JOURNAL OF APPLIED SCIENCES TECHNOLOGY AND COMPUTING
e-ISSN: 3036-0250

JASTEC

Vol. 1 No. 1 (2024) 14-28
https://publisher.uthm.edu.my/ojs/index.php/jastec

# A Review Study of the Visual Geometry Group Approaches for Image Classification

**Nurzarinah Zakaria[1], Yana Mazwin Mohmad Hassim[1]***

[1] *Faculty of Computer Science and Information Technology,*
*Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA*

*Corresponding Author: yana@uthm.edu.my
DOI: https://doi.org/10.30880/jastec.2024.01.01.003

**Article Info**

Received: 19 November 2023
Accepted: 13 February 2024
Available online: 27 February 2024

**Keywords**

Convolutional Neural Networks, computer vision, deep learning, image classification, Visual Geometry Group

**Abstract**

In the realm of advanced machine learning for image classification, Convolutional Neural Networks (CNNs) stand as a pivotal tool, with the Visual Geometry Group-16 (VGG16) model standing out for its emphasis on deepening and expanding CNNs architecture to achieve better accuracy. However, the complex design of VGG16 presents challenges regarding computational efficiency and scalability. This study addresses these issues by refining the VGG16 architecture through strategic modifications, including reducing convolution blocks, integrating batch normalization (BN) layers, and incorporating a global average pooling (GAP) layer alongside additional dense and dropout layers. The proposed architecture's effectiveness was assessed through comprehensive experiments across ten benchmark datasets, comparing its performance against the standard VGG16 architecture. The proposed architecture sped up the execution time by 63.7% on average across all benchmark datasets, compared to the standard VGG16. Furthermore, the results showed that the proposed architecture outperformed VGG16 by improving the classification accuracy by up to 30.1% based on the overall datasets. In summary, the proposed architecture is made to be compact and accurate. By adjusting parameters, it processes information quickly and accurately. It also includes features to prevent overfitting and improve classification, resulting in a significant advancement in image classification.

## 1. Introduction

Image classification is one of the applications of machine learning. It refers to the process of classifying an entire image with the aim of assigning predefined labels or categories to images based on information extracted from the images' content. Furthermore, it is essential to identify useful image information within an efficient timeframe since it will have a consequential impact on the image classification results [1]. The recent years have witnessed significant improvements in using advanced machine learning techniques, namely deep learning, for object detection and classification [2]. These advancements have greatly improved the accuracy of predictions [3]. As a result, many researchers have worked on deep learning to extract features from images [4-8].

In recent years, Visual Geometry Group (VGG) architecture has gained considerable attention due to its remarkable performance and simplicity [9], [10]. VGG architecture was designed to achieve high accuracy in image

classification tasks, with modest architecture [11]. This paper focused on the literature study to establish a proper and deeper understanding of various applications of VGG architecture for the image classification process.

Exploring the details of the VGG architecture is important as it holds the potential to enhance image classification accuracy. The highlight of this study is to understand how the VGG architecture works, challenges, and opportunities that can contribute to the improvement in image classification performance. Overall, this research contributes to the broader field of deep learning applications by comprehensively examining the role of the VGG architecture in image classification.

## 2. Visual Geometry Group (VGG)

The VGG network has several strengths that gain numerous interests in the field of image classification. It includes simplicity and uniform architecture, making it easy to understand and apply. Such simplicity emerges from the above-mentioned employment of a series of smaller-sized filters. A straightforward modification and experimentation are made possible using the uniform architecture [11]. Fig.1 shows the configuration of the VGG networks.

| ConvNet Configuration | | | | | |
|---|---|---|---|---|---|
| A | A-LRN | B | C | D | E |
| 11 weight layers | 11 weight layers | 13 weight layers | 16 weight layers | 16 weight layers | 19 weight layers |
| input (224 × 224 RGB image) | | | | | |
| conv3-64 | conv3-64<br>**LRN** | conv3-64<br>**conv3-64** | conv3-64<br>conv3-64 | conv3-64<br>conv3-64 | conv3-64<br>conv3-64 |
| maxpool | | | | | |
| conv3-128 | conv3-128 | conv3-128<br>**conv3-128** | conv3-128<br>conv3-128 | conv3-128<br>conv3-128 | conv3-128<br>conv3-128 |
| maxpool | | | | | |
| conv3-256<br>conv3-256 | conv3-256<br>conv3-256 | conv3-256<br>conv3-256 | conv3-256<br>conv3-256<br>**conv1-256** | conv3-256<br>conv3-256<br>**conv3-256** | conv3-256<br>conv3-256<br>conv3-256<br>**conv3-256** |
| maxpool | | | | | |
| conv3-512<br>conv3-512 | conv3-512<br>conv3-512 | conv3-512<br>conv3-512 | conv3-512<br>conv3-512<br>**conv1-512** | conv3-512<br>conv3-512<br>**conv3-512** | conv3-512<br>conv3-512<br>conv3-512<br>**conv3-512** |
| maxpool | | | | | |
| conv3-512<br>conv3-512 | conv3-512<br>conv3-512 | conv3-512<br>conv3-512 | conv3-512<br>conv3-512<br>**conv1-512** | conv3-512<br>conv3-512<br>**conv3-512** | conv3-512<br>conv3-512<br>conv3-512<br>**conv3-512** |
| maxpool | | | | | |
| FC-4096 | | | | | |
| FC-4096 | | | | | |
| FC-1000 | | | | | |
| soft-max | | | | | |

**Fig. 1** *Configuration of VGG networks [11]*

The VGG network has six different CNN configurations, which are VGG11, VGG11 (LRN), VGG13, VGG16 (Conv1), VGG16, and VGG19. The convolution layers in the model are represented by the numbers "11", "13", "16", and "19". The first layer of the convolution layer has 64 widths (number of channels), and after each max-pooling layer, the number of channels continues to rise by a factor of two until it reaches 512. Furthermore, VGG also includes 3 x 3 convolutional layers that are layered on top of each other to increase depth. Following that, volume size reduction is handled via max-pooling. Two fully-connected layers with 4096 nodes each are then followed by a softmax classifier. The 3 x 3 filters used in each convolution layer of this architecture were substantially smaller and were the first to be connected as a series of convolutions. This approach introduces the idea of grouping numerous convolution layers with lower kernel sizes rather than using a single convolution layer with a larger kernel size. For further research, the VGG has released two best-performing models, namely VGG16 and VGG19 [11]. When compared to previous CNNs models like AlexNet, the VGG networks are more in-depth. The depth of the network is widened by stacking multiple convolutional layers, allowing it to recognize and learn complex features from the input images [11]. Moreover, the performance and accuracy of the VGG networks on benchmark image classification datasets are outstanding [11]. The use of smaller filters and multiple convolutional layers enables more precise feature extraction, which enhances classification performance.

