# Hands Sign Language and Gesture Recognition Using CNN

Manish Kumar[a], Tarun Kumar [b], Anirudh Kumar[c], Dr. K.K. Agrawal [*],
[a] School of Computer Science & Engineering, Galgotias University, Greater Noida, UP, India
[b] School of Computer Science & Engineering, Galgotias University, Greater Noida, UP, India
[c] School of Computer Science & Engineering, Galgotias University, Greater Noida, UP, India
[*]Professor, Galgotias University, Greater Noida, UP, India

Abstract— One of the non-verbal communication techniques utilised in sign language is the hand gesture. People with speech or hearing impairments use it most frequently to converse with one another and with non-disabled persons. Several makers around the world have created various sign language systems, but they are neither adaptable nor cost-effective for end users. So, the "Hand sign and gesture recognition system software" presented in this proposal offers a system prototype that can automatically understand sign language to aid deaf and dumb individuals in communicating with each other and other people more efficiently. In sites like "YouTube" videos where there is currently no feature for automatic text generation on the basis of gestures, this approach can also be employed, likewise sign languages. Research on gesture recognition is still in its early stages. Hand gestures play a crucial role in nonverbal communication and are essential to daily living. The software aims to present a real-time system for hand gesture and sign recognition on the basis of detection of some shape-based features like orientation, Center of Mass centroid, fingers status, and thumb in positions of raised or folded fingers of hand, although these feature extraction part will be completely resolved by Convolutional Neural Network (CNN). Every frame of the video will be captured in this process, and each frame will be used to locate hands and clip them out so that our CNN may use them as input. We used ISL as a case study for our purposes. The back projection histogram approach was employed in this model to set the image's histogram. We used CNNs for training and testing, and as a result, our test accuracy was 99.89%. Our model's independence from external hardware or devices is one of its benefits.

Keywords— Convolutional Neural Network (CNN)

## I. INTRODUCTION

Our lives are growing more and more dependent on computers and other technological gadgets. The need for simple and useful computer interfaces developed as the demand for such computing devices expanded. Due to the increasing use of systems that rely on vision-based interface and control, gesture recognition is growing in popularity in the research community due to its many potential applications in human-machine interaction. Because gestures are so intuitive, any vision-based interface is more practical, comfortable, and natural than a mouse and keyboard. There are primarily three ways to perform gesture recognition: using wearable gloves, 3-dimensional placements of hand key points, and raw visual data. While producing strong results in terms of precision and speed, the first technique requires wearing a separate gadget that comes with numerous cords. The second, on the other hand, necessitates an additional step of manually extracting key points. A sliding window technique with a stride of one is used to process the video stream. The likelihood ratings for the detector, which is turned on when a gesture begins and remains on until it is over, are displayed in the top graph. The second graph uses a different colour to indicate the categorization score for each class. The third graph reduces the uncertainty between potential gesture candidates by applying weighted average filtering to raw classification results. On the bottom graph, which shows single-time activations, red arrows denote early detections and black ones, detections made after gestures had finished, respectively. additional processing time and expense. Finally, just an image-capturing sensor—such as a camera, an infrared sensor, or a depth sensor—that is not user-dependent is needed for (iii). This solution stands out as the most practical one since the user does not need to wear a cumbersome gadget in order to acquire an acceptable level of recognition accuracy and an adequate rate of computation. Any system for gesture recognition needs to have a workable infrastructure. After all, we want to apply knowledge to actual situations. We have created a vision-based gesture identification method in this work employing deep convolutional neural networks (CNNs) on raw video data to offer a workable solution. Presently, CNNs deliver the most cutting-edge results for tasks that use both images and videos, including gesture recognition, activity localisation, and image-based tasks like object detection, segmentation, and classification. There are numerous criteria that the system must meet in real-time gesture recognition applications: A single activation for each executed gesture, along with I an adequate classification accuracy, (ii) quick reaction time, (iii) resource efficiency, and (iv) single activation. For a real-time vision-based gesture

recognition programme to be successful, each of these components is of vital importance. However, the majority of earlier research ignores the remaining items and focuses solely on improving offline classification accuracy in gesture recognition. Due to the several deep CNNs that some proposed systems use on various input modalities and their inability to function in real-time, they are pushing the memory and power budgets to their limits. Convolutional Neural Networks (CNNs) have gained widespread popularity in recent years due to their impressive performance in a variety of computer vision tasks such as object recognition, image classification, and segmentation. CNNs have revolutionized the field of computer vision by enabling automated image analysis and interpretation with remarkable accuracy.

