IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 1, January 2025

DOI: 10.17148/IJARCCE.2025.14147

"Optimizing Image Classification with VGG-16: A CNN-Based Approach"

Rekha K R[1], Ravikumar A V[2], N Thanusri[3], Impana K P[4], Anusha M[5]

Professor, Department of Electronics and Communication, SJBIT, Bangalore, India[1]

Associate Professor, Department of Electronics and Communication, SJBIT, Bangalore, India[2]

Student, Department of Electronics and Communication, SJBIT, Bangalore, India[3]

Student, Department of Electronics and Communication, SJBIT, Bangalore, India[4]

Student, Department of Electronics and Communication, SJBIT, Bangalore, India[5]

Abstract: The proposed project presents the VGG16 deep learning model, a 16-layer convolutional neural network renowned for its simplicity and effectiveness, by leveraging its pre-trained foundation on the ImageNet dataset. By fine-tuning VGG16's layers, it adapts to various image processing tasks such as image classification, object detection, and image enhancement. Through rigorous experiments on benchmark datasets, the model's ability to generalize across different datasets is tested, demonstrating high accuracy in classifying images and performing well in tasks like object detection and segmentation. The project explores VGG16's capability to generate meaningful image representations, crucial for applications like image retrieval and content-based filtering, thereby showcasing its significant improvement in modern image analysis challenges.

Keywords: Caltech 101 dataset, Convolutional Neural Networks, Deep Learning, Visual Geometry Group

I. INTRODUCTION

Image classification is a fundamental task in computer vision, where images are categorized into predefined classes based on their visual features. The process begins with collecting a substantial dataset of images, which are then labeled, resized, normalized, and sometimes augmented to enhance quality and ensure consistency. Key features like shapes, textures, and colors are extracted using techniques like Convolutional Neural Networks (CNNs), which can automatically learn and hierarchically organize these features. The extracted features are fed into a machine learning model that is trained to recognize distinguishing patterns by adjusting its weights based on the training data. Once trained, the model can classify new images by comparing their features to the learned patterns.

VGG16, a type of CNN developed by the Visual Geometry Group at the University of Oxford, is designed to help computers understand and recognize images. It consists of 13 convolutional layers that detect features in the images, followed by pooling layers that reduce the size of the image representations, and three fully connected layers that classify the images. The output layer then provides the probability of the image belonging to a particular class. Image classification has numerous applications, including facial recognition, medical diagnostics, autonomous vehicles, and surveillance systems. Despite challenges like the need for high-quality data and significant computational resources, ongoing advancements in technology continue to enhance the accuracy and efficiency of image classification models.

II. OBJECTIVES

A. Improve Classification Accuracy and Training Time:

Enhance accuracy for precise results and reduce training time for quicker iterations, allowing for efficient experimentation with different datasets and parameters.[1]

B. Enhance Model Robustness and Resource Efficiency:

Increase robustness to handle diverse data effectively and develop methods to reduce computational and memory requirements, making the model deployable on a wider range of devices.[2]

C. Improve Interpretability and Real-Time Classification: Making the system time efficient:

Enhance model interpretability for trust and transparency, optimize for real-time tasks, and minimize overfitting to ensure the model generalizes well to new data across various conditions.[3]

III. PROPOSED METHODOLOGY

In this method, three distinct models and their related behaviors and impacts on performance are analyzed, focusing on CNN-based deep learning techniques capable of learning distinguishing features from large image category datasets. The empirical study investigates the concepts and effectiveness of various CNN techniques by examining the Basic CNN, the Regularized CNN, and the VGG-16 model utilizing transfer learning.

1. Dataset Preparation:
   - Collect diverse image dataset with labeled categories.
   - Preprocess images (resize, normalize) and split into training, validation, and testing sets.
   - Apply data augmentation (rotation, flipping, zooming) to increase dataset diversity.