Among the various VGG models shown in Fig 1, VGG16 and VGG19 are the best performing multi-scale models [11]. Compared to VGG19, VGG16 has a slightly lower model complexity. The three extra convolution layers in

VGG19 increase its network size, causing higher computational and memory costs. Furthermore, VGG19 has more network parameters due to its higher complexity, which raises the possibility of overfitting [41]. Hence, this research emphasized the use of VGG16 for image classification tasks.

## 3. Visual Geometry Group - 16 (VGG16)

VGG introduced Visual Geometry Group - 16 (VGG16) as a CNNs model during ILSVRC 2014 [11]. The primary difference between the standard CNNs and the VGG16 lies in its architectural depth. While standard CNNs typically consist of a moderate number of convolutional layers, VGG16 adopts a considerably deeper architecture. The VGG16 specifically emphasizes the augmentation of depth as a means to improve image classification accuracy. The architecture obtains 92.7% of the top-5 test accuracy with a dataset of over 14 million images assigned into 1000 classes. VGG16 outperformed the AlexNet architecture by gradually switching to 3 x 3 kernel-size filters from a large 11 x 11, 5 x 5, and 3 x 3 kernel-size in the first, second, and third convolution processes, respectively. The usage of stacking a smaller kernel is recommended than larger kernels because having multiple nonlinear layers helps in greater network depth for ensuring more complex learning [12]. Fig. 2 presents the flattened architecture of VGG16 while brief explanations of each layer (convolution layer, pooling layer, and fully-connected layer) are available in the following subsections.

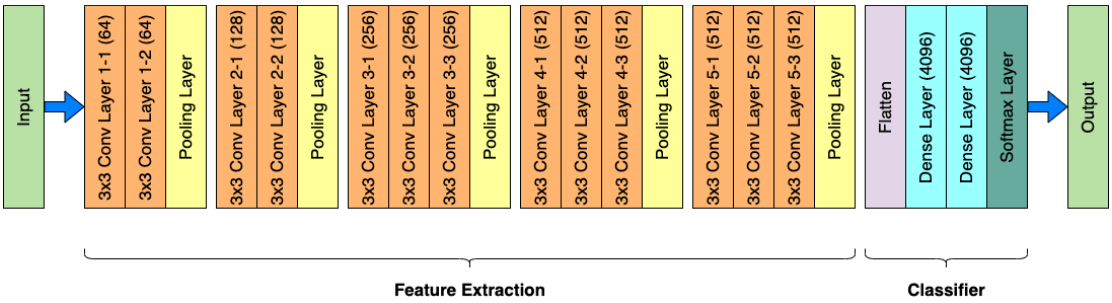


**Fig. 2** *VGG16 architecture*

### 3.1 Convolution Layer

The convolution layer is fundamental to CNNs to retrieve edges from images, which are referred to as high-level features [13]. The input image is transformed using a convolution layer to extract features from it, which is later convolved via a kernel (filter) in the transformation. Each filter uses the same bias and weight values throughout the image. The weight-sharing method allows the depiction of an entire image using the same feature [14]. Convolution has been widely used in image processing to blur and sharpen images as well as to conduct other processes such as embossing and enhancing edges. The convolution layer will operate on the input and transmit the outcome to the following layer. This layer's output commonly has a shorter length and width than the input layer but has a larger depth [15]. The training weight in this layer is kept as a parameter that can be calculated. The calculation has been modelled into a mathematical equation as shown in Equation (1), involving the form of width (m), height (n), the previous layer's filters (d), and account for all such filters (k) in the current layer. Additionally, the value of 1 is added to each filter to include the bias term.

$$\big((m * n * d) + 1\big) * k \tag{1}$$

### 3.2 Pooling Layer

Pooling layer emerges after the convolution layer, which is responsible to minimize the spatial size. This process is essential for lowering the computational cost by reducing the image dimension. It is also useful for extracting a dominant feature that is both rotationally and positional invariant [16]. If the input volume is [64* 64* 12], for example, the down sampled volume will be [32* 32*12]. As a result, it down samples the previous layer's feature maps produced from various filters in order to decrease over-fitting and network computations. The input volume in a pooling layer is given as [W1* H1* D1] where W is width, H is height, and D is depth. Two hyperparameters are written as [F,S] corresponding to the receptive field or size of the filter (F) and the stride (S), while the output volume is expressed as [W2* H2* D2] relating to the spatial dimensions of the input volume [17]. The calculation has been modelled as Equation (2), Equation (3), and Equation (4).

$$W2 = (W1 - F) \div S + 1 \tag{2}$$

$$H2 = (H1 - F) \div S + 1 \quad (3)$$

$$D2 = D1 \quad (4)$$

This layer has no learnable parameters because it only computes a specific number. As a result, the number of parameters is equal to zero.

### 3.3 Fully-Connected Layer

Fully-connected layer functions as a classifier in CNNs and contains neurons that are completely linked to neurons in the previous layer. For multi-class classification issues, this fully-connected layer is frequently retained as the final layer of CNNs that uses softmax as its activation function [17]. The measured value is delivered through the activation function before being transmitted to the next layer. This layer can determine the data class and learning non-linear combinations in high-level features produced by the convolution layer. In accordance with information from the previous layer, each node increases the connection weights and provides a bias value [14]. The calculation for the number of parameters in fully-connected layer has been modelled into the mathematical equations as shown in Equation (5), involving the multiplication of the number of neurons in the current layer (c) and the number of neurons in the previous layer (p). Similarly, the value of 1 is added to each filter to include the bias term.

$$(c * p) + 1 * c \quad (5)$$

In the context of this study, VGG16 is the main focuses on improving the VGG architecture for image classification. This study aims to enhance and optimize VGG16, specifically for faster execution time and improved the image classification performance. This involves analyzing the existing architecture, adjusting its design, and reducing parameters to produce the proposed architecture. The proposed architecture aims to improve the classification performance while significantly reducing the execution time. In addition, VGG16 acts as both the foundation and the benchmark for evaluating the proposed architecture on the speed and performance of image classification.

