Utalk: Sri Lankan Sign Language Converter Mobile App using Image Processing and Machine Learning

I.S.M Dissanayake[1], P.J Wickramanayake[2], M.A.S Mudunkotuwa[3] and P.W.N Fernando[4]
Sri Lanka Institute of Information Technology
Malabe 10115, Sri Lanka
Email: isurusmd@gmail.com[1], pasindu11.wickramanayake@gmail.com[2],
amila.mudunkotuwa@gmail.com[3], pwnirmalfernando@gmail.com[4]

2020 2nd International Conference on Advancements in Computing (ICAC) | 978-1-7281-8412-8/20/$31.00 ©2020 IEEE | DOI: 10.1109/ICAC51239.2020.9357300

***Abstract*—Deaf and mute people face various difficulties in daily activities due to the communication barrier caused by the lack of Sign Language knowledge in the society. Many researches have attempted to mitigate this barrier using Computer Vision based techniques to interpret signs and express them in natural language, empowering deaf and mute people to communicate with hearing people easily. However, most of such researches focus only on interpreting static signs and understanding dynamic signs is not well explored. Understanding dynamic visual content (videos) and translating them into natural language is a challenging problem. Further, because of the differences in sign languages, a system developed for one sign language cannot be directly used to understand another sign language, e.g., a system developed for American Sign Language cannot be used to interpret Sri Lankan Sign Language. In this study, we develop a system called Utalk to interpret static as well as dynamic signs expressed in Sri Lankan Sign Language. The proposed system utilizes Computer Vision and Machine Learning techniques to interpret sings performed by deaf and mute people. Utalk is a mobile application, hence it is non-intrusive and cost-effective. We demonstrate the effectiveness of the our system using a newly collected dataset.**

***Keywords*—Sinhala Sign Language, Computer Vision, Machine Learning**

I. Introduction

Hearing is one of the most important human senses which helps individuals to connect with the outside world in their everyday lives. Unfortunately, not every person is gifted with the hearing ability. According to the researchers, over 360 million people in the world have been affected by hearing impairment [1]. The main obstacle for people with hearing impairments is the communication with ordinary people. Generally, a sign language is used by deaf and mute people for their communication. However, most of the hearing people neither can understand the sign language nor can use it. Because of this communication barrier deaf-mute people sometimes fail to express their feelings and views as well as fail to understand the feelings and views of others. Therefore, both deaf-mute as well as hearing people face many difficulties in carrying out their essential day to day activities with each other. Ultimately, it will cause such disabled persons to be isolated from society.

There are many highly talented people suffering from hearing and speech impairments. It would be unfortunate if having such impairments becomes an obstacle to achieve their goals. Moreover, adding them to the workforce will help to improve the socio-economic development of the country. Therefore, it is imperative to assist them in making their lives more successful and providing a way to join the country's primary workforce. Many researches have proposed to utilize Computer Vision techniques to interpret signs performed by deaf and mute people and express them in natural language, so that hearing people can understand. However, most of such researches focus on interpreting static signs. Understanding dynamic signs is not well explored. Understanding dynamic visual content (videos) and translating them into natural language is a challenging problem. Further, there is no universal sign language for deaf-mute people, and different countries have their own sign language systems [2]. Hence, a system developed for one sign language cannot be directly used to understand another sign language, e.g., a system developed to interpret American Sign Language cannot be used to interpret Sri Lankan Sign Language. Furthermore, existing advanced systems to convert signs to text are not readily adaptable or affordable. For example, some of them are needed to have specific external devices such as data gloves [3]. Addressing these issues, in this work, we introduce Utalk: a Sri Lankan Sign Language Converter.

