Science Journal of Circuits, Systems and Signal Processing
2023; 11(1): 1-11
http://www.sciencepublishinggroup.com/j/cssp
doi: 10.11648/j.cssp.20231101.11
ISSN: 2326-9065 (Print); ISSN: 2326-9073 (Online)

Multiple Sign Language Identification Using Deep Learning Techniques

Ahmed Mahmoud Sultan[1, *], Waleed Makram Mohamed Zaki[3], Mohammed Kayed[1], Abdel Mgeid Amin Ali[2]

[1]Computer Science Department, Faculty of Computers and Artificial Intelligence, Beni-Suef, Egypt

[2]Computer Science Department, Faculty of Computers and Information Systems, Minia, Egypt

[3]Information System Department, Faculty of Computers and Information Systems, Minia, Egypt

Email address:

ahmed.soltan@fcis.bsu.edu.eg (Ahmed Mahmoud Sultan)

[*]Corresponding author

To cite this article:

Ahmed Mahmoud Sultan, Waleed Makram Mohamed Zaki, Mohammed Kayed, Abdel Mgeid Amin Ali. Multiple Sign Language Identification Using Deep Learning Techniques. *Science Journal of Circuits, Systems and Signal Processing*. Vol. 11, No. 1, 2023, pp. 1-11. doi: 10.11648/j.cssp.20231101.11

Received: January 21, 2023; **Accepted**: February 14, 2023; **Published**: May 29, 2023

Abstract: The research presents a general overview of sign languages, and a previous survey was conducted on all aspects of sign languages including the tools used to collect sign languages and the best algorithms to achieve the best results. A specialized database is prepared to combine the alphabet signs of the Arabic, American, and British languages, as they are the most important sign languages and the most widespread in the world. Based on different sign languages and deep learning techniques such as LeNet, VGG-16, and CapsNet, which are considered among the best methods for solving sign language problems based on our previous studies. The purpose of the research is to remove the communication gap between the deaf, and normal speaking people who speak one sign language or those who try to communicate from different countries and to identify these languages easily. We applied some of the traditional deep learning techniques such as LeNet, and then we applied VGG-16 using pre-training models and adjusted some layers to suit our problem. Also we applied CapsNet as it is perfectly suitable for solving the problem of sign language deformation, rotation, and scaling. The best results were achieved using VGG-16, as it was trained on a previous database like ImageNet, which contains millions of images. We got an accuracy of 99.69% when training the model of VGG-16, and an accuracy of 99.65% when testing the model. On the other hand, we got lower accuracies in CapsNet and LeNet compared to VGG-16. We got 96.54%, 97.45%, and 94.95% on BSL, ASL, and ArSL respectively while applying LeNet model, while we got 98.4848%, 98.4286%, and 99.5652% on ArSL, ASL, and BSL respectively while applying CapsNet model. Using VGG-16 we got 99.05%, 98.50%, and 99.69% on ArSL, ASL, and BSL respectively.

Keywords: SLI, Capsule Neural Network, Deep Learning, VGG16, LeNet

1. Introduction

Language has two different modalities: nature and Sign language [1]. SL is the second language for human's communication and interactions specially for deaf people. SL is a visual means of verbal communications using gestures expression. Linguistics' expressions are used to express natural language with movement of hands, face, lips articulation, and body movements. Fingerspelling is the main method for translating words using hand movements. According to World Health Organization (WHO) [2], 360 million people are deaf, using about 300 sign languages from different countries. Expecting the growth of this number to 466 SLs by 2020, and by 2050 it will exceed 900 million.

Each country has its own sign language like American Sign Language (ASL), British Sign Language (BSL) and Arabic Sign Language (ArSL). Unlike spoken language SL is not a global sign, so until now, there is no universal sign language unified across all the world to facilitate interaction. An effective Cooperation between the World Federation of the Deaf (WFD) and British Deaf Association (BDA) had done to generalize sign language between all nations, called " Gestuno".

Gestuno is a universal sign language composed of words from various countries like Russia, Great British, United States, and Italy. Unfortunately, Gestuno can't be generalized for global interaction for some reasons, such as no fluent or experts, no-defined grammar for teaching. Finally, it cannot be used by children or ordinary people,

The motivation of our research is using ArSL as it was excluded from previous literatures to identify it from different SLs, as signers in Arabic countries are higher than those of North America and Europe [3]. To the best of our knowledge, previous research has focused on ASL, BSL, GSL and FSL [4] and excluded other SLs. Another motivation is, models used to identify or detect two or more SL were being applied using traditional machine learning, and rare use of deep learning techniques.

Although many Sign Language Identification (SLI) models have been developed, none of them can be used to recognize multiple sign languages. At the same time, in recent decades, the need for a reliable system that could interact and communicate with people from different nations with different sign languages is a great necessity. So, we need to identify and recognize multiple signs of different SLs at the same time. As excluding deaf people and discarding their attendance will affect the whole work progress and damage their psyche which emphasizes the principle of "nothing about us without us".

Various artificial intelligent and machine learning techniques were applied to many benchmarks of sign language with high and low accuracies [5]. Deep leaning has proven how accurate its results are. Also, because the outstanding results of CNN in problems of pattern recognition [6] and image classifications [7]. So, our research applies the two CNN models: VGG16 and LeNet, in addition to CapsNet as position and rotation are important features in our studied problem. This research focuses on using deep learning models to facilitate the process of recognizing and identifying different signs from different languages such as ArSL, BSL, and ASL.

One of the main contributions of our research is the benchmark dataset, which was generated using different videos collected from YouTube. The dataset includes three sign languages: ArSL, ASL, and BSL. The variance of instructors and environments, make our dataset a comprehensive one. So, using these three datasets provided a huge amount of images, contributing to accomplish high results and accuracies, as we will discuss in the upcoming sections. The reason for creating this dataset is the lack of alphabetic dataset in different SLs and the need for a huge number of images, which describe each sign in any of the three sign languages.

Deep learning methods were used for the abundance of our large dataset images. VGG16 was also applied due to its high accuracy, but not respond correctly to images with variance in illuminations, lightening, and rotations. CapsNet is an alternative to CNN [8], CapsNet was applied to overcome previous issues, as it can keep valuable information (hand shape, pose, and location) by excluding maximum pooling layers, it also encodes instantiation parameters, by keeping the relationship between them. Finally, it applied dynamic routing between capsules by agreement [9].

