ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24MAY1891

Convolutional Neural Networks for Indian Sign Language Recognition

Manpreet Kaur Sidhu[1]; Snehal Hon[2]; Sandesh Marathe[3]; Tushar A. Rane[4]
Department of Information Technology
SCTR's Pune Institute of Computer Technology Pune, India

Abstract:- Sign Language has been a crucial means of com- munication for the deaf and mute communities worldwide since ages. In India alone, 1 percent of the population consists of hard of hearing and mute individuals. Hence, to help support these marginalized communities, it is important to make use of techno-logical advancements such as deep learning, computer vision and neural network technologies to create systems and applications that can not only help create sign language recognition software for the deaf community, but also provide means to educate others about sign languages around the world. In this paper, we present a system that utilizes Convolutional Neural Networks to recognize the alphabets A-Z of the Indian Sign Language(ISL) by accepting the real time hand signs performed by the user as input from the users' camera feed and then displays the recognized alphabet label as output in the form of text and speech. We created a custom Indian sign language dataset for all 26 alphabets for this experimentation. The extraction of key features was performed using CNN, background removal, hand segmentation and thresholding.

Keywords:- *Convolutional Neural Network(CNN), Indian Sign Language(ISL), Deep Learning, Sign Language Recogni-tion(SLR).*

I. INTRODUCTION

The increase in hearing impairments in densely populated countries such as India emphasizes the need for better communication options. Advances in speech-to-text and sign language recognition are critical for enhancing accessibility. This study investigates these technologies, focusing on the need for inclusive solutions for Indian Sign Language (ISL) users.

Deaf and hard of hearing people experience difficulties due to the restricted use of ISL. Traditional interpretation methods are expensive and inconvenient. We present a camera-based system for ISL that uses Convolutional Neural Networks (CNN) to recognize real-time ISL motions. This strategy improves affordability and accessibility.

Our dependable sign language recognition system use CNNto identify ISL motions in real time, boosting mobility by depending just on a camera-based setup that eliminates the need for extra electronics. This improves communication and social integration for ISL users.

II. LITERATURE SURVEY

There have been various methods used to recognize sign language and a significant amount of methods used CNN's for their recognition.

A proposed three-layer CNN achieves the highest recognition accuracy in [1] for numerals and alphabets and among pre-trained models, ResNet152V2 performs the best. The fine-tuning of pre-trained models and impact of hyperparameters like learning rate, batch size, and momentum on performance is studied. Human action recognition is explored in [2] using convolutional neural networks (CNNs) based on skeleton heatmaps extracted from a two-stage pose estimation model. This method combines an improved single shot detector (SSD) with convolutional pose machines (CPM) to accurately capture human skeleton data. It usesResNet as the backbone of SSD and incorporates multiscale transformations for better skeleton key point detection. A novel CNN model for Sign Language Recognition (SLR) is proposed in [3], which automatically extracts spatial-temporal features from raw video streams, utilizing multiple input channels including color, depth, and body joint positions from Microsoft Kinect. This approach eliminates the needfor hand-crafted features. CNNs and Microsoft Kinect were used in [4] and their model automates feature extraction from video data and achieves high accuracy in recognizing 20 Italian gestures by using techniques like data augmentationand dropout.

A combination of CNN and Long Short-Term Memory (LSTM) networks are utilized in [5], and the system achieved high accuracies for recognizing nine common Egyptian Sign Language(ESL) words. Using the LSA16 and RWTH-PHOENIX-Weather datasets, the authors in [6] evaluated models including LeNet, VGG16, ResNet-34, All Convolutional, and Inception, along with transfer learning techniques. The results showed that VGG16 performed the best, closely followed by LeNet, and accuracy improved significantly when hands were pre-segmented from the background. A framework for Arabic Sign Language (ArSL) recognition was introduced in [7], employing transfer learning with various deep learning models and vision transformers. The methodology encompasses pretrained models like VGG, ResNet, MobileNet, Inception, DenseNet, Big Transfer, ViT, and Swin, alongside CNN architectures with data augmentation and batch normalization. The limitations of traditional methods and introduction to modern approaches like Convolutional Neural Networks (CNNs),

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24MAY1891

YOLO, and Transfer Learning for more efficient recognition were discussed in [8]. Notably, CNNs and YOLO demonstrate impressive real-time recognition speed.

