

Classification of Real and Fake Human Faces Using Deep Learning

Fatima Maher Salman and Samy S. Abu-Naser

Department of Information Technology,
Faculty of Engineering and Information Technology,
Al-Azhar University, Gaza, Palestine

Abstract: Artificial intelligence (AI), deep learning, machine learning and neural networks represent extremely exciting and powerful machine learning-based techniques used to solve many real-world problems. Artificial intelligence is the branch of computer sciences that emphasizes the development of intelligent machines, thinking and working like humans. For example, recognition, problem-solving, learning, visual perception, decision-making and planning. Deep learning is a subset of machine learning in artificial intelligence that has networks capable of learning unsupervised from data that is unstructured or unlabeled. Deep learning is a technique used to generate face detection and recognize it for real or fake by using profile images and determine the differences between them. In this study, we used deep learning techniques to generate models for Real and Fake face detection. The goal is determining a suitable way to detect real and fake faces. The model was designed and implemented, including both Dataset of images: Real and Fake faces detection through the use of Deep learning algorithms based on neural networks. We have trained dataset which consists of 9,000 images for total in 150 epochs, and got the ResNet50 model to be the best model of network architectures used with 100% training accuracy, 99.18% validation accuracy, training loss 0.0003, validation loss 0.0265, and testing accuracy 99%.

Keywords: Artificial intelligence, Deep learning, Real and Fake Face, human images

1. Introduction

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves. The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance an adjust actions accordingly.

Deep learning is a subset of machine learning in artificial intelligence that has networks capable of learning unsupervised from data that is unstructured or unlabeled. Deep learning is a technique used to generate face detection and recognize it for real or fake by using profile images and determine the differences between them.

In this study, we will be using deep learning techniques to generate model for Real and Fake face detection. The goal is determining a suitable way to detect real and fake faces.

1.1 Fake and Real Human Face

With the development of Generative Adversarial Network (GAN) computers can generate vivid face images that can easily deceive human beings. These generated fake faces will inevitably bring serious social risks, e.g., fake news, fake evidence, and pose threats to security [1].

Thus, powerful techniques to detect these fake faces are highly desirable. However, in contrast to the intensive studies in GANs, our understanding of generated faces is fairly superficial and how to detect fake faces is still an under-explored problem. Moreover, fake faces in practical scenarios are from different unknown sources, i.e. different GANs, and may undergo unknown image distortions such as down sampling, blur, noise, and JPEG compression, which makes this task even more challenging[2].

In this study, we aim to produce new insights on understanding fake faces and propose a new architecture to tackle these challenges.

1.2 Problem Statement

According to use of social media, we will encounter a fake identity from any person through using fake profile image. The fake profile image can be occurred with using image editor, face effect, or any program using to change the facial features.

Effects on the face can change facial features and it's difficult to know true identity for someone. In this application, we hope to train a model using the dataset to recognize such fake faces.

Computerized applications can use deep learning techniques to increase accuracy and efficiency in diagnosis. These include human images, image processing techniques and data analysis.

1.3 Objectives of the Study

Main objective for this study is implementation a software model used to detect and classify face to real or fake for expert-generated high-quality photo shopped face images.

Other objectives are:

- Detect fake face images rapidly.
- Get high accuracy and validation in the testing and training images.
- Reduce cost and effort of diagnosis and repetitive images.
- Implementation network architectural to find the best model with perfect result.
- Training high dataset of fake and real human face to see an accuracy of the result.

1.4 Convolutional Neural Network (CNN)

Neural network with a convolution operation instead of matrix multiplication in at least one of the layers. It's one of the main categories to do images recognition, images classifications. Technically, deep learning CNN models to train and test, each input image will pass it through a series of convolution layers with filters (Kernels), polling, fully connected layers (FC) and apply SoftMax function to classify an object with probabilistic values between 0 and 1. The below figure is a complete flow of CNN to process an input image and classifies the objects based on values [1-5].

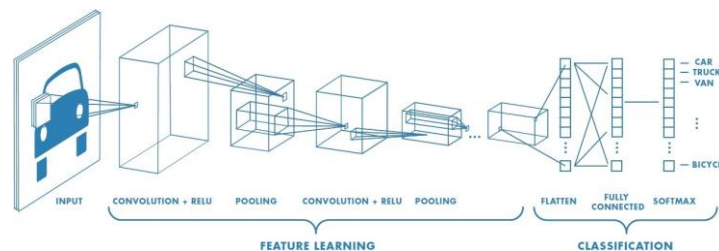


Figure 1: Convolutional Neural Network

1.4.1 Convolution Layer

Convolution is the first layer to extract features from an input image. Convolution preserves the relationship between pixels by learning image features using small squares of input data. It is a mathematical operation that takes two inputs such as image matrix and a filter or kernel [6-10].

1.4.2 Pooling Layer

Pooling layers section would reduce the number of parameters when the images are too large. Spatial pooling also called subsampling or down sampling which reduces the dimensionality of each map but retains important information. Spatial pooling can be different types: max pooling, average pooling, and sum pooling. Max pooling takes the largest element from the rectified feature map. Taking the largest element could also take the average pooling. Sum of all elements in the feature map call as sum pooling [11-14].

1.4.3 Rectified Linear Unit Layer (ReLU)

It is a type of activation function. Mathematically, it's defined as $y = \max(0, x)$. ReLU is the most commonly used function in neural networks, especially in CNNs. ReLU is linear for all positive values, and zero for all negative values. It's cheap to compute

as there is no complicated math. The model can therefore take less time to train or run. It converges faster and sparsely activated. Since ReLU is zero for all negative inputs, it's likely for any given unit to not activate at all. The advantage of not having any backpropagation errors, also for larger neural networks, the speed of building models based on ReLU is very fast. ReLU isn't without any drawbacks some of them are that ReLU is nonzero centered and is non-differentiable at zero, but differentiable anywhere else [15-19].

1.4.4 Dropout Layer

Dropout is a technique where randomly selected neurons are ignored during training. They are "dropped-out" randomly. This means that their contribution to the activation of downstream neurons is temporally removed on the forward pass and any weight updates are not applied to the neuron on the backward pass [20-24].

1.4.5 Batch Normalization Layer

It's a technique for improving the performance and stability of neural networks, and also makes more sophisticated deep learning architectures work in practice. The idea is to normalize the inputs of each layer in such a way that they have a mean output activation of zero and standard deviation of one. This is analogous to how the inputs to networks are standardized. The intention behind batch normalization is to optimize network training. It has been shown to have several benefits: networks train faster, allows higher learning rates, makes weights easier to initialize, makes more activation functions viable, simplifies the creation of deeper networks, and provides some regularization [25-30].