The main target of our research is to recognize more than sign language (alphabet characters) of static hand gestures. As shown in Figure 1, it is a sign of ArSL of character called "Baa".

The remaining of the paper and upcoming sections are listed as following; some of related works and drawbacks to be solved in Section 2. Section 3 describes preprocessing steps to be applied on image datasets which mentioned in Section 4. Section 5 discusses the proposed model. Finally, Section 6 describes results, discussions, and conclusions.

Figure 1. *ArSL sign language of alphabet character called "Baa".*

2. Related Works

Deep Learning (DL) [10] had been widely used in last and recent years due to its great results and the limitations exist in other Machine Learning [ML] algorithms. We will list these limitations in upcoming sections.

In this section we will discuss some of the state-of-the-art works to be investigated and highlighted to discover any research gap. We collected some of related works for the last 10-years related to our used model such as LeNet, VGG16 and CapsNet model, which applied to different SLs datasets. Choosing deep learning models don't require more preprocessing steps and features extraction like using traditional machine learning techniques such as HMM, SVM, and KNN classifiers [11].

According to A. Sultan *et al*, the detection and recognition tasks of different sign languages are based on three main systems. The first is based on glove-based system which contains some of built-in-sensors utilized to capture motion. The second one is vision-based system depends mainly on images captured from digital cameras. The latter is of course much cheaper, but the boom of deep learning is another reason to make it attractive. The final one is based on virtual button.

CNN architecture [7] called dense model which was applied to his own static hand gestures reaching an accuracy of 90.3%. Dataset of alphabet ASL containing more than 50,000 images, collected using same lightening and background environment. VGGNet a deep neural network [11] was applied on multiple sign languages (ASL, ISL (Irish Sign language), and ArASL (Arabic Alphabets Sign Language). Accuracy of 99% was achieved for ASL and BSL, but 98% accuracy was achieved for ArSL. The reason for high accuracy is that ASL and BSL had lower number of classes than ArASL. Enhancement to VGG16 model was proposed [12], by adding 2 dense for a dataset of 5,391

image of 26 English alphabet, measuring the influence of distance between recognition area and the screen, resulting in, that 20 to 40 cm is the best distance for better recognition which shouldn't exceed 80 cm. 99.902% was the training accuracy and 99.910% for testing accuracy.

Applied some of CNN models [13] such as VGG16, ResNet, EfficientNet, and AlexNet to recognize real-time Arabic sign language alphabets. AlexNet was the best one with training accuracy of 99.75% and 94.81% for test accuracy. A.-J. Tanseem N and A.-J. Abu-Jamie used a dataset from Kaggle include alphabets from (A to Z), space, delete, and nothing to predict ASL characters. After 20 epochs of training dataset on pre-trained model of VGG-16 the final accuracy is 100%.

LeNet-5 [15] was implemented to predict to 1500 images for each digit from 0 to 9, which was augmented to produce 3000 images for each sign, which is captured using simple and pure background with webcam. Image processing steps was applied to remove noise such as, color conversion, blurring and sharpening. The total accuracy is 99.8% with 90% as validation accuracy.

MNIST Kaggle sign language of ASL characters was used [16], except Z and J, as it required motion to describe sign. LeNet and CapsNet models were used to recognize signs. LeNet was applied with overall accuracy equals to 82% and CapsNet was applied using two versions of dataset: the MNIST dataset and an augmented version of MNIST dataset, producing accuracy of 88% and 95% respectively. Proposed a CapsNet model [17] instead of traditional CNN models. CapsNet model was able to recognize American digits from 0 to 9 and alphabets from A to Z (excluding j, Z because they required motion capture), giving a testing accuracy of 99.52% for 100*100 RGB input images and 99.94%.