Utalk is a sign language converter specially trained for the Sinhala language that converts videos into text. In contrast to existing solutions, Utalk comes as a mobile application providing users a more cost-effective and easy to use system. Another unique advantage of the proposed system is that it can interpret both static and dynamic signs. The system takes a video of the user while performing sign language as the input, extract frame segments, and then remove the background of those frames using image processing techniques. Those pre-processed frames/images are classified as static or dynamic sign frames and then fed into two separate machine learning models named *static sign classifier* and *dynamic sign classifier*. The output of these two models is going through a language model. Finally, the mobile app outputs the converted text. To understand the problem domain well, we get the assistance from Rathmanalana Deaf and Blind School.

Due to the unavailability of datasets for Sinhala Sign Language, we collected a new dataset. We evaluate the effectiveness of the proposed system using this new dataset and the results indicate that Utalk can correctly identify static as

well as dynamic Sinhala Language sigs.

The structure of the paper is organized as follows. Section 2 describes the related work in the literature. Section 3 describes the process of generating Sinhala text by taking video as the input. Section 4 illustrates the experiments carried out in this study and the results generated from the experiments, and finally, section 6 concludes the discussion with our future plans and works.

II. Related Work

There are two kinds of researches related to sign language translations. One kind translates sign language into written or spoken language [3], e.g. using a specific device to capture sign languages gestures. The other kind translates written or spoken language into sign language, e.g. using a 3d avatar [4]. Sing to natural language translations can be broadly categorized as image-based approaches and sensor-based approaches [5]. Examples for sensor-based approaches are using Kinect sensor [6] or using Leap Motion controller [7]. Sensor-based approaches require the user to wear separate devices. Hence, they can be intrusive and/or expensive. In contrast, the method proposed in this paper uses an image-based approach which non-intrusive and cheap.

Image-based approaches considers several features such as rotation, shape, angle, hand movements, and pixels. Several feature extraction methods have been used to find features and Artificial intelligence methods are used to classify those features. The most highlighted area of this paper is to review the key finding of the comparison of feature extraction methods of other existing systems based on the classification accuracy.

Prasad et al. [8] propose an approach for recognizing Indian sign language using fusion-based edge operators. There are several stages in their approach such as the pre-processing, segmentation, feature extraction and pattern recognition. In the pre-processing and segmentation stages they have used dilatation and erosion techniques to isolate hand and head portions from the image. However, unlike our study, they only used a simple background video as the input, which has made it easier to clearly identify hand and head movements.

Tolentino et al. [9] proposed static sign language recognition using deep learning. They developed that system to assist as a learning tool for starters in sign language that involves hand detection. That system also used skin-color based modeling technique and images fed into the model using Convolution Neural Network (CNN) for classification. This system gained average testing classification result of 93.67% which alphabet recognition, number recognition and static word recognition [9].

Regarding dynamic sign identification, [10] use Leap Motion techniques to classify dynamic signs in Arabic Sign Language. It consists of two main methods. First they implemented a model using K-Nearest Neighbour [11], Artificial Neural Network [12] and Support Vector Machine [13] algorithms. Then gives the single majority output using classification of each and every frame. Dynamic Time Wrapping [14] is used in the second method. This method measures and identifies the optimal alignment of two given sequences. DTW have the ability to identify the most similar classes in testing set with compared to training set.

Another research area which recognize action using dynamic image networks [15], which contains CNNs [16] to classify dynamic images. As they used single dynamic image to represent each video using CNN platform. It splits a video multiple sub-sequences and encode each and every one as a dynamic image which achieves better classification accuracy than a single dynamic image. Long Short-Term Memory [17] and Recurrent Neural Network [18] applied as well in capturing general changes within a short period of time. RNNs function is to analyze video segments and encode the frame segmentation information to their memory cells.

We understand that sign language identification techniques are updated continuously by analyzing these related works. Also, we find out that the use of a mobile device for sign language identification is infrequent. Therefore by considering all of those previous work and taking these work as our basis, we agreed to develop a system to identify Sinhala sign language using a mobile phone. Because Sinhala sign language converters are very rare, and it will be a great help for the Sri Lankan deaf people to communicate with ordinary people.