1.4.6 Fully Connected Layer (FC)

It is cheap way of learning non-linear combinations of the high-level features are represented by the output of the convolutional layer. The fully connected layer is learning a possibly non-linear function in that space [31-33].

1.4.7 SoftMax

It's a very interesting activation function because is not only maps our output to a [0,1] range but also maps each output in such a way that the total sum is 1. The output of SoftMax is therefore a probability distribution. SoftMax is used for multi-classification in logistic regression model [34-37].

1.4.8 Backpropagation

Backpropagation is the essence of neural net training. It is the practice of fine-tuning the weights of a neural net based on the error rate (i.e. loss) obtained in the previous epoch (i.e. iteration). Proper tuning of the weights ensures lower error rates, making the model reliable by increasing its generalization [38-42].

1.4.9 Adam optimization

Adam is derived from adaptive moment estimation. Adam is an optimization algorithm that can be used instead of the classical stochastic gradient descent procedure to update network weights iterative based in training data. The benefits of using Adam on non-convex optimization problems, as follows: straightforward to implement, computationally efficient, little memory requirements, invariant to diagonal rescale of the gradients, well suited for problems that are large in terms of data and/or parameters, appropriate for non-stationary objectives, appropriate for problems with very noisy/or sparse gradients, and hyper-parameters have intuitive interpretation and typically require little tuning [43-45].

1.5 Model Evaluation

There are two possibilities why your CNN is performing at a suboptimal performance, high variance and high bias. You would optimally arrive at a point where bias and variance are equally low. So, to improve your network, you need to determine whether you have high bias or variance. To do so, look at these three numbers: Benchmark accuracy, training accuracy, testing accuracy. In the case of high variance, the different between your benchmark accuracy and training accuracy will be relatively small compared to training accuracy vs testing accuracy. This is when your model is "overfitted" to your training data. To reduce bias, you can increase the test accuracy, through these ways [46-50]:

1. Get more data.

2. Try new model architecture, try something better.
3. Decrease number of features (you may need to do this manually).
4. Introduce regularization such as the L2 regularization.
5. Make your network shallower (less layers).
6. Use a smaller number of hidden units.
7. Make sure your testing and training dataset comes from the same distribution.

On the other hand, if your model is suffering from under fitting, you need to reduce the bias through increasing the training accuracy. To do so[51-54]:

1. Train longer.
2. Train a more complex/deeper model.
3. Obtain more features.
4. Decrease regularization.
5. Data augmentation.

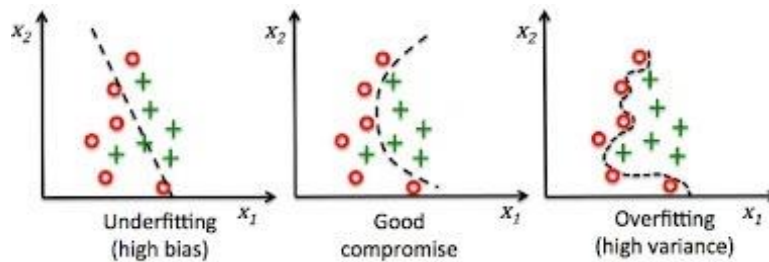


Figure 2: Performance Estimation

3.1 Previous Studies

Many research papers have been published to use artificial intelligence, expert systems, and neural networks to improve the detection of fake face. Recently, social media have begun to introduce artificial intelligence systems and applications in some disciplines to increase the accuracy of profile images detection.

Many methods and models have been introduced that have contributed to increased detection of the fake face in an efficiency.

In one of paper, the researcher has been evaluated the generalizability of the fake face detection methods through a series of studies to benchmark the detection accuracy. To this extent, they have collected a new database of more than 53,000 images, from 150 videos, originating from multiple sources of digitally generated fakes including Computer Graphics Image (CGI) generation and many tempering-based approaches. Also, they have included images (with more than 3,200) from the predominantly used Swap-Face application. Extensive experiments are carried out using both texture-based handcrafted detection methods and deep learning-based detection methods to find the suitability of detection methods [72]. **Comment on the previous study:** the difference are they have aim measurement the evaluation of generalizability on existing fake face detection techniques. So they focused to create a new database which they refer to as Fake Face in the Wild (FFW) dataset by using texture-based methods – Local Binary Patterns (LBP) with support Vector Machine (SVM) and a set of CNN architectures like AlexNet, VGG19, ResNet50, Xception, and GoogleNet/Inceptionv3

In other paper, the researcher has been proposed neural network-based classifier to detect fake human faces created by machines and humans, so they have been using ensemble methods to detect Generative Adversarial Networks (GANS) created fake images and employ pre-processing techniques to improve fake face image detection created by humans. In this study, the researcher has been focused on image contents for classification and do not use meta-data of images. The result can be effectively detected both GANS-created images and human-created fake images with 94% and 74.9% AUROC (Area Under Receiver Operating Characteristic) Score [73]. **Comment on the previous study:** we difference from this study by using large dataset of real and fake image has several changes of facial features and training it more of model in Network Architectures and get high percentage of accuracy and validation against this study.

Another paper, the researcher has been proposed a multi-feature fusion scheme that combines dynamic and static joint analysis to detect fake face attacks. According to can be easily detected the texture differences between the real and the fake faces, Local

Binary Pattern (LBP) texture operators and optical flow algorithms are often merged. Also, in the traditional optical flow algorithm is also modified by applying the multi-faction feature super-position method, which reduces the noise of the image. In the pyramid model, image processing is performed in each layer by using block calculations that form multiple block images. The features of the image are obtained via two fused algorithms (MOLF), which are then trained and tested separately by an SVM classifier. Experimental results show that this method can improve detection accuracy while also reducing computational complexity. In this paper, the researcher has been used the CASIA, PRINT-ATTACK, and REPLAY-ATTACK database to compare the various LBP (Local Binary Patter) algorithms that incorporate optical flow and fusion algorithms [74]. **Comment on the previous study:** this study focuses on photo attack, video attack, image with real eye attack, and the total three-way attack, and it is using difference model to detect the fake face attack and get the high accuracy 96.543%, also using MATLAB to implementation the result.

Generative Adversarial Networks (GANS) enable creating natural-looking human faces. Therefore, fake images can cause many potential problems, as they can be misused to abuse information, hurt people and generate fake identification. The researcher in this paper has been proposed an image forensic platform using neural networks, FakeFaceDetect, to detect various fake face images. In particular, the researcher has been focused on detecting fake images automatically created from GANS as well as manually created by humans. Also, they have been assumed a strong adversary who can arbitrarily change and remove metadata of the original images. And they have been demonstrated that FakeFaceDetect achieves high accuracy in detecting fake face images created by humans and GANS [69]. **Comment on the previous study:** this study, the researcher using FakeFaceDetect without determine the percentage of accuracy in detecting fake face images.

