Received: 20 August 2023 | *Revised:* 8 November 2023 | *Accepted:* 1 January 2024
DOI: 10.37917/ijeee.21.1.17 *Early View* | *June 2025*

Iraqi Journal for Electrical and Electronic Engineering
Original Article

Open Access
IJEEE University of Basrah College of Engineering

A Comparative Study of Deep Learning Methods-Based Object/Image Categorization

Saad Albawi*[1], Layth Kamil Almajmaie[1,2], Ali J. Abboud[1]
[1]Computer Engineering Department, Engineering College, University of Diyala, Diyala, Iraq
[2]Computer Engineering Department, University of Technology, Baghdad, Iraq

Correspondance
*Saad Albawi
Computer Engineering Department, Engineering College, University of Diyala, Iraq
Email: saadalbawi@uodiyala.edu.iq

Abstract
In recent years, there has been a considerable rise in the applications in which object or image categorization is beneficial for example, analyzing medicinal images, assisting persons to organize their collections of photos, recognizing what is around self-driving vehicles, and many more. These applications necessitate accurately labeled datasets, in their majority involve an extensive diversity in the types of images, from cats or dogs to roads, landscapes, and so forth. The fundamental aim of image categorization is to predict the category or class for the input image by specifying to which it belongs. For human beings, this is not a considerable thing, however, learning computers to perceive represents a hard issue that has become a broad area of research interest, and both computer vision techniques and deep learning algorithms have evolved. Conventional techniques utilize local descriptors for finding likeness between images, however, nowadays; progress in technology has provided the utilization of deep learning algorithms, especially the Convolutional Neural Networks (CNNs) to auto-extract representative image patterns and features for classification The fundamental aim of this paper is to inspect and explain how to utilize the algorithms and technologies of deep learning to accurately classify a dataset of images into their respective categories and keep model structure complication to a minimum. To achieve this aim, must focus precisely and accurately on categorizing the objects or images into their respective categories with excellent results. And, specify the best deep learning-based models in image processing and categorization. The developed CNN-based models have been proposed and a lot of pre-training models such as (VGG19, DenseNet201, ResNet152V2, MobileNetV2, and InceptionV3) have been presented, and all these models are trained on the Caltech-101 and Caltech-256 datasets. Extensive and comparative experiments were conducted on this dataset, and the obtained results demonstrate the effectiveness of the proposed models. The obtained results demonstrate the effectiveness of the proposed models. The accuracy for Caltech-101 and Caltech-256 datasets was (98.06% and 90%) respectively.

Keywords
Image Recognition, Deep Learning Methods, VGG19, DenseNet201, ResNet152V2, MobileNetV2, InceptionV3, Developed CNN Models.

I. Introduction

The main aim of computer vision focuses on training computers to translate the visible world similar to what human beings do. Object or image categorization represents an encouraging domain in computer vision, with various actual applications like biological identification, face recognition, self-driving, and medical diagnosis. The image categorization task indicates classifying images into various classes or categories in accordance with the images' content. Owing to the variations and high dimensionality in the data of images, it is challenging to evolve a technique capable of capturing beneficial information from images and then conducting effective classification [1]. Images consisted of numerous thousands of pixels that include a considerable amount of irrelevant infor-

mation, and this redundant information may conduct difficulty in attaining high accuracy of classification. The feature extraction process aims to convert raw pixels of images into valuable or informative features that are capable of efficiently reducing the dimensionality of images. Efficient techniques of features extraction is capable of making the performance of an image categorization technique efficient. At present, there are many conventional feature extraction techniques that have been presented to catch interesting features from an image, like the transform of scale invariant feature, speeded up robust features, histogram of oriented gradients, and local binary pattern. These features are able to reflect the useful portion of the images by catching the salient information (orientations, edges, and textures) inside them. But it is not easy to obtain encouraging results in complicated tasks of image categorization via utilizing a single technique of feature extraction because a single features kind mightn't efficiently indicate the entire image. Moreover, high image variations in background, illumination, scaling, and rotation maximize the complication in beneficial features extraction for categorization [2]. Even though the available conventional image categorization techniques have been exceedingly implemented in practical issues, there are several issues in the process of application, like non-satisfactory outcomes, weak accuracy of classification, and low adaptability. For the image categorization procedure, these techniques work on separating the processes of feature extraction and classification into two stages. While the deep learning-based models have a mighty learning capability by integrating the processes of feature extraction and classification into one stage for completing the test of image categorization, which can efficiently make the image classification accurate [3] [4]. However, these models might need abundant experience in designing the structure to certain task, and require a considerable number of computational resources and training instances. Deep learning algorithms have proven their capability for achieving high results of accuracy in various application fields like image categorization, and other complex tasks. It has been demonstrated to be quite effective and valuable to automate the typically exhausted and occasionally debatable preprocessing phase of extracting features [5] [6]. In the last few years, the Convolutional Neural Networks (CNNs) have succeeded and achieved substantial progress in various fields. In comparison to conventional image processing techniques, CNNs are capable of extracting higher-level image information and hold higher complexity. Even though CNNs have made promising accomplishments in various fields, the effectiveness issue represents a pressing issue to be solved. The effectiveness issue can be represented in the issues of prediction speed and model storage. The CNNs include a considerable number of parameters that need a considerable amount of memory to be saved, and the model prediction speed should be enhanced when CNNs are utilized practically [7]. The fundamental contribution of this paper is to inspect and explain how to utilize the algorithms and technologies of deep learning to accurately classify a dataset of images into their respective categories and keep model structure complications to a minimum. Consequently, the following objectives have been accomplished:

1. Firstly, implementing two proposed progressive CNN models.
2. Secondly, implementing the most widely utilized pre-trained CNN models (VGG19, ResNet152V2, DenseNet201, InceptionV3, MobileNetV2).
3. Finally, evaluating the performance of the adopted CNN models using two benchmark datasets (Caltech-101 and Caltech-256).

II. Related Works

Object or image categorization has continually been a hot scientific research trend all over the world, and the appearance of deep learning algorithms has encouraged the development of this domain. CNNs have progressively become the main algorithms for image categorization since 2012, and generally, the CNN architecture implemented for object localization and object detection tasks is acquired from the network architecture in image categorization. Some of the related object/image categorization schemes that are based on deep learning methods are reviewed briefly in this section. Wang et al. [8], improved various models (named expanded convolution MobileNets) based on the MobileNet model by exchanging the standard convolutional with the expanded convolution layers. These improved models increased the receptive domain of the convolutional filters for obtaining high results of categorization accuracy. The experimentation was conducted using Caltech-101, and Caltech-256 datasets and the obtained results demonstrated that the improved MobileNet models were capable of getting better accuracies than standard MobileNet. Ravi [9], proposed an object/image categorization framework in which firstly a pre-trained CNN named deep residual neural network (ResNet50)) was employed to extract features. Then, the optimal rules of categorization were learned from those features using the algorithm of Brain Storm Optimization. Finally, the classification was achieved using a fuzzy rule-based classifier. This framework was implemented using the Caltech-101 dataset and the obtained categorization accuracy was 86%. Bansal et al. [10], presented an improved system for categorizing the images in which firstly the features were extracted using the pre-trained CNN method (Visual Geometry Group (VGG19)) and four handcrafted methods (Orient Fast & Rotated Brief (ORB), Scale Invariant Features Transformation

(SIFT), Speed Up Robust Feature (SURF), Shi-Tomasi Corner Detection (SCD)). Then, the algorithm of k-means clustering was utilized to choose the significant features. After that, these selected features were classified using the diverse classifiers (eXtreme Gradient Boosting, Decision Tree, Gaussian Naïve Bayes, and Random Forest). This system was evaluated using Caltech-101dataset, and the outcomes showed that the implementation of the Random Forest classifier on the combined features outperformed the other classifiers with an accuracy of 93.73%. Rao and Mahantesh [11], presented an adopted learning model based on the architecture of VGG16 for automatically learning the features of an input image using Caltech-101 and Caltech-256 datasets. This model firstly was trained using the algorithm of gradient descent, and the weights (parameters) were optimized using the back-propagation method. Subsequently, the convolutional layers were utilized for feature learning, in addition to effective regularization utilizing dropout and the augmentation of data. Finally, the label of the class is predicted for the requested image. The achieved results proved the model's capacity to perceive high-level semantics. Zeynallı [12], presented an image categorization method in which firstly the features were extracted using VGG16. Then, the outcomes of image categorization using the diverse classifiers (Support Vector Machine, Random Forest, and Logistic Regression) were compared. This method was evaluated using Caltech-101dataset, and the outcomes showed that the implementation of the Logistic Regression classifier on the extracted features outperformed the other classifiers with an accuracy of 94.65%.

III. Main Preliminary Concept

A. Typical CNN

CNN can be effectively utilized for object or image categorization. Its architecture is built using three essential layers Convolutional, Pooling, and Fully-connected. This architecture is demonstrated in Fig. 1. The first and second type layers are responsible for extracting features, while the third layer is responsible for categorization. The activation functions (such as Rectified Linear Unit (ReLu), Sigmoid, and Softmax) and dropout layer can be considered portions of these layers [13]. The convolutional layer includes a set of filters (kernels), each like a matrix that is essentially smaller than the input data; the kernel firstly slides over the input data and implements a dot function, and the outcome represents a feature map. The feature maps concerning the subsequent layers are constructed via merging the feature maps for the former layers [14]. The pooling layer carries out the down-sampling process and thus minimizes the number of parameters while maintaining the essential features. There are several types of pooling; however, maximum pooling represents the widely utilized operation that provides the maximum-element from

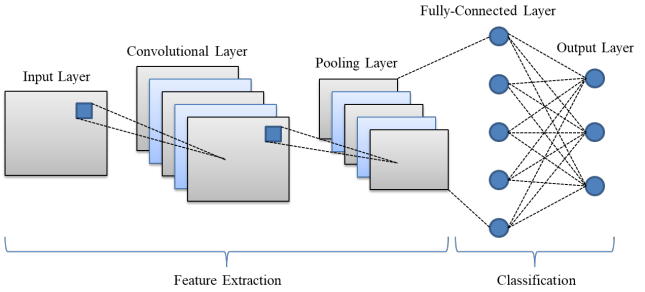

Fig. 1. First developed CNN architecture.

the feature map. While the fully connected layer accepts, the input from the previous phases to be classified [15]. When the whole features are related to the fully connected layer, this can lead to the issue of overfitting. In order to overcome this issue, a dropout layer can be utilized prior to the output layers. This layer works on randomly discarding some neurons from the network and thus produces a reduced-size model. Each activation function gets the output structure from the former layer and converts it to another structure appropriate to the following layer to be considered as input. It can be utilized in any portion of the network to make the model capable of learning the complicated patterns from the data [13].

