IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

ANIMAL SPECIES RECOGNITION USING TRANSFER LEARNING

[1]Mrs. Bhavana P, [2]Bumen Mangu, [3]Jatin Thakan, [4]Rahul Tiwari, [5]VNasir A,

[1]Assistant Professor, [2]Student, [3]Student, [4]Student, [5]Student,[1]Computer Science and Engineering,

[1]Cambridge Institute of Technology, Bengaluru, India

Abstract: The sign language is used by people with hearing / speech disabilities to express their thoughts and feelings. But normally, people find it difficult to understand hand gestures of the specially challenged people as they do not know the meaning of sign language. Our project aims to develop a system forsign language recognition using MediaPipe ,LSTM, and Keras. The proposed system utilizes a webcam to capture real-time video input of a person performing sign language gestures. MediaPipe is used to extract and track the hand landmarks and their movements in the video stream. The features are then processed using LSTM, which is a sequence modeling technique that captures the hand gestures. Finally, a deep learning LSTM model implemented in Keras and trained to recognize the different sign language gestures. The system can potentially be used to assist people with hearing. Long Short-Term Memory (LSTM) neural network architecture to get this remarkable feat. When someone performs sign language gestures in front of a camera, the system instantly recognizes and interprets those gestures.

Keywords - Animal species recognition, deep convolutional neural networks, transfer learning, camera-trap, KTH dataset.

I. INTRODUCTION

This paper advocates for hand gesture recognition technology, particularly its potential to transform sign language translation. It dissects the intricate techniques employed, categorizing them into distinct stages: data acquisition (capturing sign language gestures), pre-processing (refining the data), segmentation (isolating the hand region), feature extraction (identifying key characteristics), and classification (matching extracted features to specific signs). The paper provides a nuanced comparison of various algorithms at each stage, acknowledging both their strengths and limitations. Furthermore, it recognizes the broader challenges inherent to gesture recognition, such as variations in signing styles, along with those specific to sign language recognition, like incorporating the complementary role of facial expressions. In essence, this review offers invaluable insights for researchers striving to improve sign language recognition technology, paving the way for a future where communication transcends the constraints of physical limitations.

II. OBJECTIVE

This project Sign language recognition using Python is pivotal for converting sign gestures into text or speech, facilitating communication for the hearing impaired. Key objectives revolve around enhancing the system's capabilities and usability. Real-time recognition ensures swift communication without delays, while continuous accuracy improvement, employing algorithms such as CNNs or RNNs, refines gesture interpretation. Expanding the recognized vocabulary broadens communication possibilities.

Translation of gestures into spoken language or text bridges communication gaps between sign language users and others. Integration into diverse applications, robustness in varying environments, and user-friendly interfaces enhance accessibility and usability. Ensuring the system caters to users with different proficiency levels fosters inclusivity. Implementing evaluation mechanisms for iterative improvements and providing comprehensive documentation and educational resources empower developers and users alike. Ultimately, these objectives align to create a system that not only interprets sign language effectively but also promotes seamless communication, inclusivity, and accessibility for all individuals, regardless of their hearing abilities.

III. EXISTING SYSTEM

The existing system it involves capturing video input of sign language gestures, preprocessing the data to enhance clarity, and then employing algorithms such as deep neural networks (DNNs), convolutional neural networks (CNNs), or recurrent neural networks (RNNs) for feature extraction and recognition. These systems often struggle with accuracy, especially in complex gestures or varying environmental conditions. Additionally, limited vocabulary coverage and lack of real-time performance are common challenges. While some systems focus on specific sign languages, others attempt to generalize across multiple languages. However, interoperability and adaptability remain significant issues. Overall, while existing systems provide a foundation for sign language recognition, there's considerable room for improvement in terms of accuracy, real-time performance, vocabulary coverage, and adaptability to diverse environments and languages.

IV. SCOPE OF STUDY

The scope of study for sign language recognition encompasses several key areas. Firstly, it involves exploring and refining machine learning and computer vision techniques to improve accuracy and real-time performance in recognizing sign gestures. Additionally, expanding the vocabulary coverage and accommodating various sign languages are crucial aspects. Moreover, ensuring the system's robustness in different environmental conditions and addressing interoperability challenges across platforms are essential considerations. Furthermore, the scope includes designing user-friendly interfaces and integrating the system into diverse applications to enhance accessibility for both sign language users and non-users. Overall, the scope of study aims to advance sign language recognition systems to foster better communication and inclusivity for the hearing impaired.