CNNs are a type of artificial neural network that is inspired by the human visual system. They consist of several layers, including convolutional layers, pooling layers, and fully connected layers. Convolutional layers are responsible for detecting patterns and features in the input image. They apply a set of filters to the input image and produce a feature map that highlights the most important features. Pooling layers are used to reduce the spatial dimensionality of the feature maps, which helps to decrease the computational cost and prevent overfitting. Finally, the fully connected layers are used to classify the input image into one of the predefined categories.
One of the key strengths of CNNs is their ability to learn hierarchical representations of visual data. By using multiple layers, CNNs can automatically learn a hierarchy of features that captures both low-level and high-level visual information. For example, the first few layers might learn simple features such as edges and corners, while the higher layers might learn more complex features such as shapes and textures.
Another advantage of CNNs is their ability to handle input images of different sizes and aspect ratios. Unlike traditional computer vision methods that require images to be preprocessed to a fixed size, CNNs can learn to recognize objects and patterns in images of varying sizes and orientations.

CNNs have been successfully applied in a wide range of applications, including self-driving cars, medical image analysis, and facial recognition. They have also been used to develop sophisticated algorithms for tasks such as image captioning and visual question answering.
In conclusion, CNNs have emerged as a powerful tool for image analysis and interpretation. They have demonstrated remarkable performance in various computer vision tasks and have the potential to transform the field of artificial intelligence. As research in this area continues, we can expect to see further advancements in CNNs that will enable them to tackle even more complex visual tasks.
Hand and sign gesture detection is an important area of research in computer vision, with applications ranging from sign language interpretation to human-computer interaction. Convolutional neural networks (CNNs) have proven to be highly effective in this field, offering state-of-the-art performance on various datasets.
CNNs are a type of deep neural network that can learn hierarchical representations of images through a series of convolutional and pooling layers. These networks have been shown to be highly effective at learning features that are invariant to translation, rotation, and scaling, making them ideal for tasks such as hand and sign gesture detection.
One common approach to hand and sign gesture detection using CNNs involves training the network on a large dataset of labeled images. The dataset typically includes a wide variety of hand and sign gestures, captured from different angles and under different lighting conditions. The CNN is trained to learn discriminative features from these images, and to classify them into different categories based on the gesture being performed.
One of the key challenges in hand and sign gesture detection is dealing with the variability in hand shapes and orientations. To address this challenge, researchers have developed a range of CNN architectures that are designed to be robust to variations in hand pose and appearance. For example, some architectures incorporate recurrent neural networks (RNNs) to capture the temporal dynamics of hand movements, while others use attention mechanisms to focus on specific parts of the hand.
Another important consideration in hand and sign gesture detection is real-time performance. CNNs can be computationally intensive, and may require significant processing power to run in real-time on a mobile or embedded device. To address this issue, researchers have explored various techniques such as model compression, pruning, and quantization to reduce the computational requirements of CNNs.
In conclusion, CNNs have proven to be highly effective in hand and sign gesture detection, offering state-of-the-art performance on various datasets. These networks can learn discriminative features that are invariant to variations in hand pose and appearance, and can be designed to run in real-time on resource-constrained devices. Further research in this area is likely to focus on developing more efficient and robust CNN architectures for hand and sign gesture detection.
Hand sign and gesture recognition technology have a wide range of real-life applications across various fields. Here are some possible applications are:

1. Sign language translation: Hand sign and gesture recognition technology can be used to translate sign language into spoken language or written text, allowing people who are deaf or hard of hearing to communicate more effectively with those who do not understand sign language.
2. Human-computer interaction: Hand sign and gesture recognition technology can be used to control computers and other devices, allowing users to interact with technology in a more intuitive and natural way. For example, users can navigate through menus, zoom

in and out of images, or play games using hand gestures.