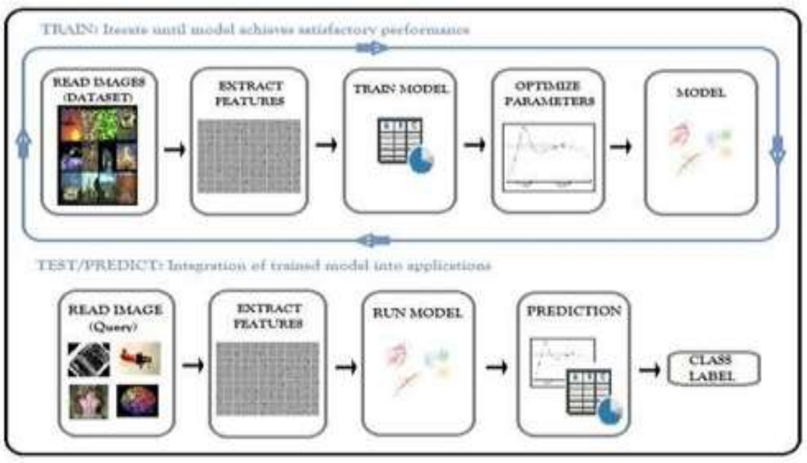
2. Basic CNN Model:
   - Construct Basic CNN with standard layers: convolutional layers, activation functions, pooling layers, and fully connected layers.
   - Train Basic CNN on training dataset using backpropagation and an optimizer (SGD, Adam).
   - Evaluate Basic CNN on validation set and adjust hyperparameters.

3. Regularized CNN Model:
   - Enhance Basic CNN architecture with dropout and batch normalization.
   - Implement dropout to randomly ignore neurons during training.
   - Normalize inputs to each layer using batch normalization.
   - Train Regularized CNN and evaluate its performance on validation set.

4. VGG-16 Model with Transfer Learning:
   - Utilize pre-trained VGG-16 model, trained on large datasets like ImageNet.
   - Fine-tune VGG-16 for specific image classification task by replacing top layers with task-specific layers.
   - Train modified VGG-16 model on training dataset, leveraging pre-learned features.
   - Evaluate VGG-16 model's performance on validation set.

5. Performance Comparison:
   - Compare models using metrics like accuracy, precision, recall, and F1-score.
   - Visualize performance results with confusion matrices and graphs.
   - Analyze strengths and weaknesses of each model based on empirical results.

IV. BLOCK DIAGRAM



The training process commences with reading a dataset of images, which forms the foundation for building the model. These images are pre-processed to ensure consistency in size and format, making it easier for the model to learn. The pre-processing steps typically involve resizing the images, normalizing pixel values, and sometimes augmenting the dataset by applying various transformations like rotations, flips, and zooms to create a more diverse and robust training

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 1, January 2025

DOI: 10.17148/IJARCCE.2025.14147

set. Once the dataset is prepared, the next step is feature extraction. This involves using convolutional layers in a Convolutional Neural Network (CNN) to detect and capture key features from the images, such as edges, textures, and shapes.

The extracted features are then fed into the model for training. During the training phase, the model learns to associate these features with specific image categories by adjusting its internal parameters, known as weights and biases, through a process called backpropagation. This optimization aims to minimize the error between the predicted labels and the actual labels of the training images. To further refine the model, various optimization techniques like stochastic gradient descent (SGD) or Adam optimizer are employed. These techniques help in efficiently navigating the error landscape to find the optimal set of parameters that result in the best performance.

After the model is trained, it is evaluated using a validation set to assess its performance and ensure it generalizes well to new, unseen data. This evaluation involves computing various metrics like accuracy, precision, recall, and F1-score, which provide insights into the model's ability to correctly classify images. If the performance is satisfactory, the model can then be deployed for real-world use. When a new image (query) is introduced, the same feature extraction process is applied to ensure consistency. The trained model processes these extracted features and makes a prediction about the image category based on the patterns it has learned during training.

In the prediction phase, the model outputs a class label that represents the most likely category of the input image. This entire workflow, from reading the dataset to feature extraction, model training, optimization, and prediction, is crucial for developing a robust image classification system. Each step plays a vital role in ensuring the model's accuracy and reliability, making it capable of handling a wide range of image classification tasks effectively. The block diagram, therefore, encapsulates this systematic approach, highlighting the interdependencies and the flow of information through the different stages of the model development and deployment process.

V. SOFTWARE USED

i. Google Collab

Google Colab, or Google Colaboratory, is a cloud-based platform that enables users to write and execute Python code in an interactive environment similar to Jupyter notebooks. It is designed for data science and machine learning tasks, offering free access to powerful computational resources like GPUs and TPUs. This makes it particularly appealing for researchers, educators, and developers who need to perform complex computations without investing in expensive hardware. Colab's seamless integration with Google Drive allows users to easily save, share, and collaborate on their notebooks, facilitating teamwork and enhancing productivity. With features like interactive visualizations, TensorBoard integration, and support for various Python libraries, Colab provides a versatile tool for a wide range of applications.