V. CONCEPT OF THE MODEL

The model captures video frames or analyzes images using computer vision techniques. Think of it as a digital eye that focuses on the scene.Next, it employs hand detection algorithms to isolate the hands from the background. This is like zooming in on the "conversation" happening through hand gestures. Once the hands are isolated, the model extracts key features. It analyzes hand posture, finger positions, and even movement patterns. Imagine these features as the "letters" of sign language. Finally, the model uses a machine learning algorithm, trained on a massive dataset of labeled sign language examples. This acts like a dictionary, matching the extracted features to the most likely corresponding sign.The complexity of the model depends on our needs. We can use simpler tools like OpenCV or MediaPipe for basic recognition, or leverage deep learning frameworks like TensorFlow for higher accuracy but with more computational power required.Ultimately, the model's effectiveness will be measured by its ability to accurately recognize signs. We'll use metrics like accuracy, precision, and recall to assess its performance. This ensures the model truly "understands" the sign language gestures it encounters.

VI. PROPOSED SYSTEM

The image classification process in this project involves several critical stages: dataset preprocessing, model training, validation, and evaluation. Initially, the dataset undergoes segmentation into three subsets: training, validation, and test sets. During training, a Convolutional Neural Network (CNN) model learns crucial features from the data and extracts relevant information essential for classification.A Python-based sign language recognition system acts as a bridge for communication by either capturing live video or using pre-recorded data. Using libraries like OpenCV and MediaPipe, the system isolates the hand and extracts features such as hand shape and key points. MediaPipe provides efficient hand tracking and landmark detection functionalities, enhancing the accuracy of feature extraction.After extracting features, they are transferred for training the machine learning model. This model, often a CNN, is trained to recognize various sign language gestures. Utilizing TensorFlow/Keras, the model iteratively learns to associate extracted features with corresponding signs.Once the model is adequately trained, it can recognize sign language gestures in real-time or from pre-recorded videos. The integration of OpenCV, MediaPipe, and TensorFlow/Keras libraries facilitates the development of a robust sign language recognition system. However, challenges such as variations in lighting conditions and complex hand gestures necessitate continuous advancements in the field.

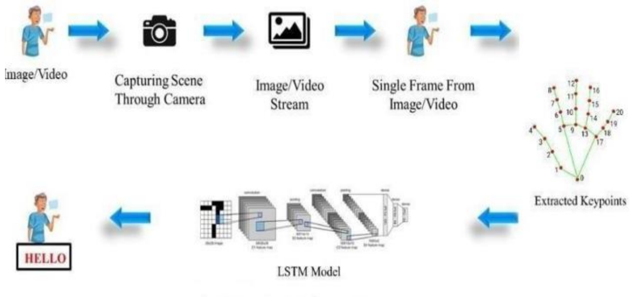


Figure1. The process flow of the image classification pipeline includes preprocessing the raw dataset, training and validating the architecture, and finally testing the finalized model with diverse sets of samples.

VII. FLOW CHART

The animal species recognition project's workflow involves collecting camera trap images of diverse animal species, labeling them accordingly, preprocessing the images, and dividing the dataset into training, validation, and test sets. These sets are then used to train various deep learning models, such as Inception V3, ResNet50, AlexNet, VGGNet-16 and VGGNet-19 to recognize animal species. After training, the models are evaluated using the validation set to select the best-performing one, which is further tested on the test set to validate its accuracy. Additionally, potential real-world applications of the chosen model are explored before deploying it for practical use. The process encompasses defining the architecture diagram, preprocessing raw data, training, and evaluating multiple deep learning algorithms, and selecting the most effective model for animal species recognition based on classification accuracy.