B. Pre Training CNN

Pre-trained CNNs represent models trained using an enormous amount of datasets concerning particular tasks. Recently, these models have been utilized in most of the object or image classification architectures, which produce higher accuracy and save time. The most widely utilized pre-trained CNNs are considered and implemented in this paper.

1. VGG19: Simonyan and Zisserman presented VGG in 2015 [16]. Its architectural depth was expanded by adding more layers that are convolutional with smaller filters (3×3) for improving its performance. There are several versions of VGG architecture (such as VGG16 and VGG19) that are trained depending on the ImageNet dataset. The architecture of VGG16 involves thirteen convolution layers and three fully connected layers. Each convolution layer involves a ReLU and a maximum pooling layer. Two of fully connected layers existing in this architecture are worked as hidden layers, and the third fully connected layer is utilized for the categorization of one thousand image classes using the ImageNet dataset. While the architecture of VGG19 involves sixteen convolution layers and three fully connected layers.

2. ResNet152V2: He et al. presented ResNet in 2015 [17] in which stacked residual blocks were utilized

rather than a normal network for solving the issue of vanishing gradient. Contrary to following the leading path in the normal network, which includes nonlinear activations and linear operators, the blocks follow the cutoff path, which instantly inserts the input activation to the output of the final layer without pulling out of the leading path. ResNet version 1 inserts the output of the final last layer in the residual network prior to the ReLU (nonlinear activation) and later the linear operation. While ResNet version 2 [18] adopts the batch normalization and ReLU as pre-activation prior to the weight layers. The second version of ResNet has provided considerable advancement in performance using different datasets. Therefore, in this paper, the model of ResNet152V2 is utilized which includes 152 layers for evaluating performance in the categorization of objects/images.

3. DenseNet201:

   In 2016, Huang et al. [19]to guarantee the utmost information flow between the layers within the network presented Dense Convolutional Network (DenseNet). This model represents an improvement of ResNet in which the layers are connected in the form of feed-forward with each other, and the obtained feature maps concerning a layer are concatenated with the succeeding feature maps.

4. InceptionV2

   The inception models were advanced to solve the issues of overfitting and the complexity of computation resulting from the restricted training dataset and rapid vanishing of gradient change. Inception version 1 [20]attempted to handle these issues via enlarging the model width including convolutional filters of multiple-sized to obtain global and local information from an input image, and including two additional classifiers to inhibit the disappearing gradient. Compared with the first version, Inception version 2 [21] involves several modifications such as; factorizing big size convolutions into smaller and asymmetric convolutions, expanding the model via increasing the filter banks, and minimizing the size of the grid (utilizing parallel pooling, convolution, and concatenation). Inception version 3 [21] added some modifications to the second version such as; factorizing (7×7) convolutions, batch normalizing of the network side layer including additional classifiers, utilizing Root mean square propagation optimizer, and presenting label smoothing regulation to inhibit the issue of overfitting. The third version of Inception has popularity since it is the principal runner-up in 2015 holding the lowest rate of error in object/image categorization using the ImageNet dataset. Therefore, in this paper, the model of InceptionV3 is utilized which includes 42 layers.

5. MobileNetV2

   Sandler et al. [22] described MobileNet version 2, which works on improving the mobile models' performance in multi-task and across a range of various model sizes. The outstanding idea concerning the models of MobileNet is to exchange the expensive layers of convolution with separable depthwise convolutional blocks, and every block involves (1×1) expansion layer, and depthwise (3×3) convolutional layer accompanied by a pointwise or projection (1×1) convolutional layer. In MobileNet V2, every layer holds batch normalization and ReLU (except the pointwise layer). The complete architecture of MobileNet V2 involves 17 bottleneck residual blocks accompanied by a normal (1×1) convolutional, a global-average pooling, and classification layers.

IV. Proposed Models

The proposed developed CNN models were constructed and trained many times to choose the suitable parameters and provide higher models performance. These models involve two essential stages (features extraction and categorization).The first proposed model is implemented using the Caltech-101 dataset. In this model, the first stage involves five blocks that are in charge of extracting the deep features, each involving one convolutional layer (with ReLU) accompanied by a maximum-pooling layer, and only the first block is accompanied by batch normalization (to reduce the network initialization's sensitivity). The maximum-pooling layer minimizes the redundant representations yielded from prior blocks and accordingly controls overfitting. Additionally, dropouts to increase the simplification of the model accompany these blocks. While the second stage involves a flattened layer (for adjusting feature maps), five dense layers (with ReLUs), and dropouts (added after the second and fourth dense layers, and after the fifth dense layer). Moreover, the final dense layer (with Softmax) is added to predict 102-class. These stages are illustrated Fig. 2 and Table I. The second developed CNN is implemented using the Caltech-256 dataset, and it involves the same architecture as the first model with some exceptions which are; in the first stage, each block involves two convolutional layers rather than one, and the second stage involves four dense layers with only one dropout in the middle, in addition to the final dense layer (with Softmax) to predict 257-class. These stages are illustrated in Table II and Fig. 3.

