Open Access **Baghdad Science Journal** **P-ISSN: 2078-8665**
Published Online First: February, 2023 **2023, 20(5): 1808-1816** **E-ISSN: 2411-7986**

DOI: https://dx.doi.org/10.21123/bsj.2023.7364

Using VGG Models with Intermediate Layer Feature Maps for Static Hand Gesture Recognition

Osamah Y. Fadhil* ***Bashar S. Mahdi*** ***Ayad R. Abbas***

Department of Computer Sciences, University of Technology, Baghdad, Iraq.
*Corresponding author: osamah.y.fadhil@uotechnology.edu.iq
E-mails address: bashar.s.mahdi@uotechnology.edu.iq, ayad.r.abbas@uotechnology.edu.iq

Received 28/4/2022, Revised 8/10/2022, Accepted 9/10/2022, Published Online First 20/2/2023, Published 1/10/2023

Abstract:

A hand gesture recognition system provides a robust and innovative solution to nonverbal communication through human–computer interaction. Deep learning models have excellent potential for usage in recognition applications. To overcome related issues, most previous studies have proposed new model architectures or have fine-tuned pre-trained models. Furthermore, these studies relied on one standard dataset for both training and testing. Thus, the accuracy of these studies is reasonable. Unlike these works, the current study investigates two deep learning models with intermediate layers to recognize static hand gesture images. Both models were tested on different datasets, adjusted to suit the dataset, and then trained under different methods. First, the models were initialized with random weights and trained from scratch. Afterward, the pre-trained models were examined as feature extractors. Finally, the pre-trained models were fine-tuned with intermediate layers. Fine-tuning was conducted on three levels: the fifth, fourth, and third blocks, respectively. The models were evaluated through recognition experiments using hand gesture images in the Arabic sign language acquired under different conditions. This study also provides a new hand gesture image dataset used in these experiments, plus two other datasets. The experimental results indicated that the proposed models can be used with intermediate layers to recognize hand gesture images. Furthermore, the analysis of the results showed that fine-tuning the fifth and fourth blocks of these two models achieved the best accuracy results. In particular, the testing accuracies on the three datasets were 96.51%, 72.65%, and 55.62% when fine-tuning the fourth block and 96.50%, 67.03%, and 61.09% when fine-tuning the fifth block for the first model. The testing accuracy for the second model showed approximately similar results.

Keywords: Convolutional Neural Networks, Deep Learning, Hand Gesture Recognition, VGG-16, VGG-19.

Introduction:

People use many ways to express meaning; they may speak, sign, or write to convey their ideas to others. However, deaf people cannot communicate with others using voiced language. Therefore, they depend on sign language, which incorporates different body and hand movements. Hand gestures represent a primary part of sign language in which different postures of hands have varying meanings. A specific hand posture may mean a single alphabetical letter, a word, or a sentence[1].

Static[2- 4] or dynamic[5] hand gestures are commonly used in recognition applications. In static hand gestures, meaning is expressed by hand postures. In contrast, in dynamic hand gestures, hand movements are also involved in conveying meaning. This study is concerned with static hand gestures, wherein each hand gesture represents the meaning of a single alphabetical letter. Hand gesture recognition has many real-life applications, such as facilitating communication with deaf people and producing ways of interaction that can be used nowadays across different applications, such as virtual environments, gaming, and appliance control.

Consequently, many methods and techniques dealing with the hand gesture recognition problem are available[6]. Hand gesture recognition systems can be categorized into two types: sensor-based (also known as glove-based) and image-based (also known as vision-based) systems. There are some limitations and drawbacks to sensor-based systems.

In particular, the signer must wear gloves attached to sensors to convey gesture information, and these wearable gloves may be cumbersome for the user. Furthermore, sensors may be expensive. These limitations have led to the use of image-based systems as an alternative to sensor-based systems[6-8]. Similar to other recognition systems, a hand-gesture recognition system consists of two parts: feature extraction and classification. Researchers have developed different methods to extract features from digital images and different classifiers. The extracted features are then fed to a classifier to recognize gestures. Traditional methods include artificial neural networks (ANNs), hidden Markov models (HMMs), support vector machines (SVMs), and transform-based models[7,8]. Recently, deep learning models have been successfully applied to digital image and speech[9] tasks, including convolutional neural networks (CNNs), prompting researchers to investigate such models in exploring recognition problems. Unlike machine learning approaches that require extracting handcrafted features[10] from data, CNNs automate the process of feature extraction. These models have hierarchical architectures and learn features with various levels of abstraction at each layer[2]. Some researchers[2,4,11] have conducted recognition experiments using CNN models and they have trained these models from scratch on hand gesture image datasets. Others[12] have used pre-trained models to train the model on the new dataset rather than the whole model. This second method is known as "transfer learning." The benefits of fine-tuning models are that less time is required to train the model, and it can be used even when the dataset size is not large enough to train the model from scratch[12, 13].

