Independent video-based Arabic Sign Language Recogition

Mostafa Balaha[1], Sara Elkady[2], Mohamed Salama[3], Eslam Emad[4], Muhammed Haassan[5], Hossam Balaha[6], Mahmoud Saafan[7]

[1] Undergraduate student at Faulty of Engineering, Mansoura University, Egypt
Mostafa19902@gmail.com
[2] Undergraduate student at Faulty of Engineering, Mansoura University, Egypt
s.elkadi97@gmail.com
[3] Undergraduate student at Faulty of Engineering, Mansoura University, Egypt
mohamed_ahmed6564@yahoo.com
[4] Undergraduate student at Faulty of Engineering, Mansoura University, Egypt
eslam.eimad@gmail.com
[5] Undergraduate student at Faulty of Engineering, Mansoura University, Egypt
muhammad.hassan12797@gmail.com
[6] Teaching Assistant at Faulty of Engineering, Mansoura University, Egypt
hossam.m.balaha@mans.edu.eg
[7] Assistant Professor of Computer Engineering, Mansoura University, Egypt
Dr. Mahmoud's email

Abstract. Over 5% of people around the world are deaf, they have difficulties to communicate with normal people. They face a real challenge to express anything without an interpreter to their signs. Nowadays there are a lot of studies around Sign Language recognition which aims to reduce this gap between deaf and normal people as it can replace the need of the interpreter. Even though there are a lot of challenges facing the signs' recognition system e.g. low accuracy, complex gestures, high noise, no standard for the videos, etc. Due to the previous reasons a lot of studies are made trying to solve these problems, however, there is a problem that could stop anyone from proceeding namely the data, as every language on earth has its own signs this would be a great challenge to cover all of the signs of all language. As for Arabic sign language, there was no real dataset available up to now. In order to solve this issue we have made the first dataset for Arabic words and presented a new CNN-RNN model that could achieve high accuracy recognizing normal life videos without the need of standards for each video e.g. same background, same clothes, etc. This model was able to achieve over 98% accuracy on the dataset and over 90% accuracy on UCF-101 dataset. The purpose of this paper is to introduce a new Arabic sign language dataset of words and a new architecture of CNN-RNN that could achieve high accuracy in action recognition specially sign language.

Keywords: Sign language recognition, Convolutional Neural Network, Recurrent Neural Network, Bidirectional Long Short-Term Memory.

I. Introduction

Key challenges in sign language are that: the minimal difference between each sign, the variance of viewpoint for the same sign, different persons and environments, complex gestures, facial expressions and the non-symmetric signs in each language as there are unique signs for each language. These challenges could complicate the communications between deaf and normal people, also they remain a great barrier for automatic recognition systems.

With the great development in computers' computations and machine learning methods and deep learning specifically combination of RNN with CNN architectures have achieved a great success and gained high attention in the process of classifying videos. Mainly this process has two operational steps, first is to extract descriptors or features vectors out of frames using CNN, these descriptors play a great role as they keep information of sequential frames as sequence, which is made for each video and the second step take place and LSTM show up trying to find the relation between these sequences and classify the videos based on these relations.

Motivated by this approach we could reduce the barrier of these systems, so in this work we are contributing in the field of sign language recognition specially in Arabic sign language recognition by presenting the biggest Arabic sign language dataset up to now and a CNN-RNN architecture which has achieved a very high accuracy; not only on our dataset but also on UCF-101 dataset.

II. Related work

Sign Language recognition (SLR) was recognized by different researchers with different methods. In this section, we introduce a quick review of several works and different methods through years. The research began in the 1990's. 1988, Tamura et al. [1] Assumed the sign word is composed of a time sequence of units called cheremes which consists of hand shape, movement, and the location of the hand. They expressed the 3-D features of these factors and converted it into 2-D image features, and classified the motion image of sign language with the 2-D features. Keskin et al. [2] created realistic 3D hand models that represent the hand with 21 different parts and trained Random decision forests (RDF). They use RDF to perform per pixel classification and assign each pixel to a hand part, which is then fed into a local mode finding algorithm to estimate the joint locations for the hand skeleton. They also describe a support vector machine (SVM) to recognize ASL digits based on this method and achieve a high recognition rate on live depth images in real-time. Nandy et al. [3] created a video database for various signs of Indian Sign Language. They used the direction histogram, which appeals for illumination and orientation invariance, as the feature for classification. They used two different approaches for recognition which are Euclidean distance and k-nearest neighbor metrics.