III. Methodology

Our proposed model, Utalk consists of several sub modules. Figure 1 illustrates the overview of the proposed model. Given the video feed from the camera, Utalk first extracts frames from the video. As a pre-processing step, we remove the background from each frame. Next, we classify frame sequence as static or dynamic for further processing. The *Static Classifier* identify static signs whereas the *Dynamic Classifier* identifies dynamic signs. Finally, results from sign classifiers are fed into the *Language model* to generate text based on input video.

Following sections provide more details about each sub module.

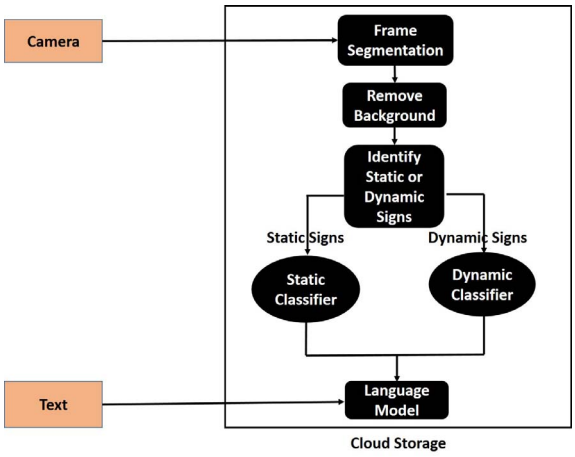

Fig. 1. Overview of the Proposed Model

Fig. 2. An example for before and after background removal

A. Extracting Frames and Removing Background

In this section, we describe the process of extracting frame segments from the input video and removing background of those frames using image processing techniques. First, Utalk reads video and then extract frames. For this, we use the OpenCv *VideoCapture* object. A video frame is nothing but an image. Each extract frame/image is processed separately in following sub modules.

Frames/images extracted as described above may not only contain the person of interest but also contains many other background objects. Using such images directly for our further processing will cause to reduce accuracy as the system might fail to identify hand gestures precisely. Therefore, as a pre-processing step to enhance system performance, we remove the background from the extract images/frames. Figure 2 shows a background removed image.

As the first step of the background removal, extracted images are converted into a grayscale images. Then we used thresh-holding method to perform edge detection correctly, such that, if pixel value changes from a larger value, considered it as an edge. Dilation and erosion operations are applied to make the detected edges more sharp [8] allowing us to identify edges of a person entity in an image accurately. Next, we extract the largest contour and consider it as the person and the rest of the image as the background. Finally, the identified background of the generated frame is filled with a plain color mask. Now the image only consists of a person entity and plain color background. Therefore, the background will not be focused and the accuracy of the system is increased.

B. Detecting Features for Static and Dynamic Sign Identification

The main task of Detecting Features for Static and Dynamic Sign Identification is to divide background removed frames to labeled static and dynamic frames. When background removed frames inserted into the Utalk system, we need to take the pixel difference from two subsequent frames by using a Python OpenCV. Then we save the images and take them into an image histogram. An image histogram is a graphical representation of the pixel distribution of the image [19]. In image histogram, we plot the number of pixels of the image against its tonal values. By this, we can further detect and identify edges correctly, and also this is essential for the system's accuracy. After taking histogram values from each frame, we need to calculate the sum of the histograms as a float value, which is the entropy of the frames. An image's entropy

Fig. 3. Frame Labeled as Static

Fig. 4. Frame Labeled as Dynamic

means corresponding states of intensity level that individual pixels can adopt [20]. By measuring entropy, we can decide which pixels are continuously spreading through each frame and which areas are highlighted due to the frame's appearance. Taking entropy value provides better compression between two frames/images [21]. Using this technique, we can identify that these frames are similar or not. Based on these identified similar frames, we go for the next step.

