International Journal of Academic Information Systems Research (IJAISR)
ISSN: 2643-9026
Vol. 6 Issue 7, July - 2022, Pages: 29-40

Classification of Sign-Language Using MobileNet - Deep Learning

Tanseem N. Abu-Jamie and Prof. Dr. Samy S. Abu-Naser

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

Abstract: *Sign language recognition is one of the most rapidly expanding fields of study today. Many new technologies have been developed in recent years in the fields of artificial intelligence the sign language-based communication is valuable to not only deaf and dumb community, but also beneficial for individuals suffering from Autism, downs Syndrome, Apraxia of Speech for correspondence. The biggest problem faced by people with hearing disabilities is the people's lack of understanding of their requirements. In this paper we try to fill this gap. By trying to translate sign language using artificial intelligence algorithms, we focused in this paper using transfer learning technique based on deep learning by utilizing a MobileNet algorithm and compared it with the previous paper results[10a], where we get in the Mobilenet algorithm on the degree of Accuracy 93,48% but the VGG16 the accuracy was 100% For the same number of images (43500 in the dataset in size 64*64 pixel) and the same data split training data into training dataset (70%) and validation dataset(15%) and testing dataset(15%) and 20 epoch .*

Keywords: *sign language Classification, Type of sign language, Deep Learning, Classification, Detection*

1. INTRODUCTION:

Communication is essential in everyday life for sharing thoughts, emotions, information, facts, ideas, opinions, and understanding. The process of exchanging information between individuals is known as communication.

Sign language is a form of communication that is used to communicate ideas. This is used by the global deaf and dumb community. Every region of the world has its own sign language with its own set of protocols. American Sign Language is the most widely used sign language. Autism, Down Syndrome, and Apraxia of Speech can all benefit from sign language .

Figure 1: Amirecan sign language

2. REVIEW OF LITERATURE SURVEY

Over the last two decades, several researchers have worked in the field of sign language recognition. Static sign language recognition systems are classified primarily based on the input captured and fed into the system. This work's literature review focuses not only on this category, but also on the use of various classifiers and their application in Machine Learning and Deep Learning-based systems [1].

In [2] researchers proposed in this paper, they have proposed a system using Eigen value weighte Euclidean distance as a classification technique for recognition of various Sign Languages of India. The system comprises of four parts: Skin Filtering, Hand Cropping, Feature Extraction and Classification.

24 signs were considered in this paper, each having 10 samples, thus a total of 240 images was considered for which recognition rate obtained was 97%.

In [3] authors are looking for efficient spatiotemporal modeling for SLR. They specifically build efficient 3D CNNs, namely 3D MobileNet-v2 for isolated SLR, and then improve performance by designing a random knowledge distillation

International Journal of Academic Information Systems Research (IJAISR)
ISSN: 2643-9026
Vol. 6 Issue 7, July - 2022, Pages: 29-40

strategy (RKD) to transfer knowledge from multiple teacher models, including R3D, R(2+1)D, and SlowFast networks. These lightweight models are also used as spatiotemporal feature extraction in the Transformer framework for the more difficult continuous SLR. The distilled models demonstrate high efficiency and strong performance on the SLR-500 and CSL benchmarks in the experiments. We conclude that the lightweight 3D MobileNet-v2 with the proposed RKD achieves a good balance of accuracy and efficiency and is well suited for SLR.

In [4] the paper aims to propose a model and build a computer system that can recognize Bangla Sign Language alphabets and translate them to corresponding Bangla letters by means of deep convolutional neural network (CNN). CNN has been introduced in this model in form of a pre-trained model called "MobileNet" which produced an average accuracy of 95.71% in recognizing 36 Bangla Sign Language alphabets.

In [5] this study they trained a model, which will be able to classify the Arabic sign language, which consists of 32 Arabic alphabet sign classes. In images, sign language is detected through the pose of the hand. In this study, they proposed a framework, which consists of two CNN models, and each of them is individually trained on the training set. The final predictions of the two models were ensembled to achieve higher results. The results they achieved on the test set for the whole data are with an accuracy of about 97%.

In [6] this study The MobileNet V1 convolutional neural network is trained against the EgoHands dataset from Indiana University's UI Computer Vision Lab to see if the dataset is sufficient to detect hands in LESCO videos from five different signers wearing short-sleeve shirts and against complex backgrounds. Those requirements are critical in determining the usefulness of the solution, as the consulted bibliography performs tests with single-color backgrounds and long-sleeve shirts, which ease classification tasks only in controlled environments. The two-step experiment yielded 1) a mean average precision of 96.1% for the EgoHands dataset and 2) an average accuracy of 91% for hand detection across the five LESCO videos.

In [7] their work they build and compare two pre-trained CNN models,MobileNet and InceptionV3 architectures. The suggested approach using the MobileNet model showed an accuracy of 99% on a custom-built dataset. The working CNN model can perform real-time recognition of ISL alphabets, numbers and a few selected gestures integrated into an Android mobile platform called SIGNify using React-Native, for better accessibility and user-friendly access.