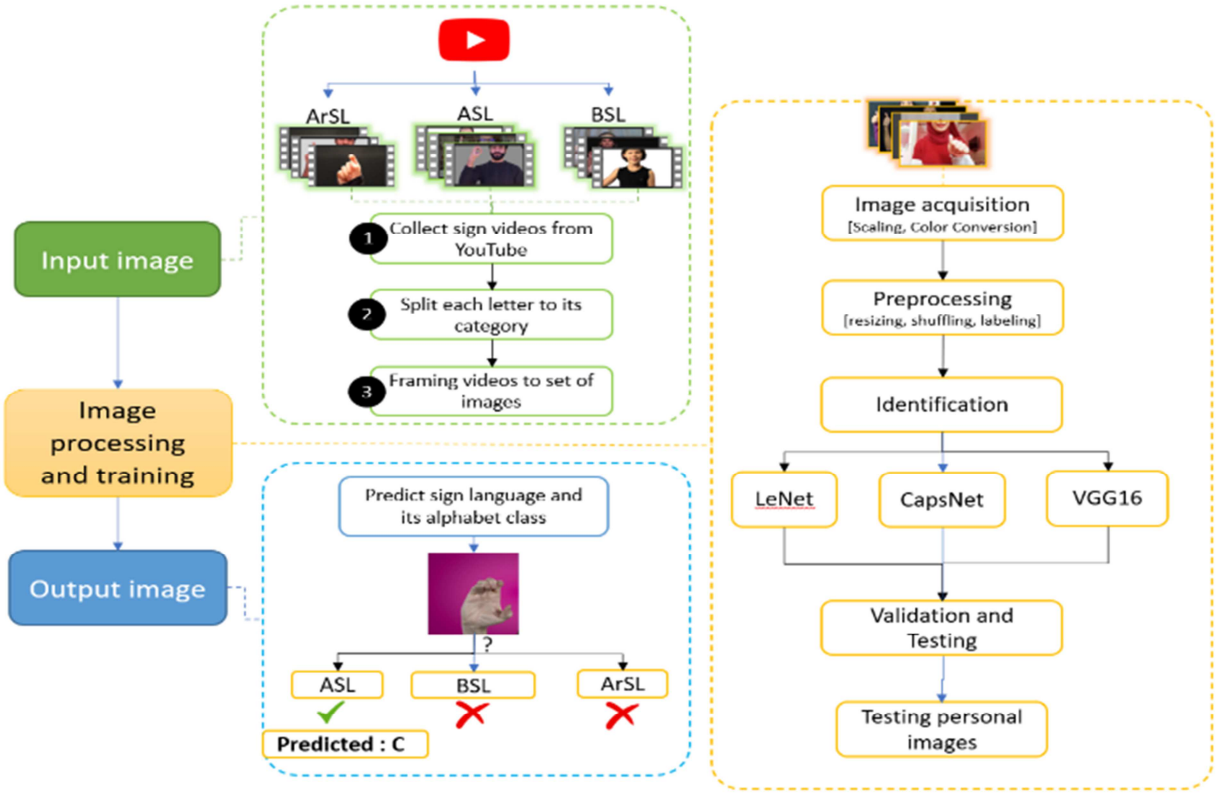


Figure 2. *Proposed workflow of our research.*

For 32*32 RGB on sign language digit dataset. While getting a test accuracy of 99.60% with 28*28 grey images of MINST dataset. Proposed a deep learning capsule model for predicting Indian sign language through signal received from IMU wearable device. Got two accuracies based on number of routings: 99.72 and 99.56 training accuracy for 3 and 5 iterations respectively.

3. Preprocessing Steps

Preprocessing is a very important step if data is not clear and has a noisy or incompatible data, as known that "Garbage in-garbage out". Our dataset is almost clear and give an amazing result, but we applied some preprocessing steps to get higher results.

Hand segmentation is a very important step as signature of alphabetics SL requires movements of hands only. Identifying image pixels which comprise hands and output them as a mask to our proposed model. Many papers had had hand segmentation on different methodology, some are based on color space and others are based on machine learning model [18].

Videos collected from YouTube were montaged and enhanced using Camtasia editor. Segment the most important region of interest ROI, which include human face and hands.

We don't need more image enhancements or more image segmentation algorithms.

We resized images to 64*64 pixel, creating index label for each class, then apply image labels using LabelBinarizer function from sklearn library. Shuffling images was applied also to images and labels followed by image normalization.

4. Dataset and Environment Initialization

Getting dataset compatible with our problem was a great issue. We crawled the web to get signs from different sources, focusing on YOUTUBE as the main source to collect dataset. Dataset has three categories of alphabets like: ASL (American Sign Language), BSL (British Sign Language), and ArSL (Arabic Sign language). Dataset was collected from specialized persons who are interested to learn and guide people how to communicate and contact with deaf people. Those specialized people are certified instructors in different learning centers.

As shown in Table 1, ASL consists of 26 letters collected from 14 signers, resulting in 41,959 images. Also, BSL consists of 61,120 images collected from 26 professional instructors and trainer on YouTube to represent 26 alphabets. ArSL alphabets contains 38,483 images demonstrated by 16 people and trainer separated into 29 classes. Table 2 shows number of each alphabet in our benchmark.

Table 1. *Our own dataset; Asl (American Sign Language), ArSL (Arabic Sign Language), and BSL (British Sign Language).*

SL	Hands	Size (image)	No of signs
ArSL	Single hand	41,959	29
ASL	Single hand	38,483	26
BSL	Double hand	61,120	26

Table 2. *Number of dataset's signs.*

#	ArSL			ASL		BSL	
	Character name in Arabic	Character name in English	Number of images	Character name in English	Number of images	Character name in English	Number of images
1	ع	Aain	1,629	A	1,141	A	1,592
2	أ	Alph	1,323	B	1,556	B	2,428
3	ب	Baa	1,225	C	1,534	C	2,370
4	ض	Dad	1,464	D	1,989	D	2,324
5	د	Dal	1,182	E	1,482	E	1,620
6	ف	Faa	1,423	F	1,586	F	2,560
7	غ	Gain	1,598	G	1,786	G	2,559
8	ج	Gem	1,511	H	1,721	H	2,189
9	ه	Ha	1,377	I	1,444	I	2,160
10	ح	Haa	1,300	J	2,133	J	2,589
11	ك	Kaf	1,117	K	1,775	K	2,714
12	خ	Khaa	1,125	L	1,276	L	2,011
13	ل	Laam	1,271	M	1,577	M	2,529
14	لا	Lam Alef	908	N	1,729	N	2,129
15	م	Meem	1,385	O	1,327	O	1,801
16	ن	Noon	1,382	P	1,765	P	2,833
17	ق	Qaf	1,378	Q	1,902	Q	3,270
18	ر	Raa	1,416	R	1,621	R	2,254
19	ص	Sad	1,288	S	1,642	S	3,277
20	س	Seen	1,210	T	1,643	T	2,588
21	ش	Sheen	1,408	U	1,316	U	1,909
22	ت	Taa	1,107	V	1,101	V	2,235
23	ط	Tah	1,227	W	1,643	W	2,256
24	ث	Thaa	1,107	X	1,886	X	1,984
25	و	Waw	1,549	Y	1,298	Y	2,647
26	ي	Yaa	1,898	Z	2,086	Z	2,292
27	ظ	Zaa	1,117	-	-	-	-
28	ز	Zain	1,319	-	-	-	-
29	ذ	Zal	1,239	-	-	-	-

Environment initialization

Experiments are applied to my personal laptop dell precision M4800 with 16G of RAM using windows 10 as the main operating system. A GPU NIVIDIA Quadro K1100M of 2GB was used. I installed and utilized it with TensorFlow and keras to run my deep learning models LeNet [19] and VGG16 [20] and saved a lot of time. Also used Google-Colab as a second environment to run CapsNet model as my own laptop doesn't have enough GPU capacity to handle and run this model.