Sensors and gloves based models, Mehdi et al. [4] has used 7-sensor glove of 5DT Company which is used to get the input data of hands movement with ANN which is used as a classifier to recognize the signs gestures. They have achieved an accuracy of 88%. López-Noriega et al. [5] has followed the same approach of [4] and also offered a GUI made with .NET.

Hidden Markov Model (HMM) based models, using gloves or images as an input for HMM, it was used widely and worked effectively in continuous and real time SLR tasks. Starner et al. [6] proposed a recognition method based on the Hidden Markov Model (HMM). They used color gloves to capture hand shape, orientation, and trajectory. They represent HMM-based systems for recognizing sentence level American Sign Language (ASL), managed to get a high word accuracy. Hienz et al.[7] have used colored cotton gloves to make it easy to extract features and then convert the sequence of videos into feature vectors and then fed to HMM in order to classify them. They have achieved an accuracy around 92% to 94%. The same approach also used in [8] and [9].

As a brief, these previous approaches were able to achieve high accuracy but these approaches can’t be used in real daily life as they require the wearing of gloves and the fixed environment which isn’t natural. Actually many of them are user dependent which means in order to generalize it, it must be trained on each user which isn’t logical and unnatural. Due to the previous reasons, Aliaa et al. [10] tended to generalize and proposed a model based on HMM that doesn’t depend on users or require gloves, but it falls into the trap of low accuracy around 82%.

Convolutional neural network (CNN) based models, with the extreme success of CNN in the field of image recognition and classification, researchers paid attention for using it with SLR. S. Masood et al. [11] proposed CNN model for ASL’s characters recognition. They are able to use CNN to achieve an overall accuracy of 96% on 2524 ASL gestures image dataset. Following same approach [12], [13] and [14] each has offered a CNN architecture to classify different languages signs alphabet with accuracies 99%, 82.5% and 100% respectively.

CNN-RNN based models, as CNN was keeping the information of 1 frame only at a time, by coupling it with RNN, both were able to keep information over time. Due to this ability, dynamic signs were able to be recognized more accurately. Yang et al. [15] proposed an effective continuous CSL recognition method, which is based on the combination of CNN and LSTM. They achieved remarkable accuracy in the experiments on their self-built dataset.

3D Convolutional Neural Network (3D-CNN) based models, instead of 2D-CNN which require another phase of RNN in order to keep information over time, 3D-CNN was able to take multi-frames of a video at once, this helped to learn the sequence between frames without the need of RNN. The work in [16] and [17] propose models based on this approach.

The approach proposed in this paper is based on “CNN-RNN” approach, specifically we use double CNN as features extractors and for RNN, BI-LSTM layers are used in order to identify the complex sequence in videos in order to avoid to problem of conflicts between different classes as almost of the signs have the same gestures with slightly difference between each sign.

III. Datasets

There is no much dataset available for Arabic sign language recognition tasks. And the available dataset is either letters or words based on specific condition such as user must wear gloves or maybe static words that can be recognized by single image. So we decide to collect the needed dataset to train our model. Our main goal when collected this dataset is to fit natural circumstances and environment, as we intended to depend on videos that anyone can easily create like. So based on statistics by 2020 almost everyone has his own smartphone with a camera, following this concept we capture our dataset using smartphone cameras without using any stabilizing tool either hardware or software. We have recorded around 8466 videos for 20 signs by 72 volunteers, the criteria we have followed is that, each volunteer has to do each sign for at least 5 times so at least around 100 videos from each volunteer. The volunteers’ age range from 20 to 23, and all of them from Faculty of Engineering, Mansoura University. Fig. 1. Shows a samples from different videos. Table 1. Shows each sign with corresponding number of each video’s class.

IV. preprocess

In this section we preview the preprocess stages made on the raw data. As mentioned in the previous section, our dataset’s videos were captured by mobile camera not a professional camera or even a fixed camera; so the videos had a huge amount of noise that we never want. By following the rule of features selection, we had to find a suitable way to extract just the necessary movement out from each frame, so the model be able to generalize on any signer under any circumstances.

.

