International Journal of Research and Technology in Electrical Industry

journal homepage: ijrtei.sbu.ac.ir

Shahid Beheshti University

An Improved MobileNet Based On Modified Attention Mechanism For Image Classification In Autonomous Vehicles

Ali Abdolahi[1], Ghazal Abdolbaghi[1], Mahdi Pourgholi Abedi[1*], Alireza Yazdizadeh[1]

[1]*Electrical Engineering Faculty, Shahid Beheshti University, Tehran, Iran*

ARTICLE INFO

Article history:
Received: 28 August 2024
Revised: 24 December 2024
Accepted: 04 January 2025

Keywords:
MobileNet
Modified attention mechanisms
Autonomous Vehicles
Image classification
Accuracy
Less-parameter algorithms

ABSTRACT

Autonomous vehicles use various sensors such as radar, LiDAR and GPS, along with computer vision algorithms, to understand their environment.These sensors gather data that needs to be analyzed for obstacle detection and navigation. However, achieving accurate object recognition is difficult due to challenges in data processing, high computational needs, and memory requirements. This study proposes a modified structure of MobileNet , called MobileNet-Att, which includes two attention mechanisms: Parallel Convolution Block Attention Module (PCBAM) and Squeeze-and-Excitation (SE) blocks. PCBAM captures multi-scale spatial features using parallel convolutions, enabling the model to focus on varying levels of spatial information. This design improves object classification and efficiency without increasing computational costs by effectively capturing richer contextual information. In the next step, SE blocks readjust the importance of each channel by "squeezing" global information through average pooling, and then "exciting" the channels based on this global context. This enables the network to emphasize essential features while minimizing the influence of irrelevant data. In essence, MobileNet-Att, with its attention mechanisms and modifications, offers a balanced approach between performance and computational loading to provide a valuable solution for object classification in autonomous vehicles. Experiments show that MobileNet-Att outperforms earlier models in accuracy and parameter efficiency on the CIFAR-10 and Caltech-101 datasets.

1. Introduction

Image analysis and classification play a crucial role in the field of computer vision. In this regard, extensive research has been done on extracting image features and developing classification algorithms for data classification purposes.

Convolutional neural networks are widely used in a variety of industrial technologies such as autonomous vehicles [1]. These networks have shown significant success in tasks like image classification [2], object detection [3] and semantic segmentation [4]. Important models like AlexNet, VGG16 and MobileNet enable autonomous systems to better understand and process visual information. Each of these models has made image classification more accurate, efficient, and faster. These features are essential for the proper functioning of self-driving vehicles.

AlexNet was a major advancement in image classification and won the ImageNet competition. It demonstrated the power of deep learning by using GPUs for efficient training. The use of ReLU activation, dropout for regularization and deep layer architectures helped overcome the limitations of previous models. However, AlexNet consists of nearly 60 million parameters, which makes it computationally intensive and This high cost limits its practicality for resource-constrained systems [5].

* Corresponding author
E-mail address: m_pourgholi@sbu.ac.ir
https://orcid.org/0000-0002-9679-1067
http://dx.doi.org/10.48308/ijrtei.2025.236727.1057

VGG16, introduced in 2014, advanced deep learning with 3x3 convolutional filters for detail capture. Its high accuracy comes with over 138 million parameters, making it computationally heavy and less suited for real-time applications such as those found in autonomous vehicles [6]. The high capacity required by these networks poses considerable challenges, including memory and computational limitations; however, accelerating the networks for implementing deep convolutional neural network models is essential. This compressing helps us to reduce parameters and effectively address the computational challenges by simplifying the computational load.

In this regard, Denil et al. demonstrated significant reduction in the parameters of deep Convolutional Neural Networks, with minimal impact on classification accuracy.They enabled researchers to successfully prune unnecessary connections and parameters in pre-trained networks [7].

MobileNet, developed in 2017, made significant advancements in improving CNN systems and introduced "DWS" , which stands for depthwise separable convolutions, as an effective method to reduce computational load while maintaining high accuracy. MobileNet is an efficient solution for autonomous vehicle applications where quick and precise decision-making are essential [8].

Similarly, ShuffleNet [9] improved pointwise channel grouping to create a structure that reduces both the number of parameters and computational cost while maintaining network accuracy. It seems that in the structure of these networks, there are still some other low-impact parameters that, by identifying and removing them,we could make the network more efficient.

As mentioned in previous sections, architectures such as AlexNet and VGG16 may not provide the high speed required for processing visual data in systems like autonomous vehicles due to their intensive structure and large number of parameters.

In recent years, advancements in Convolutional Neural Networks and their integration with attention mechanisms have improved image classification and object detection tasks. One of the most important innovations in improving CNN performance is the introduction of attention mechanisms that allow models to focus on the most relevant parts of the input [10].

The Convolutional Block Attention Module, introduced in 2018, successfully integrates attention mechanisms into CNNs by combining spatial and channel attention. CBAM helps models concentrate on relevant objects like pedestrians or vehicles while ignoring unnecessary details and enhancing accuracy in object detection and classification for self-driving cars [11].

On the other hand, SqueezeNet, developed in 2016, is an efficient CNN designed for resource-limited systems. This network achieves high performance with fewer parameters by utilizing "fire modules" that combine 1x1 and 3x3 convolutions. This approach reduces computational load while maintaining accuracy, making SqueezeNet suitable for real-time processing in autonomous vehicles where computational power and memory are crucial [12].

To illustrate that a large number of parameters does not necessarily indicate accuracy in an architecture, comparing the two architectures below is useful. VGG16, with approximately 138 million parameters, achieves high accuracy on the ImageNet dataset but requires substantial computational resources. On the other hand, MobileNet, with around 4.2 million parameters, is lighter and more efficient. Despite its smaller size, MobileNet's accuracy is only 1% lower than VGG16 on ImageNet [13]. These features makes MobileNet a good choice for applications where computational efficiency and accuracy are important.