5. Proposed Model

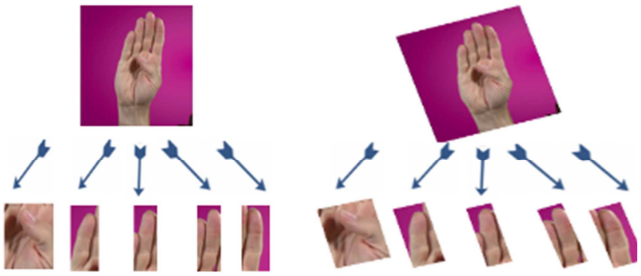
Figure 2 presents the methodology of our work through this research. Starting from the initial step of gathering dataset images to the final step of predicting personal images. The figure depicts the overall process and methodologies applied through three phases; input images, as we described previously in dataset section. Then apply some image preprocessing steps

such as: resizing, shuffling, and image labeling. We don't need more image enhancement algorithms as will be shown in results section. The next step is to choose the best CNN model suitable to our dataset. We applied LeNet model from scratch and used transfer learning to apply VGG16, finally we used CapsNet model, and we will discuss the reason for choosing each one separately in upcoming sections.

5.1. SLI-CapsNet Proposed Model

5.1.1. Rotational Equivariance

CNN models don't respond correctly to rotation or large transformations, so CNN model is not able to classify and recognize rotated images well, because of pooling layers which tends to lose more information. On the other hand, CapsNet model can recognize rotated image easily like human brain. The idea of CapsNet appears in the following example: the main idea is that how CapsNet identify images and recognize it. It captures image and break it out to individual features, such as hand's fingers, Figure 3. Alphabet "B" of ASL with breaking out its features to individuals and rotating them. See Figure 3a (ASL: B character). What about rotating the image by a give angle=30o or any value? Also, what about making image upside down image? This will need new features as shown in Figure 3b. So, this approach seems like a brute search for all possible angle rotation. CapsNet do this easily by a property called "rotational equivariance". Rotation idea can also be updated to scaling, skewing, and thickness. So, the rotated images can be recognized easily. Equivariance can be applied by three steps convolution, reshape functions, and squash function.



(a) Alphabet "B" of ASL, breaking out to individual features

(b) Alphabet "B" of ASL, rotating its individual feature.

Figure 3. *Alphabet "B" of ASL with breaking out its features to individuals and rotating them.*

5.1.2. Dynamic Routing by Agreement

CapsNet applied dynamic routing by agreement [9]. Which tries to use the minimum features required to detect and recognize the hand gesture. If you break out alphabet "B" in Figure 3a and check its label using only one of its low levels such as, the middle finger feature, it will not be enough to predict the gesture. We need to define iteratively more complex features to reach the correct label. So, we give more weightage to the features to recognize labels and can "route" the correct information to feature detector for classification process.

According to [9] the best number for routing iterations is 3, as it gives the minimum loss value and high accuracy. In our implementation it was less than 5. Figure 4 shows the complete architecture of CapsNet model. Figure 4a shows the architecture of CapsNet model starting from the convolution layer which receives the input image. Output is a feature map of an array of size 18 maps, then reshaping process of feature maps into two vectors of 18=2x9, which represents every location in the image, making sure each vector between 0 and 1 because it represents the probability of existence in location of image. The previous check is called "squashing". The primary capsule layer is used to detect objects and which capsule belongs to this object and if the object is located in image. The process is called "routing by agreement" as illustrated previously. Higher capsule layers tend to find weights called "routing weights", finding sum of wights for the first iteration. After first iteration, it tries to predict the output and compare with actual one. Again, we iterate the previous steps with second iteration, and so on of more iterations to find the predicted class correctly.

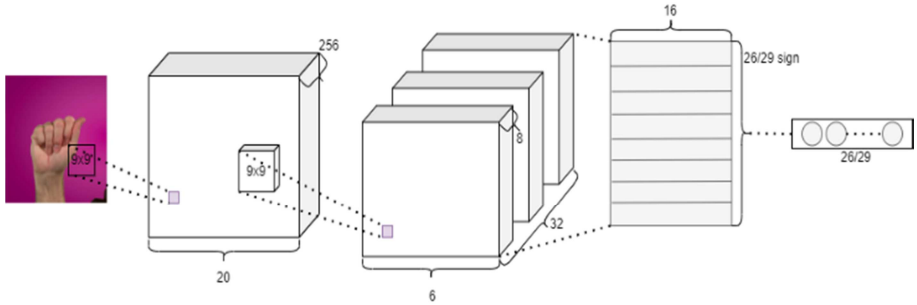
After the routing by agreement complete, we need to compute probability of each class. The original paper [9] used margin loss function to calculate the probability of each class. The vector length is computed using a layer added to the higher layer. The computed squared length of the vector is compared to two values 0.9 and 0.1. If object existed in the image, then length must be less than 0.9 and if it doesn't exist, will be less than 0.1, see Equation (1) and Equation (2).

$$|\mathcal{V}_k|^2 \geq 0.9, \text{ object exist.} \quad (1)$$

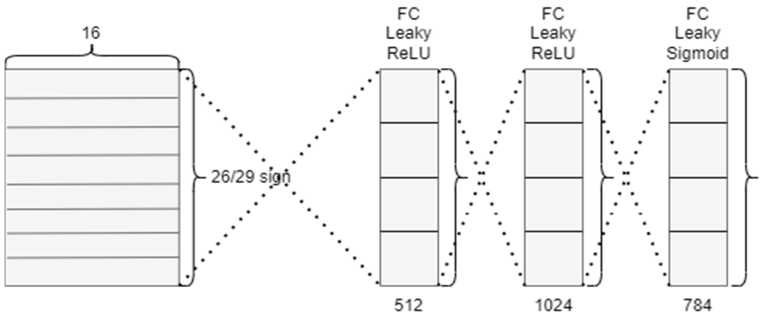
Equation for existence of object in each class.

$$|\mathcal{V}_k|^2 \geq 0.1, \text{ object doesn't exist.} \quad (2)$$

Equation for absence of object in each class.
Where, v_k is the vector length of class of object k.



(a) proposed CapsNet architecture.



(b) Proposed CapsNet decoder architecture.

Figure 4. *Complete architecture of CapsNet Model.*

A decoder network Figure 4b is added to the higher capsule layer. Which is a three fully connected layers. First two layers is activated using leaky ReLU as a new hyper parameter, as the original paper [9] used ReLU function, and sigmoid function for the last one. This decoder is used to reconstruct the input image and compute reconstructed image using the following equation Equation (3). The total loss is calculated using Equation (4). alpha equals 0.0005 as mentioned in [9] which is used to minimize the reconstruction loss.

$$Reconstruction\ Loss = (Reconstructed\ Image - Input\ Image)^2 \quad (3)$$

Equation for computing loss in reconstructed image.