OpenCV is one library using deep learning to perform face recognition on a dataset of our faces. Deep learning and OpenCV have been accomplished some of tasks such as, detect faces, compute 1280-d face embeddings to quantify a face, train a Support Vector Machine (SVM) on top of the embeddings, and recognize faces in images and video streams. Deep learning has been applied by two key steps: to apply face detection, which detects the presence and location of a face in an image, but does not identify it, and to extract the 128-d feature vectors that quantify each face in an image [75]. **Comment on the previous study:** this study, the researcher using OpenCV, Python language and using SVM classifier and implementation the deep learning in his study. The differences are using other model to detect face image and not determine the percentage of accuracy, but the author got high accuracy of images.

Binary of Auxiliary Supervision model in learning deep for face anti-spoofing. The author has been focused for the importance of auxiliary supervision to guide the learning toward discriminative and generalizable cues. A CNN-RNN model has been learned to estimate the face depth with pixel-wise supervision, and to estimate rPPG signals with sequence-wise supervision. The estimated depth and rPPG are fused to distinguish live vs. spoof faces. According all this the authors has been introducing a new face anti-spoofing database the covers a large range of illumination, subject, and pose variations. The result of this model has been achieved the state- of-the-art results on both intra- and cross-database testing [76]. **Comment on the previous study:** the researcher implementation different model using learning deep for face anti-spoofing, across testing and training, the researcher gets 94.2% in the last model on Oulu protocols.

In other paper, the authors have been development of masked fake face detection methods are necessary to guarantee a successful face verification system. They have been presented an albedo-3 method for detecting a masked fake face in a user-cooperative environment, where most practical face verification systems and employed. In a user-cooperative environment, the distance between a person to be verified and the camera, and the person's head pose, are roughly fixed. They have been showed that computation of albedo can be simplified to the measurement of radiance for a given illumination. This enables us to distinguish between the facial skins and mask materials just by measuring radiance, i.e., gray values in the image [77]. **Comment on the previous study:** the research using fisher's linear discriminant by radiance measurements and achieved 97.78% accuracy in fake face detection

4. Methodology

My Proposed methodology includes gathering the dataset, identifying the tools and language to be used, preprocessing the images in the dataset, data augmentation, and construction of the model architecture, compiling the model, training and validating the model.

4.1 Dataset

The data set in this study consists of 9,000 human faces of real-fake images. The numbers of images in this dataset are classified as follows: 1750 images that were real human faces, 1750 of fake-mid, 1750 of fake-hard and 1750 images of fake-easy. The dataset of real-fake human faces images was collected from Kaggle depository website.

Table 1 : Dataset division for training, validation, and testing

Real-Fake	Training Samples	Validation Samples	Testing Samples
Real	1750	250	250
fake-mid	1750	250	250
fake-hard	1750	250	250
fake-easy	1750	250	250
Total	7000	1000	1000

4.2 Language and tool used

Python language was used, which is an interpreted, high-level, general-purpose programming language. It was created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects [55-56].

Python is dynamically typed, and garbage collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library [57-60].

Several tools were used, the most important of which is Google Colab to write python codes, a research tool for teaching and searching for a deep learning machine, an easy to use and does not require any preparation for use. Google Colab is characterized by its speed in performance because it has very fast processors of type (GPU) [61-64].

Google Colab is a free-to-use research project that can store and read all notebooks directly from Google Drive.

4.4 Image format

Dataset was collected from a set of real-fake human faces Images for detecting whether an image is real or fake (JPG) format, in order to fit well with the model used to give the desired results.

4.5 Preprocessing

The first thing in the data preprocessing was to resize the real-fake human faces Images as the images were of various sizes, the images were resized to 256 by 256 Pixels, this image size collide with a balance between providing a high enough resolution for real-fake human faces detection by the model and efficient training. All images were normalized to ImageNet standards.

Then the images collection has been categorized into four types, uploaded to a Google Drive account and verified to be properly and accurately uploaded using Python code in the Google Colab environment [65-70].

4.6 Data augmentation

Generating more data usually means that the model will be more robust and prevent overfitting.

Having a large dataset is crucial for the performance of the deep learning model. However, I improved the performance of the model by augmenting the images that I already have without collecting new images. Deep learning frameworks usually have built-in library for data augmentation utilities; I utilized five augmentation strategies to generate new training sets, (Rotation, width shift, height shift, horizontal flip, and vertical flip) [71-73].

Rotation augmentations are done by rotating the image right or left on an axis between 1° and 360°. The safety of rotation augmentations is heavily determined by the rotation degree parameter. Shifting and flipping images are a very useful transformation to encapsulating more details about objects of interest.



Figure 3: Original Real-Fake Image



Figure 4: Real-Fake Image is rotated by 20 degrees



Figure 5: Real-Fake Image after width shift



Figure 6: Real-Fake Image after height shift

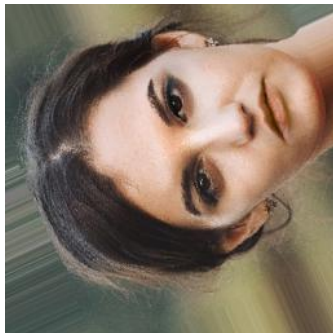


Figure 7: Real-Fake Image flipped horizontally

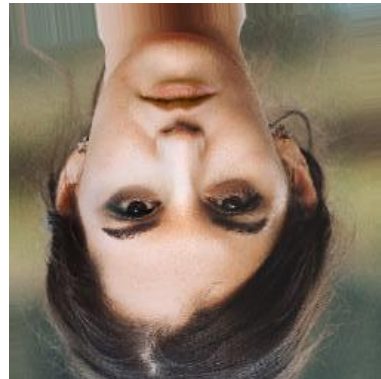


Figure 8: Real-Fake Image flipped vertically

4.7 Network Architecture

We have trained our Real-Fake dataset using a model that we created from scratch and other four pre-trained models for deep learning: VGG16, ResNet50, MobileNet, and InceptionV3.

4.8 Training and Validating the Models

4.8.1 Proposed Model

We created a model from scratch with 8 convolutional layers followed by a fully connected hidden layer (as shown in Figure

13). Output layer uses s14oftmax activation as it has to output the probability for each of the classes; to optimize the network Adam optimizer and a learning rate of 0.0001 were used.

During the model training, the model will iterate over batches of the training dataset, each of size batch_size. For each batch, gradients will be computed, and updates will be made to the weights of the network automatically. One iteration over all of the training dataset is referred to as an epoch. Training is usually run until the loss converges to a small constant.

We used checkpoint in the model to save the best validation accuracy. This is useful because the network might start overfitting after a certain number of epochs. This feature was implemented via the callback feature of the library Keras. Callback is a set of functions that is applied at given stages of training procedure like end of an epoch of training. Keras provides built-in function for both learning rate scheduling and model check-pointing.