Inspired by Le-Net-5, a study[11] proposed a CNN model, which the authors trained on a dataset of 39 classes of hand gesture images representing alphanumeric data. In addition, they reported improved recognition results over traditional methods of k-nearest neighbor and SVM. A model comprising three convolutional layers was introduced[2] to recognize hand gestures with complex backgrounds. The authors evaluated the model on two public datasets: one consisting of 10 different classes and another comprising 24 letter classes acquired under similar lighting conditions. Their results showed that using this model eliminated the need for the hand segmentation method, which is a challenging task for images with cluttered backgrounds. Overall, they showed promising results for their proposed model. Some authors[3] applied deep learning methods to recognize 24 classes of hand gesture images. They used CNNs and stacked denoising autoencoders to solve the problem. They demonstrated that their models could recognize similar hand gestures with higher recognition rates compared with other methods, such as ANN.

Meanwhile, some studies have utilized existing models to address the problem of hand gesture recognition. Others[14] modified two network architectures based on AlexNet and VGGNet, respectively, to recognize hand gestures. Using a combination of three components—hand detection, hand tracking, and hand recognition—they found that this approach is feasible in practical applications. However, they implemented the model using only six classes of hand gestures. Other researchers[15] used the inception model for hand gesture recognition, in which they fed the model with depth image data in addition to the color image data acquired by an acquisition device known as Kinect. Using transfer learning to train the last layer of the model on a target dataset of 10 classes, they reported highly accurate results for their model.

Some authors[16] proposed a CNN model and fine-tuned pre-trained Visual Geometry Group (VGG) models. They conducted different experiments on a dataset with 33 classes of static hand gestures to evaluate these models. In addition, they reported high-end accuracy for the proposed model and a further increase in performance with these fine-tuned VGG models. A study[12] used pre-trained VGG-16 and ResNet152 models to recognize 32 different classes of hand gestures. Their reported results revealed that these fine-tuned models had high recognition accuracy during the experiments. Another study[13] examined pre-trained VGG models as feature extractors and fine-tuned them on ear images. Using their methodology of training and recognition, they reported the superior performance of the fine-tuned models compared to other learning methods.

Although recent studies have introduced various pre-trained deep learning models to recognize static hand gestures, these studies have not stated whether one can select an intermediate layer within these models to extract features from hand gesture images. Therefore, this paper aims to fill this research gap. The contributions of this study are as follows:

- This study adopted two VGG models[17] as efficient deep learning models in image classification to conduct recognition experiments on hand gesture images. The models were adjusted to suit the image datasets and then trained under different learning strategies, including fine-tuning intermediate layers of the models. To the

best of our knowledge, this strategy has not been previously tackled in similar studies.

- This study introduced a new hand-gesture image dataset to evaluate the performance of models under challenging conditions.
- The models are trained on a dataset and tested on various datasets, unlike most previous studies[12, 18-21] in this field, which divided the same dataset[22] into training and testing sets.

The remainder of this paper is organized into sections. The next section explains the datasets used and considers the adapted VGG model. The training methods are also applied in this section. The results are then discussed in the next section. The details of the models are presented in Table. 1, and a comparison with other similar studies is shown in Table. 2. The training and validation accuracies are shown in graphs. The final section presents our conclusions.

Materials and Methods:

This study used VGG models with intermediate layer feature maps to recognize static hand gesture images in the Arabic sign language.

Datasets

This study focused on three datasets. The first, known as ArASL[22], is a large, public dataset prepared for image recognition and classification tasks. Fig.1(a) displays sample images from this dataset. These images are grayscale 64 × 64 pixels and fall into 32 classes corresponding to the Arabic alphabet letters. However, six classes were excluded from this dataset to match the number of classes in the other test datasets. Furthermore, the number of samples was evenly distributed among the classes, leaving 1293 samples in each class. This dataset was divided into two sets: training 70% and testing 30%.

The second and third datasets were used to evaluate the performance of the models under challenging conditions. This means that all the images of these datasets are used for testing purposes. The second dataset was collected to test the models on different samples, where 30 individuals signed the gestures as in the first dataset. These are 840 grayscale images of diverse sizes, of which 50% of the images are of the left hand and the other 50% are of the right hand. Fig. 1(b) displays sample images from this dataset. The third dataset[11] consists of 28 classes of 5771 images. These images are of size 256 × 256 pixels in color. Fig. 1(c) displays sample images from this dataset. Before testing on the second and third image datasets, these images were re-scaled to 64 × 64 to match the sizes of the training images.