The total loss is calculated using Equation (4): -

$$\text{Total Loss=Margin Loss + alpha} \quad (4)$$

* Reconstruction Loss eq (4) Equation for total loss calculation.

For the activation functions, we had tried three type such as ReLU, Siwsh, and Leaky_ReLU. Swish was the worst one. On the other hand, we got a promising result using Leaky_ReLU. G. B and S. Natarajan mentioned that Leaky_ReLU outperforms other activation functions.

5.2. CNN Architecture

CNN is a Convolutional Neural Network, which excelled in various fields such as image processing, Computer Vision (CV) and Natural Language processing (NLP), image classification, object detection and many fields.

Going deeply in CNN is absolutely necessary to extract more features from confused images. A huge and wide architectures are built based on CNN to enhance CNN performance and get higher results. One of them is LeNet architecture, which is basically consisted of 5 layers with its latest version LeNet-5 [19]. One of our proposed architectures is based on LeNet with some modifications as shown in Figure 5. It has three convolution layers, three maximum pooling layers, and two fully connected layers followed by the layer for output, which is changed to 26 for ASL and BSL and to 29 for ArSL. Image layers were flattened then we applied batch normalization three times before and after each dense layer. Drop out was used by percentage of 0.6 after the first dense layer and percentage of 0.7 after the second dense layer.

Adam optimizer was used for optimization with a loss function such as, categorical cross entropy. Using ReLU as an activation function in dense and convolution layers and SoftMax layer in output layer. The number of epochs and batch size changed with three values of 50, 100 and 150.

5.3. VGG16 Architecture

VGG is based on classical CNN architecture. VGG stands for Visual Geometry Group. Vgg16 [20] is a pretrained model, which has its own weights to be applied to our own model. It has advantages of using over other architecture such as AlexNet, GoogLeNet and ResNet. VGG has a very deep neural network (16-layers) compared to AlexNet, which is 7 layers, So VGG-16 could extract more features. VGG16 was applied to ImageNet dataset [22] which has 1000 class of different categories, reaching a test accuracy of 92.7%. We used transfer learning based on VGG16 to apply this architecture to our benchmark dataset as shown in Figure 6.

Our research method is to apply transfer learning using ImageNet weights. Transfer learning has mainly four types shown in four quadrants as depicted in Figure 7. We used Q1 as it is the similar one to our problem which is large dataset and different from pretrained model's dataset (ImageNet).

All layers were frozen and just change the size of the fully connected layers to be 512, followed by output layer of SoftMax function. Input images to the model of size 64*64 pixel. Dataset was divided to training and testing dataset with percentage of 80% and 20% respectively, then split training dataset to training and validation of 75% and 25% consecutively.

Setting up hyperparameters was not an easy step as it requires more tries to get the best values. Learning rate equals to 1e-5, activation functions used in the fully connected layers is ReLU. The output layer is updated to 26 for ASL and BSL and to 29 for ArSL using SoftMax as an activation function. RMSProp is used as an optimizer with categorical cross entropy as a loss function with 25 epoch and 128 batch size. The fully connected layers were followed by dropout and batch normalization. The dropout layers equal to 40% and 30% for the FC layers consecutively. Table 4 shortens the hyperparameters values for the deep learning models used for training the three datasets.

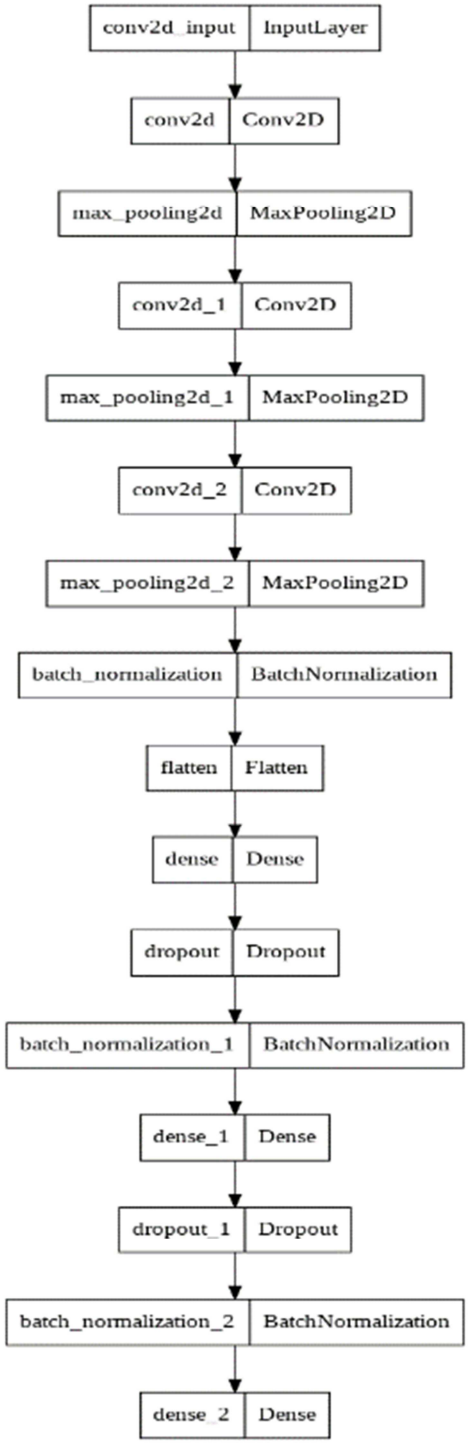


Figure 5. *CNN LeNet proposed architecture to predict ASL, ArSL, and BSL.*

6. Results Comparisons

This section compares and visualize our model's results with other state-of-the-arts. In order to test our model functionality and accordance with dataset, Table 3 shows different accuracies of applying LeNet, VGG16 and CapsNet models. We noticed that VGG-16 is the highest accuracy once trained on BSL dataset. Concluding that the huge number of BSL image cause this highest accuracy between different datasets. While other works such as K. Suri and R. Gupta performs high accuracy than our research, but it has a drawback, as they used a hand device.