A system that detects hand poses and gestures in IndianSign Language (ISL) in [9] in real time uses a Grid-based Feature Extraction approach to represent the hand's pose asa Feature Vector. Hand poses are then identified using the k-Nearest Neighbors method. For gesture classification, the motion and intermediate hand posture observation sequences are fed into Hidden Markov Model chains that correspond to the 12 pre-selected gestures in ISL. Google's Mediapipe tool was used in [10] to obtain hand landmarks, and a custom dataset of American Sign Language[ASL] was constructed forthe experiment. LSTM was used for hand gesture recognition. The neural network used in [11] is also a Convolutional Neural Network (CNN), which improves the predictability of the American Sign Language alphabet. Their interface system can decode sign language movements and hand positions into natural English. A recognition approach using Microsoft Kinect, CNNs, and GPU acceleration was presented in [12]. CNNs prioritise automated feature building over handwritten features. It is highly accurate in recognizing motions and sign language.It uses a three-module approach. Sign Language Detection, Text to Speech, and Text Scraping.

The proposed system in [13] makes use of Smart Gloves. The recommended solution uses a flex sensor and a micro- processor to transform sign language into text and audio The system analyzes the sign input using a mat lab image process- ing technique and classifies it for recognized identification. Various computer vision techniques are used in [14] suchas grayscale conversion, dilatation, and mask operation. They use a Convolutional Neural Network (CNN) to detect images taken from users' webcam. Publications on decision support and intelligent systems for sign language recognition (SLR) were retrieved and analysed from the Scopus database in [15]. The retrieved articles are analysed using bibliometric VOSViewer software to derive publication temporal and re- gional distributions, establish collaboration networks between affiliations and authors, and identify productive institutions in this context. Deep neural networks were used in [16] to provide a framework for recognizing sign language (SL) signals and converting them into English and voice. CNN and RNN are used for spatial and temporal analysis, respectively.

III. PROPOSED METHODOLOGY

To create a real time sign language recognition system, we wanted to work with Indian Sign Language. Hence, we have created our own image dataset and model for the same.

A. Dataset Collection

For our system, we decided to create and use our own Indian Sign Language(ISL) dataset for all 26 alphabets. All three members of our group have contributed to creating the dataset, and it is done so in different hand positions for added variety. We have tried to maintain a consistent background for our dataset, and avoided too many parameters (detailed background, miscellaneous items in frame, etc.) to not confuse the CNN model. Our dataset consists of approximately 40,000 images for all alphabets, which is around 1,500 images per alphabet, and all images are collected as grayscale. According to the Indian Sign Language Research and Training Center (ISLRTC) website, the English alphabets H, I, J and Y in Indian Sign Language have movement when signing. Hence for our experiment we have chosen static poses for those four alphabets by using the manual alphabet poses provided by ISLRTC's website as reference. All images were taken using the standard digital camera provided by our computer.

B. Data Preprocessing

We collected our original images in 128x128 resolution. This is done to help us reduce storage and avoid higher resolution data. Initial preprocessing steps include taking the grayscale image and adding Gaussian blur, which helps us reduce unnecessary noise and high frequency parts of the image. Our next steps include adding thresholding, which is applied to our blurred grayscale images, and it converts them into a binary inverted image. Here, we make use of Otsu's method as it helps us divide our images into foreground and background, and gives us the optimal value of thresholding we need. Another variation of our dataset was experimented with other procedures like contours for hand segmentation and preprocessing, which were discarded in the final preprocessed dataset as it didn't match our requirements. After thresholding, our dataset now consists of polarizing images, where the hand area of the image is white, thus creating a segmented hand for every image. The lighting had to be balanced the most out of all parameters, to help create the best hand segments for our dataset.

