Efficient Deep Learning Algorithm for Egyptian Sign Language Recognition

Mostafa A. Abdelrazik
Benha University
Benha, Egypt
mostafa.ahmed15@ beng.bu.edu.eg

Abdelhaliem Zekry
Ain Shams University
Cairo, Egypt
aaazekry@hotmail.com

Wael A. Mohamed
Benha University
Benha, Egypt
wael.ahmed@bhit.bu.edu.eg

Abstract— **Although most people can communicate effectively through speech, some have difficulties doing so due to physical or mental impairments. Communication is a significant obstacle for individuals with these disabilities. Methods of deep learning can aid in the elimination of communication barriers. This article proposes a model based on deep learning for detecting and recognizing words from gestures. Deep learning models such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) are used to recognize signs from Egyptian Sign Language (ESL) video frames. There are many activation functions, and every type has advantages and disadvantages. All these activation functions were applied to our dataset for ESL. To overcome the main disadvantage of the Relu activation function, we proposed a gesture recognition method for ESL using Mediapipe and modified GRU with a new activation function (Talu). The proposed model achieves approximately 94.95% accuracy across ten different signs. This method may assist people unfamiliar with Egyptian sign language in communicating with people with speech or hearing impairments.**

I. Introduction

When speech is hindered, people use tactile-kinesthetic communication. It has been estimated by the World Federation of the Deaf that there are over 70 million deaf people today. The vast majority (over 80%) of them are in third-world countries, including Egypt. They use over 300 different sign languages. Sign language helps people with speech and hearing impairments communicate—visual language. The fundamental aspects of sign language are hand shape, orientation, movement, location, and additional factors such as mouth shape and eyebrow movements. The use of bright gloves can give a voice to sign language movements. Unfortunately, people without knowledge of sign language tend to underestimate or disregard those with disabilities due to communication barriers. To address this issue, the authors propose a system that aims to eliminate communication gaps and provide equal opportunities for everyone. The proposed system involves analyzing a video of a person's hand gestures and utilizing a model to predict words one by one. The system generates a coherent sentence from these words, which can then be translated. The authors utilized Egyptian sign language in their system, which includes ten static signs such as hello, home, man, woman, etc. The system uses natural gesture input to produce sign language, which is then pre-processed and analyzed to determine the exact word associated with the gesture.

This study aimed to develop an offline sign-language recognition system. Vision-based data collection from signers was developed. This study focuses on the system's ability to recognize ESL words (10 words). The ESL dataset has ten words, each with 500 video samples from 5 male and female research participants.

The paper continues Section 2 reviews the literature (sign language recognition). Section 3 describes how to implement sign language detection and recognition. Section 4 discusses ESL and analyzes experiment results. Section 5 concludes and proposes future research.

II. Related work

Multiple methods have been suggested by the literature for recognizing ASL, including motion gloves, the Kinect Sensor, camera-based image processing, and leap motion controllers. To monitor the three-dimensional motion of 50 ASL words, an artificial neural network model was developed [1]. Compared to visual methods, using motion gloves for ASL recognition is more costly, requires specific hand anatomy, and is uncomfortable for users. Wear and tear on the gloves from repeated use also increases the calibration time and introduces the possibility of error ([2] - [4]). ASL signs are still hard to recognize with Kinect sensors alone because the signs are complicated, fingers are always in the way, there are a lot of similarities between classes, and there are also big differences between classes ([5], [6]). In addition, it is crucial that the sensory data be calibrated. Several research efforts have concentrated on angular position measurement for motion gesture prediction [7]. Another method is to recognize the sequence of glosses present in continuous video sequences ([8] – [10]) A user-dependent mode for developing an ASL recognition system was proposed by [11], and a modified kNN approach was proposed by [12]. Extensive research has also been conducted on the wearable application and sensing board for ASL recognition ([13] - [17]).