Figure 1, Samples from our Dataset

Table 1, Shows number of videos for each class

Word	baby	eat	father	finish	good	happy	hear	house	important	love	mall	me	mosque	mother	normal	sad	stop	thanks	thinking	worry
الكلمة	طفل	ياكل	أب	ينهي	جيد	سعيد	يسمع	بيت	مهم	يحب	مول	انا	مسجد	أم	عادي	حزين	توقف	شكرا	يفكر	قلق
No.	430	410	451	440	436	445	433	421	446	435	414	430	427	406	410	420	426	412	366	409

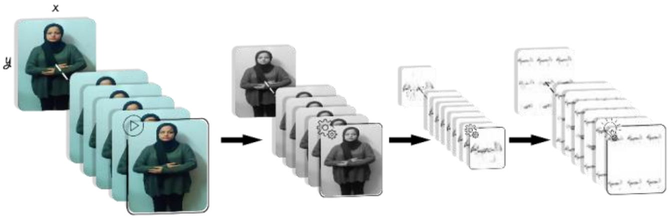


Figure 2, Preprocess overview

Raw video passes through 3 stages before it could be used with our model, first is to reduce its dimensions and to be converted into grayscale, the benefits behind this stage is to reduce the calculations time and less complexity. Output of this stages is then passed to a difference function shown in Figure 3. Function take two consecutive frames and produce just a frame holding the only the important data out from both frames. By applying this to the whole video's frames, we will end with (n-1) frames. The following and the last stage is about unifying each class features and add a unique factor to each class's videos. Output is only 30 frames out from (n-1) frames but each frame here in the stage holding 9 frames from the previous stage, Figure 5. These frames aren't selected randomly but instead it's related to index of the currently formed frame.

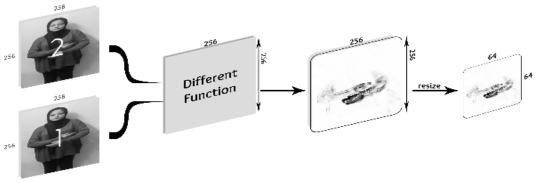


Figure 3, Difference function

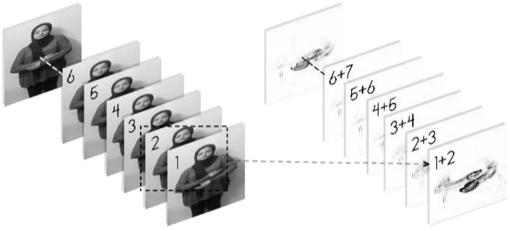


Figure 4, Second stage

Figure 4, shows the stage of applying the difference function on the whole video's frames.

A further more information about preprocess' last stage, the purpose of this stage is to reduce redundancy but at the same time without dropping any frame and keeping all information of all frames just in 30 frames. This could reduce conflicts between signs of similar movements' positions but with different operation's sequence as these frames keep the hands' positions through the time.