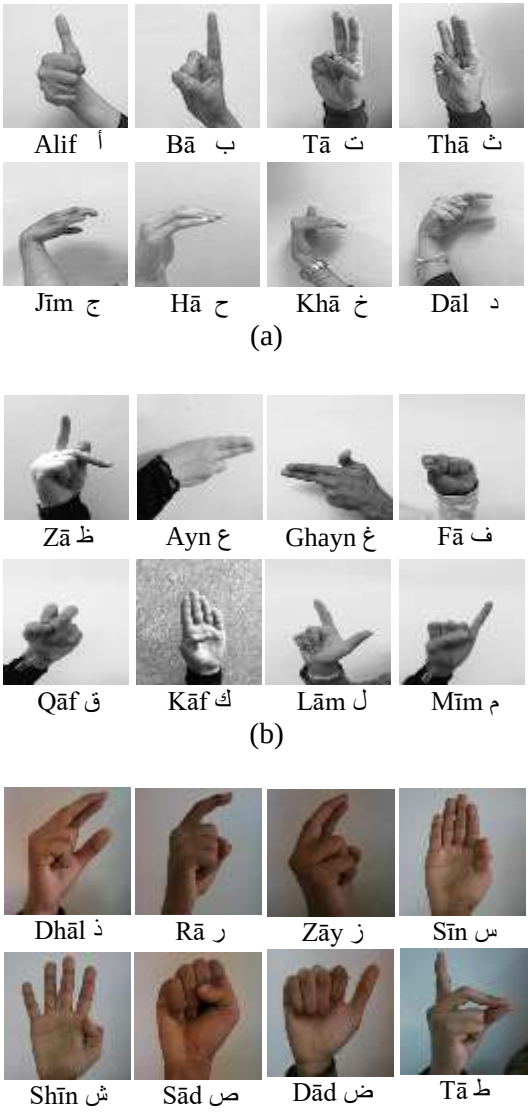


Figure 1. (a) Sample images from the ArASL dataset[22]. (b) Sample images from the prepared test dataset. (c)Sample images from the dataset[11] of colored images.

Proposed Model

This study considered two VGG models, known as VGG-16 and VGG-19. The VGG is the creator of these models, which have proven to be effective in many computer vision and image classification tasks. Each model takes an image as an input and produces a classification output. The structure of each model consists of five convolutional blocks, followed by fully connected layers (FCs). The convolutional blocks extract features from the image, and the FC layers classify an image according to those features.

The VGG models were adjusted to make them more suitable for handling the image dataset. Minor changes were made between these models, as

shown in Table. 1. These models were changed to receive 64 × 64 images to match the size of the training images, and two fully connected layers were stacked instead of three layers in the original model. This is because the two layers are sufficient for the current recognition task. See Fig. 2