Image processing is a widely accessible and effective low-cost option for vision-based sign recognition ([18] - [21]), but it takes a long time to calculate the recognition of a hand and fingers, which delays the projection of the recognition result [22]. Factors such as the subject's skin tone and ambient lighting can also have a significant impact on the reliability of data collection [23]. The palm-sized leap motion controller, on the other hand, is a more affordable and convenient alternative to motion gloves or Kinect sensors [24]. In addition to its other benefits, the leap motion controller processes data quickly is highly reliable and requires little memory [25]. The controller,

however, samples data at irregular intervals. To mitigate its impact on real-time recognition systems, post-processing is required [26]. The authors in [28] proposed an algorithm for ESL recognition using Inception-v3 as CNN stage and LSTM as RNN stage and they achieved 90% accuracy for CNN only and 72% for CNN-LSTM method.

A. Mediapipe

MediaPipe is a hybrid open-source framework that generates pipelines for processing visual data such as photos, videos, and audio. It is a comprehensive technique that uses ML for real-time hand detection and gesture identification. It offers additional hand and finger track options by identifying sign motions effectively. To get the landmarks (keypoints) from the face, hands, and body stance, we used a MediaPipe Holistic pipeline.

1) Holistic pose landmarks.

Using its BlazePose detector, the MediaPipe Holistic body pose model infers about 33 3D landmarks of coordinates (x, y, z) on input pictures of the body and extracts the person/position areas of interest (ROI) inside the frame. Pose landmarks and the cropped ROI division masks recognize postures sequentially. It properly localizes more critical places and SLR as shown in Fig.1.

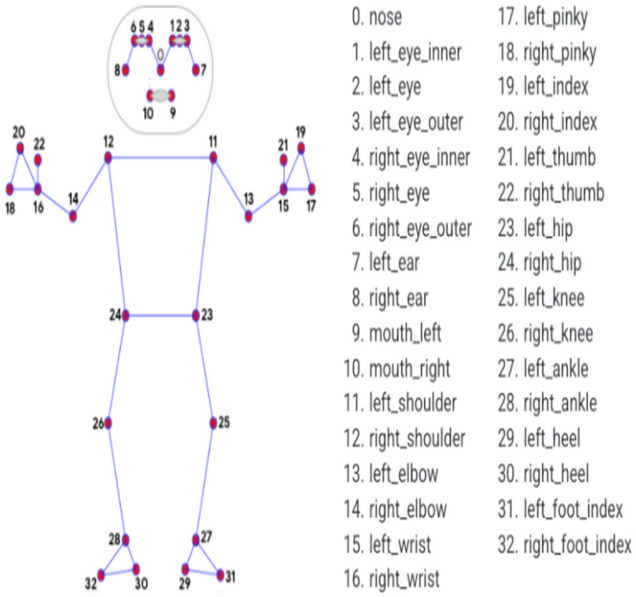

Fig. 1. Pose landmarks [27]

2) Holistic hand landmarks.

In a single frame, MediaPipe Holistic hands integrate two models the palm detection model and the hand key point localization model to infer around 21 hand landmarks consisting of (x, y, z) coordinates and to give the required output. The gadget originally employed a Blaze Palm single-shot detector.

This model highlights stiff areas in the entire picture with a bounding, such as palms and fist, for palm detection. Using palm detection output, the model localizes hand key points as shown in Fig. 2.

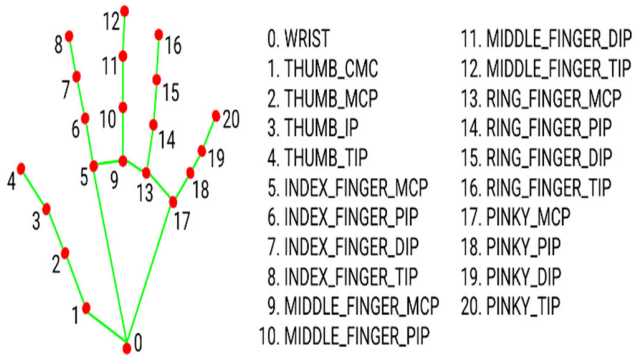

Fig. 2. Hand landmarks [27]

3) Holistic face landmarks.

Using a single camera and no depth sensor, MediaPipe face mesh creates 468 3D facial landmarks. It employs two deep neural network models: a detector for calculating face positions throughout an entire image and a 3D face landmark model. Precision face cropping reduces data augmentation processes and allows the network to concentrate on coordinate prediction accuracy as shown in Fig. 3.