We applied various data augmentation techniques such as rotation, horizontal flipping, shearing, zoom, etc. to our dataset. We performed augmentation for a specific numberof images in each class by using an interval between themand hence created multiple variations of those images, which were then combined with our original dataset. The intervalwas added because augmenting each image would increase ourdataset too much in size, which would not be managable dueto our limitations. We have made use of Image Data Generator package provided by Keras in Python for all augmentation steps. We also experimented with other functions provided like'shift_range' and 'fill_mode' on another copy of our dataset, but it resulted in excessive variation within a class, whichwe wanted to avoid. We shuffled all images before trainingthe model to create more variety within each training batch. Images were put into batch size of 128 before training, andour dataset was divided into 80/20 for training and validation respectively. More augmentation was provided to classes that had less images to balance them with respect to all other classes, and to maintain an average of approximately 1500 images per class.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24MAY1891

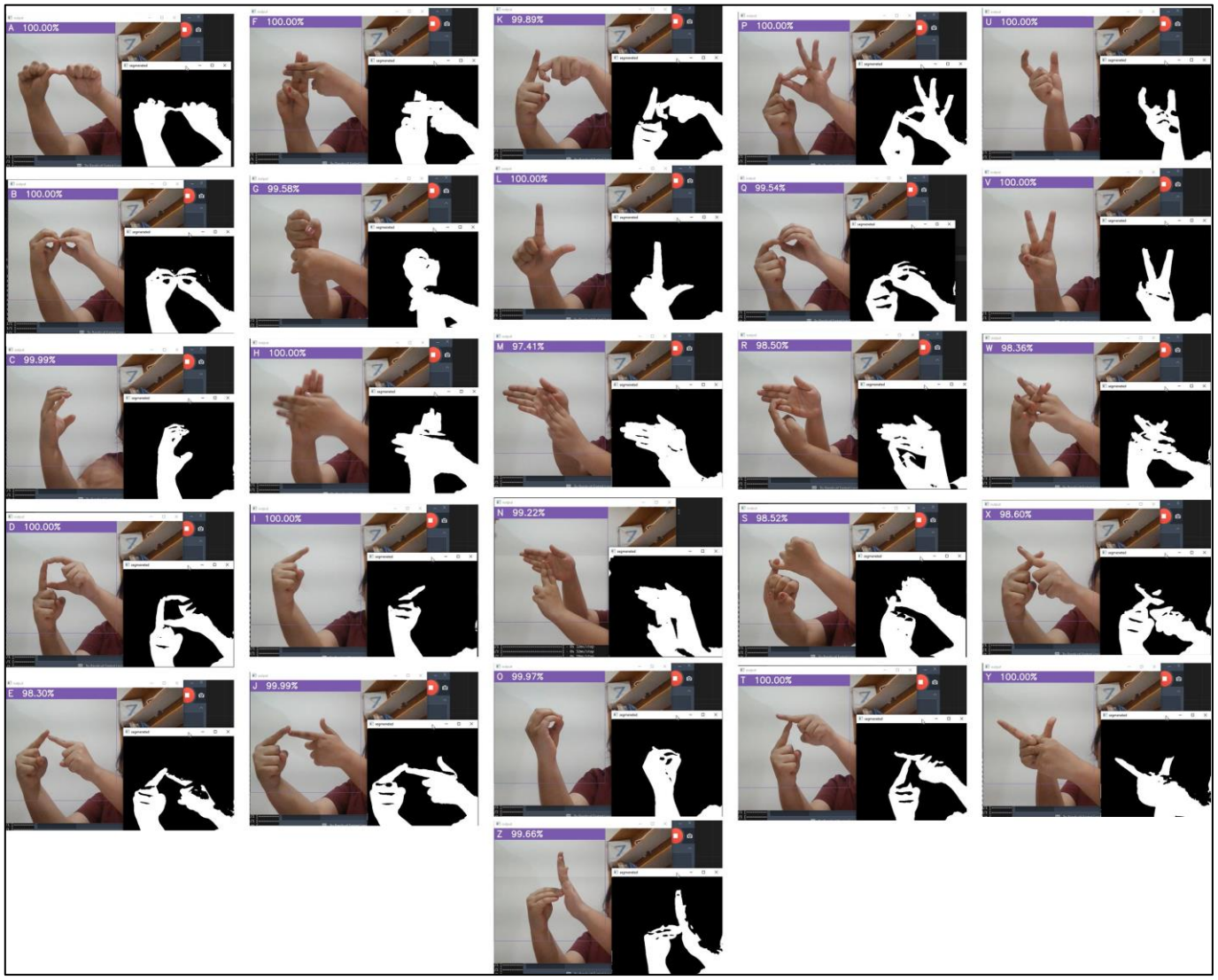

Fig 1: Screenshots of ISL Alphabet Prediction in Real Time

C. Model Building and Training

Convolutional Neural Networks are widely used for image classification, as they are good at discerning, extracting, and then classifying key and important features from the images. We use the CNN model for both extracting features and for classification, as Convolutional Neural Networks can additionaly perform good feature extraction, and we wantedto experiment a CNN's working and result for the same. CNN's lean towards better image recognition when the neural network and feature extraction gets deeper i.e. gets more layers. Hence, we make use of multiple layers for our model. I Our initial ISL dataset consisted of 13,000 images, and hence the model considered for this was a simple 2 layered CNN model which yielded good results, but as we continuedto increase our dataset, this model had to be increased in complexity to provide good results for a much larger dataset.

For our latest and final version of our dataset of 40,000 images, our CNN model initially consisted of 3 convolutional layers which didn't yield best results for our real time predictions. Hence, we changed our batch size to 128 and after analyzing different versions of the model, the final CNN modelfor our experiment consists of 4 convolutional layers and 3 dense layers. Our first layer consists of