In [8] they combine the two MobileNet convolutional network models with Random Forest to improve image recognition accuracy even further. This method first applies the MobileNet model architecture and weight files to gesture images, then trains the model and extracts image features, classifies the features extracted by the convolutional network using the Random Forest model, and finally obtains classification results. The recognition rate on the Sign Language Digital dataset, Sign Language Gesture Image dataset, and Fingers dataset was significantly improved when compared to Random Forest, Logistic Regression, Nearest Neighbor, XGBoost, VGG, Inception, and MobileNet.

3. METHODOLOGY

The proposed system for Sign Language Recognition System comprises of data to Kaggle the dataset contains sets of images in jpeg extension and prepare data to apply the architecture CNN by mobileNet model data split training data into training dataset (70%) and validation dataset (15%) and testing dataset(15%) Training and testing the dataset

A. Dataset

The proposed system employs deep learning techniques for data recognition in the training phase, which is critical to the overall system. In the first step, the number of images was very large, so we decided to only use 43,500 of the 87,000 images. This data contains 29 classes (A to Z), spaces, delete, and nothing.

Figure 2:a latters used in dataset

B. Deep Learning

Deep Learning is a subset of Machine learning [9-29]. Deep learning consists of a progressive neural network demonstration that learns the input presented during the training stage, i.e. the input is converted to a learned representation that is easier for the linear model to classify [30-40]. The primary layer extracts low-level features, such as distinct edges in a photograph, and the subsequent layers build on these highlights to form more high-level features, as shown in the following image [41-50].

Because the features are learned by neural networks with multiple layers rather than manually extracted as in machine learning, one must select a number of features and create an explicit implementation for the extraction, which is hampered by dimensionality. Given enough data and computational power, deep learning can deal with high-dimensional input because it can learn features without explicitly describing which features it should extract from an individual image in the entire dataset [51-60].

The proposed system's goal is to recognize the sign that is represented by the sign and 29 signs considered for the experiment so it is having multiple category, classes are provided as 29 target class for the sign language recognition system.

C. MobileNet

In this study, a MobileNet version 1 was used because it is a lightweight deep convolutional neural network that is much smaller in size and performs much faster than many popular neural network models. Its primary applications are image classification and detection. It is 17MB in size and employs 4.2M parameters for overall processing. It employs depth-wise, separable convolutions, which imply performing a single convolution on each color channel rather than combining all three and flattening it. It has an effect on the input channels by filtering them[61-65].

In normal convolution, the filter operates on M channels of the input image and outputs N feature maps, as shown in Fig.10. The filter, which is a cube with dimensions Dk X D M k X and an input of N, is multiplied with the corresponding element in the cube and finally outputs N feature maps[66-70].

In depth wise separable convolutions, the M single channel filters use point wise convolution, which operates on a

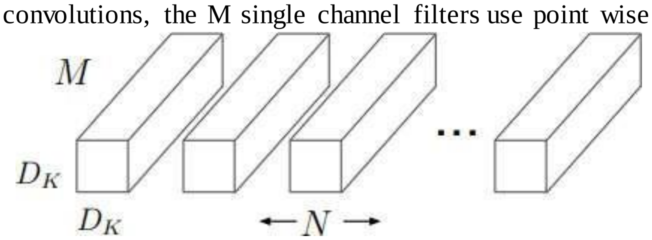

Figure 3: normal convolution

single cube in the input feature channel, reducing calculation as shown in Fig.4. Figure 5 depicts the point-wise convolution. [9]

International Journal of Academic Information Systems Research (IJAISR)
ISSN: 2643-9026
Vol. 6 Issue 7, July - 2022, Pages: 29-40

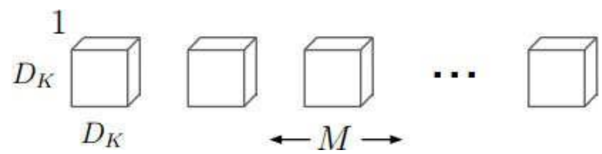

Figure 4:normal convolution

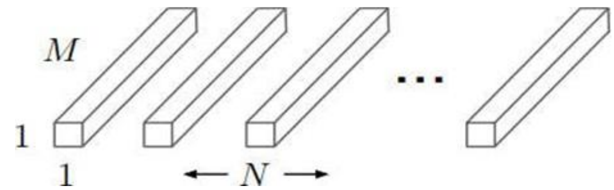

Figure 5: point-wise convolution

D. Training Algorithm

According to Table 1, the pretrained CNN MobileNet model is used. The model's final classification layer was removed, and the other layer was frozen before retraining the last layer with our dataset and fine-tuning the model's parameters. Preprocessing images with a resolution of 64x64x3 pixels. The following parameters were used to fit the model: the learning rate is 0.0001, the activation is softmax, the loss is categorical crossentropy, Adam is the optimizer, and the epoch is 20.