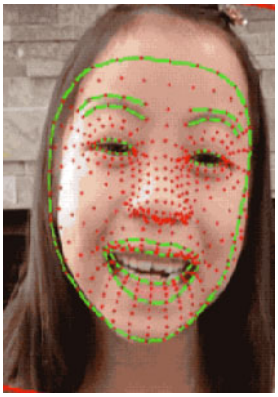

Fig. 3. Face landmarks [27]

B. GRU

Most computer vision issues include dealing with temporal connections between inputs as well as modeling short and long-term sequences. Recurrent neural networks (RNNs) are an effective method for processing sequential input. Unlike traditional neural networks, RNNs focus on modifying state neurons to learn contextual relationships within and between sequential input. However, training RNNs can be challenging due to various constraints, as well as concerns about vanishing and exploding gradients. To address these issues, researchers have developed Gated GRUs, which improve on traditional RNNs by addressing the problem of disappearing and exploding gradients. LSTM networks are the most commonly used type of RNN due to their state-of-the-art performance on numerous machine learning applications. GRUs, which are a variation of LSTMs, operate similarly and provide satisfactory results. They enhance the architecture of LSTM units by combining the three gating units into two: an update gate and a reset gate. As a result, the GRU network model parameters are significantly reduced, preserving information dependence and shortening training time. Fig. 4 depicts the overall anatomy of a typical GRU cell

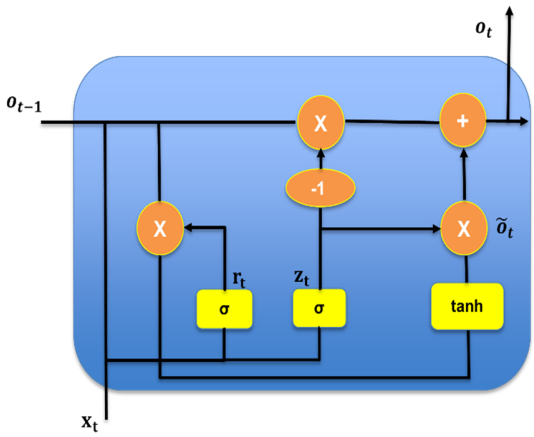

Fig. 4. GRU cell

Following are the equations used in GRU typical cell

$$r_t = \sigma(W_{xr}^T . x_t + W_{or}^T . o_{t-1} + b_r) \quad (1)$$

$$z_t = \sigma(W_{xz}^T . x_t + W_{oz}^T . o_{t-1} + b_z) \quad (2)$$

$$\tilde{o}_t = tanh(W_{x\tilde{o}}^T . x_t + W_{o\tilde{o}}^T . (r_t \otimes o_{t-1}) + b_{\tilde{o}}) \quad (3)$$

$$o_t = z_t \otimes o_{t-1} + (1 - z_t) \otimes \tilde{o}_t \quad (4)$$

$$tanh(t) = \frac{1 - e^{-2t}}{1 + e^{-2t}} \quad (5)$$

$$\sigma = \frac{1}{1 + e^{-t}} \quad (6)$$

Where W_{xr} , W_{xz} , $W_{x\tilde{o}}$ are the weights of the matrices for the corresponding connected input vector, W_{or} , W_{oz} , $W_{o\tilde{o}}$ represent the weight matrices of the previous time step and b_r , b_z , $b_{\tilde{o}}$ are bias

III. Methodology

We divided the proposed algorithm into two main parts, CNN and RNN. The CNN stage is used for features extraction from each frame and then goes to RNN to recognize and predict the word. Before these two main parts, a pre-processing stage is necessary to prepare the data.

A. Data pre-processing and feature extraction

For this stage, we used MediaPipe Holistic as MediaPipe's multistage pipeline, for data preprocessing and feature extraction from the image. Each input frame was processed by the MediaPipe Holistic, which used region-specific image resolution to handle separate models for the hands, face, and pose components. The first stage process is outlined briefly below:

1) The pose detector in BlazePose was used to estimate both the human pose and the resulting landmark model. Following that, the estimated landmarks were used to crop three regions of interest (ROIs) for the face and hands. A recrop was then used for ROI improvement.

