881/ 1,969	96.37%	97.82%
CMAL [62]	7,881/ 1,969	96.59%	98.12%
ResNet50-SAM	**7,881/ 1,969**	**96.92%**	**98.17%**

$$Overall\ Accuracy = \frac{TP_{Total}}{N_{Test}} \times 100 \tag{14}$$

where TP_{Total} is the total true prediction count and N_{Test} is testing image count. The accuracy calculated based on (10) is equivalent to (14) when the images from each class have equal proportions.

Liu et al. [68] reported 97.1% overall accuracy based on Inception-v3 [71]. Derrouz et al. [9] reported 95.2% overall accuracy using vehicle dimensions and HOG as features. Their stereo vision-based work requires two cameras and this results in higher installation costs. A novel CNN introduced by Roecker et al. [69] reported 93.94% accuracy and 93.90% overall accuracy. Hedeya et al. [70] proposed a super-learner ensemble technique based on Xception [32] and DenseNet [31]. In particular, the logits of two deep learning models were merged via a fully connected layer and 96.77% accuracy and 97.62% overall accuracy were reported. The transformer-based ViT-B [49] achieved 95.92% accuracy and 97.66% overall accuracy. APINet [44] and SwinV2-T [59] are on par by reporting 97.56% overall accuracy. For HERBS [61], a drop in ranking is seen as its performance is poorer than LRAU [60] and APCNN [42] in terms of overall accuracy. This is caused by the selector module which constrains the feature representation learning from the most discriminative responses. Consequently, other complementary visual cues are forgone and the embeddings become less diverse. APCNN consistently outflanks LRAU by claiming 96.37% accuracy and 97.82% overall accuracy. CMAL [62] remains the best network after ResNet50-SAM. It is worth noting that CMAL has high computational costs which stand at 14.04 GFLOPs since 5 forward passes are required to deduce a superior vehicle segmentation mask. For our proposal, CaffeNet-SAM achieves 95.44% accuracy and 97.41% overall accuracy. ResNet50-SAM achieves the

best performance with 96.92% accuracy and 98.17% overall accuracy.

TABLE 7. Computational complexity of SAM.

Model	#Params	FLOPs	Inference Time per frame
CaffeNet	57.00M	0.71G	1ms
CaffeNet-SAM	57.33M	0.72G	1ms
ResNet50	23.51M	4.13G	8ms
ResNet50-SAM	44.49M	5.16G	10ms

To render a holistic evaluation, a comparison between the vanilla CNN and post-SAM insertion in terms of the number of parameters, FLOPs and inference speed is tabulated in Table 7. Incorporating SAM brings negligible effect on the network size and computational cost to CaffeNet. This is attributed to the low channel count of the top-level features i.e. 256 channels. On the contrary, incorporating SAM almost doubles the size of ResNet50 and this is due to the top-level feature maps that have 2048 channels. The high channel count also translates to a larger increment in FLOPs as compared to CaffeNet and the increment approximates 1 GFLOPs, Nevertheless, ResNet50-SAM is still considered moderate in size.

In terms of inference time, CaffeNet-SAM outflanks the framework proposed by Soon et al. [4] as it is 7 times faster, which is 1 ms against 7 ms whereas the inference time of ResNet50-SAM is 10 ms. Nevertheless, we are aware that this should just serve as a reference as the machine specification is not considered. The inference time for the rest of the works is unreported.

TABLE 8. Evaluation of Stanford cars and CompCarsWeb.

Dataset	Model	Accuracy
Stanford Cars [41]	ResNet50	82.94%
	ResNet50-SAM	84.48%
CompCarsWeb [51]	ResNet50	92.88%
	ResNet50-SAM	95.96%

7) EVALUATION OF SAM ON STANFORD CARS AND COMPREHENSIVE CARS

As both Stanford Cars [41] and CompCarsWeb [51] datasets are considerably large, we choose to work with a deep CNN i.e. ResNet50 [30] in this section. The results are tabulated in Table 8. We demonstrate the benefits of SAM on these datasets where it identifies and pays more attention to critical spatial positions to render better classification performance than the baseline. ResNet50-SAM achieves 84.48% and 95.96% accuracy on Stanford Cars and CompCarsWeb. The improvement brought by SAM is 1.54% and 3.08%, respectively.

V. CONCLUSION

In this work, we propose SAM that treats each spatial position on the feature maps according to the information relevancy. It places a higher focus on spatial positions that correspond to key vehicle parts and attenuates the insignificant information to better differentiate vehicle types. We fuse SAM with CaffeNet and report 95.44% accuracy for 6 vehicle types based on the BIT-Vehicle dataset. The framework takes 1 ms during inference and it is fit for real-time implementation. Integrating SAM into deeper CNN renders even better classification performance where ResNet50-SAM achieves 96.92% accuracy. In addition, SAM leads the state-of-the-art solutions, especially those that originate from the attention domain by a considerable margin. It also exhibits a high generalization ability where it brings improvement over the baseline network by an average of 2.31% accuracy on Stanford Cars and CompCarsWeb datasets.

