ISSN: 2454-9940

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

E-Mail :
editor.ijasem@gmail.com
editor@ijasem.org

www.ijasem.org

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN 2454-9940

www.ijasem.org

Vol 19, Issue 2, 2025

Using Transfer Learning to Identify Disguised Faces

[1] M. Pradeepthi, [2] K. Soumya,

[1]Assistant Professor, Megha Institute of Engineering & Technology for Women, Ghatkesar.
[2] MCA Student, Megha Institute of Engineering & Technology for Women, Ghatkesar.

***Abstract*—**

One use of machine learning is face identification, which entails identifying faces within images or videos. Our long-term memory allows us to identify familiar faces and places, as well as many types of living and nonliving things. But how does the computer manage to achieve it with its memory? In computer vision, machine learning is a methodology or technology that may be used to make computers comprehend faces in images or videos. In order to determine which of many common Convolutional Neural Network (CNN) Model Architectures is better at recognizing disguised faces in datasets, the author of this research suggests testing them. This work will be helpful for anybody interested in exploring and developing deep learning architectures, since the author utilizes the "Recognizing Disguised Faces" dataset to identify 75 different face classes, and then tries to train and assess the machine's recognition accuracy. In order to address the issue of picture categorization, this study is anticipated to make a contribution to the area of Machine Learning. Using transfer learning in VGG Models significantly improves the experimental outcomes. Finally, we draw the conclusion that VGG Models for face recognition work best when utilizing ImageNet weight. AI, deep learning, facial recognition, and transfer learning are some of the most important terms in the field.

INTRODUCTION

Recognition of both non-living (object) and living (human, animal, event plant) things is one of the numerous fields of study in Machine Learning. One common method of biometric identification is to utilize a person's face. Because our brains and memories allow us to distinguish between individuals, we humans are able to recognize one another. But machines can't reason for themselves, so Arthur Samuel laid the groundwork for a new field: machine learning [1]. Eigenfaces [2], Principal Component Analysis (PCA) [3], and Convolutional Neural Networks (CNN) [4] are just a few of the methods that have been developed in the last decade to identify people's faces. Since then, the capacity to do so has steadily improved. As a machine learning technique, transfer learning entails starting with a training assignment to create a model, and then applying that model to a second test. The use of an existing, pre-trained model to start a new, unrelated job is what distinguishes transfer learning from more conventional forms of machine learning [5]. numerous different areas of study have made use of CNN because to its numerous benefits, such as transfer learning. Among them are picture categorization[6], item detection[8], video analysis[9], food detection[10], face recognition[4,11], and pedestrian detection [7]. Keras is an open-source neural network framework built in Python [12]. In this study, we analyze several common pre-trained CNN model architectures. We put VGG16, VGG19, ResNet50, ResNet152 v2, InceptionV3, and Inception-ResNet V2 into action. Next, we split into two parts: first, we use the vector to train the classifier model. Then, we evaluate the model's accuracy and cost function. Our goal in doing this study was to identify the most accurate Pre-Trained Architecture model in an ideal hyperparameter state, with the lowest cost function. The author draws on the "Recognizing Disguised Faces" dataset [13], which contains 75 images of people's faces when they are wearing various forms of disguise, such as a bandana, mask, false mustache, fake beard, spectacles, etc. Typically, each individual in the collection is assigned seven or eight pictures, with the final two showing their true faces.

PROPOSED METHODS

A. Artificial Neural Network (ANN)

An ANN is a kind of computational system that mimics the way neural networks in the human brain operate in many ways. In 1943, ANN was first created by McCulloch & Pitts. In order to generalize neurobiology or human cognition, artificial neural networks (ANNs) were created as mathematical models. These models are founded on the following assumptions: - Neurons are the building blocks where information processing takes place. - Each link in a network has a weight that transmits the signal.