a convolutional layer with 32 filters with a kernel size of 3x3, followed by a max pooling layer and a dropout layer. Our next layer consists of another convolutional layer with 64 filters, followed by a max pooling and dropout layer. The last two layers follow this samepattern, with the third convolutional layer using 128 filters andthe final layer using 256 filters. For all layers, a kernel sizeof 3x3 was maintained and the activation function used was 'relu', as it provides to the model efficiency and non-linearity.All max pooling layers use a pool size of 2x2 and the dropout value in all layers is 0.2. Further after flattening these layers, we add 3 dense layers, with the first dense layer consistingof 256 neurons followed by a dropout layer of value 0.5.The second dense layer consists of 128 neurons and is also followed by another dropout layer of 0.5 value. Both dense layers used the 'relu' activation function. Our final Dense layerconsists of 26 neurons corresponding to our 26 classes, and uses softmax activation function. For compiling our model weuse the Adam optimizer as it proves efficient for large numberof parameters.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24MAY1891

D. Real Time Prediction

After creating and training our model effectively we perform our final output and prediction in real time. The ISL signs are performed in front of the device camera and the live video frames are extracted from the camera feed and each frame goes through blurring and thresholding as a processing step, similar to how it was performed on the training dataset. Then the CNN model predicts the performed sign based onits training and the predicted label is displayed alongside its accuracy percentage.

The final user interface allows user to use button pressto display the predicted alphabet and be able to display the alphabets to form words. Users can also use our provided GUIto create their own dataset and train the model on custom sign language data. During the real time ongoing prediction user has to simply adjust the background to be plain and makesure the lighting is right enough to create a hand segment, and this is possible as we also display the thresholded view so the user can manage the environment according to the binary feed. We make use of tkinter package provided by pythonto create our interface. We also provide the user a text to speech feature where the predicted alphabet can be converted to speech and be spelled out for the user, and they can also view the sign language images for all alphabets if they wishto see how to perform the signs. These features provide better communication with user in our interface.