Table 1: MobileNet Model Evaluation based on number of image, Loading Time, and Accuracy

Model	**Number of image**	**Loading Time (seconds)**	**Accuracy (Percent)**
MobileNet	43500	127.583s	95.41%

4. EXPERIMENTAL RESULTS

1. **load all images in memory**

The first step was to call the library NumPy (is the fundamental package for scientific computing in Python) responsible for providing a multidimensional array object, various derived objects (such as masked arrays and arrays), and a variety of procedures for quick operations on arrays, including arithmetic and logical operations. In this library we used Numpy.random.seed() In Python, the np.random.seed function provides an input for the pseudo-random number generator. It enables us to provide a "seed" value to NumPy's random number generator [10]. We use it for performing simple tasks like splitting datasets into training and test sets requires random sampling. In turn, random sampling almost always requires pseudo-random numbers.

And we use filter to choose the best 1500 images for each class and resize to (64*64 px). So the result in this step:

   1. Training data shape: (43500, 64, 64, 3)
   2. Training labels shape: (43500, 29)

2. **Split dataset:** training data into training dataset (70%) and validation dataset(15%) and testing dataset(15%) by import train_test_split from sklearn.model_selection
3. **Call the MobileNet pre-trained Model**
   We make function contain a mobileNet model use (weights= 'imagenet', include_top=False, input_shape= (64, 64, 3)) then make MaxPooling2D and Dense then used compile (Adam(lr=0.00001), loss='categorical_crossentropy', metrics=['accuracy']) So the result in this step :
   1. Total params: 3,258,589

2. Trainable params: 3,236,701
3. Non-trainable params: 21,888

4. Train model

We used it during training fit_generator often used when datasets too large to fit into memory. They also tend to be challenging, requiring us to perform data augmentation to avoid overfitting and increase the ability of our model to generalize.

The results of a convolution neural network's by mobileNet loss rate in the training and test sets after 20 repetitions are shown in Figure 8. Which suggests that the convolutional neural network successfully learned the input and can act as a good model for understanding sign language but VGG16 100% in the previous paper [10] was better from mobileNet 95,41% accuracy

```
Epoch 1/20
61/61 [==============================] - 22s 129ms/step - loss: 5.6959 - accuracy: 0.0653 - val_loss: 5.1061 - val_accuracy: 0.0602
Epoch 2/20
61/61 [==============================] - 5s 80ms/step - loss: 3.7684 - accuracy: 0.2017 - val_loss: 3.6666 - val_accuracy: 0.1693
Epoch 3/20
61/61 [==============================] - 6s 93ms/step - loss: 2.6789 - accuracy: 0.3567 - val_loss: 2.4932 - val_accuracy: 0.3549
Epoch 4/20
61/61 [==============================] - 5s 89ms/step - loss: 1.9730 - accuracy: 0.4831 - val_loss: 1.7786 - val_accuracy: 0.5191
Epoch 5/20
61/61 [==============================] - 5s 79ms/step - loss: 1.5241 - accuracy: 0.5806 - val_loss: 1.3350 - val_accuracy: 0.6262
Epoch 6/20
61/61 [==============================] - 5s 79ms/step - loss: 1.1884 - accuracy: 0.6632 - val_loss: 1.0612 - val_accuracy: 0.7002
Epoch 7/20
61/61 [==============================] - 5s 79ms/step - loss: 0.9725 - accuracy: 0.7216 - val_loss: 0.8742 - val_accuracy: 0.7500
Epoch 8/20
61/61 [==============================] - 5s 79ms/step - loss: 0.7966 - accuracy: 0.7734 - val_loss: 0.7388 - val_accuracy: 0.7890
Epoch 9/20
61/61 [==============================] - 5s 81ms/step - loss: 0.6753 - accuracy: 0.8121 - val_loss: 0.6307 - val_accuracy: 0.8202
Epoch 10/20
61/61 [==============================] - 5s 79ms/step - loss: 0.5789 - accuracy: 0.8366 - val_loss: 0.5506 - val_accuracy: 0.8443
Epoch 11/20
61/61 [==============================] - 5s 79ms/step - loss: 0.5084 - accuracy: 0.8550 - val_loss: 0.4759 - val_accuracy: 0.8644
Epoch 12/20
61/61 [==============================] - 5s 78ms/step - loss: 0.4345 - accuracy: 0.8829 - val_loss: 0.4183 - val_accuracy: 0.8820
Epoch 13/20
61/61 [==============================] - 5s 78ms/step - loss: 0.3924 - accuracy: 0.8923 - val_loss: 0.3687 - val_accuracy: 0.8973
Epoch 14/20
61/61 [==============================] - 5s 77ms/step - loss: 0.3360 - accuracy: 0.9106 - val_loss: 0.3286 - val_accuracy: 0.9085
Epoch 15/20
61/61 [==============================] - 5s 78ms/step - loss: 0.2896 - accuracy: 0.9248 - val_loss: 0.2935 - val_accuracy: 0.9212
Epoch 16/20
61/61 [==============================] - 5s 77ms/step - loss: 0.2651 - accuracy: 0.9312 - val_loss: 0.2636 - val_accuracy: 0.9280
Epoch 17/20
61/61 [==============================] - 5s 79ms/step - loss: 0.2255 - accuracy: 0.9403 - val_loss: 0.2395 - val_accuracy: 0.9375
Epoch 18/20
61/61 [==============================] - 5s 77ms/step - loss: 0.2136 - accuracy: 0.9457 - val_loss: 0.2141 - val_accuracy: 0.9454
Epoch 19/20
61/61 [==============================] - 5s 78ms/step - loss: 0.1938 - accuracy: 0.9517 - val_loss: 0.1936 - val_accuracy: 0.9508
Epoch 20/20
61/61 [==============================] - 5s 77ms/step - loss: 0.1772 - accuracy: 0.9541 - val_loss: 0.1746 - val_accuracy: 0.9565
Time elapsed in seconds: 127.58303356170654
```