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT
ISSN 2454-9940
www.ijasem.org
Vol 19, Issue 2, 2025

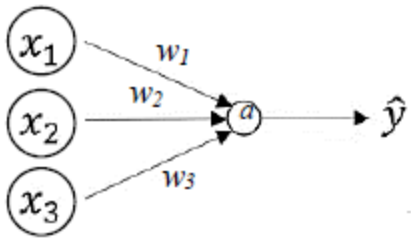


Fig. 1. Single neuron network.

The signal-passing connection between neurons is known as the connection link. In order for each neuron to decide its output, it applies an activation function to its network input, which is often non-linear. Many different parts, often referred to as neurons, units, cells, or nodes, make up an ANN's processing units. Each neuron was linked to the rest of the network and assigned a burden. The pattern classification challenge is one of the most popular applications of neural networks, and weights indicate the information that will be used to solve difficulties. The basic neural network in Figure 1 gets input from activation function neuron a and uses its output neurons, denoted as ŷ. Also, three additional activations, x1, x2, and x3, each provide input to a. The names of their neurons are represented by the symbols X1, X2, and X3. In addition, w1, w2, and w3 are the weights that link X1, X2, and X3 to activation function a. Therefore, (1) represents the output computation.

$$\hat{y} = a = w_1x_1 + w_1x_1 + w_1x_1 \qquad (1)$$

Once that is done, the network for the loss function may be calculated. You may think of the loss function as an error calculation for one step of training; it measures the difference between the predicted value and the actual value, or ground truth. You may see this function in (two).

$$L\left(\hat{y}^{(i)}, y^i\right) = \frac{1}{2}\left(\hat{y}^{(i)}, y^i\right)^2 \qquad (2)$$

We want to categorize each individual based on their face, hence we employ categorical cross-entropy as our loss function in this study. With a value of 1 for true and a value of 0 for false, this function will compare the distribution of anticipated faces. A one-hot encoded vector reflects the genuine face class; a smaller loss is obtained as the model's output vector approaches the true class. Below is the loss function:

$$L\left(X_i, Y_i\right) = -\sum_{j=1} y_{ij} * \log(p_{ij}) \qquad (3)$$

$$y_{ij} = \begin{cases} 1, & if\ i_{th} \text{ element is in class } j \\ 0, & if\ i_{th} \text{ element is not in class } j \end{cases}$$

In this class symbol C, the input vector Xi is used to create the one-hot encoded target vector Yi, and the probability of the ith element in class j is represented by pij. The most effective method for solving object identification and number detection issues at the moment is the Artificial Neural Network (ANN), of which Convolutional Neural Networks (CNNs), abbreviated as "ConvNet," is a subset of ANNs [14]. So far, several models have been built for CNNs, however in this study, we will only be covering three architectures. One, AlexNet In comparison to more conventional approaches that came before AlexNet, this 2012-built architecture is the first deep network to achieve statistically significant object classification accuracy in the ImageNet dataset. As shown in Figure 2, this network is composed of 5 convolutional layers, followed by 3 fully linked layers [15]. Two, VGG16 The VGG group at Oxford developed this design in 2013. By substituting a few 3x3 kernel filters for the 11 and 5 big kernel filters in the first and second convolutional layers, respectively, VGG was designed to enhance the AlexNet architecture. In order to learn more complicated features with a given receptive field, it is preferable to utilize small-sized stacking kernels rather than large-size kernels [12]. This is because adding several non-linear layers deepens the network. Figure 3 shows it in contrast.

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT
ISSN 2454-9940
www.ijasem.org
Vol 19, Issue 2, 2025

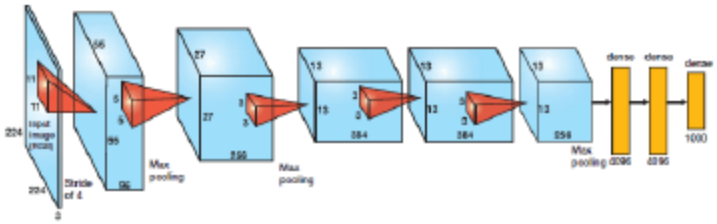


Fig. 2. Network architecture of AlexNet.