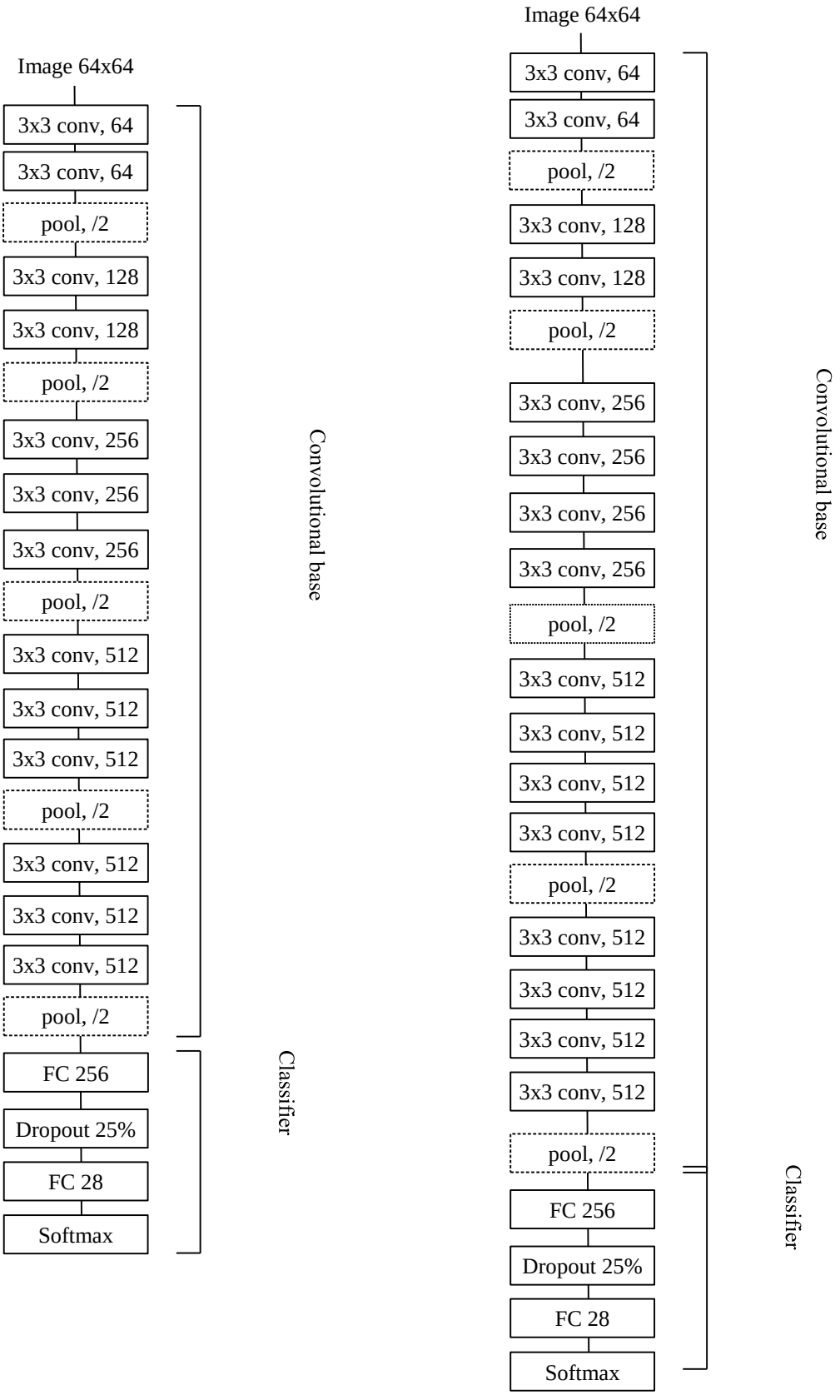


Figure 2. VGG-based models with configured settings. VGG-16 on the left, and VGG-19 on the right.
Table 1. Details of the VGG models with configured settings.

Block	VGG-16 Model	VGG-19 Model	Filter Size	Output Size
Input	Input Image (64 x 64 Gray scale)			
Block1	Convolution	Convolution	3 x 3 (64)	64 x 64
	Convolution	Convolution	3 x 3 (64)	64 x 64
	Max-Pooling			32 x 32
Block2	Convolution	Convolution	3 x 3 (128)	32 x 32
	Convolution	Convolution	3 x 3 (128)	32 x 32
	Max-Pooling			16 x 16
Block3	Convolution	Convolution	3 x 3 (256)	16 x 16
	Convolution	Convolution	3 x 3 (256)	16 x 16
	Convolution	Convolution	3 x 3 (256)	16 x 16
	Convolution	Convolution	3 x 3 (256)	16 x 16
	Max-Pooling			8 x 8
Block4	Convolution	Convolution	3 x 3 (512)	8 x 8
	Convolution	Convolution	3 x 3 (512)	8 x 8
	Convolution	Convolution	3 x 3 (512)	8 x 8
	Convolution	Convolution	3 x 3 (512)	8 x 8
	Max-Pooling			4 x 4
Block5	Convolution	Convolution	3 x 3 (512)	4 x 4
	Convolution	Convolution	3 x 3 (512)	4 x 4
	Convolution	Convolution	3 x 3 (512)	4 x 4
	Convolution	Convolution	3 x 3 (512)	4 x 4
	Max-Pooling			2 x 2
Fully Connected	Fully Connected 256 Neurons Dropout Chance 25% Fully Connected 28 Neurons Soft-Max			

Training Methods

This study considers the exact training steps described in a previous work[23]. As shown in Table. 1, each model starts with a series of convolutional blocks and ends with a fully connected classifier. First, each model was given random weights and trained from scratch for comparison purposes. Next, the pre-trained convolutional blocks of the model were employed to extract the features. A new classifier was added on top of these blocks and trained from scratch. Then, the top layers of the model were jointly trained with the added classifier. The final training method is called "fine-tuning." Due to the hierarchical learning structure of CNN, only the top layers of the model are fine-tuned. In this study, the convolutional blocks of the models were fine-tuned under three scenarios: first, beginning with the fifth block layer, then with the fourth block layer, and finally, the third block layer. The steps of fine-tuning are as follows[23]:

1. Add a new layer on top of the pre-trained base network.
2. Make the base network untrainable.
3. Train the new layer.
4. Make some layers in the base network trainable.
5. Train these layers and the newly added layer.

Training Details

A widespread problem with machine learning models is overfitting. Thus, some transformation operations were applied randomly to the training images before being fed to the models to alleviate overfitting. As a result, the model learns various kinds of patterns from the augmented dataset. The following augmentation operations were applied to the images:

1. Rescaling with a 1/255 factor.
2. Random rotation within the range of 40 degrees.
3. Random horizontal and vertical translation within the range of 0.2 of total width or height.
4. Random shearing transformations with a factor of 0.2.
5. Random zooming with a factor of 0.2.
6. Random horizontal flipping.

The stopping condition was used to interrupt training when the validation loss was no longer improving, in which a batch size of 20 and a learning rate of 10^{-5} with an RMSProp optimizer were applied. The loss function was a categorical cross-entropy suitable for multiclass, single-label classification. All experiments were conducted with the Google collaboratory environment using the Tensorflow[24] and the Keras libraries.