I trained and validated my model and I got a training accuracy of 100% and validation accuracy of 95.21%.

```

from keras import layers
from keras import models
from keras import backend as K
model = models.Sequential()
model.add(layers.Conv2D(32, (3, 3), activation='relu',
    input_shape=(256, 256, 3)))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(128, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(256, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(256, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(512, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Flatten())
model.add(layers.Dense(512, activation='relu'))
model.add(layers.Dense(4, activation='softmax'))
    
```

Figure 9: The python code of my model

Figure 14 shows the learning curve of the network during training and validation. It is seen that the validation accuracy and training accuracy were increasing, and it reached to 100 % training accuracy and 95.21% validation accuracy.

Figure 15 shows the loss curve of the network during training and validation. It is seen that the validation loss and training loss were decreasing with the number of iterations, that's great; it proved that the model works well.

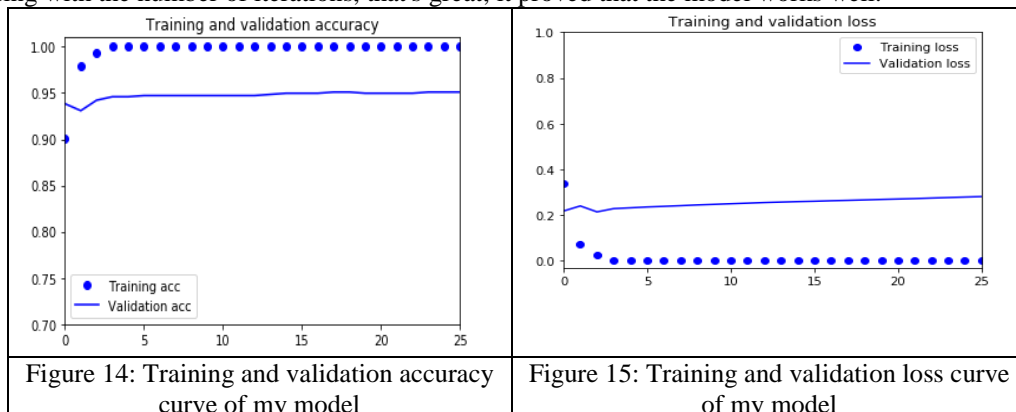


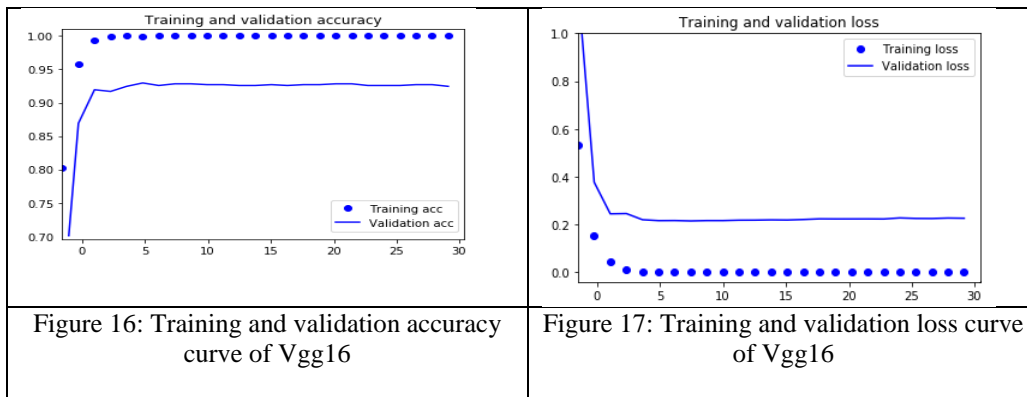
Figure 14: Training and validation accuracy curve of my model

Figure 15: Training and validation loss curve of my model

4.8.2 VGG16 model

We used a pre-trained model called VGG16 network with 16 convolutional layers followed by a fully connected hidden layer, also used dropout layers in between, dropout regularizes the networks to prevent the network from overfitting, and all layers have ReLU activation function except the output layer.

Figure 16 illustrates the accuracy of both training and validation of VGG16 model of the Real-Fake dataset. Figure 17 shows the loss of both training and validation of VGG16 model of the Real-Fake dataset.

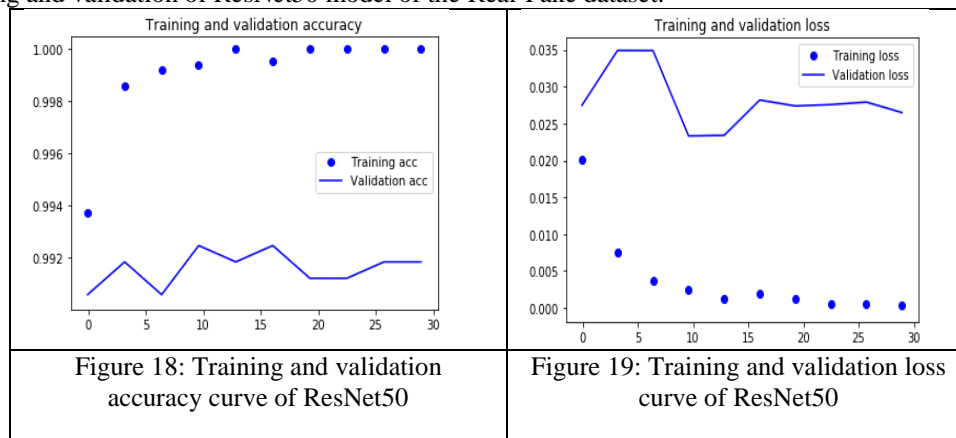


4.8.3 ResNet50 model

We used a pre-trained model called ResNet50 network with 50 convolutional layers followed by a fully connected hidden layer, also used dropout layers in between, dropout regularizes the networks to prevent the network from overfitting, and all layers have ReLU activations function.

We trained the Real-Fake dataset using the pre-trained ResNet50 model and the training accuracy reached 100% and validation accuracy reached 99.18%.

Figure 18 illustrates the accuracy of both training and validation of ResNet50 model of the Real-Fake dataset. Figure 19 shows the loss of both training and validation of ResNet50 model of the Real-Fake dataset.