Fig. 3. Network architecture of VGG16.

3) GoogLeNet / Inception

Although VGG employs a GPU (Graphic Processing Unit), its use still necessitates a lot of processing, despite its exceptional accuracy on the ImageNet dataset. It was created in 2014. The wide breadth of the convolutional layer utilized has made this inefficient. GoogLeNet is based on the premise that, due to their high degree of correlation, the majority of activations in deep networks are either unnecessary (with a zero value) or excessive. Consequently, all 512 output channels will not have links between each other [16] due to sparse connections between activations in the most effective deep network design. The Inception module, developed by GoogLeNet, resembles a narrow CNN in concept and construction (see Fig. 4). As shown before, the effectiveness of the neurons is low, therefore the kernel size's width/number of convolutional filters is maintained minimal. Additionally, this program captures features at several scales using convolution of different sizes (5x5, 3x3, 1x1). four) ResNet From what we have seen so far, it seems that increasing the network's layer depth—so long as over-fitting can be maintained—is the key to better accuracy. Adding more layers to the deep network, however, will not increase its performance. The issue of vanishing gradients makes deep networks challenging to implement; when gradients are repropagated to the preceding layer, the gradient might become very tiny due to repetitive repetition. This means that performance hits a ceiling or starts to drop sharply as the network expands. An "identity shortcut connection" that traverses one or more levels [17] is at the heart of ResNet's (Residual Network) 2015 design, as seen in Figure 5.

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT
ISSN 2454-9940
www.ijasem.org
Vol 19, Issue 2, 2025

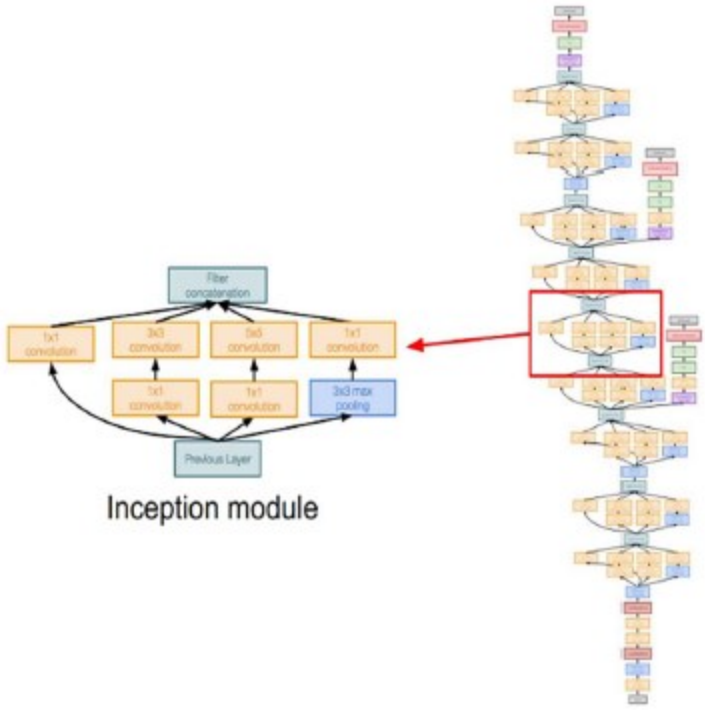


Fig. 4. Network architecture of GoogLeNet/Inception.