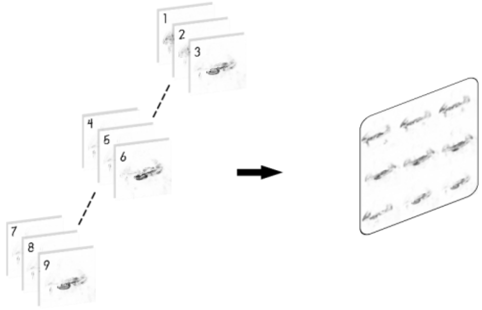


Figure 5, Last stage, combining frames

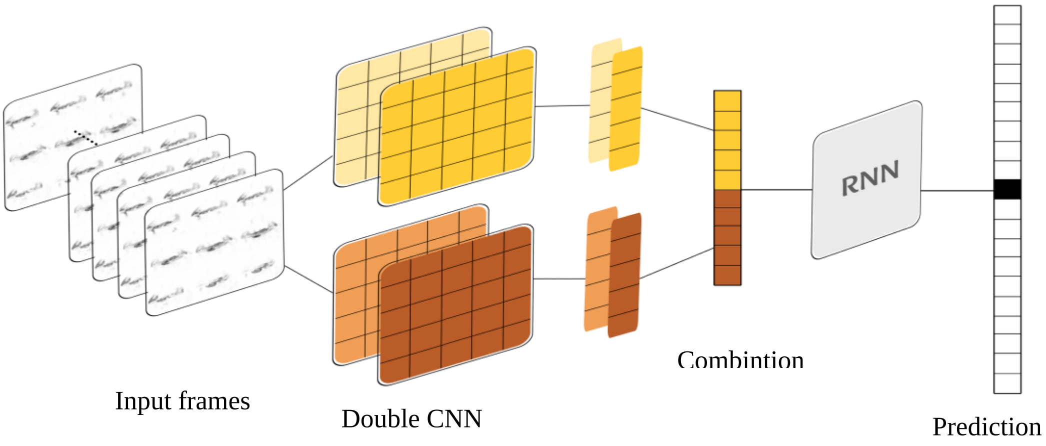


Figure 6, Architecture overview

V. Architecture

A. Architecture Overview

This paper contribute with an architecture for recognizing videos and classify them in video classification field specifically sign language recognition. The main idea behind our model is to train two different CNN independently using the same architecture but on different portions of data. By concatenating the output from each CNN into one vector which is then passed to RNN, which has a great ability to identify sequences in videos. This approach could help the network to identify different features for the same input and improves its overall confidence.

As Fig. 6 shows an overview of the model. Input frames are the result of preprocessing stage, are then passed to both CNNs. After the training each one of them could recognize the input in two different ways with different features, so we combine different features all together into just one vector of size (1 x 512) for each frame, at the end of the video we gain a sequence of (30 x 512) for each video. By applying this on each video included in dataset, we are ready now to train the RNN by the use of these generating sequences. RNN accurately learn the changes over time in each sequence and able to generalize it over the classes.

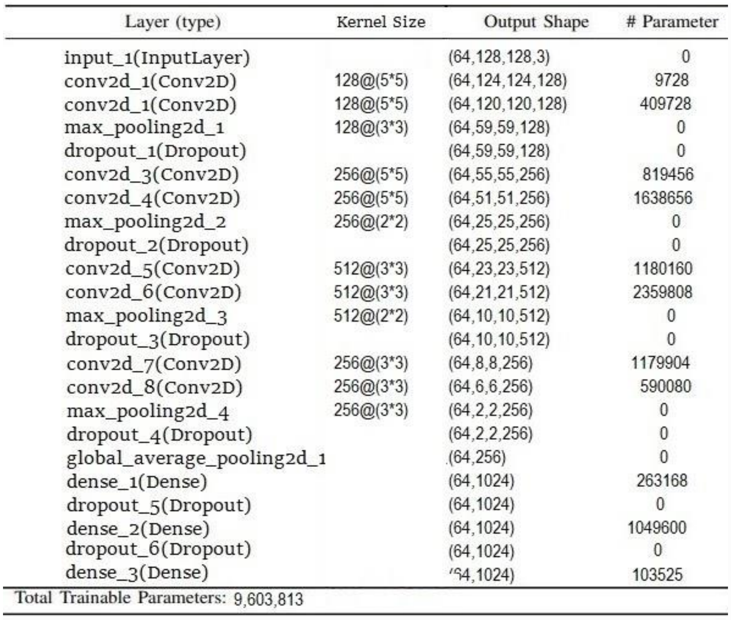
B. Convolutional Neural Network

As mentioned before CNN is used to extract spatial features. CNN consists of 4 blocks, global average pooling layer and prediction network, each block has 2 Conv. Layers, 1 pooling layer and followed by a dropout layer to reduce overfitting and increase its ability to generalize. All blocks almost have the same dimensions except for depth as follows 128, 256, 512 and 256 in order. Prediction network is about 3 layers of Fully Connected (FC) layers. It takes the output of the CNN network and use them to classify the input to its class, this step here is more important to the CNN itself rather than the model as we take the features out from the global average pooling layer. First two FCs contain 1024 neuron each one followed by dropout layer. Final FC layer contains 20 neuron with Softmax activation function.

Table 2, shows the exact output dimensions and Parameter of each layer of each block.