2) This was accomplished by cropping the input coordinates for task-specific hand and face models to the ROIs from the original, high-resolution input coordinates.

3) When everything was added up, we had more than 540 landmarks

B. Gesture recognition

In this stage, after extracting the futures, we use a modified GRU as an RNN model to recognize the gesture from the video. The learning rate for the network is debending on which activation function is used, and each one has its usage:

1) Sigmoid function: The sigmoid function is commonly used in GRU models to control the update and reset gates, which are used to regulate the flow of information through the network. The sigmoid function has the range (0,1) and is defined as

$$f(x) = 1 / (1 + exp(-x)) \quad (7)$$

2) Hyperbolic tangent function (tanh): The tanh function is another common activation function used in GRU models. It is similar to the sigmoid function but has a range of (-1,1). The tanh function is often used in the output gate of the GRU to regulate the output values. It is defined as

$$f(x) = (exp(x) - exp(-x)) / (exp(x) + exp(-x)) \quad (8)$$

3) Rectified Linear Unit (ReLU): ReLU is a popular activation function in deep learning models. It has been shown to perform well in many applications, including image recognition and natural language processing. ReLU is defined as

$$f(x) = max(0, x) \quad (9)$$

which means that it returns zero for negative inputs and the input value for positive inputs.

4) Softmax function: The softmax function is commonly used in the output layer of the GRU to produce a probability distribution over the output classes. It maps the input values to a range of (0,1) and normalizes them so that they sum to 1. The softmax function is defined as

$$f(x\_i) = exp(x\_i) / sum\_j(exp(x\_j)) \quad (10)$$

for the i-th element of the input vector.

5) Swish function: The swish function is a recently proposed activation function that has shown to improve the performance of deep neural networks. It is defined as

$$f(x) = x * sigmoid(beta * x) \quad (11)$$

where beta is a hyperparameter that controls the shape of the function. The swish function is similar to the ReLU function, but it has a smooth curve and allows negative input values to produce nonzero outputs.

The primary modification to the standard GRU cell is an enhancement to the update gate, which involves replacing the tanh activation function with the TALU activation function. Our proposed RNN model, called EGRU, improves the update gate by multiplying the original input x_t with r_t This modification leads to better learning efficiency, faster convergence, and reduced computational cost, as well as the ability to remove irrelevant information in a single screening of the complex time series data. The reset gate's output is utilized as feedback to fine-tune the update gate. By limiting the input information x_t using the reset gate, faster convergence and more efficient learning can be achieved, even when the data volume is excessive. The suggested EGRU cell structure is shown in Fig.5.

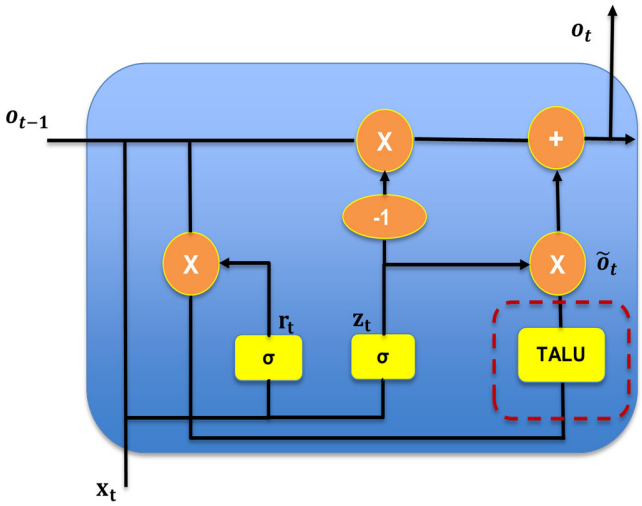

Fig. 5. EGRU cell

We applied a red dashed box which represents the standard tanh activation function that replaced with the TALU activation function. the equations remain the same as in the typical GRU except for (3), and (5). The new equations for the EGRU cell are as follows:

$$\tilde{o}_t = TALU(W_{x\tilde{o}}^T . x_t + W_{o\tilde{o}}^T . (r_t \otimes o_{t-1}) + b_{\tilde{o}}) \quad (12)$$

$$TALU(x) = \begin{cases} x, & x > 0 \\ \frac{1-e^{-2x}}{1+e^{-2x}}, & x \leq 0 \end{cases} \quad (13)$$

Fig 6 shows the network architecture for gesture recognition stage with the input and output for each step ending with the softmax layer in the final classification step with 10 classes.