Getting started with Google Colab is straightforward; you can create a new notebook directly from the [Google Colab website](https://colab.research.google.com/) and begin coding instantly. Colab supports code execution, text annotations, and rich media content within the same document, making it an excellent choice for educational purposes and collaborative projects. By leveraging its hardware accelerators, users can train deep learning models more efficiently, experiment with data analysis, and prototype machine learning algorithms. The platform's accessibility and ease of use make it an invaluable resource for anyone interested in harnessing the power of cloud computing for their data-driven projects.

ii. Caltech 101 dataset

The Caltech 101 dataset, introduced by Fei-Fei Li, Rob Fergus, and Pietro Perona in 2004, has played a significant role in the field of computer vision and machine learning. It consists of 9,146 images across 101 distinct object categories and a background clutter category, totaling 102 classes. The dataset is designed to be both comprehensive and challenging, with a variety of objects in different poses, scales, and lighting conditions. Each category typically contains between 40 to 800 images, with most categories having around 50 images. The images are generally around 200-300 pixels along their longest dimension. The primary purpose of the Caltech 101 dataset is to facilitate the development and evaluation of image classification algorithms, serving as a benchmark for comparing the performance of different models. Its diversity, encompassing a wide range of objects from animals and flowers to everyday items and vehicles, helps in testing the robustness and generalization capabilities of image recognition systems.

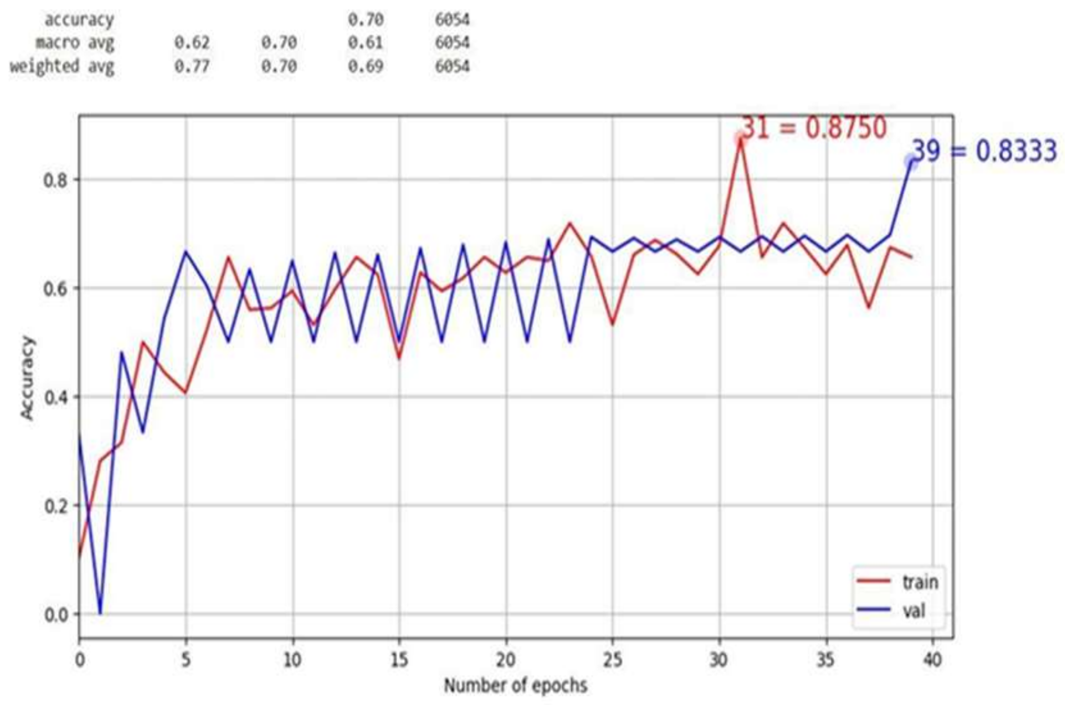
A notable advantage of the Caltech 101 dataset is its straightforward image annotation; each image is labeled with only one object, making it easy to use for supervised learning tasks. Researchers have extensively used this dataset to develop

DOI: 10.17148/IJARCCE.2025.14147

and refine various approaches, including traditional machine learning methods and more recent deep learning techniques. The dataset has been particularly valuable in advancing convolutional neural networks (CNNs), which have shown remarkable success in visual recognition tasks. One of the key contributions of the Caltech 101 dataset is its role in pioneering one-shot learning techniques, which aim to develop models capable of recognizing new object categories from very few examples by leveraging prior knowledge from other categories. Despite its age, the Caltech 101 dataset remains relevant due to its balanced mix of complexity and manageability, providing a solid foundation for experiments requiring a moderate-sized dataset with diverse categories. The insights gained from studies using this dataset have laid the groundwork for more extensive and complex datasets like Caltech 256 and ImageNet, further pushing the boundaries of object recognition capabilities.