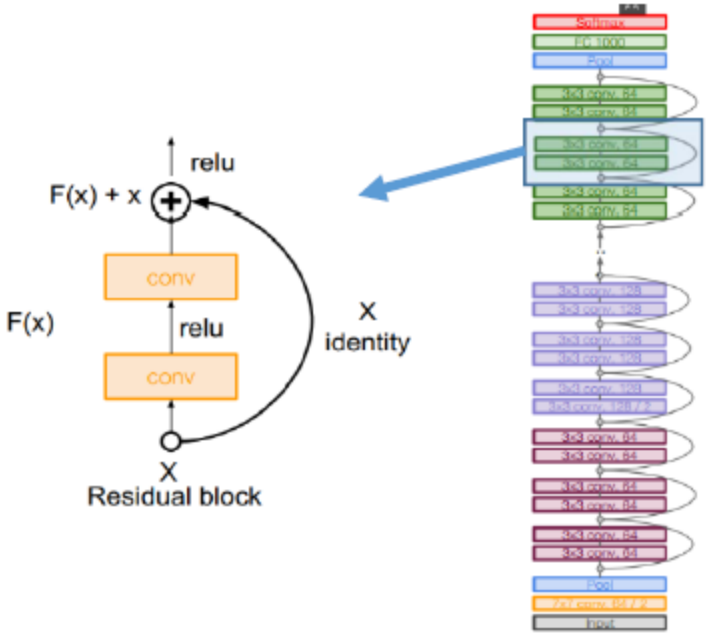


Fig. 5. Network architecture of ResNet.

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT
ISSN 2454-9940
www.ijasem.org
Vol 19, Issue 2, 2025

Fig. 6. Example of used dataset.

EXPERIMENT AND RESULTS

The author of this study will use a Convolutional Neural Network (CNN) to analyze 75 distinct picture categorization classes. After that, they will distinguish between methods that use various architectures. The author conducts tests on a machine that meets the following requirements: Intel(R) Xeon(R) dual-core processor with a 2.30 GHz model, 13 GB of RAM, and a Tesla K80 graphics processing unit (GPU) with a PCI bus id of 0000:00:04.0 and a computing capability of 3.7. Version 2.3.1 of Keras The "Recognizing Disguised Faces" dataset is used by the author. This dataset contains seventy-five images of human faces adorned with various forms of disguise, such as bandanas, maskers, false mustaches, fake beards, spectacles, etc. The data we used for assessment or testing is identical to the data we utilized for validation, and we then segment validation according to the real face. In this study, we focus only on training using grayscale images. A. Approach to Instruction and Evaluation We employ the following pre-trained models in this study: VGG16, VGG19, ResNet50, ResNet152 v2, InceptionV3, and Inception-ResNet V2. For the base layer, we build up a basic Sequential Model Architecture. Then, in order to facilitate Transfer Learning, we add a Pre-Trained weight "ImageNet" to our input model. On the first attempt, we use the same procedure as our previous input model and utilize VGG16. All trained models undergo 30 epochs in each setting. Except for the InceptionV3 Model, which is detailed in Table I.B., the parameters of the other models change once the final four layers are unfrozen. We use a data preparation approach to forestall overfitting before we begin training the model. All images are resized to 224 × 224 pixels, then flipped and rotated according to the protocol. Figure 7 displays the preprocessed picture result. For training, we use two configurations: first, freezing all CNN model layers (setup 1), and second, unfreezing the final four layers (setup 2). We utilize the weight from the previous training to train the CNN model; in this example, it's from Keras training on the ImageNet dataset, which has many labelled pictures, meaning that we freeze all layers. As illustrated in Figure 8, the second configuration involves freezing all levels but the final four. Then, to improve accuracy on the dataset we used, we train the last four layers, a technique usually referred to as Transfer Learning. As shown in Figure 9, this training yielded the highest possible accuracy. C. Turning off all CNN model layers (Setup 1) We are interested in learning the early influence of the pre-trained weight on the CNN Model (ImageNet) in this training scenario. Table II displays the output of the Keras training (or fitting) procedure. D. Defrosting the last four layers of the convolutional neural network model (setup 2) To begin, we use a pre-trained convolutional neural network (CNN) model and freeze everything except the final four levels; this allows us to train just those layers. Without regard to the layer type, we apply it to the whole Model. Following that, we are re-fitting our model. Table I displays the outcome.

Fig. 7. Output image after preprocessing.