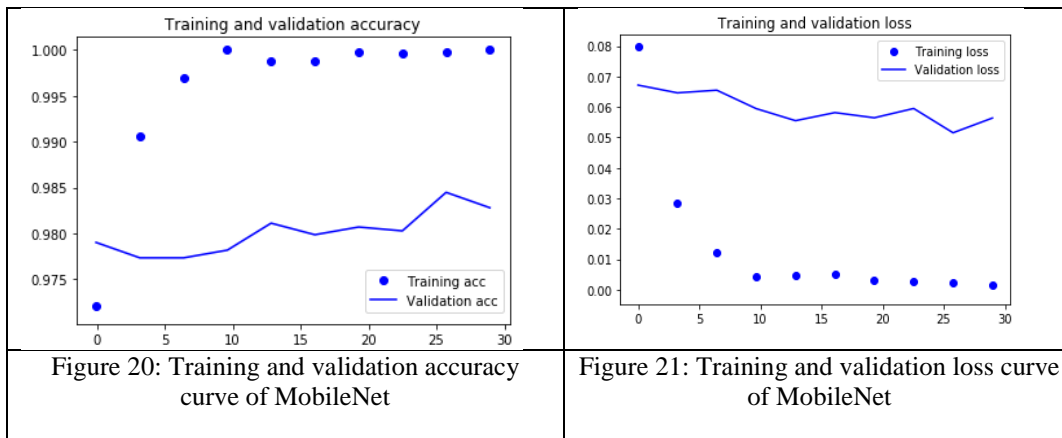
4.8.4 MobileNet model

I used a pre-trained model called MobileNet network with 28 convolutional layers followed by a fully connected hidden layer, also used dropout layers in between, dropout regularizes the networks to prevent the network from overfitting, all layers have ReLU activation function.

We trained the Real-Fake dataset using the pre-trained MobileNet model and the training accuracy reached 100% and validation accuracy reached 98.14%.

Figure 20 illustrates the accuracy of both training and validation of MobileNet model of the Real-Fake dataset. Figure 21

shows the loss of both training and validation of MobileNet model of the Real-Fake dataset.

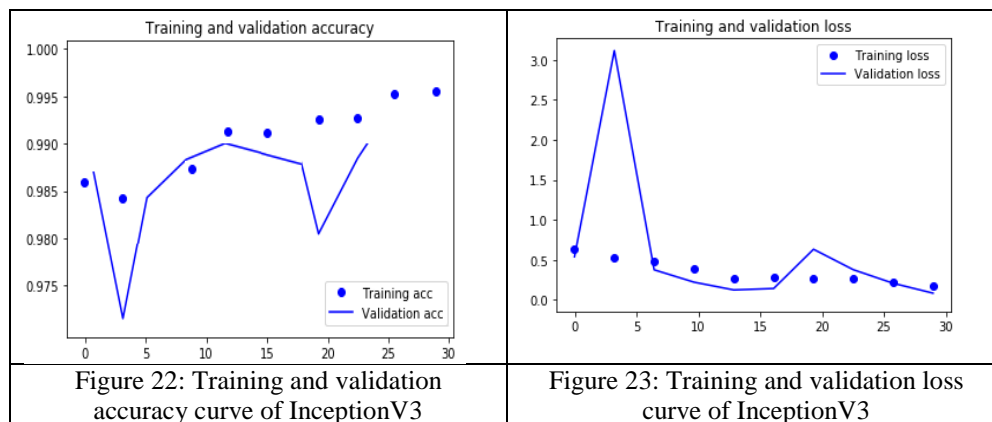


4.8.5 InceptionV3 model

I used a pre-trained model called InceptionV3 with 28 convolutional layers followed by a fully connected hidden layer, also used dropout layers in between, dropout regularizes the networks to prevent the network from overfitting, all layers have ReLU activation function.

We trained the Real-Fake dataset using the pre-trained InceptionV3 model and the training accuracy reached 99.77% and validation accuracy reached 99.03%.

Figure 22 illustrates the accuracy of both training and validation of InceptionV3 model of the Real-Fake dataset. Figure 23 shows the loss of both training and validation of InceptionV3 model of the Real-Fake dataset.



5. Evaluation of the Model

5.1 Data Set for testing the model

Testing dataset organized into one folder (Real-Fake-test) and contains 1000 of Real-Fake dataset, (JPG) format, different from the images that used in original dataset for training; they are images of four classifications of real, fake-mid, fake-hard, fake-easy distributed as in the following table:

Table 2: Distribution of Images in test dataset

Real-Fake - Categories	Number of testing images	Image size
real	250	256 x 256 pixels
fake-mid	250	256 x 256 pixels

fake-hard	250	256 x 256 pixels
fake-easy	250	256 x 256 pixels
Total	1000	

Figure 24 and 25: Shows samples of Real and Fake dataset images used for testing the 5 models.



Figure 24: Samples of the Fake images in the test dataset

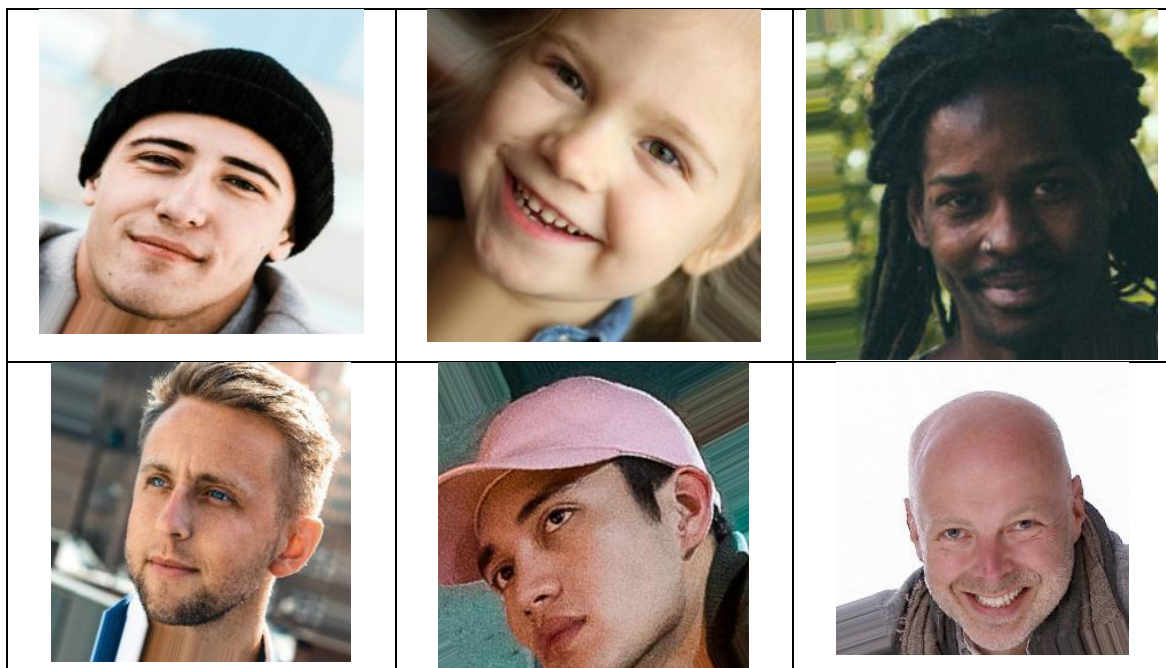


Figure 25: Samples of the Real images in the test dataset

5.2 Testing the model

After training and evaluating the model on the validation, the network is tested using 1000 Real-Fake dataset images, among them, 250 real, 250 fake-mid, 250 fake-hard, 250 fake-easy.