To improve the performance of the MobileNet model, it is possible to reduce the number of parameters and computational complexity while increasing accuracy to achieve optimal processing. To achieve this goal, we first need to briefly review the main architecture of MobileNet. We can increase the network's efficiency by identifying and removing low-impact parameters along with employing other technical approaches.

2. MobileNet Architecture

The MobileNet architecture was created by Google researchers for efficient use on resource-constrained systems. One of the main challenges with CNN is their extensive computational requirements, making them unsuitable for deployment on such systems. As a result, the researchers employed a different type of convolution layer known as "Depthwise Separable Convolution" that is more effective than standard convolution because it divides the operation into two steps: depthwise convolution and pointwise convolution. In standard convolution, a K × K filter is simultaneously applied across all input channels. This process extracts spatial features and combines information from different channels at the same time which is computationally expensive.

On the other hand, DWS initially uses depthwise convolution with a K × K filter for each channel independently to reduce computations. Then, pointwise convolution uses a 1× 1 filter to mix information across channels to focuse on channel-wise interaction. This separation reduces the computational cost from $K \times K \times Cin \times Cout$ (standard) to $K \times K \times Cin + Cin \times Cout$. Similarly, the number of parameters decreases significantly. As we know in the MobileNet structure, k and C respectively represent the size of the depthwise convolutional kernel and the number of input or output channels[14].

DWS has comparable performance to standard convolution with fewer resources. Its efficiency makes it ideal architectures with fewer parameters like MobileNet. Figure 1 illustrates the MobileNet architecture that showes the overall concept of a DWS block. This specific network is built with 13 basic blocks and contains a total of 4.2 million parameters [15].

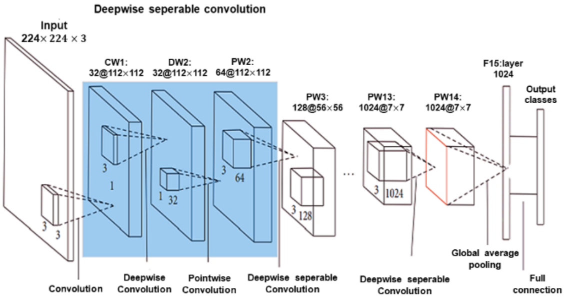


Fig. 1. Structure of MobileNet [15].

3. Attention Mechanisms[1]

Since attention mechanisms play a crucial role in this article, it is essential to briefly review them and their impact on model precision and efficiency. These mechanisms that were inspired by human cognition, enable models to focus on unique features in the dataset and prioritize important information while ignoring unnecessary details. The remarkable aspect of attention mechanisms lies in their ability to enhance model accuracy by concentrating on critical data and reducing redundancy [16]. Mathematically, this is achieved by assigning weights αij to elements of the input x{i} calculated as:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{n} \exp(e_{ik})} \qquad (1)$$

Attention mechanisms enhance the interpretability of models by identifying important parts of the data for predictions. For example, spatial attention focuses on specific regions in an image, while channel attention improves the representation of features on the feature map, as demonstrated in their respective processes. They increase model precision by enabling the model to concentrate on the most relevant parts of the input, which directly enhances its ability to make accurate predictions. For example in self attention mechanism, this is achieved through the attention formula:

$$Attention(Q,K,V) = sotmax\left(\frac{QK^T}{\sqrt{dk}}\right)V \qquad (2)$$

Here, Q,K,V represent different aspects of the input, and the dot product between Q and K computes a similarity score that determines how much attention each input should receive. The softmax function ensures that the attention model focuses more on the most relevant parts of the data. This focused attention on key information helps the model produce more accurate outputs, especially in tasks like image recognition, where relationships between elements are critical [17].

Attention mechanisms increase precision by directly connecting important parts of the input to the output. In contrast to models like RNNs, which often lose information over long sequences, attention ensures that the model remains consciously aware of the relevant context in the entire input. This helps the model make more accurate predictions. In the following sections, it will be discussed in more detail how some special attention mechanisms contribute to model precision or efficiency.

In this article, we have enhanced accuracy in our proposed architecture by modifying and integrating the Squeeze-and-Excitation block with the Convolution Block Attention Module. To effectively describe our proposed model, it is essential to first provide a comprehensive overview of these attention mechanisms.

3.1. Squeeze-and-Excitation Block

CNNs utilize the convolution operator as their fundamental building block. This allows networks to extract significant features through channel information and spatial coherence in each layer. Previous research has focused on the spatial aspect of these connections to enhance the representational capacity of CNNs. Their focus has been on improving the spatial encoding qualities in the feature hierarchy. These studies suggested a new architectural module, called Squeeze and Excitation[2].

The SE block operates in two stages. Initially, a Squeeze operation is executed, which involves performing global average pooling on each channel, thereby condensing its spatial dimensions into a singular scalar value. This is represented as:

$$z_c = F_{sq}(U_c) = \frac{1}{HW}\sum_{i-1}^{H}\sum_{j-1}^{W} U_c(i,j) \qquad (3)$$

In the subsequent step, an Excitation operation is performed on the squeeze values z_c. These values are processed through fully connected layers and subsequently subjected to a sigmoid activation function, resulting in the generation of a channel-wise attention map.

$$s = \sigma(W_2\delta(W_1 z + b_1) + b_2) \qquad (4)$$

This attention map S is used to adjust the feature map by changing the importance of each channel. This process helps the model focus on the important features. By learning which channels are more important, the SE block ensures that only the most relevant features are passed to deeper layers [18].