REFERENCES

[1] M. Won, "Intelligent traffic monitoring systems for vehicle classification: A survey," *IEEE Access*, vol. 8, pp. 73340–73358, 2020.

[2] W. Sun, X. Zhang, S. Shi, J. He, and Y. Jin, "Vehicle type recognition combining global and local features via two-stage classification," *Math. Problems Eng.*, vol. 2017, pp. 1–14, Jan. 2017.

[3] Z. Zhu and Y. Guo, "Vehicle style recognition based on image processing and neural network," in *Resent Advances in Computer Science and Information Engineering*. Cham, Switzerland: Springer, 2012, pp. 1–8.

[4] F. C. Soon, H. Y. Khaw, J. H. Chuah, and J. Kanesan, "Semisupervised PCA convolutional network for vehicle type classification," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 8267–8277, Aug. 2020.

[5] A. J. Siddiqui, A. Mammeri, and A. Boukerche, "Real-time vehicle make and model recognition based on a bag of SURF features," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 11, pp. 3205–3219, Nov. 2016.

[6] A. Martín and S. Tosunoglu, "Image processing techniques for machine vision," Florida Int. Univ., Miami, FL, USA, Tech. Rep., 2000.

[7] H. Liu, Y. Tian, Y. Wang, L. Pang, and T. Huang, "Deep relative distance learning: Tell the difference between similar vehicles," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2167–2175.

[8] B. Zhang, "Reliable classification of vehicle types based on cascade classifier ensembles," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 1, pp. 322–332, Mar. 2013.

[9] H. Derrouz, A. Elbouziady, H. A. Abdelali, R. O. H. Thami, S. E. Fkihi, and F. Bourzeix, "Moroccan video intelligent transport system: Vehicle type classification based on three-dimensional and two-dimensional features," *IEEE Access*, vol. 7, pp. 72528–72537, 2019.

[10] A. M. Narasimhamurthy, "Vehicle type classification using hybrid features and a deep neural network," *Int. J. Softw. Innov.*, vol. 10, no. 1, pp. 1–18, Mar. 2023.

[11] Y. Peng, J. S. Jin, S. Luo, M. Xu, and Y. Cui, "Vehicle type classification using PCA with self-clustering," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops*, Jul. 2012, pp. 384–389.

[12] Y. Wang, X. Ban, H. Wang, D. Wu, H. Wang, S. Yang, S. Liu, and J. Lai, "Detection and classification of moving vehicle from video using multiple spatio-temporal features," *IEEE Access*, vol. 7, pp. 80287–80299, 2019.

[13] H. Jung, M.-K. Choi, J. Jung, J.-H. Lee, S. Kwon, and W. Y. Jung, "ResNet-based vehicle classification and localization in traffic surveillance systems," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 934–940.

[14] Y. Li, B. Song, X. Kang, X. Du, and M. Guizani, "Vehicle-type detection based on compressed sensing and deep learning in vehicular networks," *Sensors*, vol. 18, no. 12, p. 4500, Dec. 2018.

[15] A. Taheri Tajar, A. Ramazani, and M. Mansoorizadeh, "A lightweight tiny-YOLOv3 vehicle detection approach," *J. Real-Time Image Process.*, vol. 18, no. 6, pp. 2389–2401, Dec. 2021.

[16] J. Zhao, S. Hao, C. Dai, H. Zhang, L. Zhao, Z. Ji, and I. Ganchev, "Improved vision-based vehicle detection and classification by optimized YOLOv4," *IEEE Access*, vol. 10, pp. 8590–8603, 2022.

[17] R. F. Rachmadi, K. Uchimura, G. Koutaki, and K. Ogata, "Single image vehicle classification using pseudo long short-term memory classifier," *J. Vis. Commun. Image Represent.*, vol. 56, pp. 265–274, Oct. 2018.

[18] O. Surinta and N. Boonsirisumpun, "Fast and accurate deep learning architecture on vehicle type recognition," *Current Appl. Sci. Technol.*, vol. 2021, pp. 1–16, May 2021.

[19] A. Arinaldi, J. A. Pradana, and A. A. Gurusinga, "Detection and classification of vehicles for traffic video analytics," *Proc. Comput. Sci.*, vol. 144, pp. 259–268, Jan. 2018.

[20] Z. Dong, Y. Wu, M. Pei, and Y. Jia, "Vehicle type classification using a semisupervised convolutional neural network," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 4, pp. 2247–2256, Aug. 2015.