VI. RESULTS AND DISCUSSION

The results of using the pre-trained VGG16 model to classify images from the Caltech101 dataset show promising performance. By leveraging transfer learning, the VGG16 model, initially trained on ImageNet, was fine-tuned for this specific task. Preprocessing involved resizing images to 224x224 pixels and applying data augmentation techniques like rotation, zoom, and horizontal flipping. The model was configured with a softmax output layer and trained for 40 epochs with early stopping to prevent overfitting. The training and validation accuracy trends showed consistent improvement, achieving satisfactory accuracy on the validation set. Metrics such as overall accuracy, precision, recall, and F1-score were used for comprehensive evaluation.Despite the positive results, further improvements could be achieved by fine-tuning VGG16 layers or exploring advanced architectures like ResNet or InceptionV3. The model achieved an overall accuracy of 0.70, with macro average precision of 0.62, recall of 0.70, and F1-score of 0.61, while weighted averages were higher. Data augmentation and early stopping contributed to the model's robustness and generalization ability. The VGG16 model demonstrated the effectiveness of pre-trained models and highlighted the benefits of transfer learning and data augmentation in image classification tasks.



```
    accuracy                         0.70      6054
   macro avg       0.62      0.70      0.61      6054
weighted avg       0.77      0.70      0.69      6054
```

Fig Average Accuracy

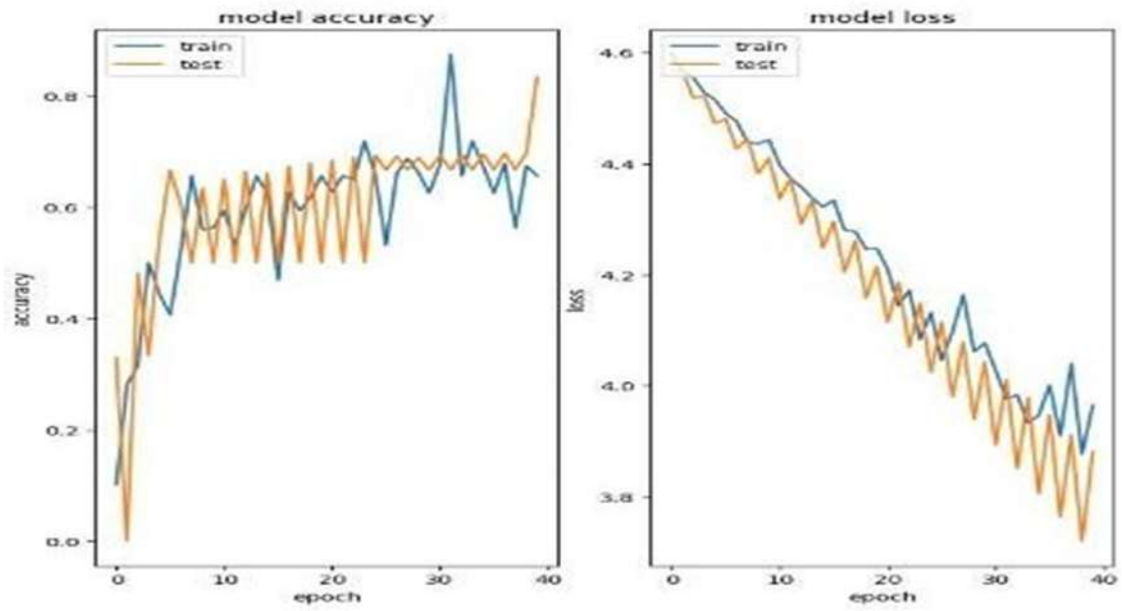
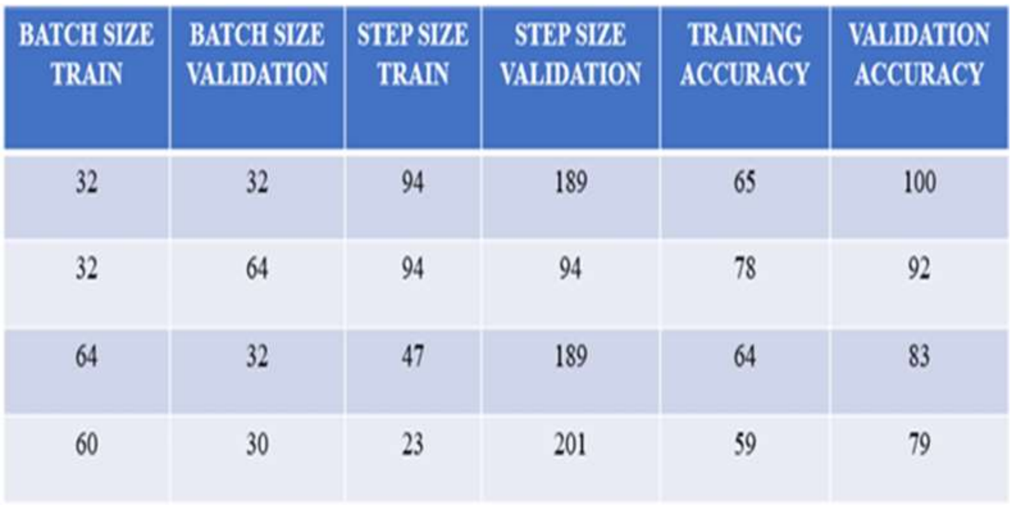