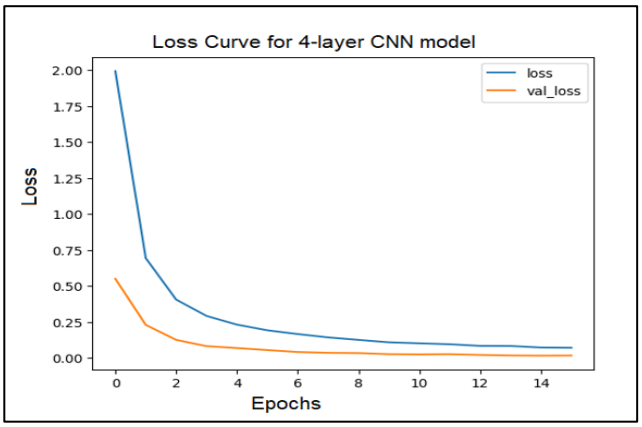

Fig 2: Loss Graph of our CNN Model

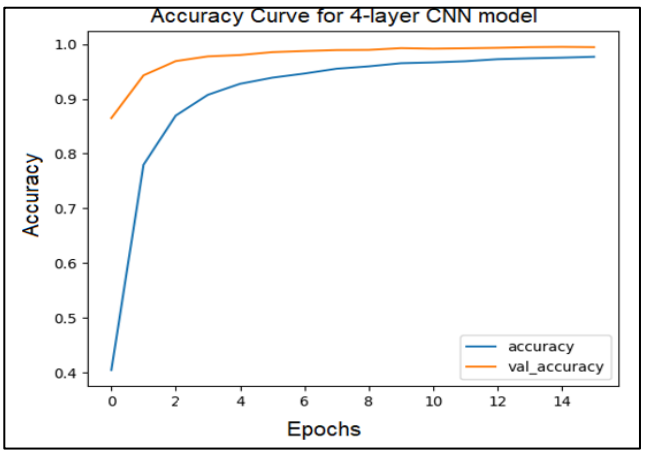

Fig 3: Accuracy Graph of our CNN Model

IV. RESULT ANALYSIS

Screenshot examples of our final real time prediction outputin a suitable environment are displayed in Fig. 1. Our proposedCNN model best predicts in real time alphabets like A, L, P, T,U, V etc., due to their distinct positioning. Our model struggleswith alphabets M, N, R because the hand positions for these signs are similar as seen in Fig. 1 and hence user must adjust hand positions accurately for these signs. We evaluated our model and output using evaluation metrics such as accuracy, precision, etc.

A. Evaluation Metrics Used

Metrics like accuracy, precision etc. are commonly usedto evaluate the performance of machine learning and deep learning models, particularly in classification. The terms in- cluded are True Positives(TP), True Negatives(TN). False Positive(FP), False Negatives(FN). Accuracy: Accuracy mea- sures the overall correctness of the model's predictions across all classes. It gives us the proportion of correctly classified instances among all instances in the dataset. Higher accuracy means a model is giving good predictions but it alone is not enough to assess the performance of a model.

$$\frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (1)$$

- Precision: Precision measures the ability of the model to avoid false positives. It provides the proportion of true positive predictions among all instances predicted as positive by the model. Precision indicates how many of the instances predicted as positive are actually positive. A high precision means that the model has a low false positive rate, which is desirable in applications where false positives are costly.

$$\frac{TP}{(TP + FP)} \quad (2)$$

- Recall: Recall measures the ability of the model to identify all relevant instances. It provides us with the proportion oftrue positive predictions among all actual positive instances in the dataset. Recall indicates how many of the actual positive instances the model can successfully identify. A high recall means that the model can capture most of the positive in- stances, which is crucial in applications where missing positiveinstances is undesirable.

$$\frac{TP}{(TP + FN)} \quad (3)$$

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24MAY1891

- F1 Score: The F1 score is the harmonic mean of precision and recall. It provides a single score that balances both precision and recall. F1 score is especially useful if one has imbalanced classes. As it combines precision and recall into a single metric, it gives equal importance to both. It provides a balanced measure of a model's performance, particularly in situations where precision and recall are equally important.