## 4. VGG Approaches on Image Classification

Table 1 presents a list of 35 articles in the field of image classification that focused on VGG16 approaches in CNNs. The review incorporated several aspects, including the key for addressing the articles, source of reference, year of publication, authors, architecture used, datasets used, and classification method.

**Table 1** *VGG16 approaches in CNNs*

| Key | Ref. | Year | Author | Architecture | Dataset | Classification Method |
|---|---|---|---|---|---|---|
| A1 | [14] | 2021 | Islam M. *et al*. | Improved VGG16 | Publicly available AEP dataset | CNNs |
| A2 | [18] | 2021 | Dheeb Albashish *et al*. | VGG16 with SVM classifiers | Public BreakHis benchmark dataset | CNNs |
| A3 | [19] | 2021 | Theetchen ya S. *et al*. | VGG16 Layered Architecture with SVM | Corel dataset | CNNs |
| A4 | [20] | 2021 | Dey R. *et al*. | VGG16 | Chars74kImg and Binary alphanumeric dataset | CNNs |
| A5 | [21] | 2021 | Baby D. *et al*. | VGG16 with KNN algorithm | Leukocytes dataset | CNNs |
| A6 | [22] | 2020 | Chen Y. *et al*. | Supplementary feature layer improved VGG16 | Thyroid nodule ultrasound images | CNNs |
| A7 | [23] | 2020 | Hsieh Y. *et al*. | Combining VGG16, Mask R-CNNs, and Inception V3 | 1600 mammography images | CNNs |
| A8 | [24] | 2020 | Panthakka n *et al*. | Binary classification method using the VGG16 | 2000 lung X-ray dataset | CNNs |

| Key | Ref. | Year | Author | Architecture | Dataset | Classification Method |
|---|---|---|---|---|---|---|
| A9 | [25] | 2020 | Toui Ogawa *et al*. | VGG16 with BN & GAP layer | Ultrasound microscopy dataset | CNNs |
| A10 | [26] | 2020 | Zeping Zhang *et al*. | A monitoring model based on VGG16 | 4 different basic facial expressions with more than 10000 images | CNNs |
| A11 | [27] | 2020 | Zhongqin Bi *et al*. | VGG16 with BN & GAP layer | German Traffic Sign Recognition Benchmark (GTSRB) Dataset | CNNs |
| A12 | [28] | 2020 | Wang Hao | VGG16 with BN layer | OpenCV computer vision library | CNNs |
| A13 | [29] | 2020 | Qian Yan *et al*. | VGG16 with GAP layer | 500 natural images of rice leaves condition | CNNs |
| A14 | [30] | 2020 | Pravitasari A. *et al*. | UNet-VGG16 | Real dataset from General Hospital (RSUD) | CNNs |
| A15 | [31] | 2020 | Setiawan W. & Damayanti | Modified CNNs 35 layer | Chest X-Ray Images from Kaggle | CNNs |
| A16 | [32] | 2020 | Dubey A. & Jain | VGG16 and concatenates additional layers | Extended Cohn-Kanade (CK+) and Japanese Female Facial Expression (JAFFE) benchmark datasets | CNNs |
| A17 | [33] | 2020 | Krishnaswamy Rangarajan & Purushothaman | The feature extractor VGG16 and the classifier MSVM | Real data collected using a smartphone camera | CNNs |
| A18 | [34] | 2020 | Lee *et al*. | Fine-tuning with the VGG16 model | Dental panoramic radiographs (DPRs) image dataset | CNNs |
| A19 | [15] | 2019 | Swasono D. *et al*. | VGG16 | Tobacco leaf pest dataset | CNNs |
| A20 | [13] | 2019 | Dewa Made Sri Arsa & Susila | VGG16 with a Random Forest | Public Batik dataset | CNNs |
| A21 | [35] | 2019 | Perdana A. & Prahara. | Light-CNNs | - ROSE-Youtu Face Liveness Detection Database<br>Taken manually with camera | CNNs |
| A22 | [36] | 2019 | Islam S. *et al*. | VGG16 with SVM and kernel method | The birds species from Google Images, Pinterest and Flickr | CNNs |
| A23 | [37] | 2019 | Lei Geng *et al*. | VGG16 with dilated convolution | Dice Similarity Coefficient (DSC) dataset | CNNs |
| A24 | [12] | 2019 | Jiang Z. | Modified VGG16 | 3,500 images in 12 categories of weed dataset from Kaggle | CNNs |
| A25 | [38] | 2019 | Oztel I. *et al*. | Alexnet and VGG16 | Facial emotions dataset | CNNs |
| A26 | [39] | 2019 | Guan Q. *et al*. | VGG16 and Inception-v3 | 279 cytological images of thyroid nodules | CNNs |
| A27 | [40] | 2019 | Hridayami *et al*. | VGG16 with a combined model | QUT Fish images | CNNs |

| Key | Ref. | Year | Author | Architecture | Dataset | Classification Method |
|---|---|---|---|---|---|---|
| | | | | of data enhancement | | |
| A28 | [41] | 2019 | Chen & Haoyu | VGG16 with SVM Classifier | Labeled Face in the Wild (LFW) and CelebFaces Attributes (CelebA) dataset | CNNs |
| A29 | [42] | 2019 | Montalbo & Hernandez | Modified VGG16 (retraining, fine-tuning, and optimization) | FishBase dataset under creative commons license | CNNs |
| A30 | [43] | 2019 | Song *et al*. | Faster R-CNN with VGG16 | Kiwifruit captured in the field throughout the day and night | CNNs |
| A31 | [44] | 2018 | Bin Liu *et al*. | VGG16-based fully convolutional structure | 3000 X-ray weld defect dataset | CNNs |
| A32 | [45] | 2018 | Hussam Qassim *et al*. | Residual Squeeze VGG16 | Large-scale MIT Places 365-Standard image dataset | CNNs |
| A33 | [46] | 2018 | Rezende E. *et al*. | VGG16 with SVM Classifier | Malware families dataset | CNNs |
| A34 | [47] | 2018 | Govindaiah A. *et al*. | VGG16 with BN layer | Age-Related Eye Disease Study (AREDS) images | CNNs |
| A35 | [48] | 2017 | Marcia Hon & Naimul Mefraz Khan | VGG16 with Inception V4 | MRI dataset | CNNs |

Based on Table 1, it can be shown that the CNNs method is widely used for image classification, as it has the unique ability to recognize critical features without human intervention. Its popularity is increasing due to its high accuracy in the image classification process. Therefore, it is safe to say that CNNs is becoming the preferred method for image classification.