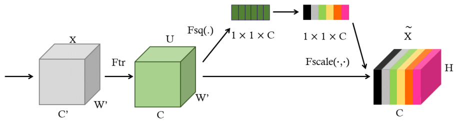


Fig. 2. Squeeze_and_Exciataion Block [18]

3.2. Convolution Block Attention Module[3]

The Convolution Block Attention Module is a simple and efficient attention mechanism that generates attention maps in both channel and spatial dimensions. These maps are multiplied with input features to dynamically adjust them. This module can be seamlessly incorporated into any CNN architecture to improve performance without adding significant computational burdens. Next, we will discuss the crucial role of spatial and channel attention in improving model performance.

[1] AM
[2] SE
[3] CBAM

3.2.1. Spatial Attention[1]

The Spatial Attention Module focuses on identifying where important informations are located. This module reduce irrelevant data reaching later layer by concentrating on specific regions of the feature map that are most relevant . This is achieved by applying max pooling and average pooling along the channel axis to extract spatial descriptors that highlight both dominant features and overall structure. These descriptors are then combined and processed to create a spatial attention map. By utilizing spatial attention, as illustrated in Figure 3, we can enhance the features of maps that improve the quality of inputs for advanced visual perception layers. This enhancement helps to boost the overall performance of the model [19].

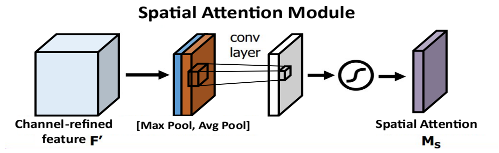


Fig. 3. Diagram of spatial attention module [19]

3.2.2. Channel Attention

The Channel Attention Module in CBAM refines feature maps by focusing on meaningful channels. Each channel of the input feature map $\mathrm{F} \in \mathbb{R}^{\mathrm{C\times H\times W}}$is considered as a feature detector. To determine channel attention, spatial features are combined using both average pooling and max pooling and produce two descriptors $F_{avg}^{c}\ and\ F_{max}^{c}$. These descriptors represent global statistics and prominent features, respectively. Both descriptors pass through a shared Multi-Layer Perceptron with a single hidden layer.

The MLP has weights $W_0 and W_1$ and its hidden layer size is reduced by a factor r for efficiency. The outputs of the MLP are summed element-wise and transferred to a sigmoid function to produce the channel attention map$\mathrm{M_c} \in \mathbb{R}^{\mathrm{C\times 1\times 1}}$ The formula is:

$$\mathrm{M_c(F)} = \sigma\left(\mathrm{MLP}(\mathrm{F^c_{avg}}) + \mathrm{MLP}(\mathrm{F^c_{max}})\right) = \sigma(\mathrm{W_1}\left(\mathrm{W_0}(\mathrm{F^c_{avg}})\right) + \mathrm{W_1}\left(\mathrm{W_0}(\mathrm{F^c_{max}})\right)) \quad (5)$$

Finally, the input feature map F is refined by multiplying it element-wise with the broadcasted channel attention map:

$$\mathrm{F}' = \mathrm{M_c(F)} \otimes \mathrm{F} \quad (6)$$

As you observed, channel attention creates a significant improvement in feature representation and effectively eliminates irrelevant data by disregarding unimportant information [20].

3.2.3. The Reason for using both structure

To achieve better results, it is recommended to use a combination of both types of attention. CBAM consists of channel and spatial attention modules, compute complementary attention, focusing on 'what' and 'where' respectively. by enhancing key features, channel attention helps the model prioritize the most informative features in each layer. On the other hand, spatial attention emphasizes relevant information within the feature map that is critical for the learning process. Considering this, the two modules can be placed in a parallel or sequential manner but it has been proven that the sequential arrangement provides better results compared to the parallel arrangement. For the sequential process, the experimental results demonstrate that the channel-first order performs slightly better than the spatial-first order. The combination of these two mechanisms maximizes the model's capacity to capture high-level and detailed features, leading to more robust predictions. The structure of CBAM is illustrated in Figure 4.

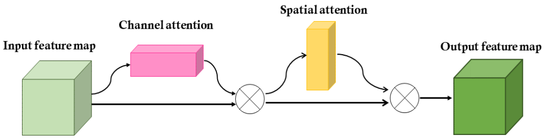


Fig. 4. CBAM structure[20]

4. Proposed network

MobileNet network generally consists of 28 layers. These layers have different impacts on the network's computational load. Previous researches have significantly contributed to remove less important layers in the MobileNet's architecture. However, there are still layers in this network that considerably increase the number of parameters while having minimal impact on the overall accuracy of MobileNet.

In the process of improving deep neural network models, the main goal is to reduce computational complexity and increase efficiency, especially when models need to be deployed on devices with limited resources. In this context, removing unnecessary layers can help improve the model. In the proposed MobileNet architecture, which consists of 28 layers, 10 layers have been selected to be removed from the original structure, as shown in the table I.

The selection of these layers for removal involved sensitivity analysis, layer impact evaluation and pruning methods. In this process, each layer was individually removed and the model's performance was evaluated after each removal. The results showed that eliminating 5 depthwise convolution layers and 5 pointwise convolution layers did not have a significant impact on the model's accuracy beacuase these layers extracted similar features in specific stages of the network that had limited result on the final performance.

Another reason for rejecting these layers was the analysis of computational costs and memory. while depthwise convolution and pointwise convolution layers are efficient in reducing the number of parameters, they can incur high computational costs in certain parts of the network, without providing a significant impact on the model's accuracy. Especially in networks with many

[1] SA

layers, adding them increases computation and memory usage.