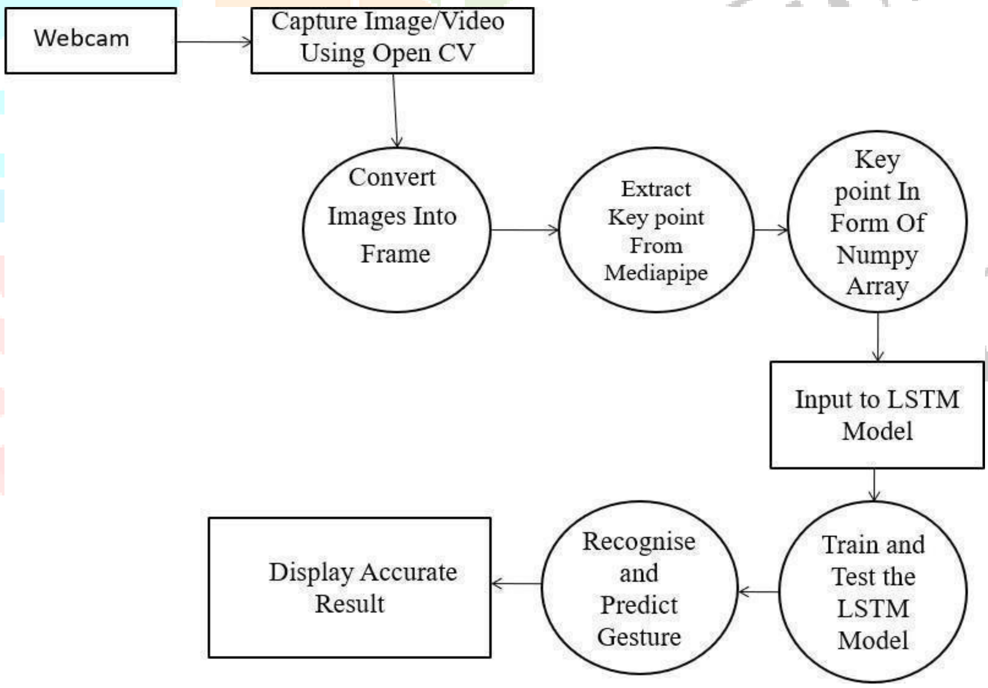


Figure 2. Flow chart of the working model

VIII. DATASET

The dataset used in this project typically consists of video recordings or images depicting various sign language gestures. These datasets can vary in size and complexity, ranging from a few hundred to several thousand samples. They may include recordings of individuals performing sign language gestures from different angles, with variations in lighting conditions and backgrounds.

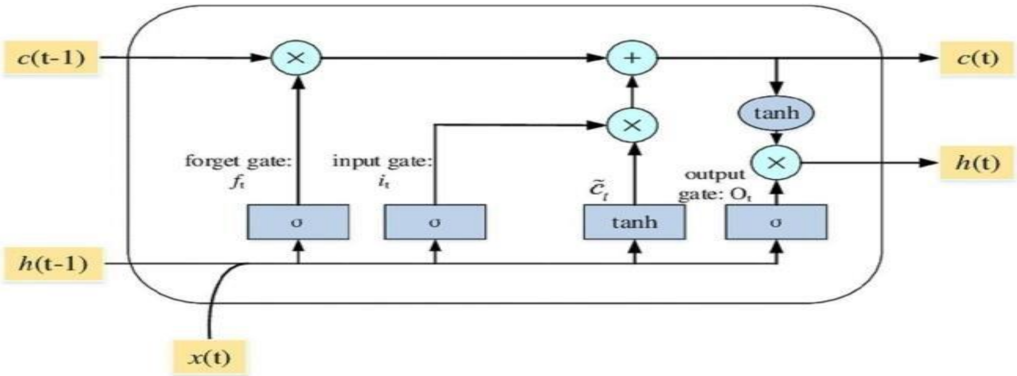
Figure 3. Sample images from dataset.

In this data acquisition and labeling process, images are initially captured using OpenCV, with time breaks introduced to accommodate hand movements. The captured images are then labeled using the LabelImg package, where hand gesture portions are identified and labeled according to their representation. Alongside each image, XML files are generated to store labeling details. Subsequently, TensorFlow records are created using the labeled images and their corresponding XML files. These records are organized into training and validation datasets, with an 80:20 split, where 80% of the images are allocated for training and 20% for validation. This process is repeated for all 26 alphabets, ensuring a comprehensive dataset for training machine learning models, potentially for tasks like hand gesture recognition.