C. Recurrent Nwural Network

Table 2, CNN Architecture



Layer (type)	Kernel Size	Output Shape	# Parameter
input_1(InputLayer)		(64,128,128,3)	0
conv2d_1(Conv2D)	128@(5*5)	(64,124,124,128)	9728
conv2d_1(Conv2D)	128@(5*5)	(64,120,120,128)	409728
max_pooling2d_1	128@(3*3)	(64,59,59,128)	0
dropout_1(Dropout)		(64,59,59,128)	0
conv2d_3(Conv2D)	256@(5*5)	(64,55,55,256)	819456
conv2d_4(Conv2D)	256@(5*5)	(64,51,51,256)	1638656
max_pooling2d_2	256@(2*2)	(64,25,25,256)	0
dropout_2(Dropout)		(64,25,25,256)	0
conv2d_5(Conv2D)	512@(3*3)	(64,23,23,512)	1180160
conv2d_6(Conv2D)	512@(3*3)	(64,21,21,512)	2359808
max_pooling2d_3	512@(2*2)	(64,10,10,512)	0
dropout_3(Dropout)		(64,10,10,512)	0
conv2d_7(Conv2D)	256@(3*3)	(64,8,8,256)	1179904
conv2d_8(Conv2D)	256@(3*3)	(64,6,6,256)	590080
max_pooling2d_4	256@(3*3)	(64,2,2,256)	0
dropout_4(Dropout)		(64,2,2,256)	0
global_average_pooling2d_1		(64,256)	0
dense_1(Dense)		(64,1024)	263168
dropout_5(Dropout)		(64,1024)	0
dense_2(Dense)		(64,1024)	1049600
dropout_6(Dropout)		(64,1024)	0
dense_3(Dense)		'54,1024)	103525
Total Trainable Parameters: 9,603,813			

Recurrent Neural Networks (RNNs) use the information in the sequence for the recognition tasks. Traditional RNNs suffer from vanishing gradients which caused them not to learn so much. Long Short-Term Memory (LSTM) is a variant of RNN, which designed to efficiently solve the vanishing and exploding gradients problems. Bidirectional LSTMs are an extension of traditional LSTMs which improve model performance on sequence classification problems. BiLSTMs train two instead of one LSTMs on the input sequence, when all time steps of the input sequence are available. This can provide additional context to the network and result in faster and even fuller learning on the problem.

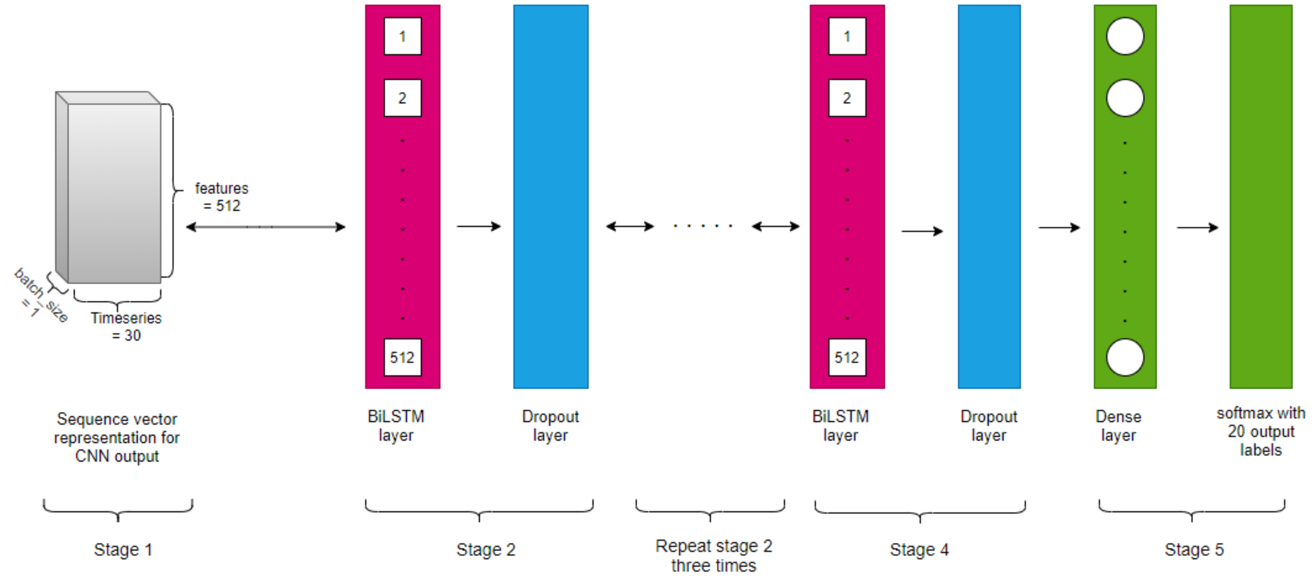


Figure 7, Architecture of the proposed RNN model

In our model, the output combined from two CNNs is fed to five layers of 512 BiLSTM units. Every one of these layers followed by dropout layer with dropout rate 0.9, to avoid the network overfitting. These layers is followed by fully connected layer with softmax activation, which is used to predict the output. To train the model, we used Adam technique, which is an optimization algorithm that used to update network weights iterative based in training instead of the classical stochastic gradient descent procedure. Adam has the advantage that it can handle spare gradients on noisy problems. We used Adam optimizer with 10^{-4} learning rate and 10^{-6} decay rate.