Finally, to ensure reliability, pruning method (removing weights and layers), was used to automatically identify and remove layers that had less effect on feature learning and predictions. Results confirmed that the 5 depthwise convolution layers and 5 pointwise convolution layers extracted repetitive features and had a high degree of similarity with other layers in the network. These layers, located in the middle stages of the network, were responsible for independently processing and combining features, and based on the empirical analysis, their removal did not significantly affect the model's accuracy.

It is important to emphasize that eliminating the layers must be done carefully to preserve the stability of the model. The initial layers, responsible for extracting fundamental features such as edges and simple shapes, are essential for the model's performance and must remain. Removing these layers would be detrimental. It is necessary to note that the removed layers, did not have these key roles and were suitable candidates for removing.

As a result, by deleting 10 layers, including 5 depthwise and 5 pointwise convolution layers, using methods like sensitivity analysis and pruning, the number of parameters was dramatically reduced. However, the network experienced only a minimal decrease in accuracy.

To address this issue, we took two important steps. In the first step, we enhanced the network structure by combining activation functions like ReLU and H-Swish. H-Swish as a non-linear activation function that is introduced in [21], can significantly improve neural network accuracy when used instead of ReLU.

Table I. Structure of MobileNet layers.

Type / Stride	**Filter Shape**	**Input Size**
Conv / s2	3 x 3 x 3 x 32	224 x 224 x 3
Conv dw / s1	3 x 3 x 32 dw	112 x 112 x 32
Conv / s1	1 x 1 x 32 x 64	112 x 112 x 32
Conv dw / s2	3 x 3 x 64 dw	112 x 112 x 64
Conv / s1	1 x 1 x 64 x 128	56 x 56 x 64
Conv dw / s1	3 x 3 x 128 dw	56 x 56 x 128
Conv / s1	1 x 1 x 128 x 128	56 x 56 x 128
Conv dw / s2	3 x 3 x 128 dw	56 x 56 x 128
Conv / s1	1 x 1 x 128 x 256	56 x 56 x 128
Conv dw / s1	3 x 3 x 256 dw	28 x 28 x 256
Conv / s1	1 x 1 x 256 x 256	28 x 28 x 256
Conv dw / s2	3 x 3 x 256 dw	28 x 28 x 256
Conv / s1	1 x 1 x 256 x 512	28 x 28 x 256
5 × Conv dw / s1	**3 x 3 x 512 dw**	**14 x 14 x 512**
5 × Conv / s1	**1 x 1 x 512 x 512**	**14 x 14 x 512**
Conv dw / s2	3 x 3 x 512 dw	14 x 14 x 512
Conv / s1	1 x 1 x 512 x 1024	14 x 14 x 512
Conv dw / s2	3 x 3 x 1024 dw	7 x 7 x 1024
Conv / s1	1 x 1 x 1024 x 1024	7 x 7 x 1024
Avg Pool / s1	Pool 7 x 7	7 x 7 x 1024
FC / s1	1024 x 1000	7 x 7 x 1024
Softmax / s1	Classifier	7 x 7 x 1024

The cost of applying H-Swish decreases as we move deeper into the network, because the activation memory of each layer is typically halved with each reduction in resolution. Therefore, we find the best results of this method are achieved only when it is used in deep layers. The goal of H-Swish is to improve network performance, and it can be calculated using the following formula.

$$h\text{-}swish = X\frac{Relu\,(x+3)}{6} \qquad (7)$$

The attention that is used in orginial structure of MobileNet is a classic example of channel attention, which primarily determines the importance of each feature channel through compression and activation. This helps the network emphasize some features and exclusively concentrates on the correlation between feature maps across various channels and ignores spatial information or pixel details that are very important for image classification.

In fact, CBAM is a combination of channel-wise attention and spatial attention[1]. This means that initially the feature map of the image is weighted by channel attention and then weighted by spatial attention to obtain the output feature map. This is a typical sequential structure. In this process, the spatial attention mechanism takes its input from the channel attention, which can interfere with spatial attention performance and thus affect the overall efficiency of the attention mechanism.

To address these issues and improve the accuracy of the architecture , we make modifications to the Convolution Block Attention Module. Ultimately, to utilize both channel and spatial information, we employ a Mixed-Domain Attention Mechanism to achieve more effective improvements in the proposed network.

The proposed structure combines the Parallel Convolution Block Attention Module with the Squeeze-and-Excitation mechanism to enhance network accuracy and improve feature representation . This design overcomes the limitations of serial mechanisms like CBAM by processing attention channel and spatially in parallel. the parallel approach improves accuracy as it allows the network to extract simultaneously important features of both channel and spatial without interference. This simultaneous processing helps the network capture complementary information from both domains. As a result, the network can generate more precise and effective feature representations.

Another factor that helps improve accuracy is preserving original information. Skip connections, which act as a direct path, transfer the main input features to the output. This ensures that essential information is not lost during the attention process and the basic features stay preserved. This helps maintain the final network's accuracy. Furthermore, the skip connection improves gradient flow during backpropagation. This reduces issues like such as vanishing gradients and allows better learning in deeper layers.

Simultaneous processing in this structure ensures that both channel and spatial attention are applied at the same time to the feature map. Channel attention, by adjusting the importance of channels, allows the network to focus

[1] SA

on important features, while spatial attention emphasizes critical pixels or regions of the feature map. This comprehensive improvement process enables the network to better identify detailed patterns and improve image classification performance.

On the other hand , SENet initially operates by applying global average pooling to compress the spatial dimensions of the feature map. This action creates a compact model that captures the global distribution of features in each channel.

This representation then passed through two fully connected layers for dimensionality reduction to identify critical inter-channel relationships and another layer for restoring the original dimensions while generating channel-specific scaling factors. These scaling factors are applied through the feature map multiplication. This process selectively emphasizes the most informative channels and reduces the impact of less relevant ones.