3. Robotics: Hand sign and gesture recognition technology can be used to control robots, allowing them to perform tasks more efficiently and effectively. For example, robots can be trained to recognize human gestures and respond accordingly, such as picking up objects or moving in a certain direction.
4. Healthcare: Hand sign and gesture recognition technology can be used to monitor patients and provide healthcare services remotely. For example, doctors can use hand gestures to control medical equipment during surgeries or patients can use hand gestures to communicate with their caregivers.
5. Education: Hand sign and gesture recognition technology can be used to improve the learning experience for students with disabilities. For example, teachers can use hand gestures to communicate with students who are deaf or hard of hearing or students can use hand gestures to interact with educational software.
6. Security: Hand sign and gesture recognition technology can be used for security purposes, such as identifying individuals based on their unique hand gestures. For example, hand gesture recognition can be used to control access to secure areas or to monitor crowds for suspicious behavior.
7. Entertainment: Hand sign and gesture recognition technology can be used to create interactive and immersive entertainment experiences. For example, users can control virtual characters or objects using hand gestures or participate in gesture-based games.

## II. LITERATURE SURVEY

In order to improve communication between deaf communities and others, gesture-based sign language recognition systems are crucial. Convolutional Neural Network (CNN) experiments have been done to recognise gestures after some pre-processing of input data from input devices. Yet, in those experiments, the complexity and diversity of hand gestures had a significant impact on the accuracy and identification rates. In their research work, the authors (Md Abdur Rahim, Jungpil Shin, and Md Rashedul Islam) present a successful solution to this issue: hand gesture detection using CNN with improved data pre-processing, such as feature fusion CNN, RGB colour input to YCbCr, binarization, erosion, and hole filling. Instead of concentrating solely on data pre-processing as other recent research papers at that time did, the authors of this study (Md. Rashedul Islam, Rasel Ahmed Bhuiyan, Ummey Kulsum Mitu, and Jungpil Shin) came up with a novel way to categorise them using Multi-class Support Vector Machine (MCSVM), which increases productivity and efficiency as the number of specimens or categories rises (types of gestures). After subdividing the multiclass problem into smaller problems, all of which are binary classification problems, the general idea of SVM is used in MCSVM. As demonstrated in the research report, this strategy improves accuracy, however data pre-processing is also employed to achieve high accuracy. They removed background images from the submitted photographs, leaving only those that showed the Area of Interest (ROI), in this case, the hands. Nevertheless, because they utilised a very ineffective technique to eliminate background images—taking two photos, one with ROI and the other without—the speed of prediction was significantly reduced, costing the model an approximate average accuracy of 95%. Afterwards, background removal, filtering, noise reduction, and grayscale conversion were used. [1]. Patel & et al. developed a static hand gesture identification method for American Sign Language using deep convolutional neural network. The system architecture weighs little, making it easy to deploy and transfer the system around. to attain high accuracy in real-time conditions, a variety of image processing methods are used to assist with background reduction and frame segmentation. The approach emphasises mobility, straightforward deployment at no cost, and no computational overhead. The approaches used in this are feature extraction, the Hand Segmentation Approach (HSA), and glove-based hand motion detection. During image processing and frame segmentation, the model is implemented using a Gaussian Mixture-based background segmentation technique (FS). The two main types of noise that were found were salt and pepper noise associated with illumination and spatial noise associated with motion. The spatial noise in the subtracted image was removed using low pass spatial filtering with a kernel size of 3, and the other forms of noise was removed using morphological opening with a structuring element of size 5. The image must then be converted to grayscale as the last stage. This eliminates any prejudice brought on by the user's skin tone or foreground lighting during recognition. They use picture recognition with the help of a convolutional neural network (CNN). In recent research papers about gesture recognition systems using CNN and Deep Learning, there were only a small number of papers that focused on Region of Interest (ROI) segmentation from image, which made the accuracy lower and especially for hand signs and gestures that have nearly more than 3000 different signs and to classify them accurately on the basis of different features that have little variation compared to others, so the image segmentation is a very crucial point that was overlooked in many of the studies. They used the Histogram Back-Projection technique to segment images and create datasets to improve the quality of the training materials. The classes created from these datasets are subsequently put into a CNN.

## III. METHODOLOGY

Raw video footage will act as a input for our software input. Raw video footage is now broken down to frames.

Then these frames will be then sent to Haar_Cascade model to filter out the Region of Interest (ROl), in our case it is hands. Then these ROI are now cropped out of frames and sent to CNN model which will classify these images.

In traditional system a different model is used to extract region of interest, but in our system, we implemented the use of Haar-Cascade algorithm to extract it.