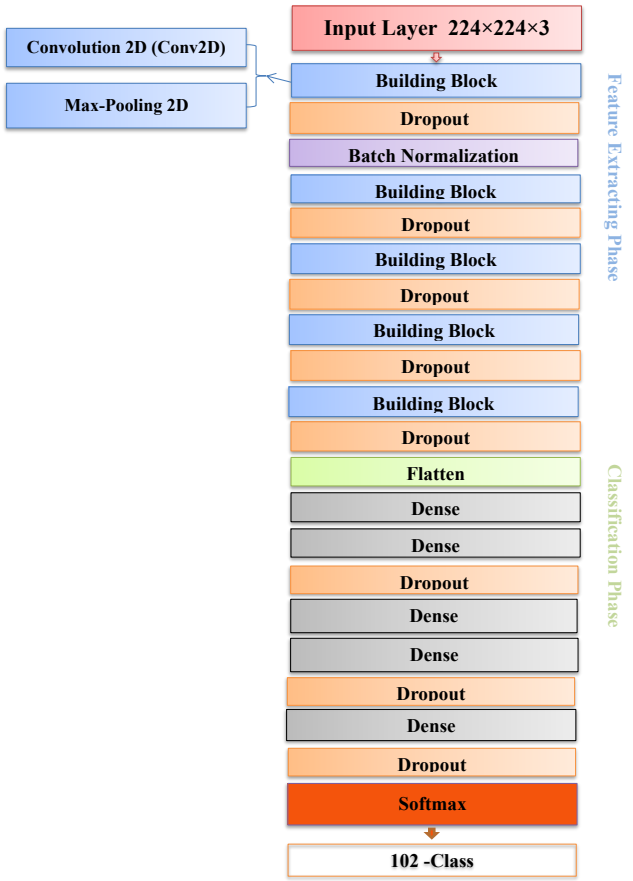

Fig. 2. First developed CNN architecture.

TABLE I.
THE LAYERS' STRUCTURE AND THE HYPERPARAMETERS OF THE FIRST DEVELOPED CNN MODEL

Layers (Types)	Shapes of output	Parameters
Image_Input (Input-Layer)	None 224 224 3	0
layer_1 (Convolutional-2D)	None 224 224 32	896
layer_2 (Max.-pooling-2D)	None 112 112 32	0
Dropout	None 112 112 32	0
Batch_Normalization	None 112 112 32	128
layer_3 (Convolutional-2D)	None 112 112 64	18496
layer_4 (Max.-pooling-2D)	None 56 56 64	0
Dropout	None 56 56 64	0
layer_5 (Convolutional-2D)	None 56 56 128	73856
layer_6 (Max.-pooling-2D)	None 28 28 128	0
Dropout	None 28 28 128	0
layer_7 (Convolutional-2D)	None 28 28 256	295168
layer_8 (Max.-pooling-2D)	None 14 14 256	0
Dropout	None 14 14 256	0
layer_9 (Convolutional-2D)	None 14 14 512	1180160
layer_10 (Max.-pooling-2D)	None 7 7 512	0
Dropout	None 7 7 512	0
Fully-connected (Flatten)	None 25088	0
layer_11 (Dense)	None 1024	25691136
layer_12 (Dense)	None 512	524800
Dropout	None 512	0
layer_13 (Dense)	None 256	131328
layer_14 (Dense)	None 128	32896
Dropout	None 128	0
layer_15 (Dense)	None 64	8256
Dropout	None 64	0
predictions (Dense)	None 102	130

V. EXPERIMENTS AND RESULTS

In this section, the categorization results of several models (VGG19, ResNet152V2, DenseNet201, InceptionV3, MobileNetV2, and developed CNNs) are compared using the datasets Caltech-101 [23] and Catech-256 [24]. The Caltech-101dataset involves (9144) images of 102 categories and every object category involves between (40 to 800) images. The Caltech-256 dataset involves (30607) images of 257 categories and every category involves between (80 to 827) images. Some selected samples of the utilized datasets are illustrated in Fig. 4. In both benchmarking datasets, we resized the images to (224×224×3) to fit as input for the networks and then separated them into two sets (80 training and 20 testing). Concerning an object/image categorization issue, there are several potential consequences; False positive (FPO), True positive (TPO), False negative (FNE), and True negative (TNE). Depending on these consequences, several evaluation metrics (Precision (P), Recall (R), F1-score (F1), and Accuracy (Acc)) are computed as shown in (1- 4)

$$P = \frac{TPO}{FPO + TPO} \tag{1}$$

$$R = \frac{TPO}{FNE + TPO} \tag{2}$$

$$F1 = \frac{RecallPrecision}{Recall + Precision} \tag{3}$$

$$Acc = \frac{TNE + TPO}{FNE + TNE + FPO + TPO} \tag{4}$$

The performance comparison between the pre-trained models and the proposed models using the evaluation metrics is