In addition, Table 2 shows a comparison of articles on image classification that looked on the use of VGG16 in CNNs. The review comprises several aspects, including the key for addressing the articles, reference sources, research objectives or aims, methods used, and the findings of each research. The table is sorted in descending order by the year of publication. According to the table, most researchers attempted to overcome similar problems and attain the same objectives, namely:

- Addressing the difficulties associated with training the network in A1, A25.
- Reducing the number of parameters in A2, A11, A13, and A32.
- Improving the convergence rate in A13.
- Increasing the accuracy rate in A1, A4, A5, A6, A7, A8, A9, A10, A12, A15, A17, A19, A20, A22, A23, A26, A28, A29, A30, A31, A33, and A34.
- Minimizing the execution time in A2, A3, A5, A11, A14, A15, A16, A19, A31, and A32.
- Handling the challenges of limited dataset in A21, A24, A27, A31, and A35.

**Table 2** *Comparison of VGG16 approaches in CNNs*

| Key | Ref. | Study Aim | Methods | Findings |
|---|---|---|---|---|
| A1 | [14] | To construct an intelligent auditory sensation system using a pre-train model. | - In the fully-connected block, new layers were added after some convolution layers were deleted.<br>- Fine-tuned the higher levels of the network | - The difficulties associated with training the network were addressed.<br>- Accuracy = 96.87% |
| A2 | [18] | High-level features should be extracted from the dataset. | - The last fully-connected layers were removed.<br>- A group of heterogeneity classifiers was used to classify the acquired features. | - Computation cost and parameters were reduced.<br>- The proposed approaches outperformed recent standard machine learning methods. |
| A3 | [19] | To construct a system for retrieving images that can handle a large volume of images all at once. | - The first layer employed CNNs for feature extraction and training.<br>- The second layer implemented SVM for classification and image retrieval. | - The execution time was minimized.<br>- Efficiency = 83.5% |
| A4 | [20] | To recognize colored characters with various orientations. | - Characters were recognized by VGG16. | - Accuracy achieved was 78.04%. |
| A5 | [21] | To detect and classify leukocytes into four types. | - VGG16 extracted the features from the segmented nucleus of the leukocyte.<br>- The K-Nearest Neighbor (KNN) model was used to analyze the extracted features. | - The error rate was reduced.<br>- The speed of computation was increased.<br>- Accuracy = 82.35% |
| A6 | [22] | To increase the classification accuracy for small dataset. | - The proposed approach used supplemental features to provide additional useful information during training.<br>- Medical features chosen by the ReliefF algorithm comprised the supplemental features. | - Classification accuracy improved from 76.68% to 78.92%. |
| A7 | [23] | To differentiate between benign and malignant clusters of breast microcalcification (MC) | - From the image, breast MC clusters were identified using VGG16.<br>- To eliminate background noise, mask R-CNN was applied to identify MC from the clusters.<br>- Inception V3 was employed to distinguish between benign and malignant MC clusters. | - Accuracy of MCs labelling was 93%, accuracy of benign was 95%, and accuracy of malignant was 91%.<br>- The precision was 87%, specificity was 89%, and sensitivity was 90%. |
| A8 | [24] | To accurately predict COVID-19. | - The computational cost of the VGG16 model was greatly minimized.<br>- The images were scaled down to a suitable size. | - Accuracy achieved was 99.5%. |
| A9 | [25] | To recognize the normal and abnormal in power equipment ultrasound images. | - BN layer was added behind all convolution layers.<br>- GAP layer became the replacement for all connected layers. | - 98.29% of accuracy, 98.96% of TPR, and 7.43% of FPR. |
| A10 | [26] | To analyze the gathered facial data and lessen | - It utilized the OpenCV Cascade Classifier. | - Accuracy up to 79.75%. |

| Key | Ref. | Study Aim | Methods | Findings |
|---|---|---|---|---|
| | | the distortion of the images | - Before training the recognition model, the plane mapping technique flattened the panoramic image.<br>- A weight model was created through VGG16 training. | |
| A11 | [27] | To increase recognition accuracy while preserving effective performance in real time. | - Some redundant convolutional layers were removed and significantly less parameters were created.<br>- The network added the BN layer and GAP layer. | - Computation costs and parameters were reduced.<br>- Accuracy = 99.21% |
| A12 | [28] | To enhance the rate of recognition accuracy and model convergence. | - ReLU activation function was used.<br>- Speed up the model's convergence by adding the BN layer. | - The correct rate was 75.6%. |
| A13 | [29] | To reduce the parameters and improve the convergence speed. | - The fully-connected layer was replaced with the GAP layer.<br>- A BN layer was added.<br>- Avoided long training time using transfer learning. | - The number of parameters was reduced and the convergence rate was improved.<br>- Accuracy = 99.01% |
| A14 | [30] | The region of interest (ROI) and non-ROI were classified using fully convolutional network. | - The proposed method was Fully Convolutional Network (FCN) by implementing U-Net architecture.<br>- The U-Net was hybridized with VGG16. | - Computation cost was reduced and the execution time was minimized.<br>- 96.1% of accuracy. |
| A15 | [31] | To identify images of pneumonia and normal chest X-rays. | - The model consisted of 35 layers, including an input layer and 8 convolutional layers with a 3x3 filter in a variety of dimensions.<br>- There were eight BN layers, eight Rectified Linear Units, seven max-pooling with stride of 2, fully-connected layer, softmax and output layer. | - The execution time was minimized<br>- 95.1% of sensitivity, 98.5% of specificity, and 96.3% of accuracy. |
| A16 | [32] | To detect and analyze emotions from human facial movements. | - The top layers were cut off.<br>- For the classification, flatten, dense, drop, and dense-softmax layers were included. | - The speed of computation was increased.<br>- The accuracy was 94.84% and 93.75% on CK+ and JAFFE. |
| A17 | [33] | To detect critical eggplant diseases using images of leaves from isolated leaf samples. | - From the eighth convolution layer, features were extracted using VGG16.<br>- By using Multi-Class Support Vector Machines (MSVM), the features were employed to classify diseases. | - Classification accuracy achieved was 99.4%. |
| A18 | [34] | To determine the effectiveness of deep CNNs. | 4 variants of CNNs model were used:<br>- CNN3, VGG16, VGG16-TR and VGG16-TR-FT | - The performance was.<br>- Accuracy = 84% |
| A19 | [15] | To automatically identify leaves affected by various pest attacks. | - Kept the original weights of VGG16 and changed the weights of the added layers. | - The execution time was minimized. |