[21] Y. Huang, R. Wu, Y. Sun, W. Wang, and X. Ding, "Vehicle logo recognition system based on convolutional neural networks with a pretraining strategy," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 4, pp. 1951–1960, Aug. 2015.

[22] Y. Gao and H. Lee, "Local tiled deep networks for recognition of vehicle make and model," *Sensors*, vol. 16, no. 2, p. 226, Feb. 2016.

[23] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.

[24] I. Bello, B. Zoph, Q. Le, A. Vaswani, and J. Shlens, "Attention augmented convolutional networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3285–3294.

[25] J. Park, S. Woo, J.-Y. Lee, and I. So Kweon, "BAM: Bottleneck attention module," 2018, *arXiv:1807.06514*.

[26] S. Woo, J. Park, J. Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 3–19.

[27] T. Chakraborti, B. McCane, S. Mills, and U. Pal, "LOOP descriptor: Local optimal-oriented pattern," *IEEE Signal Process. Lett.*, vol. 25, no. 5, pp. 635–639, May 2018.

[28] S. Parsons, "Ant colony optimization by Marco Dorigo and Thomas stützle, MIT press, 305 pp., $40.00, ISBN 0-262-04219-3," *Knowl. Eng. Rev.*, vol. 20, no. 1, pp. 92–93, Mar. 2005.

[29] Z. Luo, F. Branchaud-Charron, C. Lemaire, J. Konrad, S. Li, A. Mishra, A. Achkar, J. Eichel, and P.-M. Jodoin, "MIO-TCD: A new benchmark dataset for vehicle classification and localization," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 5129–5141, Oct. 2018.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[31] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2261–2269.

[32] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1800–1807.

[33] L. Suhao, L. Jinzhao, L. Guoquan, B. Tong, W. Huiqian, and P. Yu, "Vehicle type detection based on deep learning in traffic scene," *Proc. Comput. Sci.*, vol. 131, pp. 564–572, Jan. 2018.

[34] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.

[35] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

[36] A. Bochkovskiy, C.-Y. Wang, and H.-Y. Mark Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.

[37] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8759–8768.

[38] T.-H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma, "PCANet: A simple deep learning baseline for image classification?" *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5017–5032, Dec. 2015.

[39] J. Fang, Y. Zhou, Y. Yu, and S. Du, "Fine-grained vehicle model recognition using a coarse-to-fine convolutional neural network architecture," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 7, pp. 1782–1792, Jul. 2017.

[40] X. Ma and A. Boukerche, "An AI-based visual attention model for vehicle make and model recognition," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jul. 2020, pp. 1–6.

[41] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3D object representations for fine-grained categorization," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Dec. 2013, pp. 554–561.

[42] Y. Ding, Z. Ma, S. Wen, J. Xie, D. Chang, Z. Si, M. Wu, and H. Ling, "AP-CNN: Weakly supervised attention pyramid convolutional neural network for fine-grained visual classification," *IEEE Trans. Image Process.*, vol. 30, pp. 2826–2836, 2021.

[43] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 936–944.

[44] P. Zhuang, Y. Wang, and Y. Qiao, "Learning attentive pairwise interaction for fine-grained classification," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 13130–13137.

[45] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. 22nd ACM Int. Conf. Multimedia*, Nov. 2014, pp. 675–678.

[46] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[47] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[48] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017, *arXiv:1706.03762*.

[49] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[50] A. Jaegle, F. Gimeno, A. Brock, A. Zisserman, O. Vinyals, and J. Carreira, "Perceiver: General perception with iterative attention," 2021, *arXiv:2103.03206*.

[51] L. Yang, P. Luo, C. C. Loy, and X. Tang, "A large-scale car dataset for fine-grained categorization and verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3973–3981.

[52] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[53] G. Hinton, "Visualizing high-dimensional data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, p. 5, Jan. 2008.

[54] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1243–1252.

[55] A. Krizhevsky, "One weird trick for parallelizing convolutional neural networks," 2014, *arXiv:1404.5997*.

[56] D. F. S. Santos, G. B. De Souza, and A. N. Marana, "A 2D deep Boltzmann machine for robust and fast vehicle classification," in *Proc. 30th SIBGRAPI Conf. Graph., Patterns Images (SIBGRAPI)*, Oct. 2017, pp. 155–162.

[57] E. Baser and Y. Altun, "Detection and classification of vehicles in traffic by using Haar cascade classifier," in *Proc. 58th ISERD Int. Conf. Sci. Innov. Eng.*, Dec. 2016, pp. 19–22.

[58] S. Bai, Z. Liu, and C. Yao, "Classify vehicles in traffic scene images with deformable part-based models," *Mach. Vis. Appl.*, vol. 29, no. 3, pp. 393–403, Apr. 2017.