SENet,by enhancing the channel features, ensures that the network can extract the most meaningful information from the feature maps generated by PCBAM.

The integration of PCBAM and SENet, along with their implementation at the end of layers, significantly enhances their effectiveness. At this stage, the feature maps contain semantically rich semantic information that is essential for accurate classification. SENet refines this information while complementing PCBAM's simultaneous processing of channel and spatial attention. Together, these mechanisms enable the network to more effectively identify meaningful patterns.

In this structure, the input feature map concurrently passes through both channel attention and spatial attention processes to obtain the relevant weights. These weights are then directly applied to produce the output feature map. This process is mathematically expressed as follows.

$$F_H = M_c(F) * M_s(F) * F \qquad (8)$$

As shown in Figure 5, we have improved network accuracy by using both Parallel Convolution Block Attention Module and Squeeze-and-Excitation simultaneously at the end of the network. While this approach adds a few parameters to our proposed architecture, results show that accuracy is increased.

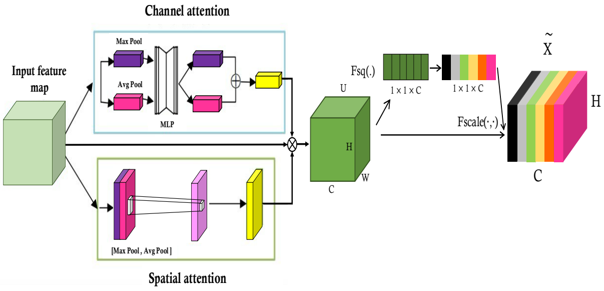


Fig. 5. Our Proposed Attention Mechanism

The final network structure, as outlined in Table II, includes the combined SENet and PCBAM attention mechanisms, the activation functions and the removed layers. Key innovation of this paper is the combination and modification of various methods to improve object classification accuracy. this approach also aims to reduce the number of parameters and computational complexity.

Table II. Final structure of proposed network.

Type / Stride	Filter Shape	Input Size	Activation Function
Conv / s2	3 x 3 x 3 x 32	224 x 224 x 3	**Relu**
Conv dw / s1	3 x 3 x 32 dw	112 x 112 x 32	**Relu**
Conv / s1	1 x 1 x 32 x 64	112 x 112 x 32	**Relu**
Conv dw / s2	3 x 3 x 64 dw	112 x 112 x 64	**Relu**
Conv / s1	1 x 1 x 64 x 128	56 x 56 x 64	**Relu**
Conv dw / s1	3 x 3 x 128 dw	56 x 56 x 128	**Relu**
Conv / s1	1 x 1 x 128 x 128	56 x 56 x 128	**Relu**
Conv dw / s2	3 x 3 x 128 dw	56 x 56 x 128	**Relu**
Conv / s1	1 x 1 x 128 x 256	56 x 56 x 128	**Relu**
Conv dw / s1	3 x 3 x 256 dw	28 x 28 x 256	**Relu**
Conv / s1	1 x 1 x 256 x 256	28 x 28 x 256	**h-swish**
Conv dw / s2	3 x 3 x 256 dw	28 x 28 x 256	**h-swish**
Conv / s1	1 x 1 x 256 x 512	28 x 28 x 256	**h-swish**
Conv dw / s2	3 x 3 x 512 dw	14 x 14 x 512	**h-swish**
Conv / s1	1 x 1 x 512 x 1024	14 x 14 x 512	**h-swish**
Conv dw / s2	3 x 3 x 1024 dw	7 x 7 x 1024	**h-swish**
Conv / s1	1 x 1 x 1024 x 1024	7 x 7 x 1024	**h-swish**
(our proposed Attention Mechanism) **Parallel Convolution Block Attention Module as the First Attention Mechanism** **Squeeze-and-Excitation Block is combined with Previous as the Second Attention Mechanism**			
Avg Pool / s1	Pool 7 x 7	7 x 7 x 1024	
FC / s1	1024 x 1000	7 x 7 x 1024	
Softmax / s1	Classifier	7 x 7 x 1000	

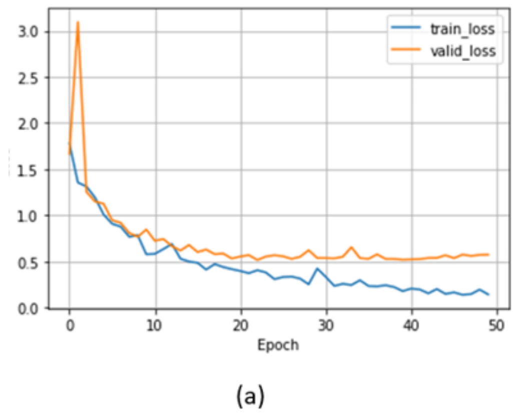
5. Results and discussion

CIFAR-10 is suitable for testing how well a model handles 7 low-resolution images in a controlled set of object classes. It's especially useful for benchmarking models that need to work with smaller, faster images, often used in mobile or embedded systems. Caltech-101 is more challenging and allows testing of the model's ability to classify a larger variety of categories with higher-resolution images. His feature makes it more suitable for real-world applications, such as autonomous vehicles or complex image recognition tasks. By evaluating the model on both datasets, we can demonstrate that the model has the ability to generalize across different levels of complexity, from simple tasks (CIFAR-10) to more challenging ones (Caltech-101).

To validate our model, we trained it using both Caltech-101 and CIFAR-10 datasets. The Caltech-101 dataset contains 9,145 images distributed across 101 classes. Each class contains between 40 to 800 images. The images in the dataset were labeled initially and then randomly shuffled for our experiments. From these, 1,500 images were randomly selected for testing, while the remaining images were used for training purposes.