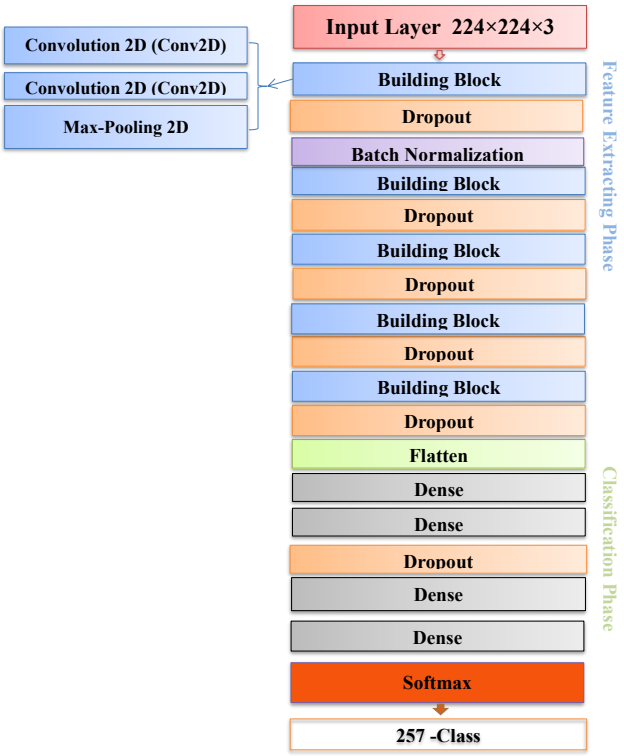

Fig. 3. Second developed CNN architecture.

illustrated in Table III. The number of the total, trainable, and non-trainable parameters for the utilized pre-trained models and the proposed model using Caltech-101 and Caltech-256 datasets are illustrated in Table IV. Fig. 5and Fig.6 illustrate the graphs of accuracy versus validation accuracy and loss versus validation loss for the utilized and proposed models using Caltech-101 and Caltech-256 datasets, respectively. Table V illustrated a comparison of the utilized feature extraction and categorization methods and the categorization accuracies between the adopted CNN models and the models in related works. Among these models, the obtained results of the proposed models reached higher values.

VI. Conclusion

Over the past few years, various pre-trained networks such as VGG, ResNet, DenseNet, InceptionV3, and lightweight MobileNet networks have appeared. The issue of reducing the network parameters and improving the effect of categorization remains one of the most popular research destinations. Meantime, various methods of deep learning consolidated with conventional ones have accomplished considerable outcomes in object/image categorization tasks. However, constructing a particular deep learning network depend on the categorization

TABLE II.
The Layers' Structure and The Hyperparameters of The Second Developed CNN Model

Layers (Types)	Shapes of output	Parameters
Image_Input (Input-Layer)	None 224 224 3	0
layer_1 (Convolutional-2D)	None 224 224 32	896
layer_2 (Convolutional-2D)	None 224 224 32	9248
layer_3 (Max.-pooling-2D)	None 112 112 32	0
Dropout	None 112 112 32	0
Batch_Normalization	None 112 112 32	128
layer_4 (Convolutional-2D)	None 112 112 64	18496
layer_5 (Convolutional-2D)	None 112 112 64	36928
layer_6 (Max.-pooling-2D)	None 56 56 64	0
Dropout	None 56 56 64	0
layer_7 (Convolutional-2D)	None 56 56 128	73856
layer_8 (Convolutional-2D)	None 56 56 128	147584
layer_9 (Max.-pooling-2D)	None 28 28 128	0
Dropout	None 28 28 128	0
layer_10 (Convolutional-2D)	None 28 28 256	295168
layer_11 (Convolutional-2D)	None 28 28 256	590080
layer_12 (Max.-pooling-2D)	None 14 14 256	0
Dropout	None 14 14 256	0
layer_13 (Convolutional-2D)	None 14 14 512	1180160
layer_14 (Convolutional-2D)	None 14 14 512	2359808
layer_15 (Max.-pooling-2D)	None 7 7 512	0
Dropout	None 7 7 512	0
Fully-connected (Flatten)	None 25088	0
layer_16 (Dense)	None 1024	25691136
layer_17 (Dense)	None 512	524800
Dropout	None 512	0
layer_18 (Dense)	None 256	131328
layer_19 (Dense)	None 128	32896
predictions (Dense)	None 257	33153

characteristics represents an extremely efficient categorization scheme. Therefore, developed CNN models have been proposed and a large number of deep learning models have been presented. Extensive and comparative experiments were conducted and the obtained results demonstrate that the proposed first and second models have better categorization accuracies on Caltech-101 and Caltech-256 datasets, respectively. In the forthcoming works, other object/image categorization datasets will be utilized and tested using the developed CNN model. Other efficient pre-trained CNN models such as NASNets and EffecientNets will be utilized as well. Furthermore, better techniques of image enhancement can be utilized for increasing the training images and hence improving the categorization accuracy of the testing images.