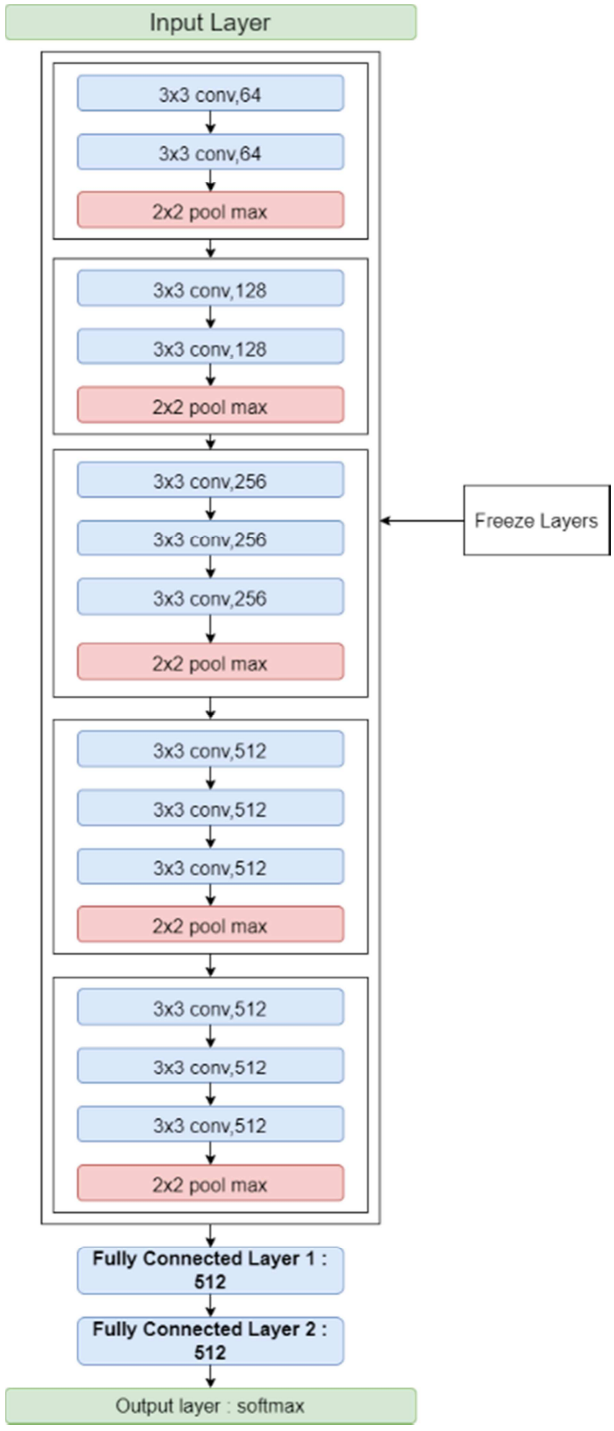


Figure 6. *VGG16 architecture used as transfer learning.*

Also [12] outperforms other state-of-the-arts but used less numbers of dataset images of 5,391 images, which is not enough for training and testing, comparable to our dataset size.

6.1. Experiments and Results of LeNet

We applied LeNet model with different number of epochs

and batch sizes. We noticed that low number of epochs and low batch size of ArSL with huge number of images produce low loss value and satisfied accuracy of 94.95%. On the other hand of BSL, increasing epochs and batch size, led to increasing of accuracy for the same number of image dataset. We got 97.45% and 0.0468 for loss, with epochs and batch size of 100. To increase results of accuracy and decrease loss, we used 150 as number of epochs and batch size. For BSL, we used an epoch and batch size of 150, to get less results with different sizes of images.

Q1	Q2
Large dataset different from the pretrained model's dataset.	Large dataset and similar to the pretrained model's dataset.
Q3	Q4
Small dataset and different from the pretrained model's dataset.	Small Dataset and similar to the pretrained model's dataset.

Figure 7. *Depicts the four quadrants of transfer learning.*

6.2. Experiments and Results of VGG16

Vgg16 was the best one for training and testing our dataset, getting the higher accuracy on BSL dataset 99.69%, because of its large number of images and a smaller number of loss values comparable to LeNet model. ArSL gives an accuracy of 99.05%, then ASL produces an accuracy of 98.5%. Also, we conclude, that VGG16's loss values are lower than LeNet model.

6.3. Experiments and Results of CapsNet

CapsNet model was one of the models used to predict and classify images specially, images with different transformations. CapsNet accuracy was very high compared to LeNet, and loss values were very low with mentioned to LeNet and VGG16. We got high accuracy of 99.56%, then 98.4848% for ArSL, 98.4286% for ASL. Concluding that large datasets led to high accuracies and low loss values.

For predicting the language category, we used some samples of each alphabet in each language to form the category of each language. So, we got 3,737 images of ArSL, 3,827 images of ASL, and 5,568 of BSL. We trained and tested these images using VGG16 only, as it was the highest accuracy to identify sign language alphabets. We trained our new images and got training accuracy of 99.99%, and 99.9 for testing. Epochs number equals 10 and batch size equals 128. Figure 8 shows loss and accuracy curves for training images. Also, figure 9 shows the true and predicted images of each sign language.

Table 3. *Results of our research compared to other results. [ASL (American Sign Language), BSL (British Sign Language), ArSL (Arabic Sign Language), ArASL (Arabic Alphabet Sign Language), IrSL (Irish Sign Language), InSL (Indian Sign Language)].*