The cascaded image is then sent to CNN model where the given image is classified. Our project deals with the OpenCV python module to record the live footage for from a device camera and it does all processing without any artificial devices for this project to run. This is what we call a video-based project, thus increasing scalability.

a) Actual Image

b) Region of Interest

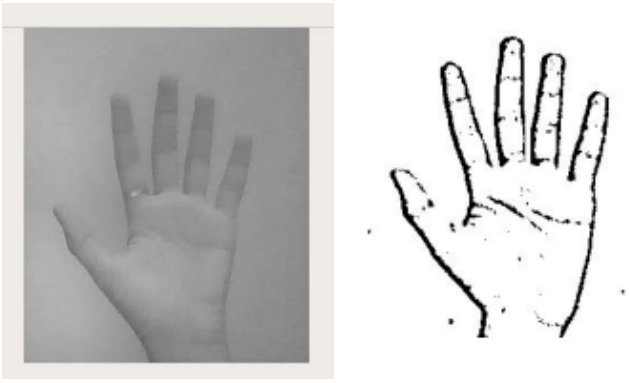
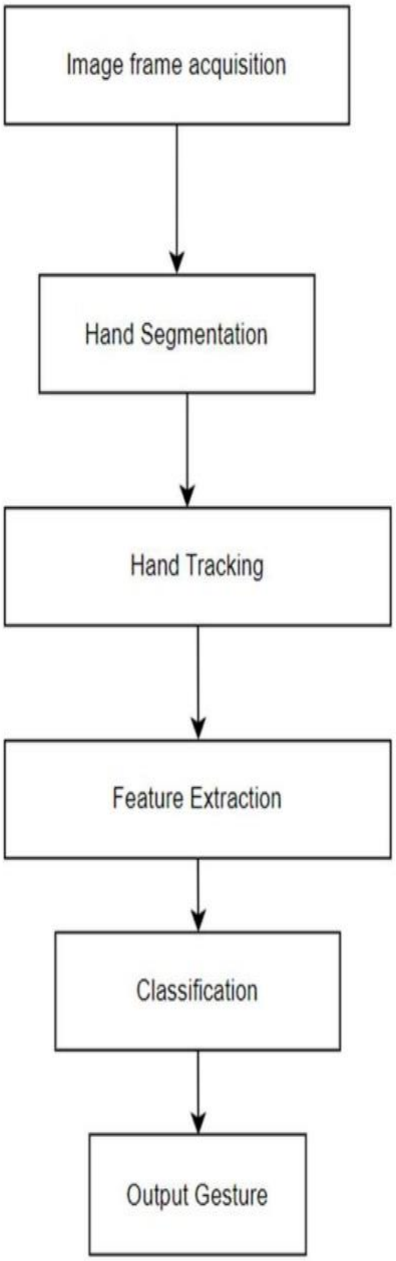


Figure1: This is image pre-processing for easier recognition and feature outlining. a) actual image b)Region of Interest

Image is taken frame by frame from video input and a section of it is cropped out, which will be then passed to next stage. In this stage image is then turned to grey scale thus reducing processing power of the system which in turn boosts latency time of our output. This greyscale image is now then sent to next stage for feature outlining which is then sent to the model. This feature outlining is done by Gaussian blur. The image after applying gaussian blur looks like below.




Flow Diagram

**Dataset**

Dataset is a valuable part of our project. In order to increase accuracy of our model we faced multiple challenges which are as follows:

- Any online dataset will ruin our Haar-cascade algorithm which crops the images which will result in image augmentation during testing phase. Causing inaccuracy.
- Other dataset doesn't have that diversity for our project. Only contained limited number of gestures (eg. Alphabets and digits) which defeats the objective of this software.

So, the solution we got is to create our own dataset for training, which will solve the above problems. We did this by following steps:

- We created 200 images of each each gestures and ran through Haar-cascade then turned into grey scale then applied Gaussian Blur which will become input for our model.
- This procedure will ensure there is not much variations for input data in our model.

**GESTURE CLASSIFICATION AND WORKING ALGORITHM**

This project works on the working of two separate algorithms integrated together to reduce latency for processing and output generation.

1. Algorithm 1 (Pre-processing of raw input)
    a. During a live footage input we can detect Region of Interest i.e. hand in our case using Haar-Cascade model and crop that, which is then sent to next step.
    b. The raw cropped image i.e. is now in RGBw format is now converted into Grey Scale which is very crucial as it reduces dimension of image by 3 folds. Which reduces processing burden on our model.
    c. In this last stage the image is then converted into a form where feature outlining is done by Gaussian Blur method. This image is then sent to model for our Recognition system to work.
2. Algorithm 2 (Recognition and output processing)
    a. Our software loads the pre-trained model for faster execution and reduced latency.
    b. The received Gaussian Blurred image is then fed into the model where the last layer of our model predicts the gesture on the basis of probability.