| Key | Ref. | Study Aim | Methods | Findings |
|---|---|---|---|---|
| | | | - The newly added layer served as a classifier. | - The proposed approach was capable of accurately classifying all data. |
| A20 | [13] | To classify Batik types. | - VGG16 was used as feature extraction.<br>- Imagenet was utilized to train the pre-trained network.<br>- Random Forest (RF) as classifier. | - The accuracy, F-score, recall, and precision exceed 97%. |
| A21 | [35] | To recognize face with a small dataset. | - The VGG16-based Light-CNN architecture worked well for small datasets.<br>- Some layers were removed to make the architecture light and compact. | - The challenges of limited dataset were addressed.<br>- Accuracy = 94.4% |
| A22 | [36] | Bangladeshi birds identification based on their species. | - The model was built using VGG16 and used to extract features.<br>- Different machine learning algorithms were implemented (Support Vector Machine (SVM) and Random Forest K-Nearest Neighbor (KNN)). | - Accuracy = 89% |
| A23 | [37] | To accurately segment the lung parenchyma. | - Convolution and input pooling were performed using the first three blocks of VGG16.<br>- The network had a sufficient receptive field because of several sets of dilated convolutions.<br>- Each pixel was calculated by MLP and multi-scale convolution features were combined. | - The lung parenchymal area was efficiently segmented using this approach. |
| A24 | [12] | To identify images of weeds in the field. | - The first 14 layers of VGG16 were fixed. | - The challenges of limited dataset were addressed.<br>- Accuracy = 91.08% |
| A25 | [38] | To evaluate the efficiency of transfer learning and training from scratch for the task of recognizing facial emotions. | - Both AlexNet and VGG16 were compared by training from scratch and using transfer learning. | - The difficulties associated with training the network were addressed.<br>- Achieved the highest average accuracy. |
| A26 | [39] | Implementing cytological imaging to identify papillary thyroid cancer (PTC). | - Inception-v3 and VGG16 underwent training and testing to produce differential diagnoses. | - For images that were fragmented, the accuracy was 97.66% and 92.75%, respectively.<br>- The accuracy on patients were 95% and 87.5%, respectively. |
| A27 | [40] | To determine the species of fish using VGG16. | - Four types of settings were used: RGB color space, canny filter, blending, and blending combined with RGB. | - The challenges of limited dataset were addressed.<br>- By combining images with RGB image datasets, the pre-trained VGG16 produced the best performance. |
| A28 | [41] | To provide an SVM-based VGG architecture | - VGG16 extracted the features of the face image. | - Reducing to 400 for the feature dimension results in the best |

| Key | Ref. | Study Aim | Methods | Findings |
|---|---|---|---|---|
| | | that extracts facial features from a face recognition method. | - The principal component analysis (PCA) approach was used to minimize the dimensionality of the extracted features.<br>- The SVM approach was used for face recognition. | accuracy when comparing from the range of 100 to 600 feature dimensions. |
| A29 | [42] | Using a modified VGG16 network to classify fish species. | - To achieve better accuracy, the VGG16 underwent retraining, fine-tuning, and optimization. | - The model reached an overall accuracy of 99%. |
| A30 | [43] | To create a harvesting machine using vision system. | - A faster R-CNN model developed with VGG16 was created using kiwifruit images taken at various times with or without flash. | - The performance was improved.<br>- The Faster R-CNN-based VGG16 model demonstrated excellent detection accuracy. |
| A31 | [44] | To identify the weld defect image. | - 1 x 1 kernels were used to minimize the dimension.<br>- Output mapping using Mean-pooling kernels.<br>- The fully convolutional structure comprised 1 x 1 kernels and mean-pooling. | - The execution time was minimized.<br>- The challenges of limited dataset were addressed.<br>- Accuracy = 97.6% |
| A32 | [45] | To handle the VGG16 issue of larger size and extremely long execution time. | - The proposed models compressed the VGG16 network.<br>- Contained 12 fire modules and 4 squeeze convolutional layers.<br>- Convolutional layer sequences linked to residual connections at those points with no intermediate pooling. | - The proposed approach had a size that was 88.4% smaller and a training time that was 23.86% faster. |
| A33 | [46] | Using the bottleneck features of the VGG16 network to categorize the malware family. | - Grayscale byte plot images were used to represent malware samples.<br>- The convolutional layers in VGG16 were employed to extract the bottleneck features.<br>- For the malware family classification task, the SVM classifier was trained using these features. | - The accuracy achieved was 92.97% |
| A34 | [47] | To identify people with a possibility of Age-related Macular Degeneration (AMD). | - Added a BN layer to the last fully-connected layers of VGG16. | - The accuracy achieved was 92.5%. |
| A35 | [48] | To detect Alzheimer's Disease (AD). | - Pre-trained weights from huge benchmark datasets were used to initialize the VGG16 and Inception V4 models.<br>- Only a few MRI images were used to retrain the fully-connected layer. | - The challenges of limited dataset were addressed.<br>- This study showed that training sizes almost ten times smaller than existing model were sufficient to obtain equivalent or even better results. |

Despite the advocating results reported by numerous past studies in the literature, several possible enhancements to the VGG16 difficulties exist regarding the image classification problems. The findings in Table 2 indicate that most articles contributed to increase the classification accuracy and reduce the execution time by improving (A1, A4, A6, A8, A10, A11, A15, A16, A18, A19, A21, A23 - A27, A29 - A31, and A35) or hybridizing (A2,

A3, A5, A7, A9, A12 - A14, A17, A20, A22, A28, and A32 - A34) the VGG16 architecture with other algorithms to achieve a better performance in image classification. These findings are undeniably promising with the proposed methodologies achieving outstanding classification accuracy and increasing the speed of the execution time.

Furthermore, it is interesting to note that there is a notion of removing or replacing some layers to make the architecture become light and compact in order to produce high classification accuracy and improve the execution time. The VGG16 2021 architecture improved the standard VGG16 by reducing the convolutional layer at the fifth block in the feature extraction process [14]. Three dense layers were used in the top layer of the architecture instead of two dense layers in the standard VGG16 architecture because adding extra layers to the dense section can increase the architecture's robustness and classification performance. A dropout layer was added after each dense layer to simplify the architecture and prevent overfitting. The VGG16 2021 architecture is shown in Fig. 3.