Results and Discussion

The main purpose of this study was to test the recognition performance of the pre-trained VGG models with different intermediate layers on hand-gesture images. These models were tested on the abovementioned datasets, and two popular metrics, accuracy and top-5 accuracy, were evaluated in the validation set. The accuracy measure shows the percentage of correctly classified samples, while the top-5 accuracy indicates the percentage of classifications in which the correct label appears in the five classes with the highest scores. The obtained results are shown in Table. 2.

Table. 2. A comparison of accuracy and top-5 accuracy for the two VGG-based models on the three datasets. The results are given in percentages.

Method	Model	Dataset[22]		Test Dataset		Dataset[11]	
		Accuracy	Top-5 Accuracy	Accuracy	Top-5 Accuracy	Accuracy	Top-5 Accuracy
Scratch	VGG-16	91.53	99.18	56.71	87.34	22.49	56.25
Training	VGG-19	92.04	99.26	57.96	86.40	27.65	73.59
Feature	VGG-16	56.69	91.95	34.68	73.43	25.46	68.75
Extraction	VGG-19	55.09	91.73	31.09	72.18	27.03	68.59
Fine-Tuning	VGG-16	96.50	99.71	67.03	92.18	61.09	89.99
Block 5	VGG-19	95.78	99.47	64.84	91.25	62.65	92.03
Fine-Tuning	VGG-16	96.51	99.71	72.65	94.84	55.62	83.12
Block 4	VGG-19	95.67	99.59	68.43	92.18	62.65	90.93
Fine-Tuning	VGG-16	87.10	99.05	62.50	89.84	34.99	72.96
Block 3	VGG-19	3.57	17.88	3.90	18.43	2.34	17.34

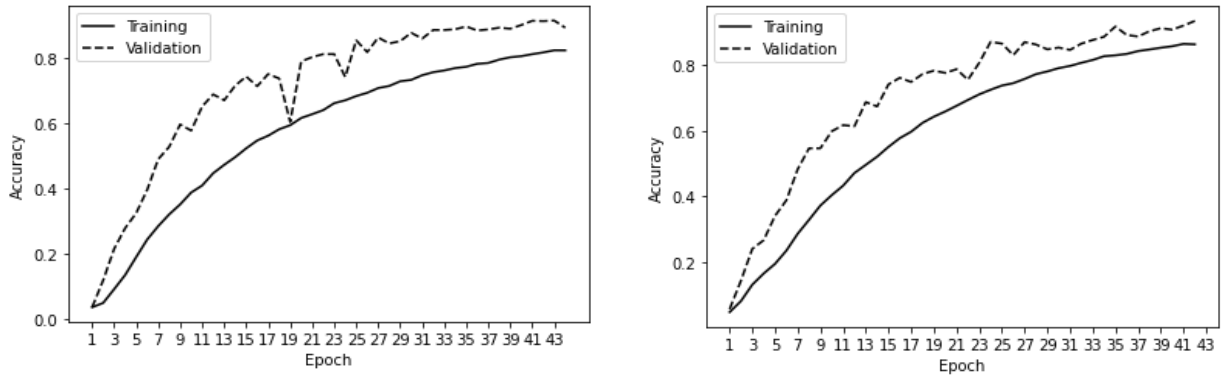
Regarding the training methods, it can be seen from Table. 2 that the models achieved the best performance on all datasets when fine-tuning the fourth and fifth blocks. Focusing on these two training cases, the performance on the first and second datasets indicated that block five contained unnecessary information to recognize the images. These results confirmed the experimental results in a previous work[13], in which the authors demonstrated the superior performance of fine-tuning over training from scratch and feature extraction. However, the results also revealed that the performance dropped, particularly with VGG-19, when fine-tuning the third block. This case indicated that the fourth block contained the necessary feature information for the classifier to recognize the images. When the models were trained from scratch, the performance was lower than the two best cases of fine-tuning. Moreover, this came at the expense of training time because more training iterations were required for the model to stop training. Thus, the performance of the models as feature extractors revealed that training the classifier alone was not sufficient to extract feature information.

Turning now to the datasets, the models showed variations in their performance, as shown in Table. 2. The models were trained on a subset of sample images from the first dataset, and these images showed a high degree of similarity. Obviously, these models achieved the highest recognition accuracy when tested on sample images from the first dataset. These datasets created a challenge for the models, as the conditions of the sample images in the second and third datasets varied. A comparison of the performance with other published studies on the same dataset is shown in Table. 3.