Subtle body movements that cannot easily be detected by the human eye are called micro-movements. When a person performs a sign gesture, there can micro-movements that the person does, but it is not a part of that particular movement. Therefore, we have to ignore those and take the movements with a more extended time duration in our required frames. These longer time duration frames can be static signs or dynamic signs. If it is a static sign, the movement will stop with fewer frames, and movement will be in a transition period until it goes for another movement. Then we can assume it as a transition from one static frame to another static frame. If the frames' movement is continuously changing, that means the person is still performing some gesture. These frames we can take as dynamic frames. According to that after getting the image entropy, add a loop to compare each frame and set a default entropy value as the place to separate the static and dynamic segments in the frame.In case, added 10 as the default entropy value. If the entropy value is less than 10, have considered it a static frame, and if the entropy value is greater than 10, have considered it a dynamic sign. However, within these dynamic frames, there can be transition frames. By removing these transition frames, we can separate the dynamic frames. Then these static and dynamic frames are labeled and sent to the static and dynamic classifiers in the Utalk system. Figure 4 shows the final output for the Static and Dynamic sign identification.

C. Static Sign Classification

We had identified, there are two types of signs have in the Sinhala Sign Language. The first one is the static sign which is not any movement and the second one is dynamic signs

which having movements. So these signs should be identified when the deaf man shows to the app.

Static sign classification [22] is one of an essential component of the Utalk system, which mainly affects the output. Because every meaningful sentence has at least one or more static signs to support the idea like numbers, letters, and symbols. In some cases, one sign is sufficient to give the whole sentence meaning in SLSL. So this module will develop the classifier to classify static sings by using visual features generated by the previous member.

Data set collection for static SLSL was done by taking photos with the help of a camera. After that images were automatically cropped and converted to 70 * 70 pixels grayscale samples using python. cv2.imread [23] function was used to convert images to grayscale and cv2.resize function used for resizing the images. Each class contained between 400 and 600 images.

A convolution neural network [24] is used for developing the static sign classifier. CNN networks basically consist of three main parts; Conv, Pooling, and Dense layers. Conv layers consist of filters and feature maps. The Pooling layer reduces the feature obtained in the previous Conv layer. The Dense layer is the normal feed-forward network layer. The problem with the first model built using these key layers is low validation accuracy rates. Because of over-fitting (excessive adaptation) and high variance. In that case, we used max pooling, Relu, Sigmoid, SoftMax layers with basic layers for that model building process. Moreover, we were rich enough to check the accuracy rate by switching layers and using more than one layer. We ended that model with a Softmax layer. Because it gives the predicted probabilities for each class label.as well as Each class was trained individually over the network.

The data set was divided into two segments as training and testing using the algorithm to see the performance of the model. It is 80% for training and 20% for testing of the total data set. The Network was implemented and trained via Keras and TensorFlow using a Graphics Processing Unit. We had used 100 epochs with a 200 batch size for train the model and images ware resized to 70*70, 1. Finally, we had used the object detector to predict the static sign and it also used a pre-processed model like mentioned above.

D. Dynamic Sign Classification

Dynamic sign classification's primary function is to identify the differences and changes between every frame segment. The dynamic sign classification model is implemented using convolutional neural network (CNN) classification algorithm which classifies dynamic signs using input and output layers as well as multiple hidden layers. Dynamic sign classification is done through a particular process. The process as follows,

First we have to read and store frames of videos in train data set. There are set of videos included into one folder which stand for one recognized dynamic sign. Like wise ,there are sets of video folders included for our selected dynamic signs. We call those video folders as classes. We have to frame the videos,when creating the data set. We add all these frames into one single csv file, which separate file names with labeled classes. After that load the image and keep target size as (224,224,3). Then we should convert the data set into an array. In order to do that we should normalize the pixel values and append the images into train image list.

Next main objective is convert the train data set into a numpy array. The function of a numpy array is to convert the 3 Dimentional array into a 1 Dimentional array. As we are using CNN, that creating a numpy array is a must. It generates a numeric array.