IV. Experiment Results

A. Experimental Setup

We used our own dataset which consists of ten classes acted by 5 actors each one with 100 clips. These classes represent the words (hello, good morning, good night, home, phone, thanks, bye, friend, man, and woman). We split the data set into 75% for training, and 25% for validation.

The experiments of this paper were carried out with the following setup: 11th Gen Intel core i7 processor @2.30 GHz with 32 GB ram, and Nvidia Geforce RTX 3060 with 6 GB memory. The platform was Jupyter Notebook and python 3.8 on windows 11.

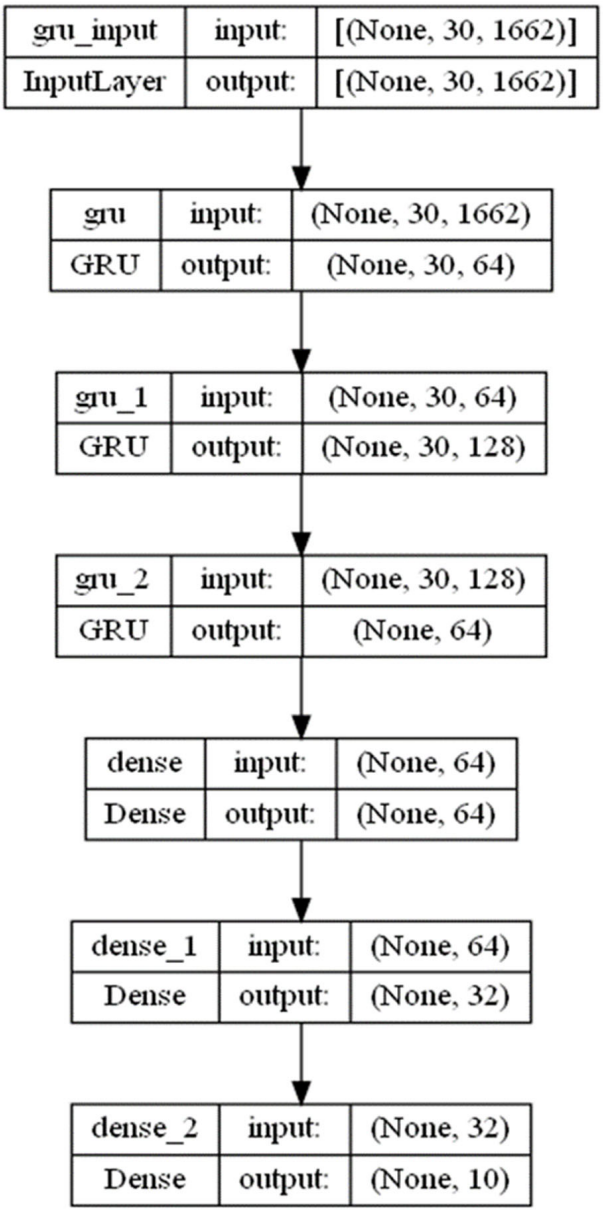

Fig. 6. Architecture of the network

B. Results

The results show that the proposed algorithm achieves better accuracy when compared with the standard GRU, GRU with RELU activation function, standard SLTM, and simple RNN. Table I summarize the accuracy of all networks.