**Code and Explanation**

- Importing necessary packages for project

```
import cv2
import numpy as np
import mediapipe as mp
import tensorflow as tf
from tensorflow.keras.models import load_model
```

- Initialise Mediapipe class object to import the Haar-Cascade for detecting Region of Interest

```
# initialize mediapipe
mpHands = mp.solutions.hands
hands = mpHands.Hands(max_num_hands=1,
                      min_detection_confidence=0.7)
mpDraw = mp.solutions.drawing_utils
```

- Loading the gesture model along with the last layer nodes label for recognition and initialising the web cam.

```
# Load the gesture recognizer model
model = load_model('mp_hand_gesture')

# Load class names
f = open('gesture.names', 'r')
classNames = f.read().split('\n')
f.close()
print(classNames)

# Initialize the webcam
cap = cv2.VideoCapture(0)
```

- Image pre-processing

```
# Read each frame from the webcam
_, frame = cap.read()
x, y, c = frame.shape

# Flip the frame vertically
frame = cv2.flip(frame, 1)
framergb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

# Get hand landmark prediction
result = hands.process(framergb)

# print(result)
className = ''
```

- Now the image is now detected. And then image transformation

```
if result.multi_hand_landmarks:
    landmarks = []
    for handslms in result.multi_hand_landmarks:
        for lm in handslms.landmark:
            # print(id, lm)
            lmx = int(lm.x * x)
            lmy = int(lm.y * y)
            landmarks.append([lmx, lmy])
```

- Now drawing landmark on the screen along with prediction and displaying the text

```
        prediction = model.predict([landmarks])
        # print(prediction)
        classID = np.argmax(prediction)
        className = classNames[classID]
# show the prediction on the frame
cv2.putText(frame, className, (10, 50),
            cv2.FONT_HERSHEY_SIMPLEX,
              1, (0,0,255), 2, cv2.LINE_AA)
# Show the final output
cv2.imshow("Output", frame)
if cv2.waitKey(1) == ord('q'):
    break
```

## IV. CONCLUSION AND FUTURESCOPE

Raw video footage will act as a input for our software input. Raw video footage is now broken down to frames.
Then these frames will be then sent to Haar_Cascade model to filter out the Region of Interest(ROl), in our case it is hands. Then these ROI are now cropped out of frames and sent to CNN model which will classify these images.

In traditional system a different model is used to extract region of interest, but in our system we implemented the use of Haar-Cascade algorithm to extract it.

The cascaded image is then sent to CNN model where the given image is classified. Our project deals with the OpenCV python module to record the live footage for from a device camera and it does all processing without any artificial devices for this project to run. This is what we call a video based project, thus increasing scalability. This system can also be used in platforms like “YouTube”, “Netflix” etc., videos where there is currently no feature for auto-text generation on the basic of gestures and sign languages.
• Even in video conference we can embed our system of better communication. • Can also be used in places like smart devices to control them via gestures rather than voice (for dumb people) • In other words, this proposed “Hand sign and gesture recognition system” can be used for both public welfare and commercial use.

**Future scope**

We have used Haar-Cascade model to detect Region of Interest but we can replace this method with latest version of YOLO v7 for faster detection of object which then thus reducing the latency of image pre-processing.
Not only that we can also use different feature outlining method other than gaussian blur method, which is faster then it.

REFERENCE


1. Dynamic Hand Gesture Based Sign Word Recognition Using Convolutional Neural Network with Feature Fusion
2. https://ieeexplore.ieee.org/document/8777621

3. A Static Hand Gesture Based Sign Language Recognition System using Convolutional Neural Networks
4. Hand Gesture Feature Extraction Using Deep Convolutional Neural Network for Recognizing American Sign Language

5. https://ieeexplore.ieee.org/document/8858563
6. https://ieeexplore.ieee.org/document/9057853 7. https://paperswithcode.com/paper/fast-and-robustdynamic-hand-gesture
8. https://paperswithcode.com/paper/real-time-handgesture-detection-an
9. https://paperswithcode.com/paper/deep-learning-forhand-gesture-recognition-on