Then we have to split the data set into trained and test data set. We allocate 80% for training and 20% for testing. We give high percentage for training to increase the accuracy of the model.

Next creating dummies of target variables. As this model is a video classification one, we have to initialize the training and testing data sets using dummy values.

Then we create a base model. Here we used VGG 16 base model which is used for video classification for some more. It has pre-trained using a generic data set. It increases the accuracy of the model.

Then reshaping the training and testing data set. Reshape it into a single dimention, converts the array into single dimention which has video one side and file name other.

After that normalize the pixel values. It means load all pixel values and take the maximum and minimum of pixel values. We give this range to the model,then model doesn't considerate about other ranges(outliers). It helps to increase the model accuracy.

Then define the model architecture. It is done throug adding dense layers to the model. Model consist of 4 hidden layers. Input shape is 61440 array and output shape is 8 because, here we consider only 8 dynamic signs with 8 video classes.

Next step is to define a function to save the weights of the model using Keras. Then compile the model which we have initially defined. Next we train the model.

Finally, save the model into a h5 file. When we make predictions, we take the trained h5 file. We train the model only once.

Sign conversion into text happens in real-time. The user capture video with the mobile app, and the video will be uploaded to the back-end application on server. After that, the video clip will be split into frames, and each frame's background is removed. These frames will separate into static and dynamic frames with the numbered sequence they were in the original video. Then static frames are put through the image classifier, and the output is recorded with the frame number. Dynamic frames are put through the dynamic classifier, and the output is recorded with the frame number. The outputs recorded with the frame number arranged in order of the original video frame numbers and redundancies are removed. Then remaining words that are identified from the classifiers are sent back to the mobile application as the request response. In the mobile application, once the response arrives

TABLE I
CLASSIFICATION ACCURACY FOR STATIC AND DYNAMIC SIGN CLASSIFICATION

Static sign	Dynamic signs
0.97	0.95

TABLE II
ACCURACY COMPARISON WITH AND WITHOUT BACKGROUND

	with background	without background
Static	0.67	0.97
Dynamic	0.80	0.95

from the back-end with the translation, it is displayed as a text.

IV. EXPERIMENTS AND RESULTS

Due to unavailability of datasets for Sinhala Sign Language, we collected a dataset with the participation of nine volunteers to evaluate the proposed Utalk system. Nine volunteers consisted of two females and seven males. Further, volunteers were selected such that three age groups (kid, young and middle-aged) are covered. The participants were asked to perform four static sings and eight dynamic signs. The four static signs are Ayubowan (A Sri Lankan greeting), House, Love and School. The eight dynamic signs are He or She, Hello, Here, Me, Name, Teacher and You. Each user were asked to perform each sign for three times, different background locations and different lightning conditions. Hence, each sign have 27 images and video segments. Sample images from the collected dataset are shown in 5.

Fig. 5. Sample images from the collected dataset

We evaluate Utalk systems using 4 metrics, classification accuracy, precision, recall and F1 score. Table I shows classification accuracy values for static and dynamic sign classification. We can see that Utalk can perform well in both static and dynamic sign classification.

We remove background of the images as a pre-processing step. Table II shows the impact of background removal. Results indicates that background removal and separation of static and dynamic signs leads to better classification accuracy in both static and dynamic sign classification.

Further, collection of large datasets for sign language is very expensive. However, to successfully train a deep learning model we need a large dataset. Hence, in this work we perform data augmentation to increase the number of images in our dataset. We use flipping, rotation, translation and zooming as data augmentation methods. In Table III we compare impact of data augmentation on classification accuracy. Results demonstrate that data augmentation greatly helps to improve classification accuracy.