TABLE III.
THE EVALUATION OF CNN MODELS' PERFORMANCE (CATEGORIZATION REPORT)

CNN Models	Acc		Metrics	Caltech-101 Dataset			Caltech-256 Dataset		
	Support: 1863	Support: 6285		P	R	F1	P	R	F1
VGG19	0.84	0.51	Macro Avg	0.83	0.78	0.77	0.65	0.48	0.49
			Weight Avg	0.87	0.84	0.84	0.67	0.51	0.53
ResNet152V2	0.92	0.76	Macro Avg	0.91	0.89	0.89	0.82	0.74	0.75
			Weight Avg	0.93	0.92	0.92	0.84	0.76	0.76
DenseNet201	0.92	0.75	Macro Avg	0.91	0.89	0.89	0.82	0.73	0.74
			Weight Avg	0.94	0.92	0.93	0.83	0.75	0.76
InceptionV3	0.91	0.73	Macro Avg	0.93	0.91	0.91	0.73	0.71	0.74
			Weight Avg	0.93	0.91	0.91	0.74	0.73	0.74
MobileNetV2	0.91	0.73	Macro Avg	0.92	0.91	0.90	0.73	0.70	0.71
			Weight Avg	0.93	0.91	0.91	0.74	0.72	0.73
Proposed	0.98	0.98	Macro Avg	0.98	0.98	0.98	0.99	0.98	0.98
			Weight Avg	0.98	0.98	0.98	0.99	0.98	0.98

TABLE IV.
THE NUMBER OF PARAMETERS FOR THE UTILIZED PRE-TRAINED AND PROPOSED MODELS

Models	Caltech-101 Dataset			Caltech-256 Dataset		
	Total parameters	Trainable parameters	Non-trainable parameters	Total parameters	Trainable parameters	Non-trainable parameters
VGG19	22,583,462	2,559,078	20,024,384	26,472,257	6,447,873	20,024,384
ResNet152V2	68,567,654	10,236,006	58,331,648	84,122,369	25,790,721	58,331,648
DenseNet201	27,918,246	9,596,262	18,321,984	42,500,801	24,178,817	18,321,984
InceptionV3	27,025,286	5,222,502	21,802,784	34,961,441	13,158,657	21,802,784
MobileNetV2	8,655,256	6,397,542	2,257,984	31,125,665	31,125,601	2,257,984
Proposed Models	27,957,250	27,957,186	64	31,125,665	31,125,601	64

TABLE V.
A COMPARISON BETWEEN THE ADOPTED CNN MODELS AND THE MODELS IN RELATED WORKS

Author(s), 2025, Reference	Methods of Feature Extraction	Methods of Categorization	Accuracy on Caltech-101	Accuracy on Caltech-256
Wang et al., 2020, [8]	Expanded Convolution MobileNets	Softmax Classifier	78.73%	65.16%
Ravi, 2020, [9]	ResNet50	Fuzzy Rule-based Classifier	86%	-
Bansal et al., 2021, [10]	VGG19, ORB, SIFT, SURF, and SCD	Random Forest Classifier	93.73%	-
Rao and Mahanthesh, 2021, [11]	Modified VGG16	Softmax Classifier	78.42%	57.57%
Zeynalli, 2021, [12]	VGG16	Logistic Regression	94.65%	-
VGG19	VGG19	Softmax Classifier	90.62%	45.44%
ResNet152V2	ResNet152V2	Softmax Classifier	95.76%	74.35%
DenseNet201	DenseNet201	Softmax Classifier	93.97%	76.30%
InceptionV3	InceptionV3	Softmax Classifier	94.64%	70.18%
MobileNetV2	MobileNetV2	Softmax Classifier	97.54%	90.37%
Proposed Models	Developed CNN	Softmax Classifier	98.06%	94.00%

(a)

(b)

Fig. 4. Image samples of (a) Caltech-101, and (b) Catech-256.

Conflict of Interest

The authors have no conflict of relevant interest to this article.

References

[1] C. Kamusoko, *Image Classification*. Springer Geography. Springer, Singapore, 2019.

[2] Q. Fan, Y. Bi, B. Xue, and M. Zhang, "Genetic programming for feature extraction and construction in image classification," *Applied Soft Computing*, vol. 118, p. 108509, 2022.

[3] Òscar Lorente, I. Riera, and A. Rana, "Image classification with classic and deep learning techniques," *Computer Science, Computer Vision and Pattern Recognition*, 2021.

[4] J. e Liu and F.-P. An, "Image classification algorithm based on deep learning-kernel function," *Scientific Programming*, vol. 2020, pp. Article ID 7607612, 14 pages, 2020.

[5] M. Xin and Y. Wang, "Research on image classification model based on deep convolution neural network," *EURASIP Journal on Image and Video Processing*, vol. 40, 2019.