Testing the models was done through loading the test images and predicting their classes using the `model.predict_classes()` function, probabilities of each image belonging to a specific class were calculated, by the following classification: real, fake-mid, fake-hard, fake-easy.

Note the classification rates of network during testing, it was a surprise, the probability of results in the classification for our model, VGG16, ResNet50, MobileNet, and InceptionV3 were 95%, 93%, 99%, 98%, 99% correct respectively for all test images.

5.3 Result and Discussion

We trained our custom model on the training dataset using 9,000 images for a total of 150 epochs, the results were as follows:

Table 3: Analysis of the model used in the training, validation and testing

Criterion	Proposed Model	VGG16	ResNet50	MobileNet	InceptionV3
Training Accuracy	100%	100%	100%	100%	99.77%
Validation Accuracy	95.21%	93.05%	99.18%	98.14%	99.03%
Training loss	0.000005	0.0002	0.0003	0.0016	0.0075
Validation loss	0.2906	0.2237	0.0265	0.0563	0.0269
Testing Accuracy	95%	93%	99%	98%	99%

Based on the previous table of the results, InceptionV3 and ResNet50 were the best models then come MobileNet model in 3rd place, then my model 4th place which we were created from scratch with 95.21%. VGG16 was ranked in the 5th place.

This study presents a deep learning approach for detecting if an image is real or faked and if faked is it easy, mid, or hard fake.

We accomplished this using the proposed model and other four different pre-trained models: VGG16, ResNet50, MobileNet, and InceptionV3 architectures, these model were able to correctly predict Real-Fake images with a balanced accuracy approximately 99% on the validation and the test set after only training for 150 epochs, due to the techniques such as Adam optimization, data augmentation, dropout and others, it was possible to enhance the accuracy of model without sacrificing training efficiency, because one of the problems with machine learning, including deep learning, is overfitting. Overfitting occurs when the trained model does not generalize well to unseen cases but fits the training data well. This becomes more apparent when the training sample size is small.

Assessment of the plot training can be used to assess the possibility of overfitting. From the curve, it is apparent that the data loss is similar for both validation and training datasets. If there were overfitting, the loss on the training data would be much greater than that of the validation data. In addition, for this reason, the cases were split three groups (training, validation, and test).

It is conceivable that the use of larger training datasets, additional image augmentation methods, and additional machine learning approaches with more ensembles could improve this result.

6.1 Conclusion

The advance of image capability and image generation techniques have now provided the ability to create security-less and convincing fake face images. The challenging nature of data, whether in terms of visual perception or algorithm discovery, is present in recent works. The main problem that has not yet been taken into consideration is the assessment the overall capability of existing fake face detection techniques. In order to answer the question of generalizability, in this work, we have trained dataset using 9,000 images for total of 150 epochs, and get the ResNet50 model is the best model of network architectures with 100% training accuracy, 99.18% validation accuracy, training loss 0.0003, validation loss 0.0265, and testing accuracy 99%. It was development by using Google Colab and Python language that supported for a deep learning machine. Thus, advances in real-fake face detection technology require that the correct validation of datasets and accuracy be included in all future research as a condition for publication.