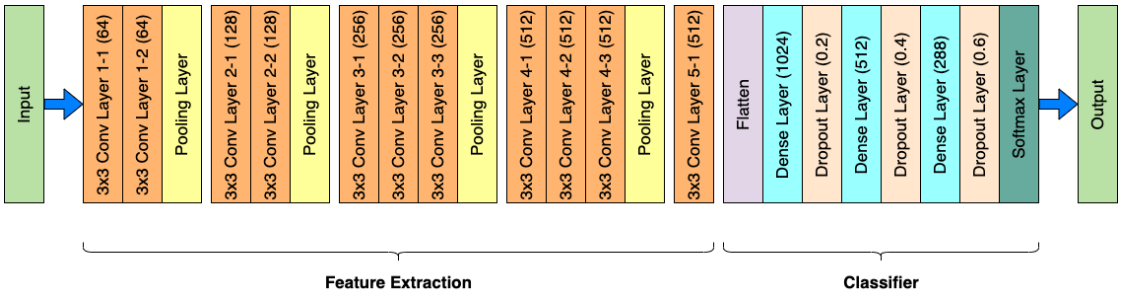


**Fig. 3** *The VGG16 2021 architecture*

Meanwhile, the proposed VGG16 2020 architecture in article A11 [27] was improved by reducing one convolutional layer in the third block and removing the parameters in the last two blocks. A BN layer was added after each pooling layer to obtain zero mean and unit variance in order to optimize the model by normalizing the input map. Following the convolutional layer, this article proposed on the use of GAP layer instead of flatten layer for the classifier since the standard VGG16 architecture connects two 4096-dimensional fully-connected layers, subsequently gaining a massive number of parameters. The VGG16 2020 architecture is shown in Fig. 4.

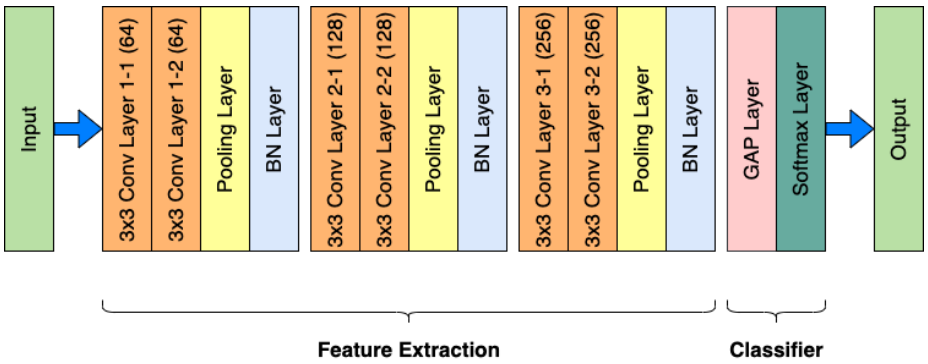


**Fig. 4** *The VGG16 2020 architecture*

These studies on VGG16 2021 and VGG16 2020 provide valuable motivation and support for this study, which aimed to produce an improved VGG16 architecture. VGG16 2021 discussed the need for optimizing deep learning architectures to achieve higher accuracy and efficiency in image classification tasks. It highlighted the VGG16 architecture as a widely used and effective model but also acknowledges its limitations in terms of computational cost and parameter redundancy. Recognizing the shortcomings of the original VGG16 architecture provides a strong motivation for this study to explore potential improvements that can address these issues. Whereas, VGG16 2020 presented a modified VGG16 model that reduced the total amount of parameters but retained competitive accuracy. It demonstrates the importance of adapting the VGG16 architecture to different deployment scenarios and provides further motivation for this study to enhance the architecture's efficiency and effectiveness.

Following a thorough review of empirical findings reported by previous research, this study aimed to address the limitations and challenges that have been identified in the standard VGG16 architecture. Such investigation is crucial for further advancement and contributes to the development of improved deep learning models that can deliver better performance, efficiency, and adaptability in various real-world applications.

## 5. Conclusion

The field of image classification has been greatly assisted by the use of many VGG approaches. The deep convolutional layers and small kernel sizes of the VGG architecture have demonstrated prominent effectiveness in collecting complex image features and delivering excellent results on large-scale datasets. However, there are certain drawbacks to the VGG16 architecture, including computational complexity, memory requirements, and potential overfitting issues. These drawbacks stand as a motivation for more research that focuses on resolving these difficulties and improving the effectiveness and efficiency of VGG-based models for image classification tasks.

A number of research attempted to improve the VGG architecture by addressing its limitation, including the reduction of model complexity while sustaining the classification performance [27], [35]. These initiatives attempt to overcome VGG16's drawbacks and enable its application in real-world situations where efficiency of computation and scalability are important. This study has undertaken numerous initiatives to improve the VGG architecture, concentrating on key aspects such as difficulties in training the network, reduction of parameters, and improvement in convergence rates. Moreover, efforts have been directed towards enhancing the accuracy rate across multiple dimensions of image classification. The focus also extends to minimizing execution times and handling challenges associated with limited datasets.

The growing attempts to improve the VGG approaches and address its limitations have resulted in continuous efforts and developments in the field of image classification. This has promoted the advancement of more accurate and efficient models for various applications in image classification tasks.

## Acknowledgement

This research was supported by Ministry of Higher Education (MOHE) through Fundamental Research Grant Scheme (FRGS/1/2020/ICT02/UTHM/02/1).

## References

[1] P. Wang, E. Fan, and P. Wang, "Comparative analysis of image classification algorithms based on traditional machine learning and deep learning," Pattern Recognition Letter, vol. 141, 2021, doi: 10.1016/j.patrec.2020.07.042.

[2] Z. Liu et al., "Improved Kiwifruit Detection Using Pre-Trained VGG16 with RGB and NIR Information Fusion," IEEE Access, vol. 8, 2020, doi: 10.1109/ACCESS.2019.2962513.

[3] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553. Nature Publishing Group, pp. 436–444, May 27, 2015. doi: 10.1038/nature14539.

[4] H. Azarmdel, A. Jahanbakhshi, S. S. Mohtasebi, and A. R. Muñoz, "Evaluation of image processing technique as an expert system in mulberry fruit grading based on ripeness level using artificial neural networks (ANNs) and support vector machine (SVM)," Postharvest Biol Technol, vol. 166, 2020, doi: 10.1016/j.postharvbio.2020.111201

[5] A. Jahanbakhshi and K. Kheiralipour, "Evaluation of image processing technique and discriminant analysis methods in postharvest processing of carrot fruit," Food Sci Nutr, vol. 8, no. 7, 2020, doi: 10.1002/fsn3.1614.