TABLE III
COMPARISON OF CLASSIFICATION ACCURACY WITH AND WITHOUT DATA AUGMENTATION

	Without data augmentation	With data augmentation
Static	0.27	0.97
Dynamic	0.60	0.95

TABLE IV
PRECISION, RECALL AND F1 SCORE VALUES FOR STATIC SIGN CLASSIFICATIONS

	Precision	Recall	F1 Score
Ayubowan	0.95	1.00	0.97
Love	1.00	0.91	0.95
House	1.00	1.00	1.00
School	0.98	1.00	0.99

TABLE V
PRECISION, RECALL AND F1 SCORE VALUES FOR DYNAMIC SIGN CLASSIFICATIONS USING CNN

	Precision	Recall	F1 Score
He or She	1.00	0.95	0.98
Hello	0.94	0.92	0.93
Here	0.89	1.00	0.94
Me	1.00	1.00	0.95
Name	1.00	0.74	0.83
Teacher	0.92	1.00	0.96
You	0.77	0.74	0.75

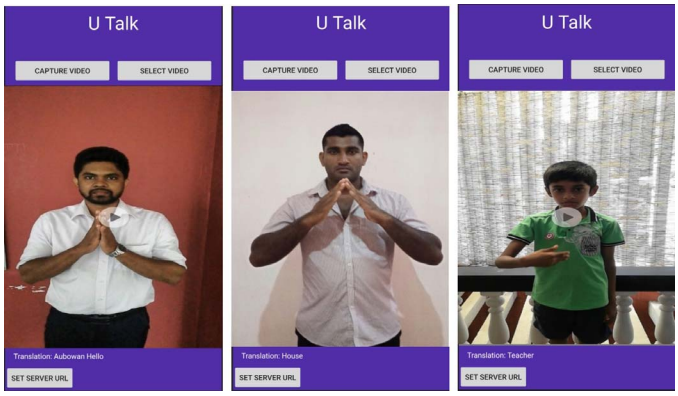

Fig. 6. Mobile app outputs

In Table IV and V we show precision and recall and F1 score each static and dynamic sign respectively. As we can see in In Table IV, Utalk achieves high precision and recall values (over 0.90) for all the static signs that it was evaluated for. Further, Table V shows that Utalk can also achieve higher precision and recall values for dynamic signs. These results

indicate that the proposed Utalk system can be successfully used for both static and dynamic sign identification.

Final output from the Utalk mobile app for static "Ayubowan" sign and dynamic "Hello" sign, static "House" and dynamic "Teacher" signs are shown in the Figure 6.

V. Conclusion

In this paper, we introduced a system called UTalk which can interpret static as well as dynamic signs expressed in Sri Lankan Sign Language. The proposed system accepts a video feed as the input, uses computer vision and machine learning techniques to interpret the signs observed in the video and finally translates interpreted signs into text. Due to lack of datasets of Sinhala Sign language, we collected a new dataset in this work. Experimental results on the collected dataset indicate that Utalk can correctly identify static as well as dynamic Sinhala Signs. Further, Utalk is implemented as a mobile application. Hence, Utalk is non-intrusive and cheaper.In future work, we will translate Sinhala Sign Language gestures into voice in addition to text making Utalk more user-friendly.

References

[1] F. R. Khan, H. F. Ong, and N. Bahar, "A sign language to text converter using leap motion," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 6, no. 6, pp. 1089–1095, 2016.

[2] O. Vedak, P. Zavre, A. Todkar, and M. Patil, "Sign language interpreter using image processing and machine learning," 2019.

[3] M. U. Kakde, M. G. Nakrani, and A. M. Rawate, "A review paper on sign language recognition system for deaf and dumb people using image processing," *International Journal of Engineering Research & Technology (IJERT)*, vol. 5, no. 03, 2016.

[4] Y. Bouzid and M. Jemni, "An avatar based approach for automatically interpreting a sign language notation," in *2013 IEEE 13th International Conference on Advanced Learning Technologies*. IEEE, 2013, pp. 92–94.