TABLE I. NETWORKS ACCURACIES AND LOSSES

Network	Accuracy	Losses
Standard GRU	91.23	0.32
GRU with RELU	92.51	0.35
Standard LSTM	90.97	0.38
Simple RNN	87.53	0.45
Proposed EGRU	94.95	0.25

Fig. 7 shows samples of the good morning class, marked with MediaPipe holistic for hand, pose, and face

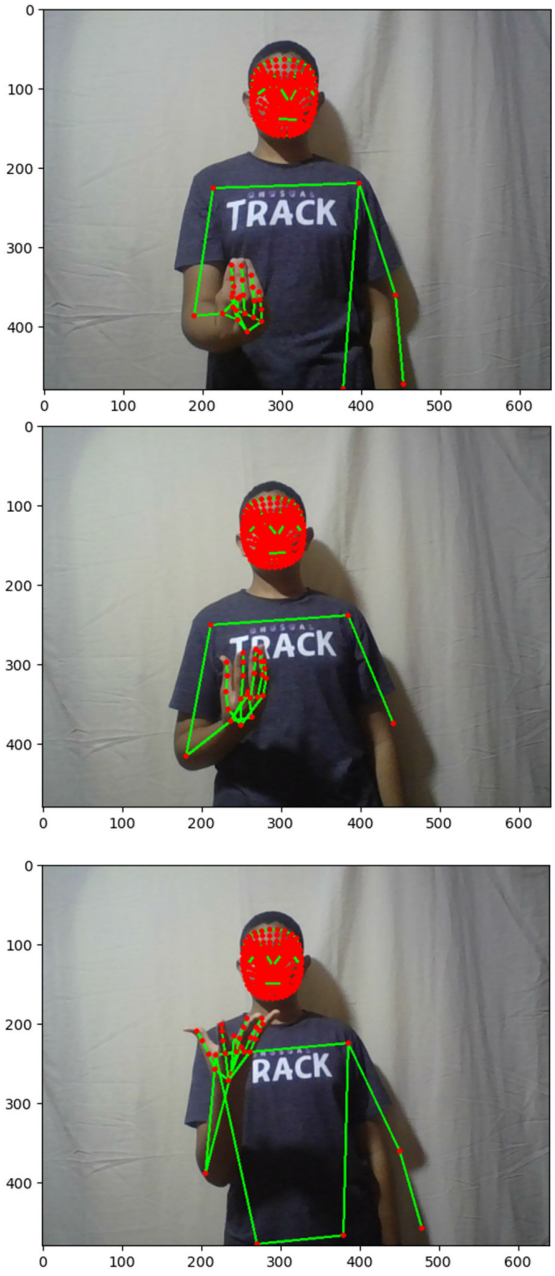

Fig. 7. Good morning class

Fig. 8 and Fig. 9 show the accuracies and losses charts respectively. The accuracy in the first 3 epochs are under expectations but in the next four epochs it gets better and from epoch 8 it starts to slightly increase

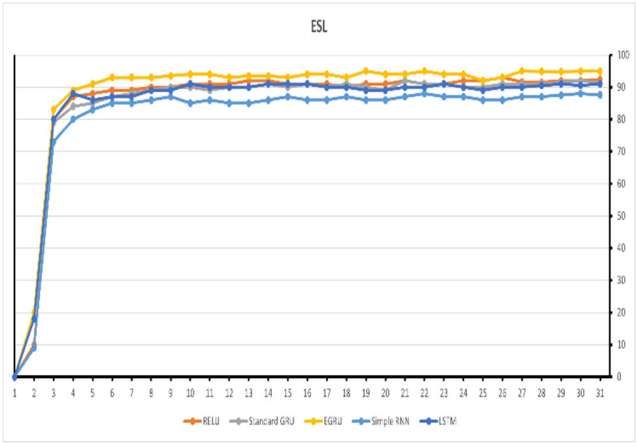

Fig. 8. Networks Accuracies

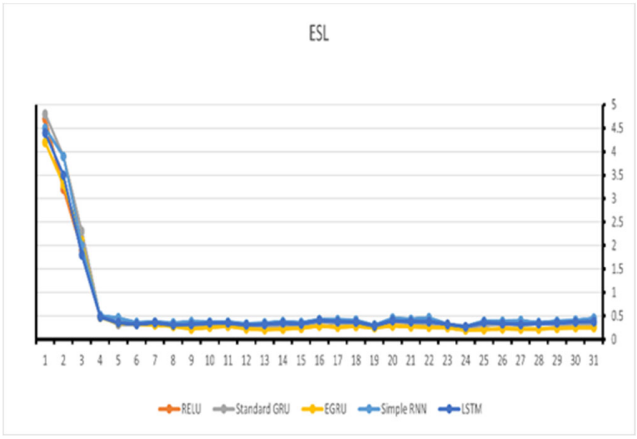

Fig. 9. Networks Losses

Table II shows the precession, recall, and accuracy results for the proposed algorithm