Decreasing the number of BiLSTM layers with keeping the same number of BiLSTM units is experimented, and using only 3 BiLSTM layers with 2048, 1024, 2048 units respectively is also experimented. We tested these on our dataset and found that 5 BiLSTM layers with 512 hidden units is performed best.

VI. Experminted Results

The experimental results are shown in Figure 8 and Table 3. Figure 3 shows the accuracy increasing and loss decreasing with 20 epochs over training and validation data. Our model achieved over 98% on validation and 91% on test as 1st prediction and around 97% on test as 2nd prediction using Top-N accuracy matrix. Table 3 shows F1 score per class for validation and test sets. By analyzing performance of classes using confusion matrix as shown in figure 9, we found that there are two pair of classes make most of conflict. These classes are “Thanks” with “Thinking” and “Good” with “Hear”. This happen because they have a great chance of similarity in their sequence.

VII. Concolution

In this paper, we proposed an effective continuous CSL recognition method, which is based on the combination of CNN and LSTM. With the powerful feature extraction of CNN, and the ability of LSTM network to learn from contextual information, this method achieved remarkable accuracy in the experiments on our self-built dataset. And the CNNLSTM model has been proved suitable for continuous SLR without any external devices.

References

[1] Tamura, S. and Kawasaki, S., “Recognition of sign language motion images”, Pattern Recognition, 21(4), 343-353 (1988)

[2] Keskin, C., Kirac, F., Kara, Y.E., and Akarun, L., "Real time hand pose estimation using depth sensors", Proceedings of the IEEE International Conference on Computer Vision, 1228-1234 (2011)

[3] Nandy, A., Prasad, J.S., Mondal, S., Chakraborty, P. and Nandi, G.C, “Recognition of isolated Indian Sign Language gesture in real time”, Inf. Process. Manag., 102–107 (2010)