[59] Z. Liu, H. Hu, Y. Lin, Z. Yao, Z. Xie, Y. Wei, J. Ning, Y. Cao, Z. Zhang, L. Dong, F. Wei, and B. Guo, "Swin transformer v2: Scaling up capacity and resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 11999–12009.

[60] A. Boukerche and X. Ma, "A novel smart lightweight visual attention model for fine-grained vehicle recognition," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 13846–13862, Aug. 2022.

[61] P.-Y. Chou, Y.-Y. Kao, and C.-H. Lin, "Fine-grained visual classification with high-temperature refinement and background suppression," 2023, *arXiv:2303.06442*.

[62] D. Liu, L. Zhao, Y. Wang, and J. Kato, "Learn from each other to classify better: Cross-layer mutual attention learning for fine-grained visual classification," *Pattern Recognit.*, vol. 140, Aug. 2023, Art. no. 109550.

[63] J. Ngiam, Z. Chen, S. Bhaskar, P. Koh, and A. Ng, "Sparse filtering," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 24, 2011, pp. 1125–1133.
[64] H. Touvron, M. Cord, A. Sablayrolles, G. Synnaeve, and H. Jégou, "Going deeper with image transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 32–42.
[65] H. Kang, S. Mo, and J. Shin, "ReMixer: Object-aware mixing layer for vision transformers and mixers," in *Proc. ICLR Workshop Elements Reasoning, Objects, Struct. Causality*, 2022, pp. 1–12.
[66] K. Yuan, S. Guo, Z. Liu, A. Zhou, F. Yu, and W. Wu, "Incorporating convolution designs into visual transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 559–568.
[67] T. Ridnik, H. Lawen, A. Noy, E. Ben, B. G. Sharir, and I. Friedman, "TResNet: High performance GPU-dedicated architecture," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 1399–1408.
[68] P. Liu, H. Fu, and H. Ma, "An end-to-end convolutional network for joint detecting and denoising adversarial perturbations in vehicle classification," *Comput. Vis. Media*, vol. 7, no. 2, pp. 217–227, Jun. 2021.
[69] M. N. Roecker, Y. M. G. Costa, J. L. R. Almeida, and G. H. G. Matsushita, "Automatic vehicle type classification with convolutional neural networks," in *Proc. 25th Int. Conf. Syst., Signals Image Process. (IWSSIP)*, Jun. 2018, pp. 1–5.
[70] M. A. Hedeya, A. H. Eid, and R. F. Abdel-Kader, "A super-learner ensemble of deep networks for vehicle-type classification," *IEEE Access*, vol. 8, pp. 98266–98280, 2020.
[71] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.

SHI HAO TAN received the B.Eng. degree (Hons.) in electrical and electronics engineering from Universiti Teknologi PETRONAS, Malaysia, in 2018. He is currently pursuing the Ph.D. degree with the Department of Electrical Engineering, University of Malaya, Malaysia. His research interests include deep learning, computer vision, and intelligent transportation systems.

JOON HUANG CHUAH (Senior Member, IEEE) received the B.Eng. degree (Hons.) from Universiti Teknologi Malaysia, the M.Eng. degree from the National University of Singapore, and the M.Phil. and Ph.D. degrees from the University of Cambridge. He is currently the Head of the VIP Research Group and an Associate Professor with the Department of Electrical Engineering, Faculty of Engineering, University of Malaya. His current research interests include image processing, computational intelligence, IC design, and scanning electron microscopy. He is a fellow and was the Honorary Secretary of the Institution of Engineers, Malaysia (IEM). He is the Chairman of the Institution of Engineering and Technology (IET) Malaysia Network. He was the Honorary Treasurer of the IEEE Computational Intelligence Society (CIS) Malaysia Chapter and the Honorary Secretary of the IEEE Council on RFID Malaysia Chapter. He is a Chartered Engineer registered under the Engineering Council, U.K., and also a Professional Engineer registered under the Board of Engineers, Malaysia.

CHEE-ONN CHOW (Senior Member, IEEE) received the B.Eng. (Hons.) and M.S.E. degrees from the University of Malaya, Malaysia, in 1999 and 2001, respectively, and the D.Eng. degree from Tokai University, Japan, in 2008. He joined as a Tutor with the Department of Electrical Engineering, in 1999, and subsequently been offered a Lecturer position, in 2001. He has been an Associate Professor with the Department of Electrical Engineering, since 2015, where he has been the Head of Department, since 2020. His research interest includes wireless communications. He is a Chartered Engineer of IET, U.K., and a Professional Engineer of BEM, Malaysia.

JEEVAN KANESAN is currently an Associate Professor with the Department of Electrical Engineering, Faculty of Engineering, University of Malaya, Malaysia. He has published over 40 papers in peer-reviewed journals and conferences. His current research interests include optimal control, expert systems, and nature-inspired metaheuristics.