TABLE I I. ACCURACY RESULTS FOR PROPOSED EGRU

Network	Precision	Recall	Accuracy
Proposed EGRU	94.94	95.06	94.95

The confusion matrix is shown in Fig.10 for the proposed network with 10 words (good morning, good night, friend, home, phone, bye, man, woman, hello, and thanks). The network gives an accuracy of 97% for words (woman, man, and good morning), while the word bye has the least accuracy of 83%.

V. CONCLUSION

In our study to simulate the human brain performance in recognition of all actions and objects and to help deaf peoples

communicate with the community, we proposed a modified GRU model with Mediapipe Holistic to recognize Egyptian Sign Language, and the results show that the proposed algorithm gives an efficient model for with an accuracy of 94.95. In future work, we want to apply the algorithm to a complete dataset for Egyption sign Language and develop an android version to operate on mobile phones.

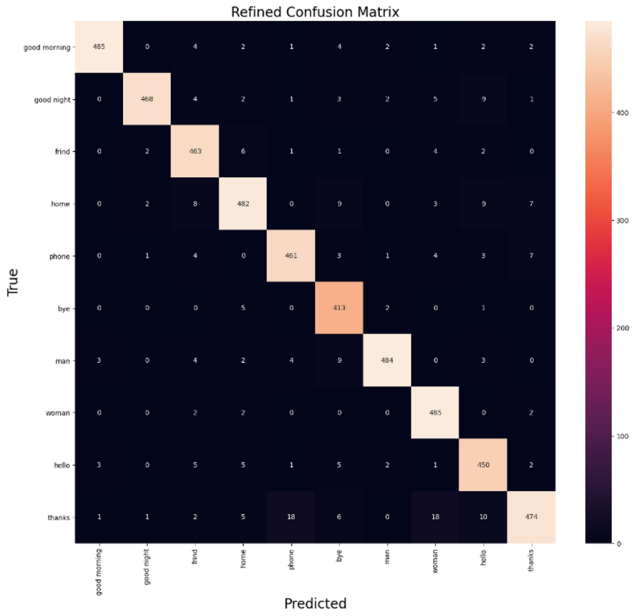

Fig. 10. Proposed network confusion matrix

VI. References

[1] Oz, C., & Leu, M. C. (2011). American Sign Language word recognition with a sensory glove using artificial neural networks. Engineering Applications of Artificial Intelligence, 24(7), 1204–1213.

[2] Huenerfauth, M., & Lu, P. (2010). Accurate and accessible motion-capture glove calibration for sign language data collection. ACM Transactions on Accessible Computing (TACCESS), 3(1), 1–32.. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[3] Luzanin, O., & Plancak, M. (2014). Hand gesture recognition using low-budget data glove and cluster-trained probabilistic neural network. Assembly Automation, 34(1), 94–105.

[4] Oz, C., & Leu, M. C. (2007). Linguistic properties based on American Sign Language isolated word recognition with artificial neural networks using a sensory glove and motion tracker. Neurocomputing, 70(16-18), 2891–2901.

[5] Sun, C., Zhang, T., Bao, B.-K., Xu, C., & Mei, T. (2013). discriminative exemplar coding for sign language recognition with kinect. IEEE Transactions on Cybernetics, 43(5), 1418–1428.

[6] 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.

[7] Fujiwara, E., Ferreira Marques Dos Santos, M., & Suzuki, C. K. (2014). Flexible optical fiber bending transducer for application in glove-based sensors. IEEE Sensors J., 14 (10), 3631–3636.

[8] O. Koller, C. Camgoz, H. Ney, and R. Bowden, “Weakly supervised learning with multi-stream cnn-lstm-hmms to discover sequential parallelism in sign language videos,” IEEE transactions on pattern analysis and machine intelligence, 2019.