Author	Method	Dataset	Device	Size (image)	Classes	batch size /epochs	Loss	Training Accuracy
Our own	LeNet	ArSL	Camera	26,100	29	50 / 50	0.1600	94.95%
		ASL	Camera	13,000	26	150 / 150	0.0468	98.43%
			Camera	13,000	26	100 / 100	0.0788	97.45%
			Camera	13,000	26	150 / 150	0.0387	98.79%
		BSL	Camera	18,200	26	150 / 150	0.1536	95.02%
			Camera	50,906	26	150 / 150	0.1124	96.54%
	VGG-16	ArSL	Camera	41,959	29	128 / 25	0.0358	99.05%
		ASL	Camera	38,483	26	128 / 25	0.0649	98.50%
		BSL	Camera	61,120	26	128 / 25	0.0135	99.69%
	CapsNet	ArSL	Camera	26,100	29	50 / 10	0.026618	98.4848%
		ASL	Camera	23,400	26	50 / 10	0.033218	98.4286%
		BSL	Camera	36,010	26	50 / 10	0.012218	99.5652%
[11]	VGGNET	ASL	Camera	58,114	29	32 / 40	-	99%
		ArASL	Camera	54,049	32	32 / 40	-	98%
		IrSL	Camera	58,120	26	32 / 40	-	99%
[12]	VGG-16	ASL	Camera	5,391	26	- / 50	-	99.902%
[13]	VGG-16	ArSL	Camera	54,049	32	32 / 30	0.1957	94.05%
[14]	VGG-16	ASL	Camera	43,500	29	- / 20	0.0020	99.99%
[16]	LeNet	ASL	Camera	34,627	24	128 / 30	-	82%
	CapsNet		Camera			128 / -		95%
[17]	CapsNet	ASL	Camera	34,627	34	50 / -	-	-
[23]	CapsNet	InSL	IMU Device	2000 sentence	20 sentences	- / -	0.01	99.72%

Table 4. *Hyper parameters of the trained models.*

Hyperparameters	LeNet Model	VGG16	CapsNet
Activation Function	ReLU/ SoftMax	ReLU/ SoftMax	Leaky ReLU
Learning Rate	-	1e-5	-
Epochs	150	25	10
Batch Size	150	128	50
Loss Function	Sparse categorical cross entropy	Categorical Cross Entropy	Margin loss (MSE)
Optimizer	Adam	RMSprop	Adam

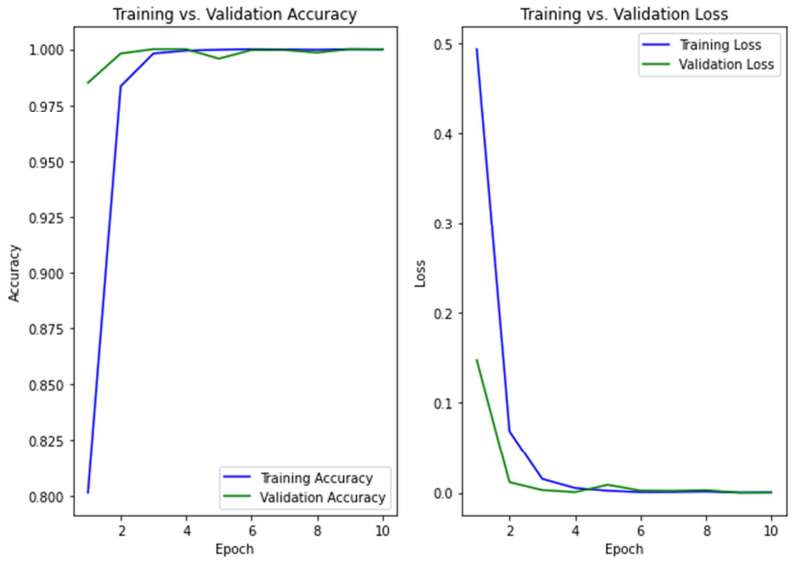


Figure 8. *loss and accuracy of ASL, BSL, and ArSL using VGG16.*

7. Conclusion

In our paper we applied a different deep learning models to recognize and identify static hand gestures of ArSL, BSL, and ASL. The LeNet was developed from scratch to train and test every one of our datasets. Transfer learning was implemented using VGG16 which gives the highest accuracy especially on BSL dataset. Also, to overcome image rotation, scaling, and different transformations of image, we applied Capsule network which is a deep learning model depending on concept of Routing by agreement. Our datasets perform perfectly on the three models, which emphasizes that our dataset is a good one different applications of sign language. As a future work, U-net model can be used to preprocess incoming images to be identified more efficiently. Recognize and identify real-time sign language images and videos using deep learning models.

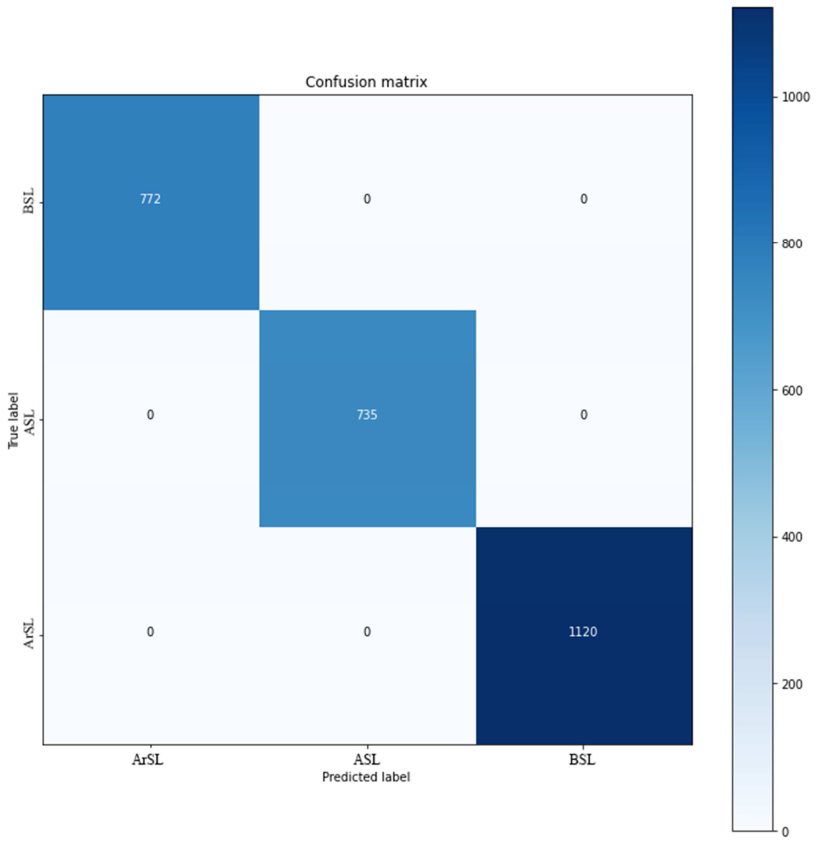


Figure 9. *Confusion matrix of true and predicted images in ASL, BSL, and ArSL.*

Data Availability

The dataset used is available for any researcher upon a reasonable request to the paper author.