[6] M. Mateen, J. Wen, Nasrullah, S. Song, and Z. Huang, "Fundus image classification using VGG-19 architecture with PCA and SVD," Symmetry (Basel), vol. 11, no. 1, Jan. 2019, doi: 10.3390/sym11010001.

[7] S. Tammina, "Transfer learning using VGG-16 with Deep Convolutional Neural Network for Classifying Images," International Journal of Scientific and Research Publications (IJSRP), vol. 9, no. 10, p. p9420, Oct. 2019, doi: 10.29322/ijsrp.9.10.2019.p9420.

[8] Y. Wang and Z. Wang, "A survey of recent work on fine-grained image classification techniques," J Vis Commun Image Represent, vol. 59, 2019, doi: 10.1016/j.jvcir.2018.12.049.

[9] N. Ahmad and K. Dimililer, "Brain Tumor Detection Using Convolutional Neural Network," in ISMSIT 2022 - 6th International Symposium on Multidisciplinary Studies and Innovative Technologies, Proceedings, 2022. doi: 10.1109/ISMSIT56059.2022.9932741.

[10] A. Younis, L. Qiang, C. O. Nyatega, M. J. Adamu, and H. B. Kawuwa, "Brain Tumor Analysis Using Deep Learning and VGG-16 Ensembling Learning Approaches," Applied Sciences (Switzerland), vol. 12, no. 14, 2022, doi: 10.3390/app12147282.

[11] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings, pp. 1–14, Sep. 2014, [Online]. Available: http://arxiv.org/abs/1409.1556.

[12] Z. Jiang, "A Novel Crop Weed Recognition Method Based on Transfer Learning from VGG16 Implemented by Keras," IOP Conf Ser Mater Sci Eng, vol. 677, no. 3, 2019, doi: 10.1088/1757-899X/677/3/032073.

[13] D. M. S. Arsa and A. A. N. H. Susila, "VGG16 in Batik Classification based on Random Forest," Proceedings of 2019 International Conference on Information Management and Technology, ICIMTech 2019, vol. 1, no. August, pp. 295–299, 2019, doi: 10.1109/ICIMTech.2019.8843844.

[14] M. N. Islam et al., "Diagnosis of hearing deficiency using EEG based AEP signals: CWT and improved-VGG16 pipeline," PeerJ Comput Sci, vol. 7, p. e638, 2021, doi: 10.7717/peerj-cs.638.

[15] D. I. Swasono, H. Tjandrasa, and C. Fathicah, "Classification of tobacco leaf pests using VGG16 transfer learning," Proceedings of 2019 International Conference on Information and Communication Technology and Systems, ICTS 2019, pp. 176–181, 2019, doi: 10.1109/ICTS.2019.8850946.

[16] Y. Lecun, E. Bottou, Y. Bengio, and P. Haffner, "Gradient-Based Learning Applied to Document Recognition," 1998.

[17] I. Gogul and V. S. Kumar, "Flower species recognition system using convolution neural networks and transfer learning," 2017 4th International Conference on Signal Processing, Communication and Networking, ICSCN 2017, pp. 1–6, 2017, doi: 10.1109/ICSCN.2017.8085675.

[18] D. Albashish, R. Al-Sayyed, A. Abdullah, M. H. Ryalat, and N. Ahmad Almansour, "Deep CNN Model based on VGG16 for Breast Cancer Classification," 2021 International Conference on Information Technology, ICIT 2021 - Proceedings, pp. 805–810, 2021, doi: 10.1109/ICIT52682.2021.9491631.

[19] S. Theetchenya, S. Ramasubbareddy, S. Sankar, and S. M. Basha, "Hybrid approach for content-based image retrieval," International Journal of Data Science, vol. 6, no. 1. p. 45, 2021. doi: 10.1504/ijds.2021.117467.

[20] R. Dey, R. C. Balabantaray, J. Piri, and D. Singh, "Offline Natural Scene Character Recognition Using VGG16 Neural Networks," Proceedings of the 3rd International Conference on Inventive Research in Computing Applications, ICIRCA 2021, pp. 946–951, 2021, doi: 10.1109/ICIRCA51532.2021.9544539.

[21] D. Baby, S. J. Devaraj, and A. M. M. Raj, "Leukocyte classification based on transfer learning of VGG16 features by K-Nearest neighbor classifier," 2021 3rd International Conference on Signal Processing and Communication, ICPSC 2021, no. May, pp. 252–256, 2021, doi: 10.1109/ICSPC51351.2021.9451707.

[22] Y. Chen, X. Zhang, D. Li, J. Jin, and Y. Shen, "Classification of a Small-data-set Thyroid Nodules Based on Supplementary Feature Layer Improved VGG16," Chinese Control Conference, CCC, vol. 2020-July, pp. 7316–7321, 2020, doi: 10.23919/CCC50068.2020.9188671.

[23] Y. C. Hsieh, C. L. Chin, C. S. Wei, I. M. Chen, P. Y. Yeh, and R. J. Tseng, "Combining VGG16, Mask R-CNN and Inception V3 to identify the benign and malignant of breast microcalcification clusters," 2020 International Conference on Fuzzy Theory and Its Applications, iFUZZY 2020, pp. 1–4, 2020, doi: 10.1109/iFUZZY50310.2020.9297809.

[24] A. Panthakkan, S. M. Anzar, S. Al Mansoori, and H. Al Ahmad, "Accurate Prediction of COVID-19 (+) Using AI Deep VGG16 Model," 2020 3rd International Conference on Signal Processing and Information Security, ICSPIS 2020, vol. 19, pp. 4–7, 2020, doi: 10.1109/ICSPIS51252.2020.9340145.

[25] T. Ogawa, H. Lu, A. Watanabe, I. Omura, and T. Kamiya, "Identification of normal and abnormal from ultrasound images of power devices using VGG16," International Conference on Control, Automation and Systems, vol. 2020-Octob, no. Iccas, pp. 415–418, 2020, doi: 10.23919/ICCAS50221.2020.9268275.