IX. METHODOLOGY

Our proposed system is sign language recognition using system convolution neural networks which recognizes various hand gestures by capturing video and converting it into frames. Then the hand pixels are segmented and the image it obtained and sent for comparison to the trained model. Thus our system is more robust in getting exact text labels of letters. the hand pixels are segmented and the image it obtained and sent for comparison to the trained Cell state is crucial for LSTM, it resembles a conveyor belt. The chain follows with minimal interactions, allowing information to easily pass through unchanged. LSTMs use gates to control the addition or removal of information from a cell`s state. Gates selectively pass information. They're made with a sigmoid NN layer & point multiplication.

Cell state is crucial for LSTM, it resembles a conveyor belt. The chain follows with minimal interactions, allowing information to easily pass through unchanged. LSTMs use gates to control the addition or removal of information from a cell`s state. Gates selectively pass information. They're made with a sigmoid NN layer & point multiplication. The sigmoid layer produces numbers from to 1, showing how much of each component to pass through. A esteem of zero implies "skip nothing," and a esteem of one implies "skip everything." An LSTM has three of these entryways, to ensure and control the cell state. The primary step in our LSTM is to decide what data we'll



evacuate from the cell state. This choice is made by the "overlooked entryway layer," the sigmoid layer. analyzes ht1 and xt-1 and returns the esteem of each number between and 1 in cell Ct1's state. 1 implies "keep totally" and implies "get freed of totally". The next step is to decide what unused data we are going store within the cells. It is separated into two parts. To begin with, a sigmoid layer known as the "input gating layer" decides which values got to be overhauled. The tanh layer is then utilized to produce a vector Ct of unused candidate values that can be included to the state. Within the following step, we'll combine the two to make a status upgrade. It is time to move from the ancient cell state.

Implementation involves executing a plan, method, or design to achieve a particular goal.The step that comes after the initial planning phase. During implementation, libraries such as Keras are imported for training LSTM models and recognizing sign language gestures. Sign language recognition with deep learning models is vital for aiding communication among speech-impaired and hearing-impaired individuals. These systems interpret natural hand gestures through MediaPipe Holistic, which integrates pose, hand, and face keypoints with high accuracy and low latency in real-world scenarios. LSTM networks are important in capturing temporal dependencies in sequences as part of RNNs. They excel at recognizing sign language gestures, especially those with sequential movements. For example, they have studied American Standard Sign Language (ASSL). By training an LSTM model with TensorFlow and Keras, they achieved an accuracy of 98.50% in detecting ASSL gestures. Real-time gesture detection involves capturing frames and predicting sign language gestures on-screen. The system has a userfriendly interface where users can perform gestures in front of a camera, and it instantly detects and interprets the gestures.

X. RESULTS AND DISCUSSION

The developed system is able to detect Indian Sign Language alphabets in real-time. The system has been created using TensorFlow object detection API. The pre-trained model that has been taken from the TensorFlow model using 320x320. It has been trained using transfer learning on the created dataset which contains 650 images in total, 25 images for each alphabet10 The total loss incurred during the last part of the training, at 10,000 steps was 0.25, localization loss was 0.18, classification loss was 0.13, and regularization loss was 0.10 The result of the system is based on the confidence rate and the average confidence rate of the system is 85.45%. For each alphabet, the confidence rate is recorded and tabulated in the result as shown in Table 1. The confidence rate of the system can be increased by increasing the size of the dataset which will boost up the recognition ability of the system. Thus, improving the result of the system and enhancing it.

.

A	B	C	D	E	F	G	H	I
94%	98%	90%	90%	70%	96%	73%	97%	95%
J	K	L	M	N	O	P	Q	R
57%	87%	93%	91%	55%	78%	95%	95%	83%
S	T	U	V	W	X	Y	Z	
86%	81%	87%	86%	87%	88%	90%	80%	

The state-of-the-art method of the Indian Sign Language Recognition system achieved 93-96% accuracy [4]. Though being highly accurate, it is not a real-time SLR system. This issue is dealt with in this paper. In spite of the dataset being small, our system has achieved an average confidence rate of 85.45%.