References

[1] J. Quer and M. Steinbach, "Handling sign language data: The impact of modality," Frontiers in Psychology, 2019.

[2] R. Kushalnagar, "Deafness and hearing loss," Human–Computer Interaction Series, pp. 35-47, 2019.

[3] B. I. II, "Deafness in the Arab world: a general investigation, with applications to Lebanon," scholarship.tricolib.brynmawr.edu, 2018.

[4] C. D. Monteiro, C. M. Mathew, R. Gutierrez-Osuna and F. Shipman, "Detecting and identifying sign languages through visual features," 2016 IEEE International Symposium on Multimedia (ISM), 2016.

[5] A. Sultan, W. Makram, M. Kayed and A. A. ALi, "Sign language identification and recognition: A comparative study," Open Computer Science, pp. 191-210, 2022.

[6] M. Mohandes, Junzhao Liu and M. Deriche, "A survey of image-based Arabic sign language recognition," 2014 IEEE 11th International Multi-Conference on Systems, Signals & Devices (SSD14), 2014.

[7] R. Daroya, D. Peralta and P. Naval, "Alphabet sign language image classification using deep learning," TENCON 2018 - 2018 IEEE Region 10 Conference, 2018.

[8] E. Goceri, "Analysis of capsule networks for Image Classification," Proceedings of the 15th International Conference on Computer Graphics, Visualization, Computer Vision and Image Processing (CGVCVIP 2021), the 7th International Conference on Connected Smart Cities (CSC 2021) and 6th International Conference on Big Data A, 2021.

[9] S. Sabour, N. Frosst and H. Geoffrey E, "Dynamic routing between capsules," Advances in neural information processing systems, 2017.

[10] Y. LeCun, B. Yoshua and H. Geoffrey, "Deep learning," nature, vol. 521, no. 7553, pp. 436-444, 2015.

[11] A. El Zaar, N. Benaya and A. El Allati, "Sign language recognition: High performance deep learning approach applyied to multiple sign languages," E3S Web of Conferences, 2022.

[12] H. T. Nguyen, L. T. Pham, T. T. Mai, T. K. Vo and T. T. Dien, "Letter recognition in hand sign language with VGG-16," Intelligent Systems and Networks, pp. 410-417, 2022.

[13] Z. Alsaadi, E. Alshamani, M. Alrehaili, A. A. Alrashdi, S. Albelwi and A. O. Elfaki, "A real time Arabic Sign Language Alphabets (arsla) recognition model using deep learning architecture," Computers, vol. 11, no. 5, p. 78, 2022.

[14] A.-J. Tanseem N and A.-J. Abu-Jamie, "Classification of Sign-language Using VGG16," 2022.

[15] V. Shreya and Y. Shaik Sohail, "Sign Language Interpreter Using Computer Vision and LeNet-5 Convolutional Neural Network Architecture," International Journal of Innovative Science and Research Technolog, vol. 6, 2021.

[16] M. Bilgin and K. Mutludogan, "American sign language character recognition with capsule networks," 2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), 2019.

[17] H. Xiao, Y. Yang, K. Yu, J. Tian, X. Cai, U. Muhammad and J. Chen, "Sign language digits and alphabets recognition by Capsule Networks," Journal of Ambient Intelligence and Humanized Computing, vol. 13, no. 4, pp. 2131-2141, 2021.

[18] A. M. B. S. A. K. M. B. and M., "Different techniques of hand segmentation in the real time," IJCAIT, pp. 45-49, 2013.

[19] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard and L. Jackel, "Backpropagation applied to handwritten zip code recognition," Neural Computation, pp. 541-551, 1989.

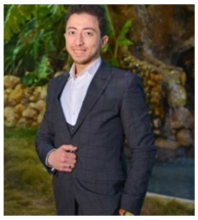
[20] S. Karen and Z. Andrew, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv: 1409.1556, 2014.

[21] G. B and S. Natarajan, "Hyperparameter optimisation for Capsule Networks," EAI Endorsed Transactions on Cloud Systems, 2019.

[22] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009.

[23] K. Suri and R. Gupta, "Continuous sign language recognition from wearable Imus using deep capsule networks and game theory," Computers & Electrical Engineering, vol. 78, pp. 493-503, 2019.

Biography



Ahmed Mahmoud Sultan is currently pursuing the M.Sc. degree from faculty Computers and Information, Minia University, Minia, Egypt. He received his bachelor's degree from Computer Science department, faculty of computers and artificial intelligence, Beni-Suef University, Egypt in 2017. Since 2018 he is a teaching assistant at the same faculty where he received his bachelor's degree. His research interests include image processing, artificial intelligence, machine learning applications and deep learning.

Mohammed Kayed received the M.Sc. degree in Computer Science from Minia University, Minia, Egypt, in 2002 and the Ph.D. degree in Computer Science from Beni-Suef University, Beni-Suef, Egypt, in 2007. From 2005 to 2006, he was a Research & Teaching Assistant in Department of Computer Science and Information Engineering at the National Central University, Taiwan. He is currently a professor and the dean of faculty of computers and artificial intelligence, Beni-Suef University, Egypt. He is the author of more than 40 articles. His research interests include Web mining, Opinion Mining, Information Extraction and Information Retrieval.

Waleed Makram Mohamed Zaki received his Ph.D. degree in information system from the Minia University, Egypt in 2018. He is currently a CIO of Minia University, Director of the Information Technology Center, Director of the Center for Management Information Systems (MIS), and Lecturer at the Faculty of Computers and Information, Information System Dept. He has worked on several research topics. Waleed has contributed technical papers in the areas of Big Data, Data Mining, Data Warehouse and, and Machine learning in international journals.

Abdel Mgeid Amin Ali is currently a Professor and the dean of faculty of computers and Information, Minia University, Minya, Egypt. He has published over 100 research papers in prestigious international journals and conference proceedings. He has supervised over 60 Ph.D. and M.Sc. students. His research interests include information retrieval, software engineering, image processing, data security, metaheuristics, the IoT, digital image steganography, and data warehousing.