Fig Modal Accuracy

The graphs depict the training and validation performance of a deep learning model using accuracy and loss metrics over multiple epochs. The accuracy graph shows that both training an validation accuracy steadily improve as the model learns to recognize patterns in the data. Initially, accuracy is low but increases over time, indicating successful learning. However, slight fluctuations in validation accuracy suggest variability in the model's generalization, likely influenced by data augmentation techniques (e.g., rotations, flips, and zooms) that introduce diversity in the training data. The loss graph demonstrates a consistent decrease in both training and validation loss, confirming themodel's ability to minimize error during optimization. While the training loss decreases smoothly, the validation loss shows minor oscillations in the later epochs, hinting at slight overfitting, where the model starts to memorize training-specific features rather than generalizing effectively. This behavior can be addressed through regularization techniques, early stopping, or learning rate adjustments. Overall, the model demonstrates effective learning and convergence, with a reasonable balance between training and validation performance, achieving good accuracy while maintaining a relatively low loss.



BATCH SIZE TRAIN	BATCH SIZE VALIDATION	STEP SIZE TRAIN	STEP SIZE VALIDATION	TRAINING ACCURACY	VALIDATION ACCURACY
32	32	94	189	65	100
32	64	94	94	78	92
64	32	47	189	64	83
60	30	23	201	59	79

The overall performance of the machine learning model was evaluated using different configurations of batch sizes and step sizes during training and validation. Training batch sizes ranged from 32 to 64, and validation batch sizes varied from 30 to 64. Training step sizes ranged from 23 to 94, and validation step sizes ranged from 94 to 201. The model achieved a training accuracy between 59% and 78%, and a validation accuracy between 79% and 100%. Notably, the highest training accuracy of 78% and validation accuracy of 100% were observed when both batch sizes were set to 32, with training and validation step sizes of 94 and 189, respectively, indicating that specific batch and step size combinations significantly impact model performance.

ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 1, January 2025

DOI: 10.17148/IJARCCE.2025.14147

VII. CONCLUSION

The VGG16 model was configured with a softmax output layer for multi-class classification and used categorical cross-entropy as the loss function, making it suitable for classifying images into one of the 101 categories in the Caltech101 dataset. Training was conducted for 40 epochs with an early stopping mechanism to monitor validation performance and prevent overfitting, ensuring the model generalized well to unseen data. Throughout the training process, training and validation accuracy were visualized, helping to monitor the model's learning progress and adjust hyperparameters as necessary. The model showed satisfactory performance on the validation set, demonstrating its effectiveness in distinguishing between different categories.

Despite the positive outcomes, there is room for improvement. Fine-tuning the frozen layers of VGG16, initially keeping the bottom layers frozen and only fine-tuning the top layers, can improve accuracy by learning more specific features relevant to the Caltech101 dataset. Additionally, exploring more advanced architectures such as ResNet or InceptionV3 could offer better performance by capturing more complex patterns and details from images. These models incorporate deeper networks and different architectural innovations that might lead to higher classification accuracy.