On the other hand, the CIFAR-10 dataset consists of 60,000 color images with dimensions of 32x32 pixels. These images are categorized into 10 classes, with each class containing 6,000 images. This dataset is divided into 50,000 training images and 10,000 test images.The CIFAR-10 dataset is divided into five training sets and one test set, each containing 10,000 images, with 1,000 images randomly selected per class in the test set.The training sets consist of the remaining images, selected randomly. As a result, some categories may have more images than others within the training sets. Overall, the training sets collectively contain 5,000 images from each class.

Initially, we trained the proposed model on the CIFAR-10 dataset for 50 epochs. Figures 6a and 6b present loss and accuracy curves, respectively. As illustrated in the figure below, the loss curve gradually decreases, while the accuracy curve reaches its maximum value in the final epochs.



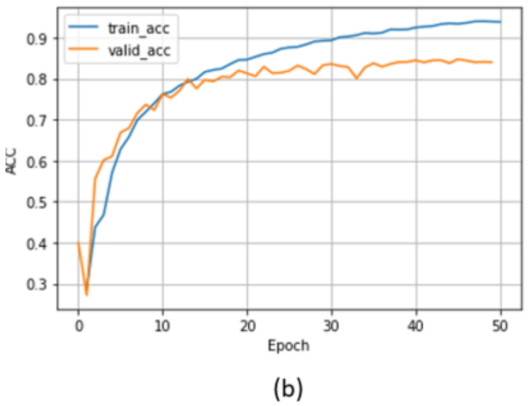


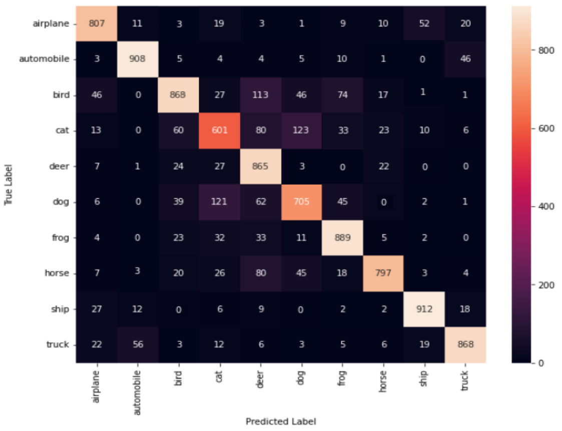
Fig. 6.
(a) loss, and (b) accuracy of the train model vs. valid model on CIFAR-10 dataset.

According to table III it can be seen that in CIFAR-10 dataset, over 92% of the cases that our proposed model predicted as positive were actually positive And has correctly identified over 86% of the true positive cases. The balance between precision and recall, which is represented by the F1_score, was also 89%.

Table III. Inforrmation of models on CIFAR-10 dataset.

Network	precision	Recall	F1-score
Proposed Model	0.923	0.864	0.889

Confusion matrix as a valuable tool for evaluating the performance of classification models , provides detailed insights into the model's ability to correctly and incorrectly identify different classes. As shown in CIFAR-10 confusion matrix, the proposed model correctly classified 908 instances of the car class and accurately identified 705 instances of the dog class, which can be considered as potential obstacles for the cars.



	airplane	automobile	bird	cat	deer	dog	frog	horse	ship	truck
airplane	807	11	3	19	3	1	9	10	52	20
automobile	3	908	5	4	4	5	10	1	0	46
bird	46	0	868	27	113	46	74	17	1	1
cat	13	0	60	601	80	123	33	23	10	6
deer	7	1	24	27	865	3	0	22	0	0
dog	6	0	39	121	62	705	45	0	2	1
frog	4	0	23	32	33	11	889	5	2	0
horse	7	3	20	26	80	45	18	797	3	4
ship	27	12	0	6	9	0	2	2	912	18
truck	22	56	3	12	6	3	5	6	19	868

Fig. 7. Confusion matrix of the network on the CIFAR-10 dataset.

In the next stage, we trained our proposed network on Caltech-101 dataset, which contains more classes and higher-quality images. The aim of this work was to demonstrate the efficiency of the proposed model on another benchmark dataset. As illustrated in the figure 8a and 8b, the performance of the trained network on the Caltech_101 dataset is also remarkable. In fact, as the number of training epochs increased, the model achieved higher accuracy and lower loss.

As seen in Table VI The proposed model achieves a good balance between Precision and Recall based on the F1-score. Moreover, the results obtained in terms of accuracy and recall on the CALTECH-101 dataset demonstrate the effectiveness of our proposed model.

The results of the confusion matrix in figure 9 indicate that the proposed model has performed well overall on the Caltech 101 dataset and has been able to accurately classify the categories. The high number of True Positives and True Negatives prove that the model has mostly made correct predictions. In contrast, low number of False Positives and False Negatives suggest that the model has not made many mistakes in classifying certain categories.