$$\frac{2 \cdot Precision \cdot Recall}{(Precision + Recall)} \quad (4)$$

We also observed during experimentation that the lesser data we had within our dataset, the simpler the model needed to be to get good accuracy. A complex model for a smaller number of images in a dataset led to training errors like overfitting. Hence balancing the model complexity to the size of ourdataset was an important part in our process. A 4-layer deep CNN model gave us the best accuracy out of all versions of CNN experimented with. The training and validation loss and accuracy curve for the model is displayed in Fig. 2 and Fig.3 respectively.

Our CNN model gives us an overall accuracy of above99%. The models' accuracy on training dataset is above 97% and a validation accuracy of above 99%. The precision, recall. F1 score and accuracy of our CNN model can be seenin Table 1.

Table 1: Metrics Table

Model	Accuracy	Precision	Recall	F1 Score
4-layer CNN	99.4%	99.5%	99%	99.2%

For further evaluation we considered two different custom datasets. Dataset 1 led to Accuracy of 99.96% and dataset 2 led to an accuracy of 98.75% as seen in Fig. 4 and 5.

Our CNN model is trained on both single-handed and double-handed signs and is able to provide a good accuracy for both types of hand signs.

```
loss,accuracy=model.evaluate(testing_data)
print(f'Test Loss: {loss*100}, Test Accuracy: {accuracy*100}')

21/21 [==============================] - 4s 193ms/step - loss: 0.0012 - accuracy: 0.9996
Test Loss: 0.12135505676269531, Test Accuracy: 99.96204972267151
```

Fig 4: Test Accuracy of Model on Dataset 1

```
loss,accuracy,precision,recall=model.evaluate(validation_generator)
print(f'Test Loss: {loss*100}, Test Accuracy: {accuracy*100}')
print(f'Test precision: {precision*100}, Test recall: {recall*100}')
✓ 15.4s

66/66 [==============================] - 15s 230ms/step - loss: 0.0494 - accuracy: 0.9875
Test Loss: 4.940858483314514, Test Accuracy: 98.75075221061707
```

Fig 5: Test Accuracy of Model on Dataset 2

V. LIMITATIONS AND FUTURE SCOPE

Our current system can recognise 26 alphabets with a database of 40,000 but due to device limitations training the model was outstretched and time consuming. Hence adding further updates like GPU can help make the process smoother and provide faster training which can lead to better testing and more time for experimenting. Further we have certain environment limitations in our experiment in real time like background and lighting and hence applying methods that can help deal with those issues can create better sign language recognition systems. Sign language recognition can also be implemented into video calls for providing further accessibility features to deaf and mute people. Our model works with a database of images of static poses, which can be expanded intoworking with video database of hand poses that

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24MAY1891

have move- ment. Our model can also be used to recognise alphabets in other Sign Languages such as American Sign Language(ASL),British Sign Language(BSL), etc if trained with respectivedatasets.

VI. CONCLUSION

In our paper, we display our experimentation and approach to recognise Indian sign language input in real time using Convolutional Neural Network with an overall accuracy ofthe model being above 99%. Our model provides quality and favourable recognition for both one-handed and two-handed signs done by user with limited real time delay and can be expanded to work with larger datasets. It can be determined that multiple layers of Convolutional Neural Networks providebetter accuracy for larger datasets, as it did for our custom created dataset. CNN proves to be very competent and useful to work with image dataset and extracts relevant features,good enough to predict proficiently in real time. Thus through our research we can establish that using Neural network technologies like CNN can help create systems for deaf, mute and disabled communities which help make communication easier for them through such technological advancements.