Overall, using VGG16 for image classification on the Caltech101 dataset showcased the significant benefits of pre-trained models, reducing training time and achieving reliable results with a relatively smaller dataset. This project highlighted the versatility and power of deep learning techniques in practical applications, reaffirming VGG16's status as a robust tool for image classification. The successful deployment of VGG16 in this context paves the way for further advancements and applications in the field of computer vision.

REFERENCES

[1]. Christopher B Choy, JunYoung Gwak, Silvio Savarese, and Manmohan Chandraker. Universal correspondence network. In Proc. Neural Information Processing Systems (NeurIPS), pages 2414– 2422, 2016.
[2]. Rowley, H.A., Baluja, S. and Kanade, T., 1998. Neural network-based face detection. IEEE Transactions on pattern analysis and machine intelligence, 20(1), pp.23-38.
[3]. A Newman, M. Kilmer, L. Horesh, Image classification using local tensor singular value decompositions (IEEE, international workshop on computational advances in multi-sensor adaptive processing. IEEE,Willemstad, 2018), pp. 1–5.
[4]. X. Wang, C. Chen, Y. Cheng, et al, Zero-shot image classification based on deep feature extraction. United Kingdom: IEEE Transactions on Cognitive & Developmental Systems, 10(2), 1–1 (2018).Huang, G., Liu, Z., Van Der Maaten, L., Weinberger,K.Q. (2017).
[5]. Densely connected convolutional networks.in Proceedings of the IEEE Conference on Computer Vision Pattern Recognition, Honolulu, HI, USA, pp. 4700-470. https://doi.org/10.1109/CVPR.2017.243
[6]. Deep Learning Image Classification Using VGG-19 Model Chandra Bhushana Rao Killi, Narayanan Balakrishnan, Chinta Someswara Rao https://doi.org/10.18280/isi.280228
[7]. Transfer learning for image classification using VGG19: Caltech-101 image data set Monika Bansal1 · Munish Kumar2 · Monika Sachdeva3 · Ajay Mittal4 Received: 7 April 2021 / Accepted: 31 August 2021 / Published online: 17 September 2021
[8]. Mingyuan X, Wang Y (2019) Research on image classification model based on deep convolution neural network. EURASIP J Image Video Process 2019(1):1–11
[9]. Pandey A, Puri M, Varde A (2018) Object detection with neural models, deep learning and common sense to aid smart mobility. In: 2018 IEEE 30th international conference on tools with artificial intelligence (ICTAI), pp 859–863. https:// doi.org/ 10. 1109/ ICTAI. 2018.00134
[10]. Y. Qian, J. Dong, W. Wang, and T. Tan, "Learning and transferring representations for image steganalysis using convolutional neural network," in IEEE ICIP, 2016, pp. 2752–2756.
[11]. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in NIPS, 2012, pp. 10971105
[12]. Bosch, X. Munoz, and R. Mart , "Which is the best way to organize/classify images by content?" Image and Vision Computing, vol. 25, no. 6, pp. 778–791, 2007.
[13]. Hosny, K.M., Mortda, A.M., Fouda, M.M., Lashin, N.A. (2022). An efficient CNN model to detect image forgery. https://doi.org/10.1109/ACCESS.2022.3172273 IEEE Access, 10: 48622-48632.
[14]. Albawi, S., Mohammed, T.A., Al-Zawi, S. (2017). Understanding of a convolutional neural network. In 2017 International Conference on Engineering and Technology (ICET), Antalya, Turkey, pp. 1-6. https://doi.org/10.1109/ICEngTechnol.2017.8308186
[15]. H. Drucker, R. Schapire, and P. Simard, "Boosting Performance in Neural Networks," Int'l J. Pattern

Recognition and Artificial Intelli gence, vol. 7, no. 4, pp. 705-719, 1993.

[16]. A.A.M. Al-Saffar, H. Tao, M.A. Talab, Review of deep convolution neural network in image classification (International conference on radar, antenna, microwave, electronics, and telecommunications. IEEE, Jakarta, 2018), pp. 26–31.

[17]. Shima Y. Image augmentation for object image classification based on combination of pre- trained CNN and SVM. International Conference on Informatics, Electronics and Vision & 2017, International Symposium in Computational Medical and Health Technology. 2018:1–6.

[18]. Z. Yan, V. Jagadeesh, D. Decoste, et al., HD-CNN: hierarchical deep convolutional neural network for image classification. Eprint Arxiv 4321-4329 (2014).