Figure 6: Loss and Accuracy rate

5. Training and validation accuracy

We must understand the difference between each:

- Training dataset: the data used to fit the model.
- Validation dataset: during the training process, the data is used to validate the model's generalization ability or to stop it early.
- Testing dataset: the data used to for other purposes other than training and validating

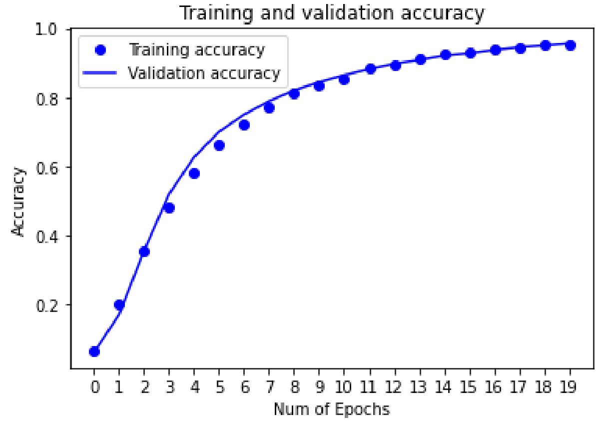

Figure 7: Training and validation accuracy

In Figure 7 show us no overfitting (When a model is trained with a large amount of data, it begins to learn from the noise and incorrect data entries in our data set) or underfitting(A statistical model or a machine learning algorithm is said to have underfitting when it cannot capture the underlying trend of the data) this means that the model is well trained[11].

6. Training and validation loss

The training loss is a metric that measures how well a deep learning model fits the training data. That is, it evaluates the model's error on the training set. It should be noted that the training set is a subset of the dataset used to train the model initially. The training loss is computed by adding the sum of errors for each example in the training set.

Validation loss, on the other hand, is a metric used to evaluate the performance of a deep learning model on the validation set. The validation set is a subset of the dataset set aside to test the model's performance. The validation loss, like the training loss, is calculated by adding the errors for each example in the validation set. It is also worth noting that the training loss is calculated after each batch.

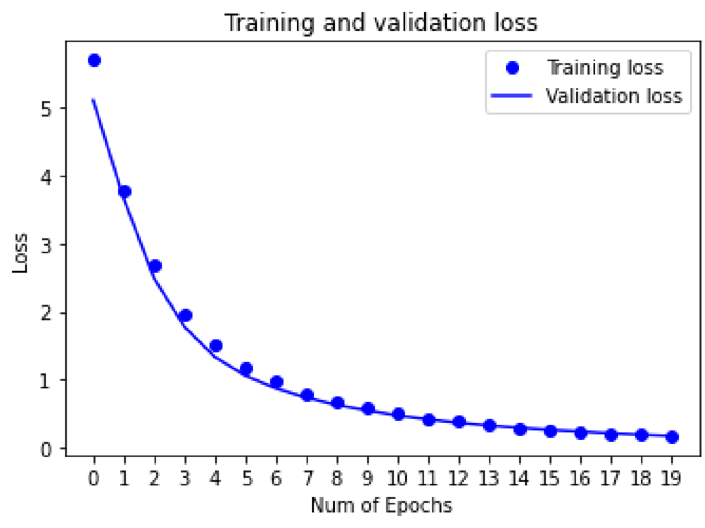

Figure 8: Training and validation loss

7. Evaluating the model on the training, validating sets

The evaluating the model process of analyzing a training model to ensure that it is delivered effectively and efficiently is known as training evaluation. Training evaluation identifies training gaps and even opportunities for model improvement By seeing each of the following values validating accuracy, validating loss ,training loss ,training accuracy If every value in training accuracy and validating accuracy has a high value, and every value in validating loss and training loss is low, it means that the model is well trained.

Table 2: Evaluating the model on the training, validating sets

Evaluating the model	Training Accuracy	Training Loss	Validating Accuracy	Validating Loss
Precent	97,95%	10,26%	95,65%	17,46%

8. Plot ROC curve

The ROC curve is a short form for Receiver Operating Characteristic curve. The performance of a classification model is represented by ROC curves. In terms of predicted probability, ROC tells us how good the model is at distinguishing between the given classes. The area under the curve (AUC) can be used as an indicator of the performance of the model.