[26] Z. Zhang, J. He, and Z. Zhang, "Emotion recognition algorithm based on panorama-plane mapping dataset and VGG16 in prison monitoring system," J Phys Conf Ser, vol. 1627, no. 1, 2020, doi: 10.1088/1742-6596/1627/1/012010.

[27] Zhongqin Bi, Ling Yu, Honghao Gao, Ping Zhou, and Hongyang Yao, "Improved VGG model based efcient trafc sign recognition for safe driving in 5G scenarios.pdf." 2020.

[28] H. Wang, "Garbage recognition and classification system based on convolutional neural network vgg16," Proceedings - 2020 3rd International Conference on Advanced Electronic Materials, Computers and Software Engineering, AEMCSE 2020, pp. 252–255, 2020, doi: 10.1109/AEMCSE50948.2020.00061.

[29] Q. Yan, B. Yang, W. Wang, B. Wang, P. Chen, and J. Zhang, "Apple leaf diseases recognition based on an improved convolutional neural network," Sensors (Switzerland), vol. 20, no. 12, pp. 1–14, 2020, doi: 10.3390/s20123535.

[30] A. A. Pravitasari et al., "UNet-VGG16 with transfer learning for MRI-based brain tumor segmentation," Telkomnika (Telecommunication Computing Electronics and Control), vol. 18, no. 3, pp. 1310–1318, 2020, doi: 10.12928/TELKOMNIKA.v18i3.14753.

[31] W. Setiawan and F. Damayanti, "Layers Modification of Convolutional Neural Network for Pneumonia Detection," J Phys Conf Ser, vol. 1477, no. 5, 2020, doi: 10.1088/1742-6596/1477/5/052055.

[32] A. K. Dubey and V. Jain, "Automatic facial recognition using VGG16 based transfer learning model," Journal of Information and Optimization Sciences, vol. 41, no. 7, pp. 1589–1596, Oct. 2020, doi: 10.1080/02522667.2020.1809126.

[33] A. Krishnaswamy Rangarajan and R. Purushothaman, "Disease Classification in Eggplant Using Pre-trained VGG16 and MSVM," Sci Rep, vol. 10, no. 1, Dec. 2020, doi: 10.1038/s41598-020-59108-x.

[34] K. S. Lee, S. K. Jung, J. J. Ryu, S. W. Shin, and J. Choi, "Evaluation of transfer learning with deep convolutional neural networks for screening osteoporosis in dental panoramic radiographs," J Clin Med, vol. 9, no. 2, Feb. 2020, doi: 10.3390/jcm9020392.

[35] A. B. Perdana and A. Prahara, "Face Recognition Using Light-Convolutional Neural Networks Based on Modified Vgg16 Model," 2019 International Conference of Computer Science and Information Technology, ICoSNIKOM 2019, pp. 14–17, 2019, doi: 10.1109/ICoSNIKOM48755.2019.9111481.

[36] S. Islam, S. I. A. Khan, M. Minhazul Abedin, K. M. Habibullah, and A. K. Das, "Bird species classification from an image using VGG-16 network," PervasiveHealth: Pervasive Computing Technologies for Healthcare, pp. 38–42, 2019, doi: 10.1145/3348445.3348480.

[37] L. Geng, S. Zhang, J. Tong, and Z. Xiao, "Lung segmentation method with dilated convolution based on VGG-16 network," Computer Assisted Surgery, vol. 24, no. sup2, pp. 27–33, 2019, doi: 10.1080/24699322.2019.1649071.

[38] I. Oztel, G. Yolcu, and C. Oz, "Performance Comparison of Transfer Learning and Training from Scratch Approaches for Deep Facial Expression Recognition."

[39] Q. Guan et al., "Deep convolutional neural network VGG-16 model for differential diagnosing of papillary thyroid carcinomas in cytological images: A pilot study," J Cancer, vol. 10, no. 20, pp. 4876–4882, 2019, doi: 10.7150/jca.28769.

[40] P. Hridayami, I. K. G. D. Putra, and K. S. Wibawa, "Fish species recognition using VGG16 deep convolutional neural network," Journal of Computing Science and Engineering, vol. 13, no. 3, pp. 124–130, 2019, doi: 10.5626/JCSE.2019.13.3.124.

[41] H. Chen and C. Haoyu, "Face Recognition Algorithm Based on VGG Network Model and SVM," in Journal of Physics: Conference Series, Institute of Physics Publishing, May 2019. doi: 10.1088/1742-6596/1229/1/012015.

[42] F. J. P. Montalbo and A. A. Hernandez, "Classification of fish species with augmented data using deep convolutional neural network," in 2019 IEEE 9th International Conference on System Engineering and Technology, ICSET 2019 - Proceeding, Institute of Electrical and Electronics Engineers Inc., Oct. 2019, pp. 396–401. doi: 10.1109/ICSEngT.2019.8906433.

[43] Z. Song, L. Fu, J. Wu, Z. Liu, R. Li, and Y. Cui, "Kiwifruit detection in field images using Faster R-CNN with VGG16," in IFAC-PapersOnLine, Elsevier B.V., 2019, pp. 76–81. doi: 10.1016/j.ifacol.2019.12.500.

[44] B. Liu, X. Zhang, Z. Gao, and L. Chen, "Weld defect images classification with VGG16-based neural network," Communications in Computer and Information Science, vol. 815, pp. 215–223, 2018, doi: 10.1007/978-981-10-8108-8_20.

[45] H. Qassim, A. Verma, and D. Feinzimer, "Compressed residual-VGG16 CNN model for big data places image recognition," 2018 IEEE 8th Annual Computing and Communication Workshop and Conference, CCWC 2018, vol. 2018-Janua, pp. 169–175, 2018, doi: 10.1109/CCWC.2018.8301729.

[46] E. Rezende, G. Ruppert, T. Carvalho, A. Theophilo, F. Ramos, and P. de Geus, "Malicious Software Classification Using VGG16 Deep Neural Network's Bottleneck Features," in Advances in Intelligent Systems and Computing, Springer Verlag, 2018, pp. 51–59. doi: 10.1007/978-3-319-77028-4_9.

[47] A. Govindaiah, M. A. Hussain, R. T. Smith, and A. Bhuiyan, "Deep convolutional neural network based screening and assessment of age-related macular degeneration from fundus images," in Proceedings - International Symposium on Biomedical Imaging, IEEE Computer Society, May 2018, pp. 1525–1528. doi: 10.1109/ISBI.2018.8363863.

[48] Marcia Hon and Naimul Mefraz Khan, 2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). IEEE, 2017.