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT
ISSN 2454-9940
www.ijasem.org
Vol 19, Issue 2, 2025

TABLE I. TRAINABLE PARAMETER OF CNN MODEL

CNN Model	Freezing all layer (parameter)	Unfreezing last 4 layer (parameter)
VGG16	8,466,507	15,545,931
VGG19	8,466,507	15,545,931
ResNet50	102,838,347	103,893,067
ResNet152 v2	102,838,347	103,893,067
Inception v3	52,506,699	52,506,699
Inception-ResNet v2	100,741,195	103,937,611

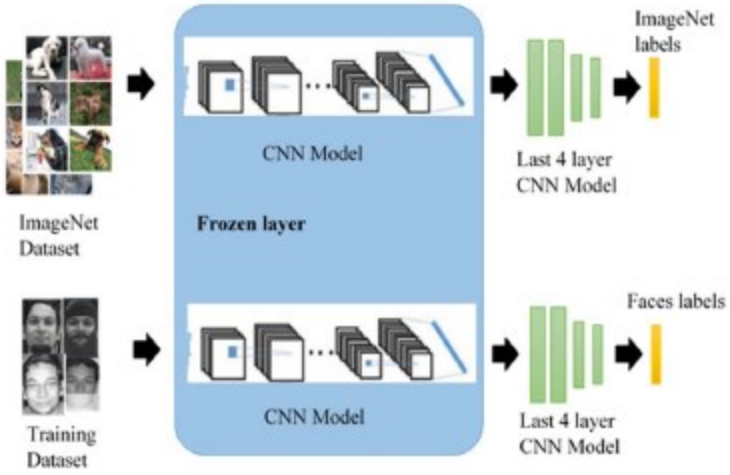


Fig. 8. Training model for transfer learning.

TABLE II. MEASURE PERFORMANCE ON FREEZE ALL LAYER

CNN Model	Loss Value	Accuracy (%)	Validation Loss Value	Validation accuracy (%)
VGG16	3.41	20.49	3.14	45.24
VGG19	3.91	11.93	3.81	26.19
ResNet50	1.74	53.6	4.66	1.6
ResNet152 v2	1.57	61.0	6.16	1.6
Inception v3	3.51	16.3	3.43	3.9
Inception-ResNet v2	4.01	7.3	4.15	1.6

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT
ISSN 2454-9940
www.ijasem.org
Vol 19, Issue 2, 2025

Fig. 9. Accuracy on ResNet152 v2.

TABLE III. MEASURE PERFORMANCE ON UNFREEZING LAST 4 LAYER

CNN Model	Loss Value	Accuracy (%)	Validation Loss Value	Validation accuracy (%)
VGG16	2.17	43.12	0.98	80.16
VGG19	2.39	37.92	1.39	67.47
ResNet50	1.58	58.0	4.86	1.6
ResNet152 v2	1.44	62.0	6.00	4.7
Inception v3	3.56	13.0	3.54	6.3
Inception-ResNet v2	3.51	15.0	4.13	5.5

Based on the data in the table, it is clear that this training configuration leads to lower loss values and more accurate results in the VGG16 Model in particular. However, in the second configuration, Inception V3, the model degrades. Table IV shows the different performance for easier understanding. In the table, a negative number indicates that the value has decreased after the second setup, and the opposite is also true. Results from the second configuration in the training and validation set show that the VGG16 Model achieves much improved accuracy, as shown in Table IV, Figure 10, and Figure 11. The ResNet Model is an exception; it performs well on the training set but fails miserably on the validation phase. Then, during testing, we make predictions using the train and validation sets' face data using Keras's predict command. Additionally, the outcome is almost identical to the training outcome improvement. ables V and VI show the outcome. The second configuration of the VGG16 Model is shown in Figure 12, which is a running forecast.