References

1. Al Barsh, Y. I., et al. (2020). "MPG Prediction Using Artificial Neural Network." International Journal of Academic Information Systems Research (IJAISR) 4(11): 7-16.
2. Alajrami, E., et al. (2019). "Blood Donation Prediction using Artificial Neural Network." International Journal of Academic Engineering Research (IJAER) 3(10): 1-7.
3. Alajrami, E., et al. (2020). "Handwritten Signature Verification using Deep Learning." International Journal of Academic Multidisciplinary Research (IJAMR) 3(12): 39-44.
4. Al-Araj, R. S. A., et al. (2020). "Classification of Animal Species Using Neural Network." International Journal of Academic Engineering Research (IJAER) 4(10): 23-31.
5. Al-Atrash, Y. E., et al. (2020). "Modeling Cognitive Development of the Balance Scale Task Using ANN." International Journal of Academic Information Systems Research (IJAISR) 4(9): 74-81.
6. Alghoul, A., et al. (2018). "Email Classification Using Artificial Neural Network." International Journal of Academic Engineering Research (IJAER) 2(11): 8-14.
7. Abu Nada, A. M., et al. (2020). "Age and Gender Prediction and Validation through Single User Images Using CNN." International Journal of Academic Engineering Research (IJAER) 4(8): 21-24.
8. Abu Nada, A. M., et al. (2020). "Arabic Text Summarization Using AraBERT Model Using Extractive Text Summarization Approach." International Journal of Academic Information Systems Research (IJAISR) 4(8): 6-9.
9. Abu-Sager, M. M., et al. (2020). "Type of Grapefruit Classification Using Deep Learning." International Journal of Academic Information Systems Research (IJAISR) 4(1): 1-5.
10. Abana, M., et al. (2018). "Artificial Neural Network for Forecasting Car Mileage per Gallon in the City." International Journal of Advanced Science and Technology 124: 51-59.
11. Al-Araj, R. S. A., et al. (2020). "Classification of Animal Species Using Neural Network." International Journal of Academic Engineering Research (IJAER) 4(10): 23-31.
12. Al-Atrash, Y. E., et al. (2020). "Modeling Cognitive Development of the Balance Scale Task Using ANN." International Journal of Academic Information Systems Research (IJAISR) 4(9): 74-81.
13. Alghoul, A., et al. (2018). "Email Classification Using Artificial Neural Network." International Journal of Academic Engineering Research (IJAER) 2(11): 8-14.
14. Al-Kahlout, M. M., et al. (2020). "Neural Network Approach to Predict Forest Fires using Meteorological Data." International Journal of Academic Engineering Research (IJAER) 4(9): 68-72.
15. Alkronz, E. S., et al. (2019). "Prediction of Whether Mushroom is Edible or Poisonous Using Back-propagation Neural Network." International Journal of Academic and Applied Research (IJAAR) 3(2): 1-8.
16. Al-Madhoun, O. S. E.-D., et al. (2020). "Low Birth Weight Prediction Using JNN." International Journal of Academic Health and Medical Research (IJAHMR) 4(11): 8-14.
17. Al-Massri, R., et al. (2018). "Classification Prediction of SBRCTs Cancers Using Artificial Neural Network." International Journal of Academic Engineering Research (IJAER) 2(11): 1-7.
18. Al-Mobayed, A. A., et al. (2020). "Artificial Neural Network for Predicting Car Performance Using JNN." International Journal of Engineering and Information Systems (IJEIS) 4(9): 139-145.
19. Al-Mubayyed, O. M., et al. (2019). "Predicting Overall Car Performance Using Artificial Neural Network." International Journal of Academic and Applied Research (IJAAR) 3(1): 1-5.
20. Alshawwa, I. A., et al. (2020). "Analyzing Types of Cherry Using Deep Learning." International Journal of Academic Engineering Research (IJAER) 4(1): 1-5.
21. Al-Shawwa, M., et al. (2018). "Predicting Temperature and Humidity in the Surrounding Environment Using Artificial Neural Network." International Journal of Academic Pedagogical Research (IJAPR) 2(9): 1-6.
22. Ashqar, B. A., et al. (2019). "Plant Seedlings Classification Using Deep Learning." International Journal of Academic Information Systems Research (IJAISR) 3(1): 7-14.
23. Bakr, M. A. H. A., et al. (2020). "Breast Cancer Prediction using JNN." International Journal of Academic Information Systems Research (IJAISR) 4(10): 1-8.
24. Barhoom, A. M., et al. (2019). "Predicting Titanic Survivors using Artificial Neural Network." International Journal of Academic Engineering Research (IJAER) 3(9): 8-12.
25. Belbeisi, H. Z., et al. (2020). "Effect of Oxygen Consumption of Thylakoid Membranes (Chloroplasts) From Spinach after Inhibition Using JNN." International Journal of Academic Health and Medical Research (IJAHMR) 4(11): 1-7.
26. Dallfa, M. A., et al. (2019). "Tic-Tac-Toe Learning Using Artificial Neural Networks." International Journal of Engineering and Information Systems (IJEIS) 3(2): 9-19.
27. Dawood, K. J., et al. (2020). "Artificial Neural Network for Mushroom Prediction." International Journal of Academic Information Systems Research (IJAISR) 4(10): 9-17.
28. Dheir, I. M., et al. (2020). "Classifying Nuts Types Using Convolutional Neural Network." International Journal of Academic Information Systems Research (IJAISR) 3(12): 12-18.
29. El-Khatib, M. J., et al. (2019). "Glass Classification Using Artificial Neural Network." International Journal of Academic Pedagogical Research (IJAPR) 3(2): 25-31.
30. El-Mahelawi, J. K., et al. (2020). "Tumor Classification Using Artificial Neural Networks." International Journal of Academic Engineering Research (IJAER) 4(11): 8-15.
31. El-Mashharawi, H. Q., et al. (2020). "Grape Type Classification Using Deep Learning." International Journal of Academic Engineering Research (IJAER) 3(12): 41-45.
32. Elzamy, A., et al. (2015). "Classification of Software Risks with Discriminant Analysis Techniques in Software planning Development Process." International Journal of Advanced Science and Technology 81: 35-48.
33. Elzamy, A., et al. (2015). "Predicting Software Analysis Process Risks Using Linear Stepwise Discriminant Analysis: Statistical Methods." Int. J. Adv. Inf. Sci. Technol 38(38): 108-115.
34. Elzamy, A., et al. (2017). "Predicting Critical Cloud Computing Security Issues using Artificial Neural Network (ANNs) Algorithms in Banking Organizations." International Journal of Information Technology and Electrical Engineering 6(2): 40-45.
35. Habib, N. S., et al. (2020). "Presence of Amphibian Species Prediction Using Features Obtained from GIS and Satellite Images." International Journal of Academic and Applied Research (IJAAR) 4(11): 13-22.
36. Harz, H. H., et al. (2020). "Artificial Neural Network for Predicting Diabetes Using JNN." International Journal of Academic Engineering Research (IJAER) 4(10): 14-22.
37. Hassanein, R. A. A., et al. (2020). "Artificial Neural Network for Predicting Workplace Absenteeism." International Journal of Academic Engineering Research (IJAER) 4(9): 62-67.
38. Heriz, H. H., et al. (2018). "English Alphabet Prediction Using Artificial Neural Networks." International Journal of Academic Pedagogical Research (IJAPR) 2(11): 8-14.
39. Jaber, A. S., et al. (2020). "Evolving Efficient Classification Patterns in Lymphography Using EasyNN." International Journal of Academic Information Systems Research (IJAISR) 4(9): 66-73.
40. Kashf, D. W. A., et al. (2018). "Predicting DNA Lung Cancer using Artificial Neural Network." International Journal of Academic Pedagogical Research (IJAPR) 2(10): 6-13.
41. Khalil, A. J., et al. (2019). "Energy Efficiency Predicting using Artificial Neural Network." International Journal of Academic Pedagogical Research (IJAPR) 3(9): 1-8.
42. Kweik, O. M. A., et al. (2020). "Artificial Neural Network for Lung Cancer Detection." International Journal of Academic Engineering Research (IJAER) 4(11): 1-7.
43. Maghari, A. M., et al. (2020). "Books' Rating Prediction Using Just Neural Network." International Journal of Engineering and Information Systems (IJEIS) 4(10): 17-22.
44. Mettleq, A. S. A., et al. (2020). "Mango Classification Using Deep Learning." International Journal of Academic Engineering Research (IJAER) 3(12): 22-29.
45. Metwally, N. F., et al. (2018). "Diagnosis of Hepatitis Virus Using Artificial Neural Network." International Journal of Academic Pedagogical Research (IJAPR) 2(11): 1-7.
46. Mohammed, G. R., et al. (2020). "Predicting the Age of Abalone from Physical Measurements Using Artificial Neural Network." International Journal of Academic and Applied Research (IJAAR) 4(11): 7-12.
47. Musleh, M. M., et al. (2019). "Predicting Liver Patients using Artificial Neural Network." International Journal of Academic Information Systems Research (IJAISR) 3(10): 1-11.
48. Oriban, A. J. A., et al. (2020). "Antibiotic Susceptibility Prediction Using JNN." International Journal of Academic Information Systems Research (IJAISR) 4(11): 1-6.
49. Qwaider, S. R., et al. (2020). "Artificial Neural Network Prediction of the Academic Warning of Students in the Faculty of Engineering and Information Technology in Al-Azhar University-Gaza." International Journal of Academic Information Systems Research (IJAISR) 4(8): 16-22.
50. Sadek, R. M., et al. (2019). "Parkinson's Disease Prediction Using Artificial Neural Network." International Journal of Academic Health and Medical Research (IJAHMR) 3(1): 1-8.
51. Salah, M., et al. (2018). "Predicting Medical Expenses Using Artificial Neural Network." International Journal of Engineering and Information Systems (IJEIS) 2(20): 11-17.
52. Salman, F. M., et al. (2020). "COVID-19 Detection using Artificial Intelligence." International Journal of Academic Engineering Research (IJAER) 4(3): 18-25.
53. Samra, M. N. A., et al. (2020). "ANN Model for Predicting Protein Localization Sites in Cells." International Journal of Academic and Applied Research (IJAAR) 4(9): 43-50.
54. Shawarib, M. Z. A., et al. (2020). "Breast Cancer Diagnosis and Survival Prediction Using JNN." International Journal of Engineering and Information Systems (IJEIS) 4(10): 23-30.
55. Zaqout, I., et al. (2015). "Predicting Student Performance Using Artificial Neural Network: in the Faculty of Engineering and Information Technology." International Journal of Hybrid Information Technology 8(2): 221-228.
56. Saleh, A. et al. (2020). "Brain Tumor Classification Using Deep Learning." 2020 International Conference on Assistive and Rehabilitation Technologies (iCareTech). IEEE, 2020.
57. Almadhoun, H. et al. (2021). "Classification of Alzheimer's Disease Using Traditional Classifiers with Pre-Trained CNN." International Journal of Academic Health and Medical Research (IJAHMR) 5(4):17-21.
58. El-Habil, B. et al. (2022). "Cantaloupe Classification Using Deep Learning." International Journal of Academic Engineering Research (IJAER) 5(12): 7-17.
59. Alkahlout, M. A. et al. (2022). "Classification of Fruits Using Deep Learning." International Journal of Academic Engineering Research (IJAER) 5(12): 56-63.
60. Alfarrar, A. H. et al. (2022). "Classification of Pineapple Using Deep Learning." International Journal of Academic Information Systems Research (IJAISR) 5(12): 37-41.
61. Al-Masawabe, M. M. et al. (2022). "Papaya maturity Classification Using Deep Convolutional Neural Networks." International Journal of Engineering and Information Systems (IJEIS) 5(12): 60-67.
62. Abu-Jamie, T. N., et al. (2022). "Six Fruits Classification Using Deep Learning." International Journal of Academic Information Systems Research (IJAISR) 6(1): 1-8.
63. Aish, M. A., et al. (2022). "Classification of pepper Using Deep Learning." International Journal of Academic Engineering Research (IJAER) 6(1): 24-31.
64. Alsaqqa, A. H., et al. (2022). "Using Deep Learning to Classify Different types of Vitamin." International Journal of Academic Engineering Research (IJAER) 6(1): 1-6.
65. M. Alnajjar, M. K. and S. S. Abu-Naser (2022). "Heart Sounds Analysis and Classification for Cardiovascular Diseases Diagnosis using Deep Learning." International Journal of Academic Engineering Research (IJAER) 6(1): 7-23.
66. Aldammagh, Z., Abdeljawad, R., & Obaid, T. (2021). Predicting Mobile Banking Adoption: An Integration of TAM and TPB with Trust and Perceived Risk. Financial Internet Quarterly, 17(3), 35-46.
67. Obaid, T. (2021). Predicting Mobile Banking Adoption: An Integration of TAM and TPB with Trust and Perceived Risk. Available at SSRN 3761669.
68. Joula, H., Abu Jarad, A., Obaid, T., Abu Mdallah, S., & Awaja, A. (2020). Mobile Banking Adoption: Decomposed Theory of Planned Behavior with Perceived Trust. Available at SSRN 3660403.
69. Obaid, T., Abdaljawad, R., & Mdallah, S. A. (2020). Factors Driving E-Learning Adoption In Palestine: An Integration of Technology Acceptance Model And IS Success Model. Available at SSRN 3686490.
70. Obaid, T. F., & Eneizan, B. M. (2016). Transfer of training and post-training on job performance in Middle Eastern countries. Review of Public Administration and Management, 400(3786), 1-11.
71. Obaid, T. F., Zainon, M. S., Eneizan, P. D. B. M., & Wahab, K. A. TRANSFER OF TRAINING AND POST-TRAINING ON JOB PERFORMANCE IN MIDDLE EASTERN COUNTRIES.
72. Musleh, M. M., et al. (2019). "Predicting Liver Patients using Artificial Neural Network." International Journal of Academic Information Systems Research (IJAISR) 3(10): 1-11.
73. Qwaider, S. R., et al. (2020). "Artificial Neural Network Prediction of the Academic Warning of Students in the Faculty of Engineering and Information Technology in Al-Azhar University-Gaza." International Journal of Academic Information Systems Research (IJAISR) 4(8): 16-22.
74. Mettleq, A. S. A., et al. (2020). "Mango Classification Using Deep Learning." International Journal of Academic Engineering Research (IJAER) 3(12): 22-29.
75. Salman, F. M., et al. (2020). "COVID-19 Detection using Artificial Intelligence." International Journal of Academic Engineering Research (IJAER) 4(3): 18-25.
76. Metwally, N. F., et al. (2018). "Diagnosis of Hepatitis Virus Using Artificial Neural Network." International Journal of Academic Pedagogical Research (IJAPR) 2(11): 1-7.