To achieve the best model, we want to increase our True Positive Rate while decreasing our False Positive Rate (TPR = 1, FPR = 0) this means that our model will be able to correctly separate the classes. Such models are referred to as skilled models. This is never achieved in real life [12].

The ROC curve is a plot of the true positive rate (TPR: calculated as the number of true positives divided by the sum of the number of true positives and the number of false negatives.) (y-axis) vs for the false positive rate (FPR: calculated as the number of false positives divided by the sum of the number of false positives and the number of true negatives.) (x-axis) a number of different candidate threshold values between 0.0 and 1.0.

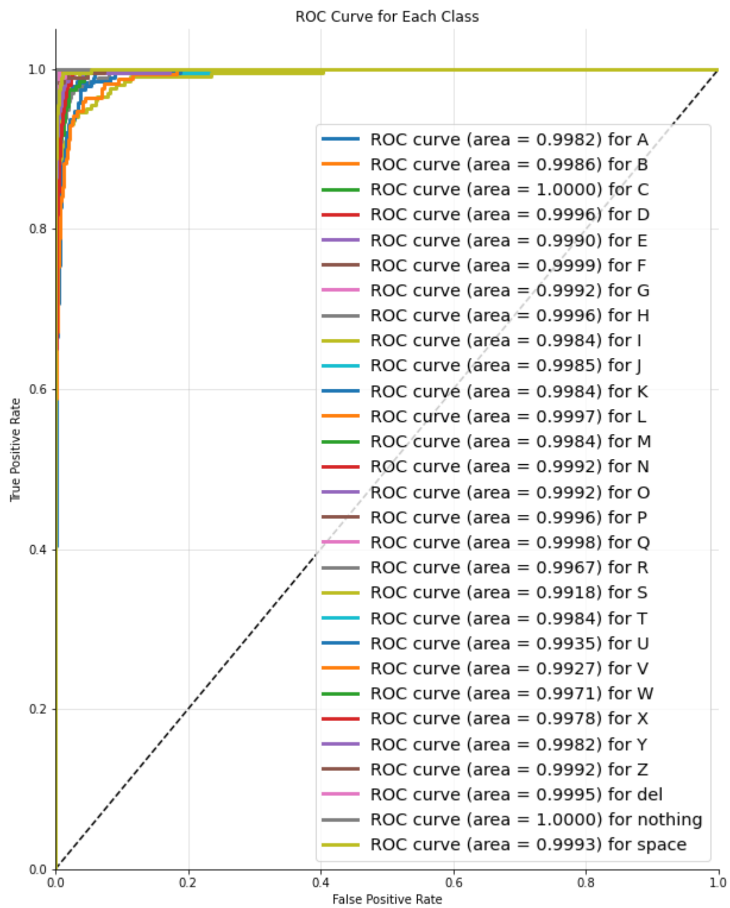

Figure 9: ROC curve

9. Classification Report

The classification report contains information about the key metrics in a classification problem. For each class you're looking for, in this model performance evaluation of the proposed system was computed using parameters accuracy, precision, recall and F1 score

- The ***Accuracy*** is defined as the number of correctly classified signs divided by the total number of signs used for classification. tells us about how well the model was trained and how well it will perform in general.[13]

International Journal of Academic Information Systems Research (IJAISR)
ISSN: 2643-9026
Vol. 6 Issue 7, July - 2022, Pages: 29-40

Accuracy = (TP+TN)/(TP+FP+FN+TN)

1. **True Positives (TP) -** The model correctly predicted the sign as the actual class.
2. **True Negatives (TN) -** The model correctly predicted the sign that does not belong to the class.
3. **False Positives (FP)** occur when the actual class of sign is false and the predicted class is true.
4. **False Negatives (FN)** Model prediction is contradicts as actual class of sign is true but predicted class in false[14].

- The ***precision*** will be defined as "how many people in that class are correctly classified " measures only the rate of false positives[13]

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

- The ***recall*** is defined as "how many of this class you find out of the total number of elements in this class." is the opposite of precision; it measures false negatives against true positives [13]

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

- The ***f1-score*** represents the harmonic mean of precision and recall [13].

$$\text{F1} = 2 \times \frac{Precision * Recall}{Precision + Recall}$$

- The ***support*** is the number of occurrence of the given class in your dataset