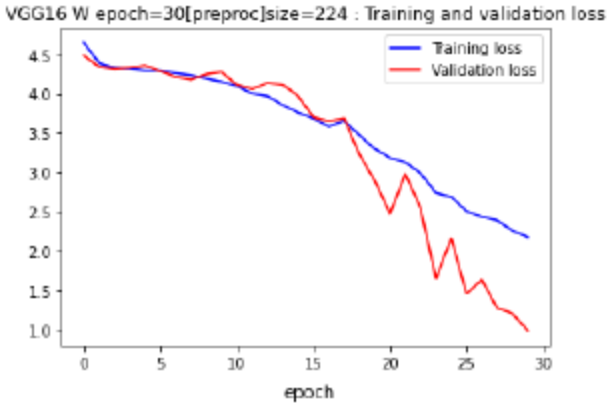
INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT
ISSN 2454-9940
www.ijasem.org
Vol 19, Issue 2, 2025

TABLE IV. DIFFERENCE PERFORMANCE AFTER UNFREEZING LAST 4 LAYER

CNN Model	Performance Setup 2 – Performance Setup 1			
	Loss Value	*Accuracy (%)*	*Validation Loss Value*	*Validation accuracy (%)*
VGG16	-1.24	22.63	-2.16	34.92
VGG19	-1.52	25.99	-2.42	41.28
ResNet50	-0.16	4.4	0.20	0.0
ResNet152 v2	-0.13	1.0	-0.16	3.1
Inception v3	0.05	-3.3	0.11	2.4
Inception-ResNet v2	-0.50	7.7	-0.02	3.9

TABLE V. TESTING PERFORMANCE ON TRAIN SET

CNN Model	Train set	
	Accuracy on setup 1 (%)	*Accuracy on Setup 2 (%)*
VGG16	57.34%	75.69%
VGG19	29.51%	65.75%
ResNet50	1.53%	1.53%
ResNet152 v2	4.43%	7.95%
Inception v3	3.67%	3.98%
Inception-ResNet v2	1.99%	3.98%



INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT
ISSN 2454-9940
www.ijasem.org
Vol 19, Issue 2, 2025

Fig. 10. Loss and accuracy graph on VGG16 model on setup 2.

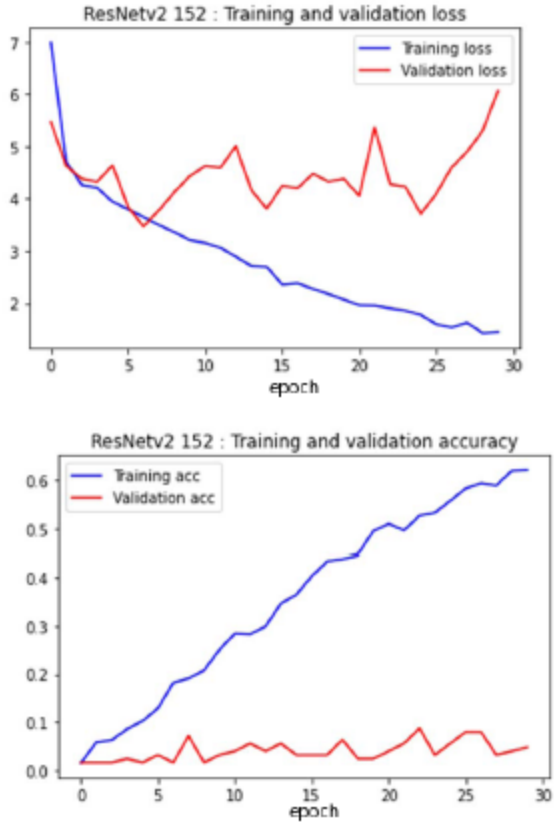


Fig. 11. Loss and accuracy graph on ResNet152 v2 model on setup 2.

TABLE VI. TESTING PERFORMANCE ON VALIDATION SET

CNN Model	Validation set	
	Accuracy on setup 1 (%)	*Accuracy on Setup 2* (%)
VGG16	45.24	80.16
VGG19	26.19	67.46
ResNet50	1.59	0.00
ResNet152 v2	1.59	4.76
Inception v3	3.97	5.56
Inception-ResNet v2	1.59	5.56

```
Found 126 images belonging to 75 classes.
Found 654 images belonging to 75 classes.
66/65 [==============================] - 4s 67ms/step
13/12 [==============================] - 1s 65ms/step
---------- VGG16 W epoch=30[preproc]size=224 -----------------
---------------- predict on train set ----------------
Total photos success prediction  = 495 of 654
Total photos fail prediction     = 159 of 654
accuration = 75.688 %
---------------- predict on validation set ------------
Total photos success prediction  = 101 of 126
Total photos fail prediction     = 25 of 126
accuration = 80.159 %
```

Fig. 12. Running predict command example on CNN Model VGG16.