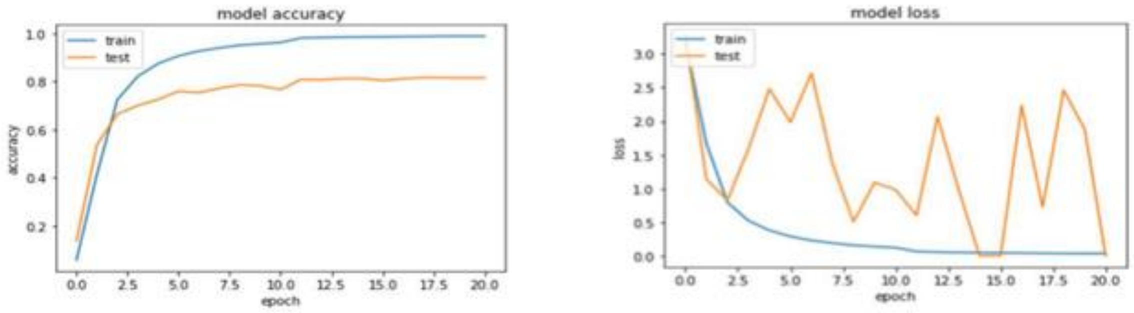
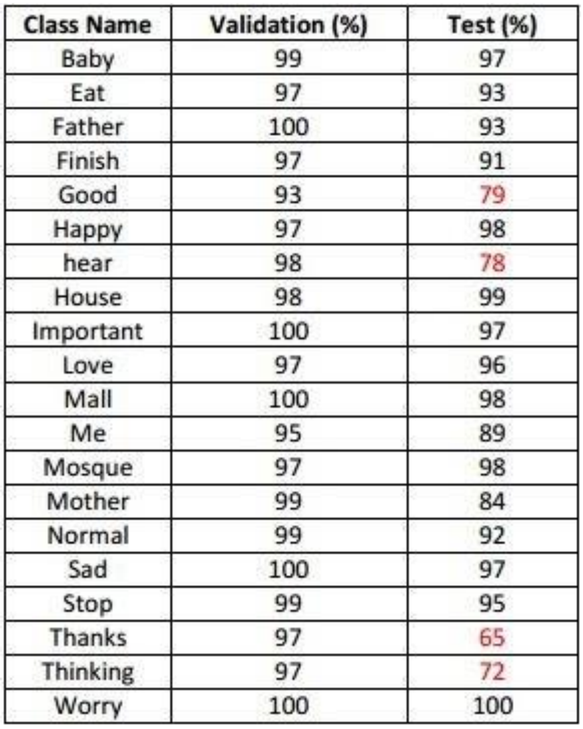
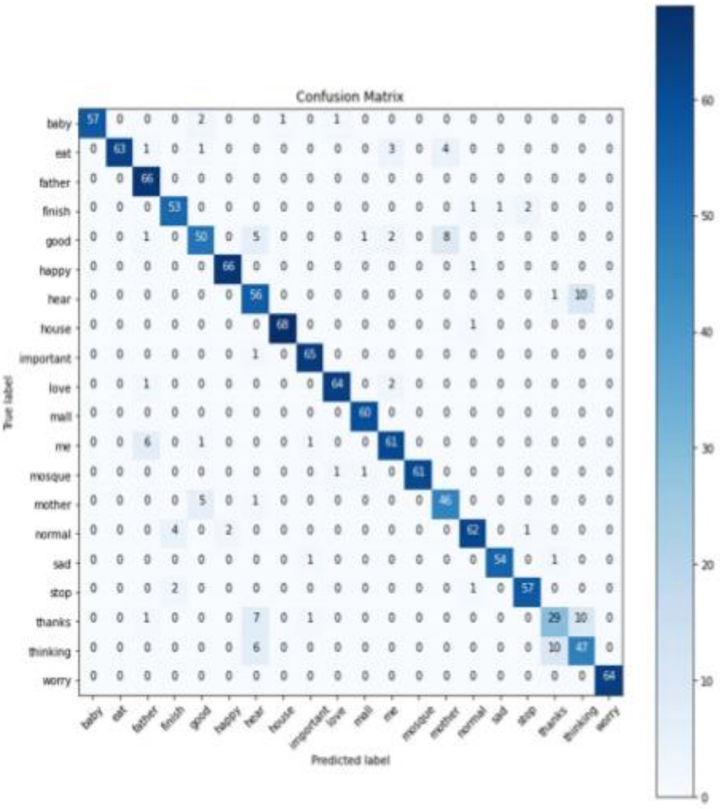


Figure 8, shows changes of accuracy and loss over time

Table 3, shows "F1 score per class" for test and validation sets



Class Name	Validation (%)	Test (%)
Baby	99	97
Eat	97	93
Father	100	93
Finish	97	91
Good	93	79
Happy	97	98
hear	98	78
House	98	99
Important	100	97
Love	97	96
Mall	100	98
Me	95	89
Mosque	97	98
Mother	99	84
Normal	99	92
Sad	100	97
Stop	99	95
Thanks	97	65
Thinking	97	72
Worry	100	100



	baby	eat	father	finish	good	happy	hear	house	important	love	mall	me	mosque	mother	normal	sad	stop	thanks	thinking	worry
baby	57	0	0	0	2	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0
eat	0	63	1	0	1	0	0	0	0	0	0	3	0	4	0	0	0	0	0	0
father	0	0	66	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
finish	0	0	0	53	0	0	0	0	0	0	0	0	0	0	1	1	2	0	0	0
good	0	0	1	0	50	0	5	0	0	0	1	2	0	8	0	0	0	0	0	0
happy	0	0	0	0	0	66	0	0	0	0	0	0	0	0	1	0	0	0	0	0
hear	0	0	0	0	0	0	56	0	0	0	0	0	0	0	0	0	0	1	10	0
house	0	0	0	0	0	0	0	68	0	0	0	0	0	0	1	0	0	0	0	0
important	0	0	0	0	0	0	1	0	65	0	0	0	0	0	0	0	0	0	0	0
love	0	0	1	0	0	0	0	0	0	64	0	2	0	0	0	0	0	0	0	0
mall	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0	0	0	0	0
me	0	0	6	0	1	0	0	0	1	0	0	61	0	0	0	0	0	0	0	0
mosque	0	0	0	0	0	0	0	0	0	1	1	0	61	0	0	0	0	0	0	0
mother	0	0	0	0	5	0	1	0	0	0	0	0	0	46	0	0	0	0	0	0
normal	0	0	0	4	0	2	0	0	0	0	0	0	0	0	62	0	1	0	0	0
sad	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	54	0	1	0	0
stop	0	0	0	2	0	0	0	0	0	0	0	0	0	0	1	0	57	0	0	0
thanks	0	0	1	0	0	0	7	0	1	0	0	0	0	0	0	0	0	29	10	0
thinking	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	10	47	0
worry	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	64