  The results of the present research work has been projected through Fig.10

International Journal of Academic Information Systems Research (IJAISR)
ISSN: 2643-9026
Vol. 6 Issue 7, July - 2022, Pages: 29-40

	precision	recall	f1-score	support
A	0.9027	0.9126	0.9076	183
B	0.8498	0.9427	0.8938	192
C	0.9946	0.9786	0.9865	187
D	0.9714	0.9392	0.9551	181
E	0.9227	0.9176	0.9201	182
F	0.9548	0.9694	0.9620	196
G	0.9319	0.9570	0.9443	186
H	0.9843	0.9641	0.9741	195
I	0.9368	0.8717	0.9030	187
J	0.9565	0.9724	0.9644	181
K	0.9119	0.9312	0.9215	189
L	0.9752	0.9336	0.9540	211
M	0.9462	0.9362	0.9412	188
N	0.9096	0.9500	0.9293	180
O	0.9484	0.9528	0.9506	212
P	0.9673	0.9628	0.9650	215
Q	0.9590	0.9639	0.9614	194
R	0.8808	0.8673	0.8740	196
S	0.8667	0.8366	0.8514	202
T	0.9375	0.9116	0.9244	181
U	0.7879	0.8168	0.8021	191
V	0.7849	0.7941	0.7895	170
W	0.9176	0.8520	0.8836	196
X	0.8571	0.8867	0.8717	203
Y	0.9215	0.9215	0.9215	191
Z	0.9667	0.9613	0.9640	181
del	0.9471	0.9521	0.9496	188
nothing	0.9948	1.0000	0.9974	191
space	0.9254	0.9394	0.9323	198
accuracy			0.9243	5547
macro avg	0.9245	0.9240	0.9240	5547
weighted avg	0.9250	0.9243	0.9244	5547

Figure 10:Classification Report

10. Confusion Matrix

A confusion matrix, as shown in Fig.11, provides a graphical summary of the performance of a trained model of prediction results on a classification problem. It shows how many inputs are predicted correctly and incorrectly as a result of class confusion. The predictions are listed and divided into classes using count values. It also summarizes the model's testing results for further examination. The diagonal values of the confusion matrix represent the number of signs correctly classified by the models, while the off-diagonal values represent the number of incorrectly classified signs. The model will perform better if the diagonal values are higher.

```
[[167   6   0   0   5   1   0   0   0   0   2   1   0   0   0   0   0   0
    0   0   0   0   0   0   0   0   0   0   1]
 [  2 181   0   0   0   0   0   0   1   0   0   0   1   1   0   0   0   2
    1   0   1   1   0   0   0   0   0   0   1]
 [  0   0 183   0   0   0   0   0   0   0   0   1   0   0   0   0   1   0
    1   0   0   0   0   0   0   1   0   0   0]
 [  2   1   0 178   0   2   0   0   0   0   1   1   1   0   3   0   0   0
    0   0   0   0   0   0   0   0   0   0   0]
 [  1   7   0   0 167   0   2   0   1   0   0   1   0   0   0   0   0   0
    0   0   0   0   0   3   0   0   0   0   0]
 [  2   1   0   0   1 190   0   0   0   0   0   0   0   0   1   0   0   0
    0   0   0   0   0   1   0   0   0   0   0]
 [  2   0   0   0   0   0 178   1   0   0   0   1   1   0   0   0   1   0
    0   0   0   1   0   0   0   0   1   0   0]
 [  0   0   0   0   0   0   3 188   0   1   0   0   0   0   0   0   1   0
    0   0   0   1   0   0   0   0   0   0   1]
 [  2   4   0   3   4   1   0   0 163   1   0   0   0   0   1   1   1   2
    0   0   3   0   1   0   0   0   0   0   0]
 [  0   0   0   0   0   0   1   0   0 176   0   0   0   1   0   0   0   0
    0   0   0   0   0   0   2   0   0   0   1]
 [  0   3   0   0   0   0   0   0   3   0 176   0   0   1   0   0   0   4
    0   0   2   0   0   0   0   0   0   0   0]
 [  2   0   0   0   0   0   0   0   2   0   2 197   0   1   0   0   0   0
    1   2   0   0   0   2   0   1   0   0   1]
 [  0   0   0   0   1   0   1   0   0   1   0   0 176   4   0   0   0   0
    1   0   0   1   0   0   1   0   1   1   0]
 [  0   2   0   0   0   0   0   0   0   0   0   0   3 171   1   0   0   0
    1   0   0   1   0   1   0   0   0   0   0]
 [  0   0   0   1   0   0   0   1   1   0   0   0   0   5 202   0   0   0
    0   0   0   1   0   0   0   0   1   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0 207   3   0
    0   0   0   0   0   0   1   0   3   0   1]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   3 187   0
    0   0   0   0   0   0   0   0   4   0   0]
 [  0   0   0   0   0   2   1   0   1   0   3   0   0   1   1   0   0 170
    1   0  10   1   1   4   0   0   0   0   0]
 [  1   0   0   0   1   2   0   0   0   0   1   0   2   0   0   2   0   1
  169   2   3   2   1  12   2   0   0   0   1]
 [  0   0   0   0   0   0   0   0   0   0   1   0   1   0   0   0   0   0
    5 165   1   1   0   1   2   2   0   0   2]
 [  0   0   0   0   2   0   0   0   0   0   4   0   0   1   1   0   0   6
    2   1 156  15   1   2   0   0   0   0   0]
 [  0   2   0   0   0   0   1   0   0   1   1   0   0   0   0   0   0   4
    2   0  16 135   4   1   0   0   0   0   3]
 [  2   5   0   1   0   0   1   0   2   0   1   0   1   0   0   0   0   1
    2   0   2   8 167   1   0   1   0   0   1]
 [  0   1   0   0   0   1   0   0   0   0   0   0   0   0   0   1   0   3
    6   1   4   1   2 180   2   1   0   0   0]
 [  0   0   0   0   0   0   0   0   0   2   0   0   0   1   0   0   0   0
    1   3   0   1   5   1 176   0   0   0   1]
 [  0   0   1   0   0   0   0   0   0   0   0   0   0   1   1   0   0   0
    0   2   0   0   0   1   1 174   0   0   0]
 [  0   0   0   0   0   0   1   0   0   1   0   0   0   0   2   0   1   0
    1   0   0   0   0   0   2   0 179   0   1]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    0   0   0   0   0   0   0   0   0 191   0]
 [  2   0   0   0   0   0   2   1   0   1   1   0   0   0   0   0   0   0
    1   0   0   2   0   0   2   0   0   0 186]]
```