Table 3. Comparison of the testing accuracies obtained for the adapted models with other models on the same dataset as Reference [22]. The results are given in percentages

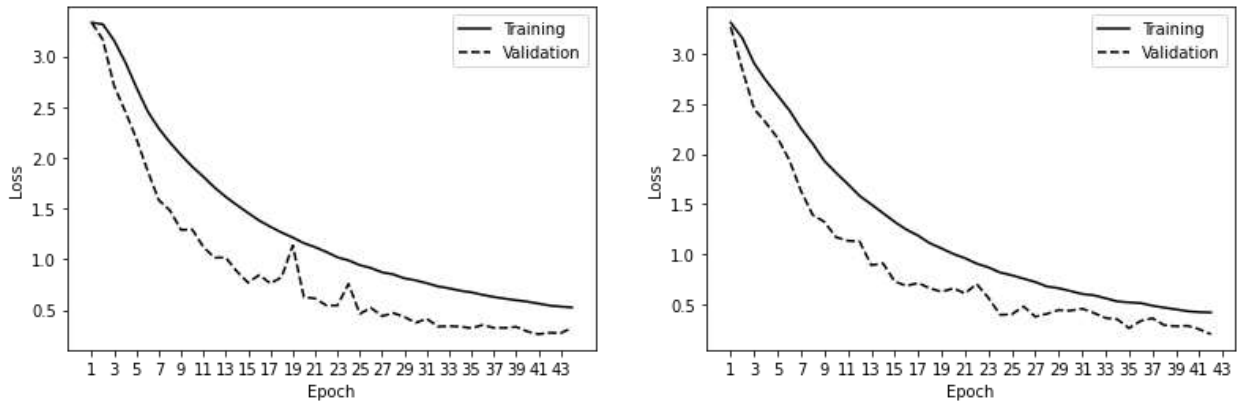
Reference	Model	Accuracy
12	VGG16	99.26
	ResNet152	99.57
18	ArSL-CNN	96.59
	ArSL+SMOTE	97.29
19	Proposed CNN	97.6
20	CNN-2	96.4
21	AlexNet	93
	GoogleNet	88
	VGGNet	97
25	ResNet50	97.5
	MobileNetV2	97.1
	ResNet50 + MobileNetV2	98.2
26	EfficientNetB4	95
Best Results with this study	VGG16 fine-tuning block 5	96.50
	VGG19 fine-tuning block 5	95.78
	VGG16 fine-tuning block 4	96.51
	VGG19 fine-tuning block 4	95.67

Figures. 3 and 4 show the training and validation accuracies as well as the training and validation losses of the configured VGG models during scratch training, respectively. As shown in Fig. 4, the validation losses were lower than their corresponding training losses. Thus, it can be concluded that the scratch training of these models did not result in overfitting.



(a) **(b)**

Figure 3. Training and validation accuracies for (a) the VGG-16 and (b) the VGG-19 models under scratch training.



(a) **(b)**

Figure 4. Training and validation losses for (a) the VGG-16 and (b) the VGG-19 model under scratch training.

Table. 4 shows the number of trainable parameters and the training time for each VGG model. The original VGG models have been proposed for the ImageNet classification task, which has 1000 classes. However, the datasets in this study have only 28 classes, resulting in fewer parameters in the fully connected layers. As a result, there are fewer trainable parameters in this table than in the original VGG models.

Finally, there are some limitations to these experiments that should be mentioned. First, the models were trained on a dataset of images with a high degree of similarity, resulting in low recognition accuracy when testing on the other two datasets. Therefore, there is a need for a hand-gesture dataset comprising varied images. Second, the experiments were conducted in an environment that depended upon the speed of the internet connection, which affected the training time. Thus, if the training was achieved on a local machine that satisfied the requirements of the experiments, the training time would decrease.

Table 4. The number of trainable parameters and the training time for the two VGG-based models.

Method	Model	Trainable Parameters	Training Time (Milliseconds Per Step)
Scratch Training	VGG-16	15,246,428	51
	VGG-19	20,556,124	139
Feature Extraction	VGG-16	531,740	60
	VGG-19	531,740	69
Fine-Tuning	VGG-16	7,611,164	69
5-Blocks	VGG-19	9,970,972	79
Fine-Tuning	VGG-16	8,004,380	64
4-Blocks	VGG-19	10,364,188	69
Fine-Tuning	VGG-16	5,677,084	57
3-Blocks	VGG-19	6,267,164	64

Conclusion:

The hand gesture recognition system aims to forge an interaction between humans and computers. Most previous studies overcame this issue by proposing new model architectures or fine-tuning pre-trained models. In this study, two versions of VGG models were trained on hand-gesture images using different strategies. First, the models were trained from scratch, then they were used as feature extractors, and the pretrained models were finally fine-tuned with intermediate layers. The models were slightly modified and tested on different datasets to create challenging conditions. The first dataset, which was publicly available, was divided into two subsets: one for training and the other for testing. The other two datasets were used for testing purposes only. One of these testing datasets was created as part of this study. The third dataset consisted of colored images, unlike the first and second ones, which consisted of grayscale images.