77. Shahroz Tariq, Sangyup Lee, Hoyoung Kim, Youjin Shin, Simon S. Woo, Detecting Both Machine and Human Created Fake Face Images In the Wild, Proceedings of the 2nd International Workshop on Multimedia Privacy and Security pages 81-87, October 15-19, 2018, New York, NY, USA. Retrieved from (<https://dl.acm.org/citation.cfm?id=3267367>)
78. Haiqing Liu, Shiqiang Zheng, Shuhua Hao, and Yuancheng Li, "Multifeature Fusion Detection Method for Fake Face Attack in Identity Authentication," Advances in Multimedia, vol. 2018, Article ID 9025458, 10 pages, 2018. Retrieved from (<https://doi.org/10.1155/2018/9025458>)
79. Shahroz Tariq, Sangyup Lee, Hoyoung Kim, Youjin Shin, Simon S. Woo, GAN is a friend or foe? a framework to detect various fake face images, Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing Pages 1296-1303, April 08-12, 2019, New York, NY, USA. Retrieved from (<https://dl.acm.org/citation.cfm?id=3297410>)
80. Adrian Rosebrock, OpenCV Face Recognition, September 24, 2018 in deep leaning, faces, tutorials. (<https://www.pyimagesearch.com/2018/09/24/opencv-face-recognition/>)
81. Yaojie Liu, Amin Jourabloo, Xiaoming Liu; Learning Deep Models for face Anti-Spoofing: Binary of Auxiliary Supervision; The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 389-398. Retrieved from (http://openaccess.thecvf.com/content_cvpr_2018/html/Liu_Learning_Deep_Models_CVPR_2018_paper.html)
82. Youngshin Kim, Jaekeun Na, Seongbeak Yoon, and Juneho Yi. Masked fake face detection using radiance measurements. JOSA A, 2009. Retrieved from (https://www.researchgate.net/publication/24251875_Masked_fake_face_detection_using_radiance_measurements)