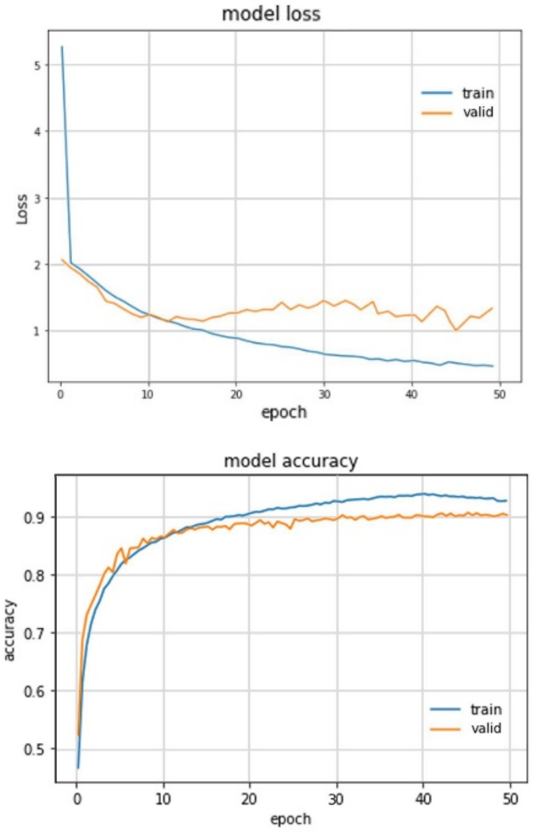
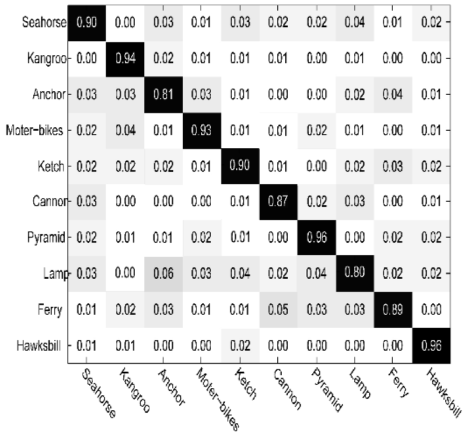


Fig. 8. (a) loss, and (b) accuracy of the train model vs. valid model on Caltech-101 dataset.

Table VI. Information of models on Caltech-101 dataset.

Network	Precision	Recall	F1-score
Proposed Model	0.893	0.847	0.834



	Seahorse	Kangroo	Anchor	Moter-bikes	Ketch	Cannon	Pyramid	Lamp	Ferry	Hawksbill
Seahorse	0.90	0.00	0.03	0.01	0.03	0.02	0.02	0.04	0.01	0.02
Kangroo	0.00	0.94	0.02	0.01	0.01	0.01	0.00	0.01	0.00	0.00
Anchor	0.03	0.03	0.81	0.03	0.01	0.00	0.00	0.02	0.04	0.01
Moter-bikes	0.02	0.04	0.01	0.93	0.01	0.01	0.02	0.01	0.00	0.01
Ketch	0.02	0.02	0.02	0.01	0.90	0.01	0.00	0.02	0.03	0.02
Cannon	0.03	0.00	0.00	0.00	0.01	0.87	0.02	0.03	0.00	0.01
Pyramid	0.02	0.01	0.01	0.02	0.01	0.00	0.96	0.00	0.02	0.02
Lamp	0.03	0.00	0.06	0.03	0.04	0.02	0.04	0.80	0.02	0.02
Ferry	0.01	0.02	0.03	0.01	0.01	0.05	0.03	0.03	0.89	0.00
Hawksbill	0.01	0.01	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.96

Fig. 9. Confusion matrix of the network on the Caltech-101 dataset (10 classes of 101 classes).

Therefore, based on these results, it can be said that the model has performed well overall.

In this article, we analyzed the performance of various networks on two datasets and compared them with our proposed network. Additionally, a closer look at the number of parameters in other networks shows how efficient our proposed network is in reducing computational and graphical processing load. The comparison table (V) clearly shows that despite having fewer parameters, our proposed network achieves significantly higher accuracy compared to its counterparts.

Table V. Comparison of different information of different types of networks: (network's volume and parameters number are estimated on 1000 classes).

	Network accuracy on Caltech-101 dataset	Network accuracy on CIFAR-10 dataset	Network size (MB)	Number of network parameter
mobileNet	76.73%	84.2%	16.27	4.264.808
shuffleNet	77.3%	83.4%	9.50	2.491.504
GoogleNet	91.05%	92.9%	26.70	6.998.552
mobileNetV2	78.2 %	87.3%	13.50	3.489.552
VGG16	89.37%	91.4%	527.79	138.3 57.54
Proposed Model	90.07%	92.6%	12.25	3.211.245

Our proposed network outperforms others on both datasets in terms of accuracy and number of used parameters, as shown in Table V.

To introduce an application of this network in autonomous vehicles, we considered multi-object images containing humans and vehicles. The objects in These Images were localized by the pre-trained Single-Shot MultiBox Detector MobileNet then the single objects were given to our proposed network for classification. It is important to note that the SSD MobileNet was only used in the final section for lockalization in order to obtaine graphic outputs for for presenting a practical application. As shown in Figure. 10.

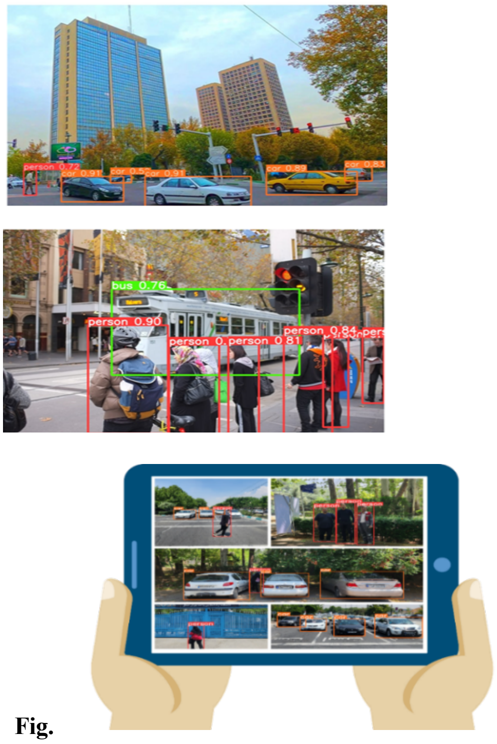


Fig. 10. Classifying of Localized Objects by Proposed Network.