The experimental results revealed that the best recognition accuracy of these models could be obtained when the models were fine-tuned at the fourth and fifth blocks. Moreover, significantly reducing the number of training parameters leads to a reduction in the training time. Furthermore, the accuracies of the models were decreased on the second and third datasets because the images in these datasets were more different from the images in the first dataset. Therefore, the proposed models could be used in practical applications if trained on datasets with more varied images. However, this study is different from other studies in its field because a similar strategy applied to hand-gesture images was not found during our investigation. Therefore, future works could extend these experiments to investigate other deep learning models, such as ResNet. Furthermore, there is a need to understand what these layers have learned to determine the most relevant regions of the hand gesture that led the model to make its decision during the recognition process.

Authors' declaration:

- Conflicts of Interest: None.
- We hereby confirm that all the Figures and Tables in the manuscript are mine ours. Besides, the Figures and images, which are not mine ours, have been given the permission for re-publication attached with the manuscript.
- Ethical Clearance: The project was approved by the local ethical committee in University of Tecnology.

Authors' contributions statement:

O. Y. F. design, acquisition of data, analysis, interpretation, and drafting the MS. A. R. A. design, analysis, interpretation, revision and proofreading. B. S. M. revision and proofreading.

References:

1. Bragg D, Koller O, Bellard M, Berke L, Boudreault P, Braffort A, et al. Sign language recognition, generation, and translation: An interdisciplinary perspective. The 21st Int Acm Sigac- Cess Conf Comp Access. 2019; 16-31.
2. Venugopalan A, Reghunadhan R. A Deep Convolutional Neural Network Approach for Static Hand Gesture Recognition. Procedia Comput Sci. 2020; 171: 2353-2361.
3. Oyedotun OK, Khashman A. Deep learning in vision-based static hand gesture recognition. Neural Comput Appl. 2017; 28: 3941-3951.
4. Sharma S, Singh S. Vision-based hand gesture recognition using deep learning for the interpretation of sign language. Expert Syst Appl. 2021. 182: 1-12.
5. Ding I J, Zheng N W, Hsieh M C. Hand gesture intention-based identity recognition using various recognition strategies incorporated with VGG convolution neural network-extracted deep learning features. J Intell Fuzz Syst. 2021; 40: 7775-7788.
6. Oudah M, Al-Naji A, Chahl J. Hand Gesture Recognition Based on Computer Vision: A Review of Techniques. J Imaging. 2020; 6: 1-29.
7. Alzohairi R, Alghonaim R, Alshehri W, Aloqeely S. Image based Arabic Sign Language recognition system. Int J Adv Comput Sci Appl. 2018; 9: 185-194.
8. Suharjito Anderson R, Wiryana F, Ariesta M C, Kusuma G P. Sign Language Recognition Application Systems for Deaf-Mute People: A Review Based on Input-Process-Output. Procedia Comput Sci. 2017; 116: 44- 448.
9. Asroni A, Ku-Mahamud KR, Damarjati C, Slamat HB. Arabic Speech Classification Method Based on Padding and Deep Learning Neural Network. Baghdad Sci J [Internet]. 2021Jun.20 [cited 2022Sep.11]; 18(2(Suppl.): 0925. Available from: https://bsj.uobaghdad.edu.iq/index.php/BSJ/article/view/6213.
10. Mahmood RAR, Abdi A, Hussin M. Performance Evaluation of Intrusion Detection System using Selected Features and Machine Learning Classifiers. Baghdad Sci.J [Internet]. 2021Jun.20 [cited 2022Sep.11]; 18(2(Suppl.): 0884. Available from: https://bsj.uobaghdad.edu.iq/index.php/BSJ/article/view/6210.
11. Hayani S, Benaddy M, El Meslouhi O, Kardouchi M. Arab Sign Language Recognition with Convolutional Neural Networks. Int Conf Comp Sci Renew Energies. 2019; 1-4.
12. Saleh Y, Issa GF. Arabic sign language recognition through deep neural networks fine-tuning. Int J Online Biomed Eng. 2020; 16: 71-83.