Figure 9,, Confusion Matrix

[4] S. A. Mehdi and Y. N. Khan, "Sign language recognition using sensor gloves," Proceedings of the 9th International Conference on Neural Information Processing, 2002. ICONIP '02., Singapore, 2002, pp. 2204-2206 vol.5, doi: 10.1109/ICONIP.2002.12018

[5] J. E. López-Noriega, M. I. Fernández-Valladares and V. Uc-Cetina, "Glove-based sign language recognition solution to assist communication for deaf users," 2014 11th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE), Campeche, 2014, pp. 1-6, doi: 10.1109/ICEEE.2014.6978268.

[6] Starner, T. and Pentl, A., "Visual Recognition of American Sign Language Using Hidden Markov Models", International Workshop on Automatic Face & Gesture Recognition, 189-194 (1995)

[7] Hienz H., Bauer B., Kraiss K. (1999) HMM-Based Continuous Sign Language Recognition Using Stochastic Grammars. In: Braffort A., Gherbi R., Gibet S., Teil D., Richardson J. (eds) Gesture-Based Communication in Human-Computer Interaction. GW 1999. Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence), vol 1739. Springer, Berlin, Heidelberg

[8] K. Grobel and M. Assan, "Isolated sign language recognition using hidden Markov models," 1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation, Orlando, FL, USA, 1997, pp. 162-167 vol.1, doi: 10.1109/ICSMC.1997.625742.

[9] Parcheta Z., Martínez-Hinarejos CD. (2017) Sign Language Gesture Recognition Using HMM. In: Alexandre L., Salvador Sánchez J., Rodrigues J. (eds) Pattern Recognition and Image Analysis. IbPRIA 2017. Lecture Notes in Computer Science, vol 10255. Springer, Cham

[10] Aliaa A.A Youssif, Amal Elsayed Aboutabl and Heba Hamdy Ali, "Arabic Sign Language (ArSL) Recognition System Using HMM " International Journal of Advanced Computer Science and Applications(IJACSA), 2(11), 2011.

[11] Masood, S., Thuwal, H.C. and Srivastava, A., "American sign language character recognition using convolution neural network", In: Proceedings of Smart Computing and Informatics, pp. 403–412. Springer, Singapore (2018)

[12] Wadhawan, A., Kumar, P. Deep learning-based sign language recognition system for static signs. Neural Comput & Applic (2020).

[13] Bheda, Vivek & Radpour, Dianna. (2017). Using Deep Convolutional Networks for Gesture Recognition in American Sign Language.

[14] Tao, W., Leu, M. C., & Yin, Z. (2018). American Sign Language alphabet recognition using Convolutional Neural Networks with multiview augmentation and inference fusion. Engineering Applications of Artificial Intelligence, 76, 202-213. doi:10.1016/j.engappai.2018.09.006

[15] Yang, S. and Zhu, Q., "Continuous Chinese Sign Language Recognition with CNN-LSTM", Ninth International Conference on Digital Image Processing (ICDIP 2017), doi:10.1117/12.2281671 (2017)

[16] Jie Huang, Wengang Zhou, Houqiang Li and Weiping Li, "Sign Language Recognition using 3D convolutional neural networks," 2015 IEEE International Conference on Multimedia and Expo (ICME), Turin, 2015, pp. 1-6, doi: 10.1109/ICME.2015.7177428.

[17] M. Al-Hammadi, G. Muhammad, W. Abdul, M. Alsulaiman, M. A. Bencherif and M. A. Mekhtiche, "Hand Gesture Recognition for Sign Language Using 3DCNN," in IEEE Access, vol. 8, pp. 79491-79509, 2020, doi: 10.1109/ACCESS.2020.2990434.