[9] R. Cui, H. Liu, and C. Zhang, “A deep neural framework for continuous sign language recognition by iterative training,” IEEE Transactions on Multimedia, 2019.

[10] I. Papastratis, K. Dimitropoulos, D. Konstantinidis, and P. Daras, “Continuous sign language recognition through cross-modal alignment of video and text embeddings in a joint-latent space,” IEEE Access, vol. 8, pp. 91 170–91 180, 2020.

[11] Tubaiz, N., Shanableh, T., & Assaleh, K. (2015). Glove-based continuous arabic sign language recognition in user-dependent mode. IEEE Trans. Human-Mach. Syst., 45(4), 526–533.

[12] Aly, W., Aly, S., & Almotairi, S. (2019). User-independent american sign language alphabet recognition based on depth image and PCANet Features. IEEE Access, 7, 123138–123150.

[13] Lee, B. G., & Lee, S. M. (2018). Smart wearable hand device for sign language interpretation system with sensors fusion. IEEE Sensors J., 18(3), 1224–1232.

[14] Paudyal, P., Lee, J., Banerjee, A., & Gupta, S. K. (2019). A Comparison of Techniques for Sign Language Alphabet Recognition Using Armband Wearables. ACM Transactions on Interactive Intelligent Systems (TiiS), 9, 14.

[15] Wu, J., & Jafari, R. (2017). Wearable Computers for Sign Language Recognition. In S. U. Khan, A. Y. Zomaya, & A. Abbas (Eds.), Handbook of Large-Scale Distributed Computing in Smart Healthcare (pp. 379–401).

[16] Wu, J., Sun, L., & Jafari, R. (2016). A wearable system for recognizing american sign language in real-time using IMU and surface EMG sensors. IEEE Journal of Biomedical and Health Informatics, 20(5), 1281–1290.

[17] Wu, J., Tian, Z., Sun, L., Estevez, L., & Jafari, R. (2015). Real-time American Sign Language Recognition using wrist-worn motion and surface EMG sensors. In 2015 IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks (BSN) (pp. 1–6).

[18] iaramello, F. M., & Hemami, S. S. (2011). A Computational Intelligibility Model for Assessment and Compression of American Sign Language Video. IEEE Transactions on Image Processing, 20, 3014-3027.

[19] Starner, T., Weaver, J., & Pentland, A. (1998). Real-time American sign language recognition using desk and wearable computer-based video. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20, 1371-1375.

[20] D. Li, C. Rodriguez, X. Yu, and H. Li, “Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison,” in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2020, pp. 1459–1469.

[21] Rastgoo, R., Kiani, K., & Escalera, S. (2020). Hand sign language recognition using multi-view hand skeleton. Expert Systems with Applications, 150, 113336.

[22] Khelil, B., Amiri, H., Chen, T., Kammüller, F., Nemli, I., & Probst, C. (2016). Hand gesture recognition using leap motion controller for recognition of arabic sign language. Sci: Lect Notes Comput.

[23] Bheda, V., & Radpour, D. (2017). Using deep convolutional networks for gesture recognition in American sign language. arXiv preprint arXiv:1710.06836.

[24] Chuan, C.-H., Regina, E., & Guardino, C. (2014). In American sign language recognition using leap motion sensor (pp. 541–544). IEEE.

[25] Naglot, D., & Kulkarni, M. (2016). Real time sign language recognition using the leap motion controller (Vol. 3, 1–5.

[26] Guna, J., Jakus, G., Poga ˇcnik, M., Toma ˇ zi ˇ c, S., & Sodnik, J. (2014). An analysis of the precision and reliability of the leap motion sensor and its suitability for static and dynamic tracking. Sensors, 14, 3702-3720

[27] https://google.github.io/mediapipe/solutions/solutions.html

[28] Ahmed Elgohary, Rawan Galal Elrayes (2021) “Egyption Sign Language Recognition Using CNN and LSTM” Computer Vision and Pattern Recognition (cs.CV). https//doi.org/10.48550/arXiv.2017.1364