6. Conclusion

The results of this study show that the proposed MobileNet-Att model, utilizing modified SE (Squeeze-and-Excitation) and PCBAM (Parallel Convolution Block Attention Module), has achieved considerable improvements in accuracy and computational efficiency. These improvements include reducing the number of layers and parameters in the network, along with the use of effective activation functions and parallel architectures in attention mechanisms.

Evaluation of the model on the CIFAR-10 and Caltech-101 datasets revealed that MobileNet-Att provides remarkable accuracy in object classification while reducing computational load.These results confirm that the proposed model is not only suitable for autonomous systems but can also contribute to enhance the safety and efficiency of these systems. This research presents a practical approach to improving the performance and efficiency of convolutional neural networks in real-world applications.

It should be noted that our proposed model may have lower performance with complex datasets, such as multi-modal data. To address this issue, AutoML or dynamic attention mechanisms can enhance accuracy. Using multi-head attention mechanisms, which allow the model to simultaneously focus on multiple features or different parts of the data, can also improve efficiency.

In addition to reducing dependence on labeled data, unsupervised or semi-supervised learning methods can be used. In real-time applications, computational complexity can be reduced by using model compression or distributed parallel processing. Furthermore, combining models with recurrent neural networks or using multimodal networks can enhance performance in sequential or time-series data.

References

[1]. T. Turay, T. Vladimirova, “Toward performing image classification and object detection with convolutional neural networks in autonomous driving systems”, A survey. IEEE Access, 10, pp.14076-14119, 2022.

[2]. P. Lakshmi Prasanna. D. Raghava Lavanya. T. Sasidhar. S. Babu, “Image classification using convolutional neural networks. International Journal of Pure and Applied Mathematics”, 119(17), pp.1307-1319. 2018.

[3]. Z. Su, C. Hu, J. Hao, P. Ge, B. Han. December, “Target Detection in Single-Photon Lidar Using CNN Based on Point Cloud Method. In Photonics. Vol. 11, No. 1, p. 43. MDPI. 2023.

[4]. M. Ansari, A. Choudhary, S. Bhosale. “Recent Researches on Image Classification Using Deep Learning Approach. pp. 481-488. IEEE, 2023.

[5]. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in Proceedings of the Neural Information Processing Systems Conference, pp. 1–9, Lake Tahoe, NV, USA, December 2012.

[6]. K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in Proceedings of the International Conference on Learning Representations (ICLR), pp. 1–14, San Diego, CA, USA, May 2015.

[7]. M. Denil, B. Shakibi, L. Dinh, M. A. Ranzato, and N. De Freitas, “Predicting parameters in deep learning,” in Proceedings of the Advances in Neural Information Processing Systems, pp. 2148–2156, Lake Tahoe, NV, USA, December 2013.

[8]. B. Khasoggi, E. Ermatita, and S. Samsuryadi, “Efficient MobileNet architecture as image recognition on mobile and embedded devices,” in Indonesian Journal of Electrical Engineering and Computer Science, vol. 16, no. 1, pp. 389–394, October 2019.

[9]. X. Zhang, X. Zhou, M. Lin, and J. Sun, “Shufflenet: an extremely efficient convolutional neural network for mobile devices,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6848–6856, Salt Lake City, UT, USA, June 2018.

[10]. W. A. AlZoubi, G. B. Desale, S. Bakyarani, U. K. C. R., D. Nimma, K. Swetha, and B. Kiran Bala, “Attention-based deep learning approach for pedestrian detection in self-driving cars,” in International Journal of Advanced Computer Science and Applications, vol. 15, no. 8, pp. 923–929, 2024.

[11]. S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, “CBAM: convolutional block attention module,” in Proceedings of arXiv Preprint, arXiv:1807.06521, July 2018.

[12]. F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, “SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5 MB model size,”2016,

[13]. J. Wang, F. Deng, and B. Wei, “Defect detection scheme for key equipment of transmission line for complex environment,” in Electronics, vol. 11, no. 15, pp. 2332, July 2022.

[14]. Wang T, Ray N. Compact depth-wise separable precise network for depth completion. IEEE Access. Jul 11.2023.

[15]. Hassan, M. Ali, N. M. Durrani, and M. A. Tahir, “An empirical analysis of deep learning architectures for vehicle make and model recognition,” in IEEE Access, vol. XX, pp. 1–13, 2021.

[16]. D. Soydaner, “Attention mechanism in neural networks: where it comes and where it goes,” in Proceedings of arXiv Preprint, arXiv:2204.13154, April 2022.

[17]. M.-H. Guo, T.-X. Xu, J.-J. Liu, Z.-N. Liu, P.-T. Jiang, T.-J. Mu, S.-H. Zhang, R. R. Martin, M.-M. Cheng, and S.-M. Hu, “Attention mechanisms in computer vision: a survey,” in Proceedings of arXiv Preprint, arXiv:2111.07624, November 2021.

[18]. J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-Excitation Networks," in IEEE Transactions

on Pattern Analysis and Machine Intelligence, vol. 42, no. 5, pp. 1123–1135, May 2020.

[19]. M. A. B. Abbass and H.-S. Kang, "Violence detection enhancement by involving convolutional block attention modules into various deep learning architectures," in IEEE Access, vol. XX, pp. 1–12, 2023.

[20]. Woo, S. P, Jongchan, L. Joon-Young" In So, CBAM: Convolutional Block Attention Module. ECCV" https://doi.org/ 10.48550/arXiv.1807.06521, 2018

[21]. D. Hendrycks, K. Gimpel, "Bridging nonlinearities and stochastic regularizers with gaussian error linear units", CoRR, abs/1606.08415, 2016.