REFERENCES

[1]. Prachi Sharma, Radhey Shyam Anand.“A comprehensive evalua- tion of deep models and optimizers for Indian sign language recognition”. Graphics and Visual Computing 5 (2021) 200032. https://doi.org/10.1016/j.gvc.2021.200032

[2]. Ruiqi Sun, Qin Zhang, Chuang Luo, Jiamin Guo, Hui Chai. “Human ac- tion recognition using a convolutional neural network based on skeleton heatmaps from two-stage pose estimation” Biomimetic Intelligence and Robotics 2 (2022) 100062. https://doi.org/10.1016/j.birob.2022.100062

[3]. PRABHAKARA R UYYALA. “SIGN LANGUAGE RECOGNITION USING CONVOLUTIONAL NEURAL NETWORKS” Journal of In- terdisciplinary Cycle Research Volume XIV, Issue I, January/2022 ISSNNO: 0022-1945.

[4]. Lionel Pigou(B), Sander Dieleman, Pieter-Jan Kindermans, Benjamin Schrauwen. “Sign Language Recognition Using Convolutional Neural Networks” ELIS, Ghent University, Ghent, Belgium (2015)

[5]. Ahmed Adel Gomaa Elhagry, Rawan Gla Elrayes. “Egyptian Sign Language Recognition Using CNN and LSTM” Computer Vision and Pattern Recognition(2021)

[6]. Quiroga, Facundo — Antonio, Ramiro — Ronchetti, Franco — Lan- zarini, Laura Cristina — Rosete, Alejandro “A Study of Convolutional Architectures for Handshape Recognition applied to Sign Language” CACIC 2017

[7]. Nojood M. Alharthi, Salha M. Alzahrani.“Vision Transformers and Transfer Learning Approaches for Arabic Sign Language Recognition”. Applied Sciences(2023).

[8]. Xianwei Jiang, Yanqiong Zhang1,Juan Lei and Yudong Zhang. “A Survey on Chinese Sign Language Recognition: From Traditional Meth-ods to Artificial Intelligence” Computer Modeling in Engineering & Sciences Tech Science Press(2024)

[9]. Kartik Shenoy, Tejas Dastane, Varun Rao, Devendra Vyavaharkar “Real-time Indian Sign Language (ISL) Recognition” 9th ICCCNT 2018, IISC, Bengaluru

[10]. Sundar B. , Bagyammal T. “American Sign Language Recognition for Alphabets Using MediaPipe and LSTM ” 4th International Confer- ence on Innovative Data Communication Technology and Application. 10.1016/j.procs.2022.12.066

[11]. Ahmed KASAPBAS Ahmed Eltayeb AHMED ELBUSHRA Omar AL- HARDANEE Arif YILMAZ “DeepASLR: A CNN based human com- puter interface for American Sign Language recognition for hearing- impaired individuals ” Computer Methods and Programs in Biomedicine Update 2 (2022) 100048. https://doi.org/10.1016/j.cmpbup.2021.100048

[12]. Piyush Kapoor, Hema N2 “Sign Language and Common Gesture Using CNN ”. International Journal of Advanced Trends in Computer Science and Engineering ISSN 2278-3091

[13]. Karan Bhavsar, Raj Ghatiya, Aarti Gohil, Devanshi Thakkar, Bhumi Shah. .“Sign Language Recognition”. International Journal of Research Publication and Reviews Vol (2) Issue (9) (2021) Page 771-777.

[14]. Rachana Patil, Vivek Patil, Abhishek Bahuguna, and Mr. Gaurav Datkhile. “Indian Sign Language Recognition using Convolutional Neu-ral Network”. ITM Web of Conferences 40, 03004 (2021) ICACC-2021.

[15]. I.A. Adeyanju, O.O. Bello b, M.A. Adegboyega. “Machine learning methods for sign language recognition: A critical review and analysis”.

[16]. Intelligent Systems with Applications 12 (2021) 200056

[17]. Sai Bharath Padigala, Gogineni Hrushikesh Madhav,Saranu Kishore Kumar, . International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056Dr. Narayanamoorthy M. “VIDEO BASED SIGN LANGUAGE RECOGNITION USING CNN-LSTM”. Interna- tional Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056