Figure 11: confusion matrix

5. CONCLUSION

In this paper we evaluated the performance of pretrained MobileNet model of Deep Learning on datasets of 29 classes to signs we utilized pretrained model MobileNet. We use a pre trained CNN Model MobilelNet fine-tuned. We trained and validated the proposed model and tested its performance with un-seen dataset for testing. The Accuracy rate we achieved was 95.41%% to 20 epoch. This indicates that our proposed model can effectively predicate and classify different sign language without error and with full performance but the VGG16 achieved higher accuracy 100% in pervious paper [10].

References:

[1] Nadgeri, Sulochana and Kumar, Arun, An Analytical Study of Signs Used in Baby Sign Language Using Mobilenet Framework (June 6, 2020).

[2] Singha, Joyeeta, and Karen Das. "Indian sign language recognition using eigen value weighted euclidean distance based classification technique." arXiv preprint arXiv:1303.0634 (2013).

[3] X. Han, F. Lu and G. Tian, "Sign Language Recognition Based on Lightweight 3D MobileNet-v2 and Knowledge Distillation," ICETIS 2022; 7th International Conference on Electronic Technology and Information Science, 2022, pp. 1-6.

[4] Angona, Tazkia Mim, et al. "Automated Bangla sign language translation system for alphabets by means of MobileNet." TELKOMNIKA (Telecommunication Computing Electronics and Control) 18.3 (2020): 1292-1301.

[5] Alnuaim, Abeer, et al. "Human-Computer Interaction with Hand Gesture Recognition Using ResNet and MobileNet." Computational Intelligence and Neuroscience 2022 (2022).

[6] Zamora-Mora, Juan, and Mario Chacón-Rivas. "Real-time hand detection using convolutional neural networks for costar can sign language recognition." 2019 International Conference on Inclusive Technologies and Education (CONTIE). IEEE, 2019.

[7] S. Mohammed Siddiq, S. Roopashree, M. Suha, M. Ruthvik and K. Divyasree, "SIGNify – A mobile solution for Indian sign language using MobileNet architecture," 2021 2nd Global Conference for Advancement in Technology (GCAT), 2021, pp. 1-9, doi: 10.1109/GCAT52182.2021.9587639.

[8] Wang, Fei, Ronglin Hu, and Ying Jin. "Research on gesture image recognition method based on transfer learning." Procedia Computer Science 187 (2021): 140-145..

[9] Andrew G. Howard, Menglong Zhu, Bo Chen ,Dmitry Kalenichenk,o Weijun Wang ,Tobias Weyand ,Marco Andreetto ,Hartwig Adam, " MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications". doi: arxiv:1704.04861

[10] Abu-Jamie, T. N., et al. (2022). Classification of Sign-language Using VGG16. International Journal of Academic Engineering Research (IJAER), 6(6): 36-46.

[11] 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.

[12] 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.

[13] Abu-Saqer, M. M., et al. (2020). "Type of Grapefruit Classification Using Deep Learning." International Journal of Academic Information Systems Research (IJAISR) 4(1): 1-5.

[14] Afana, 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.

[15] 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.

[16] Alajrami, E., et al. (2019). "Blood Donation Prediction using Artificial Neural Network." International Journal of Academic Engineering Research (IJAER) 3(10): 1-7.

[17] Alajrami, E., et al. (2020). "Handwritten Signature Verification using Deep Learning." International Journal of Academic Multidisciplinary Research (IJAMR) 3(12): 39-44.

[18] 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.

[19] 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.

[20] Alfarra, A. H. et al. (2022). "Classification of Pineapple Using Deep Learning." International Journal of Academic Information Systems Research (IJAISR) 5(12): 37-41.

[21] Alghoul, A., et al. (2018). "Email Classification Using Artificial Neural Network." International Journal of Academic Engineering Research (IJAER) 2(11): 8-14.