[6] H. Q. Flayyih, J. Waleed, and S. Albawi, "A systematic mapping study on brain tumors recognition based

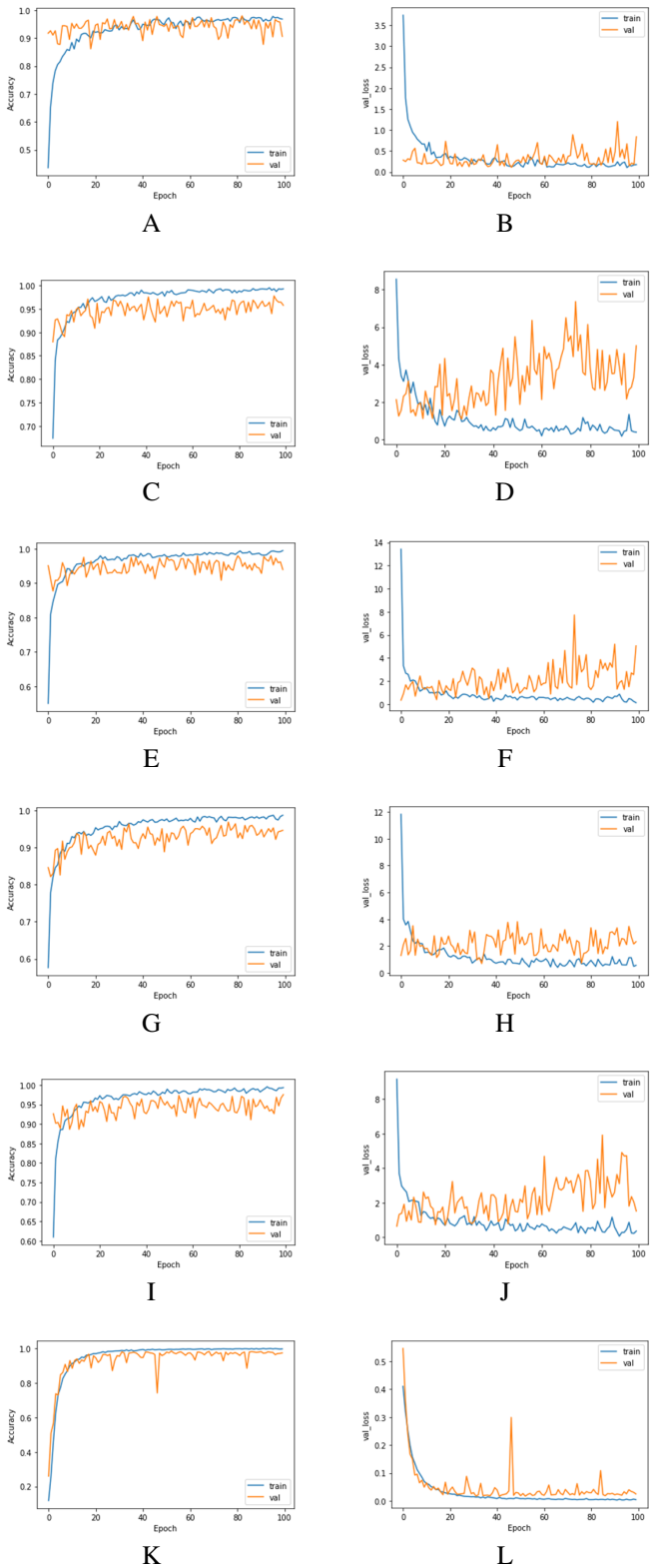

Fig. 5. The first column (a, c, e, g, i and k) accuracy and validation accuracy. The second column (b, d, f, h, j and l) loss and validation loss per epoch using Caltech-101, for VGG19, ResNet152V2, DenseNet201, InceptionV3, MobileNetV2, and the first proposed model respectively.