XI. CONCLUSION

Improved accessibility for the deaf people ,hard-of hearing community in various aspects of life, such as education, communication, and public services . Facilitating real-time communication between people who use sign language and those who don't, breaking down language barriers. Integration into devices like smartphones, tablets, and computers to enable direct communication through video calls, messaging, and social media. Integration into robotics and automation, allowing machines to interpret sign language for improved humanrobot interaction. Development of assistive technologies and devices that can interpret sign language, aiding in various contexts like healthcare, customer service, and public spaces. Further research in the sign language linguistics and recognition could lead to advancements in understanding the complexities and nuances of sign languages, potentially contributing to linguistic studies. Empowering deaf community by providing tools that support their language and culture, promoting inclusivity and equality.

REFERENCES

[1] Lean Karlo; Ronnie Juan; August C. Thio-ac; Maria Pamahoy;

[2] Joni Rose Forteza; Xavier Garcia; "Static Sign Language Recognition Using Deep Learning" ; International Journal of

[3] Machine Learning and Computing, Vol. 9, No. 6, December 2019.

[4] Brandon Garcia; Sigberto Alarcon Viesca; "Sign Language Recognition Using Image Based Hand Gesture Recognition"; ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538Volume 10 Issue VIII

August 2022M

[5] Dhivyasri S; Krishnaa Hari K B; Akash M; "Efficient Approach for Interpretation of Sign Language using Machine

[6] Learning"; 2021 3rd International Conference on Signal

[7] Processing and Communication (ICPSC) | 13 – 14 May 2021

[8] Kumus Tripathi; Neha baranwal; G C Nandi; "Continuous Indian Sign Language Gesture Recognition and Sentence Formation"; Eleventh International Multi-Conference on

[9] Information Processing-2015

[10] Ming Jin Cheok ; Zaid Omar1 ; Mohamed Hisham Jaward ; "Hand Gesture Recognition for Sign Language: A review" ; Int. J. Mach. Learn. & Cyber. DOI 10.1007/s13042-017-0705-5 ; July 2017

[11] Shreyasi Narayan Sawant ; M. S. Kumbhar ; "Real Time SignLanguage Recognitionusing PCA" ; IEEE International Conference on Advanced Communication Control and Computing Technologies (lCACCCT) 2014

[12] Daniele Cippitelli, Davide Cipolla, Proposed a DeepASL: Enabling Ubiquitous and Non Intrusive Mobile Sign Language

[13] Recognition, IEEE 2018

[14] Thad Starner, Mohammed J. Islam, Proposed a Sign Language Recognition with Microsoft Kinect, IEEE 2013.

[15] Alex Graves, Santiago Fernández, Proposed a Sign Language Recognition Using a Convolutional Neural Network, IEEE 2018.

[16] Juyoung Shin, Joo H. Kim, Proposed a Sign Language Translation and Recognition using Wearable Myoelectric Sensors, IEEE (2017)

[17] E. Assogba and P. H. S. Amoudé, Proposed a Deep Learning for Sign Language Recognition and Translation, IEEE 2019

[18] Oscar Koller, David Ney, Proposed a Neural Machine Translation for Sign Language: A Survey,IEEE2020

[19] Hrishikesh Kulkarni, Suchismita Saha, Proposed a Deep Learning-Based American Sign Language (ASL) Recognition

[20] System, IEEE 2020

[21] Chien-Wei Wu, Eugene Lai, Proposed a Sign Language Recognition Using 3D Convolutional Neural Networks, IEEE 2019

[22] Siawpeng Er, Jie Zhang, Proposed a Sign Language Recognition and Translation: A Multimodal Deep Learning Approach, IEEE 2020

[23] Zixia Cai, Yu Che, Proposed A Survey Sign Language Recognition and Translation Systems,IEEE 2021.

[24] Sevigi Z Gurbaj , Evie A. malaia, Proposed a American Sign Langauge recognition using RF sensing, IEEE 2021.

[25] Jiming Pan , Yuxuan, Proposed A wireless multi-channel capasitive sensor system for efficient globe gesture recognition with AI at the edge IEEE,2020.