[5] H. Bhavsar and J. Trivedi, "Review on feature extraction methods of image based sign language recognition system," *Indian Journal of Computer Science and Engineering*, vol. 8, no. 3, pp. 249–259, 2017.

[6] C. Dong, M. C. Leu, and Z. Yin, "American sign language alphabet recognition using microsoft kinect," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2015, pp. 44–52.

[7] H. Li, L. Wu, H. Wang, C. Han, W. Quan, and J. Zhao, "Hand gesture recognition enhancement based on spatial fuzzy matching in leap motion," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 1885–1894, 2019.

[8] M. Prasad, P. Kishore, E. K. Kumar, and D. A. Kumar, "Indian sign language recognition system using new fusion based edge operator." *Journal of Theoretical & Applied Information Technology*, vol. 88, no. 3, 2016.

[9] L. K. S. Tolentino, R. O. S. Juan, A. C. Thio-ac, M. A. B. Pamahoy, J. R. R. Forteza, and X. J. O. Garcia, "Static sign language recognition using deep learning," *International Journal of Machine Learning and Computing*, vol. 9, no. 6, 2019.

[10] H. Luqman, S. A. Mahmoud *et al.*, "Transform-based arabic sign language recognition," *Procedia Computer Science*, vol. 117, pp. 2–9, 2017.

[11] K. M. Leung, "k-nearest neighbor algorithm for classification," *Polytechnic University Department of Computer Science/Finance and Risk Engineering*, 2007.

[12] A. Dey, G. Miyani, and A. Sil, "Application of artificial neural network (ann) for estimating reliable service life of reinforced concrete (rc) structure bookkeeping factors responsible for deterioration mechanism," *Soft Computing*, vol. 24, no. 3, pp. 2109–2123, 2020.

[13] Y.-D. Cai, P.-W. Ricardo, C.-H. Jen, and K.-C. Chou, "Application of svm to predict membrane protein types," *Journal of theoretical biology*, vol. 226, no. 4, pp. 373–376, 2004.

[14] H.-S. Lee, "Application of dynamic time warping algorithm for pattern similarity of gait," *Journal of exercise rehabilitation*, vol. 15, no. 4, p. 526, 2019.

[15] H. Bilen, B. Fernando, E. Gavves, A. Vedaldi, and S. Gould, "Dynamic image networks for action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3034–3042.

[16] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," *Insights into imaging*, vol. 9, no. 4, pp. 611–629, 2018.

[17] D. Soutner and L. Müller, "Application of lstm neural networks in language modelling," in *International Conference on Text, Speech and Dialogue*. Springer, 2013, pp. 105–112.

[18] Y. Gal and Z. Ghahramani, "A theoretically grounded application of dropout in recurrent neural networks," in *Advances in neural information processing systems*, 2016, pp. 1019–1027.

[19] Y. Wang, Q. Chen, and B. Zhang, "Image enhancement based on equal area dualistic sub-image histogram equalization method," *IEEE Transactions on Consumer Electronics*, vol. 45, no. 1, pp. 68–75, 1999.

[20] J. Lee, S. Cho, and S.-K. Beack, "Context-adaptive entropy model for end-to-end optimized image compression," *arXiv preprint arXiv:1809.10452*, 2018.

[21] M. A. Aljanabi, Z. M. Hussain, and S. F. Lu, "An entropy-histogram approach for image similarity and face recognition," *Mathematical Problems in Engineering*, vol. 2018, 2018.

[22] A. Wadhawan and P. Kumar, "Deep learning-based sign language recognition system for static signs," *Neural Computing and Applications*, pp. 1–12, 2020.

[23] N. Obasi, A. Egbuonu, P. Ukoha, and P. Ejikeme, "Comparative phytochemical and antimicrobial screening of some solvent extracts of samanea saman pods," *African journal of pure and applied chemistry*, vol. 4, no. 9, pp. 206–212, 2010.

[24] H.-C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers, "Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning," *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1285–1298, 2016.