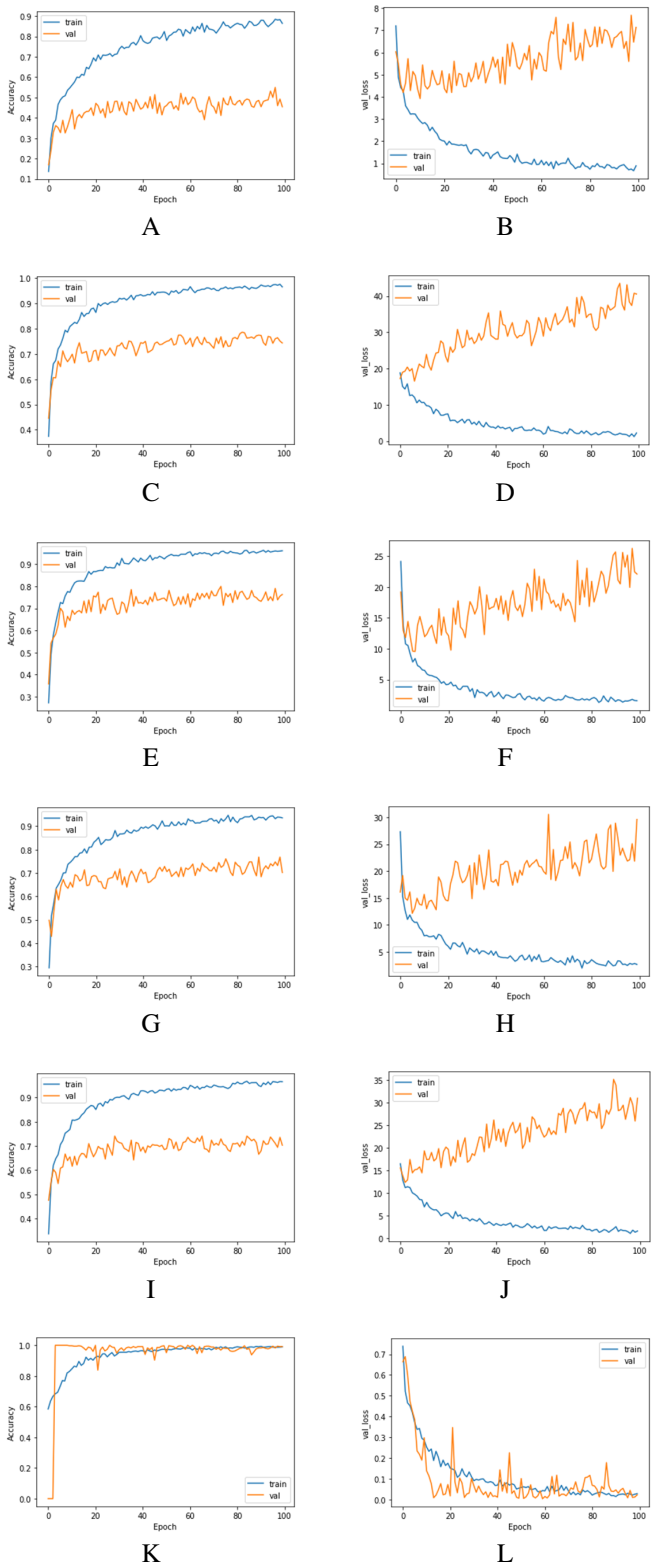

Fig. 6. The first column (a, c, e, g, i and k) accuracy and validation accuracy. The second column (b, d, f, h, j and l) loss and validation loss per epoch using Caltech-256, for VGG19, ResNet152V2, DenseNet201, InceptionV3, MobileNetV2, and the second proposed model respectively.

on machine learning algorithms," in *2020 3rd International Conference on Engineering Technology and its Applications (IICETA)*, pp. 191–197, 2020.

[7] J. Waleed, S. Albawi, H. Q. Flayyih, and A. Alkhayyat, "An effective and accurate cnn model for detecting tomato leaves diseases," in *2021 4th International Iraqi Conference on Engineering Technology and Their Applications (IICETA)*, pp. 33–37, 2021.

[8] W. Wang, Y. Hu, T. Zou, H. Liu, J. Wang, and X. Wang, "A new image classification approach via improved mobilenet models with local receptive field expansion in shallow layers," *Computational Intelligence and Neuroscience*, vol. 2020, pp. Article ID 8817849, 10 pages, 2020.

[9] C. Ravi, "Image classification using deep learning and fuzzy systems," *Journal of Healthcare Engineering*, vol. 2022, pp. Article ID 6216273, 13 pages, 2022.

[10] M. Bansal, M. Kumar, M. Sachdeva, and A. Mittal, "Transfer learning for image classification using vgg19: Caltech-101 image data set," *Journal of Ambient Intelligence and Humanized Computing*, 2021.

[11] A. S. Rao and K. Mahantesh, "Learning semantic features for classifying very large image datasets using convolution neural network," *SN Computer Science*, vol. 2, no. 187, 2021.

[12] E. Zeynallı, "Analysis the image classification problem based on transfer learning," in *14th International Conference on Theory and Application of Fuzzy Systems and Soft Computing – ICAFS-2020, Advances in Intelligent Systems and Computing*, vol. 1306, Springer, Cham, 2021.

[13] H. M. Shyni and E. Chitra, "A comparative study of x-ray and ct images in covid-19 detection using image processing and deep learning techniques," *Computer Methods and Programs in Biomedicine Update*, vol. 2, p. 100054, 2022.

[14] J. Waleed, T. Abbas, and T. M. Hasan, "Facemask wearing detection based on deep cnn to control covid-19 transmission," in *2022 Muthanna International Conference on Engineering Science and Technology (MICEST)*, pp. 158–161, 2022.

[15] V. Kumar, A. Zarrad, R. Gupta, and O. Cheikhrouhou, "Cov-dls: Prediction of covid-19 from x-rays using enhanced deep transfer learning techniques," *Journal of Healthcare Engineering*, vol. 2022, pp. Article ID 6216273, 13 pages, 2022.

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

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Computer Vision – ECCV 2016. Lecture Notes in Computer Science*, vol. 9908, Springer, Cham, 2016.

[19] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269, 2017.

[20] C. Szegedy and et al., "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, 2015.

[21] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, 2016.

[22] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, 2018.

[23] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories," *Computer Vision and Image Understanding*, vol. 106, no. 1, pp. 59–70, 2007.

[24] G. Griffin, A. Holub, and P. Perona, "The caltech-256 object category dataset," Tech. Rep. CNS-TR-2007-001, Caltech, Pasadena, Calif, USA, 2007.