13. Alshazly H, Linse C, Barth E, Martinetz T. Ensembles of Deep Learning Models and Transfer Learning for Ear Recognition. Sens. 2019; 19: 1-26.
14. Chung H Y, Chung Y L, Tsai W F. An Efficient Hand Gesture Recognition System Based on Deep CNN. IEEE Int Conf Ind Technol. 2019; 853-858.
15. Sokhib T, Whangbo TK. A combined method of skin- and depth-based hand gesture recognition. Int Arab J Inf Technol. 2020; 17: 137-145.
16. Odartey LK, Huang Y, Asantewaa EE, Agbedanu PR. Ghanaian Sign Language Recognition Using Deep Learning. PRAI 19: Proceedings of the 2019 the International Conference on Pattern Recognition and Artificial Intelligence. 2019; 81-86.
17. Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. ArXiv. 2015; 1-14.
18. Alani AA, Cosma G. ArSL-CNN: A convolutional neural network for Arabic sign language gesture recognition. Indones J Electr Eng Comput Sci. 2021; 22: 1096-1107.
19. Latif G, Mohammad N, AlKhalaf R, AlKhalaf R, Alghazo J, Khan MA. An Automatic Arabic Sign Language Recognition System based on Deep CNN: An Assistive System for the Deaf and Hard of Hearing. Int J Comput Digit Syst. 2020; 9: 715-724.
20. Alshomrani S, Aljoudi L, Arif M. Arabic and American Sign Languages Alphabet Recognition by Convolutional Neural Network. Adv Sci Technol Res J. 2021; 15: 136-148.
21. Duwwairi RM, Halloush ZA. Automatic recognition of Arabic alphabets sign language using deep learning. Int J Electr Comput Eng. 2022; 12: 2996-3004.
22. Latif G, Mohammad N, Alghazo J, AlKhalaf R, AlKhalaf R. ArASL: Arabic Alphabets Sign Language Dataset. Data Brief. 2019; 23: 1-4.
23. Chollet F. Deep Learning with Python. 1 st ed. Shelter Island: Manning Publications. 2018 Chap.5. p. 154.
24. Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. Arxiv. 2016 Mar; Available from: https://arxiv.org/abs/1603.04467..
25. Alnuaim A, Zakariah M, Hatamleh WA, Tarazi H, Tripathi V, Amoatey ET. Human-Computer Interaction with Hand Gesture Recognition Using ResNet and MobileNet. Comput Intell Neurosci. 2022; 2022: 1-16.
26. Zakariah M, Alotaibi YA, Koundal D, Guo Y, Elahi MM. Sign Language Recognition for Arabic Alphabets Using Transfer Learning Technique. Comput Intell Neurosci. 2022; 2022: 1-15.

استخدام نماذج VGG بخرائط ميزات الطبقة المتوسطة للتعرف على الايماءات الثابتة لليد

اسامة يونس فاضل* **بشار سعدون مهدي** **اياد روضان عباس**

قسم علوم الحاسوب، الجامعة التكنولوجية، بغداد، العراق

الخلاصة:

يوفر نظام التعرف على إيماءات اليد حلاً قويًا ومبتكرًا للاتصال غير اللفظي من خلال التفاعل بين الإنسان والحاسوب. تتمتع نماذج التعلم العميق بإمكانيات ممتازة لاستخدامها في تطبيقات التعرف. للتغلب على المشكلات ذات الصلة ، اقترحت معظم الدراسات السابقة هياكل نموذجية جديدة أو قامت بضبط النماذج المدربة مسبقًا. علاوة على ذلك ، اعتمدت هذه الدراسات على مجموعة بيانات قياسية واحدة للتدريب والاختبار. وبالتالي، فإن دقة هذه الدراسات معقولة. على عكس هذه الأعمال، تبحث الدراسة الحالية في نموذجين للتعلم العميق مع طبقات وسيطة للتعرف على صور إيماءات اليد الثابتة. تم اختبار كلا النموذجين على مجموعات بيانات مختلفة ، وتم تعديلهما ليناسب مجموعة البيانات ، ثم تم تدريبهما وفقًا لأساليب مختلفة. أولاً ، تمت تهيئة النماذج باستخدام أوزان عشوائية وتم تدريبها من نقطة الصفر. بعد ذلك ، تم فحص النماذج المدربة مسبقًا على أنها مستخرجة من الميزات. أخيرًا ، تم ضبط النماذج المدربة مسبقًا باستخدام طبقات وسيطة. تم إجراء الضبط الدقيق على ثلاثة مستويات: المستوى الخامس والرابع والثالث على التوالي. تم تقييم النماذج من خلال تجارب التعرف باستخدام إيماءات اليد في لغة الإشارة العربية المكتسبة في ظل ظروف مختلفة. توفر هذه الدراسة أيضًا مجموعة بيانات جديدة لصورة إيماءات اليد المستخدمة في هذه التجارب، بالإضافة إلى مجموعتي بيانات أخريين. تشير النتائج التجريبية إلى أنه يمكن استخدام النماذج المقترحة مع الطبقات المتوسطة للتعرف على صور إيماءات اليد. علاوة على ذلك ، أظهر تحليل النتائج أن الضبط الدقيق للكتلتين الخامسة والرابعة من هذين النموذجين حقق أفضل نتائج دقة. على وجه الخصوص ، كانت دقة الاختبار على مجموعات البيانات الثلاث 96.51٪ و 72.65٪ و 55.62٪ عند ضبط الكتلة الرابعة و 96.50٪ و 67.03٪ و 61.09٪ عند ضبط الكتلة الخامسة للنموذج الأول. أظهرت دقة الاختبار للنموذج الثاني نتائج مماثلة تقريبًا

الكلمات المفتاحية: الشبكات العصبية التلافيفية، التعلم العميق، التعرف على إيماءات الي ، في جي جي -١٦، في جي جي -١٩