[22] Alkahlout, M. A. et al. (2022). "Classification of Fruits Using Deep Learning." International Journal of Academic Engineering Research (IJAER) 5(12): 56-63.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] Al-Masawabe, M. M. et al. (2022). "Papaya maturity Classification Using Deep Convolutional Neural Networks." International Journal of Engineering and Information Systems (IJEAIS) 5(12): 60-67.

[28] 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.

[29] Al-Mobayed, A. A., et al. (2020). "Artificial Neural Network for Predicting Car Performance Using JNN." International Journal of Engineering and Information Systems (IJEAIS) 4(9): 139-145.

[30] 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.

[31] Alshawwa, I. A., et al. (2020). "Analyzing Types of Cherry Using Deep Learning." International Journal of Academic Engineering Research (IJAER) 4(1): 1-5.

[32] 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.

[33] Ashqar, B. A., et al. (2019). "Plant Seedlings Classification Using Deep Learning." International Journal of Academic Information Systems Research (IJAISR) 3(1): 7-14.

[34] 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.

[35] Barhoom, A. M., et al. (2019). "Predicting Titanic Survivors using Artificial Neural Network." International Journal of Academic Engineering Research (IJAER) 3(9): 8-12.

[36] 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.

[37] Dalffa, M. A., et al. (2019). "Tic-Tac-Toe Learning Using Artificial Neural Networks." International Journal of Engineering and Information Systems (IJEAIS) 3(2): 9-19.

[38] Dawood, K. J., et al. (2020). "Artificial Neural Network for Mushroom Prediction." International Journal of Academic Information Systems Research (IJAISR) 4(10): 9-17.

[39] 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.

[40] El-Habil, B. et al. (2022). "Cantaloupe Classification Using Deep Learning." International Journal of Academic Engineering Research (IJAER) 5(12): 7-17.

[41] El-Khatib, M. J., et al. (2019). "Glass Classification Using Artificial Neural Network." International Journal of Academic Pedagogical Research (IJAPR) 3(2): 25-31.

[42] El-Mahelawi, J. K., et al. (2020). "Tumor Classification Using Artificial Neural Networks." International Journal of Academic Engineering Research (IJAER) 4(11): 8-15.

[43] El-Mashharawi, H. Q., et al. (2020). "Grape Type Classification Using Deep Learning." International Journal of Academic Engineering Research (IJAER) 3(12): 41-45.

[44] Elzamly, 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.

[45] Elzamly, 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.

[46] Elzamly, 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.

[47] 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.

[48] 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.

[49] 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.

[50] Heriz, H. H., et al. (2018). "English Alphabet Prediction Using Artificial Neural Networks." International Journal of Academic Pedagogical Research (IJAPR) 2(11): 8-14.

[51] 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.

[52] 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.

[53] Khalil, A. J., et al. (2019). "Energy Efficiency Predicting using Artificial Neural Network." International Journal of Academic Pedagogical Research (IJAPR) 3(9): 1-8.

[54] 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.

[55] Maghari, A. M., et al. (2020). "Books' Rating Prediction Using Just Neural Network." International Journal of Engineering and Information Systems (IJEAIS) 4(10): 17-22.

[56] Mettleq, A. S. A., et al. (2020). "Mango Classification Using Deep Learning." International Journal of Academic Engineering Research (IJAER) 3(12): 22-29.

[57] 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.

[58] 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.

[59] 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.

[60] Oriban, A. J. A., et al. (2020). "Antibiotic Susceptibility Prediction Using JNN." International Journal of Academic Information Systems Research (IJAISR) 4(11): 1-6.

[61] 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.

[62] 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.

[63] Salah, M., et al. (2018). "Predicting Medical Expenses Using Artificial Neural Network." International Journal of Engineering and Information Systems (IJEAIS) 2(20): 11-17.

[64] Saleh, A. et al. (2020). "Brain Tumor Classification Using Deep Learning." 2020 International Conference on Assistive and Rehabilitation Technologies (iCareTech). IEEE, 2020.

[65] Salman, F. M., et al. (2020). "COVID-19 Detection using Artificial Intelligence." International Journal of Academic Engineering Research (IJAER) 4(3): 18-25.

[66] 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.

[67] Shawarib, M. Z. A., et al. (2020). "Breast Cancer Diagnosis and Survival Prediction Using JNN." International Journal of Engineering and Information Systems (IJEAIS) 4(10): 23-30.

[68] 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.

[69] 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.
[70] Obaid, T. (2021). Predicting Mobile Banking Adoption: An Integration of TAM and TPB with Trust and Perceived Risk. Available at SSRN 3761669.
[71] Jouda, H., Abu Jarad, A., Obaid, T., Abu Mdallalah, S., & Awaja, A. (2020). Mobile Banking Adoption: Decomposed Theory of Planned Behavior with Perceived Trust. Available at SSRN 3660403.
[72] Obaid, T., Abdaljawad, R., & Mdallalahc, S. A. (2020). Factors Driving E-Learning Adoption In Palestine: An Integration of Technology Acceptance Model And IS Success Model. Available at SSRN 3686490.
[73] 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.
[74] 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.