CONCLUSIONS

The purpose of this research is to provide a comparison of six well-known convolutional neural network (CNN) models for face verification using the "Recognizing Disguised Faces" datasets, and to draw conclusions on the usefulness of Transfer Learning in this context. The training results demonstrate that the VGG model achieves a balanced training and validation accuracy; nevertheless, in the train set, the ResNet152 v2 Model outperforms VGG. However, when compared to other CNN models, the VGG model performed the best in the tests. Finally, we draw the conclusion that the ImageNet weight may be used for VGG Model Face Recognition via Transfer Learning. One of the key reasons why Deep Learning CNN has been trending recently is because of how successful this Convolutional Neural Network is.

REFERENCES

[1] A. Samuel, "Some Studies in Machine Learning Using the Game of Checkers," *IBM Journal of Research and Development,* vol. 3, no. 3,p. 210–229, 1959.
[2] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces,"*IEEE Conference on Computer Vision and Pattern Recognition,* vol.1, p. pp. 586–591, 1991.

[3] J. Yang, D. Zhang, A. F. Frangi and J. Yang, "Two dimensional PCA: a new approach to appearance-based face representation and recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 26, no. 1, p. 131–137, 2004.
[4] Y. Taigman, M. Yang, M. Ranzato and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," *CVPR,* p. 1701–1708, 2014..
[5] J. West, D. Ventura and S. Warnick, "A Theoretical Foundation for Inductive Transfer," Spring Research Presentation, 2007.
[6] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama and a. T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *ACM MM*, 2014.
[7] D. Tomé, F. Monti, L. Baroffio, L. Bondi, M. Tagliasacchi and S. Tubaro, "Deep Convolutional Neural Networks for pedestrian detection," *Signal Processing: Image Communication,* vol. 47, pp. 482-489, 2016.
[8] Z. Deng, H. Sun, S. Zhou, J. Zhao and H. Z. Lin Lei, "Multi-scale object detection in remote sensing imagery with convolutional neural networks," *ISPRS Journal of Photogrammetry and Remote Sensing,* vol. 145, no. Part A, pp. 3-22, 2018.
[9] M. Perez, S. Avila, D. Moreira, D. Moraes, V. Testoni, E. Valle, S. Goldenstein and A. Rocha, "Video pornography detection through deep learning techniques and motion information," *Neurocomputing,* vol. 230, pp. 279-293, 2017.
[10] R. D. Yogaswara and A. D. Wibawa, "Comparison of Supervised LearningImage Classification Algorithms for Food and Non-Food Objects," in *CENIM*, Surabaya, 2018.
[11] Z. Yang and R. Nevatia, "A multi-scale cascade fully convolutional network face detector," in *International Conference on Pattern Recognition (ICPR)*, Mexico, 2016.
[12] F. Chollet, *Keras,* GitHub: GitHub repository, 2015.
[13] T. I. Dhamecha, A. Nigam, R. Singh and M. Vatsa, "Disguise detection and face recognition in visible and thermalspectrums," in *IEEE International Conference on Biometrics, pp. 1–8*, 2013.
[14] Q. V. Le, J. Ngiam, Z. Chen, D. Chia, P. Koh and A. Y. Ng., "Tiled Convolutional Neural Networks," in *Neural Information Processing Systems Foundation*, 2010.
[15] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional," 2012.
[16] C. Szegedy, W. Liu, Y. Jia, P. Sermanet and S. Reed, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA , 2015.
[17] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016.