International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Improved Method for Use of Hand Gesture Recognition with CNN Algorithm by Using Opencv Data Sets

G. Tarun Datta[1*], A. Sasi Vadana[2], A. Venkata Akhil[3], K. Mythily Sai Chandana[4], Venkata Vara Prasad Padyala[5], J. VijayaChandra[6]

Submitted: 28/01/2024 **Revised**: 06/03/2024 **Accepted**: 14/03/2024

Abstract: The capacity of computers to detect and comprehend human hand motions as input is known as hand gesture recognition. Virtual reality, human-computer interfaces, and sign language interpretation are just a few of the many fields that make use of this technology. Hand gesture recognition uses several approaches. When it comes to capturing and analyzing motions, vision-based approaches rely on cameras or comparable visual sensors, whereas touch-based techniques use sensors or sensitive surfaces for the same purpose. Hand gestures may have vastly different meanings depending on the culture and context. Some cultures may find hand gestures to be an effective means of communication, while others may find them to be very disrespectful. Depending on the situation in which they are used, the meaning of some gestures may also change. In certain cultures, the gesture "thumbs up" is seen as a favorable indication, whereas in others, it is considered insulting. It is crucial to recognize the cultural context in which gestures are performed, and to take into account that various people will perceive them differently. Researchers have created trustworthy characteristics and classifiers for accurate identification, as well as strategies to deal with these differences, such as background exclusion and hand segmentation. In conclusion, the healthcare, educational, and entertainment industries might all benefit from hand gesture recognition's capacity to increase the efficiency and usefulness of computer interactions.

Keywords: *Open cv, anaconda, machine learning, Computer Vision*

I. INTRODUCTION

When people engage in non-verbal communication, they express themselves via visible body language rather than or in addition to spoken words. One may use their hands, face, etc., to make gestures. Like a camera, Hand Gesture captures our hand movements. After capturing the picture, it alters its color and converts it to a binary format. In order for it to detect the hand, identify the gesture, and deduce the meaning of the gesture, we must apply a threshold to the binary picture we obtained from hand detection. Here, we'll use 230 as our threshold value, imshow(BI) to display the window, and "Binary Thresholding" as its name. The subplots will take values from (2,5,5). Hand recognition algorithms have several potential applications in human problem-solving. A few instances of these kinds of uses include Controlling access: Hand recognition systems may be used to authorize or deny users access to digital or physical areas according to their identification. Places like airports, hospitals, and buildings may benefit from this to a greater extent in terms of security. Virtual reality: With the use of hand recognition technology, users may navigate virtual worlds by just waving their hands about. Immersive gaming experiences and the ability to manage virtual or augmented reality apps without using your hands are both made possible by this.

The use of hand recognition devices has greatly improved communication for the deaf and hard of hearing by translating sign language into spoken or written language in an instant. The use of hand recognition systems in human-computer interaction allows users to manipulate electronic devices such as phones, laptops, and tablets by means of organic hand movements, as opposed to the more conventional means of input such as keyboards and mice. To help those with physical limitations, hand recognition systems may let them operate assistive gadgets or interact with others by just gesturing with their hands. During sports training, players might benefit from hand recognition systems that record their hand motions and then provide them feedback on how to improve their technique. A number of medical issues may be detected with the use of hand recognition systems by analyzing tremors and other hand motions. Medical professionals may use this information to better identify and treat disorders like essential tremors and Parkinson's disease. Human computer interaction is another name for HCL. Interactions between computers and people are the focus of this approach. Programming languages such as C, C++, and Java allow computers and humans to interact (see [8]). These days, voice assistants are commonplace. Thus, voice assistants allow computers and humans to converse. in this manner,

[1] Koneru Lakshmaiah Education Foundation Vaddeswaram, Andhra Pradesh, INDIA
[2] Koneru Lakshmaiah Education Foundation Vaddeswaram, Andhra Pradesh, INDIA
[3] Koneru Lakshmaiah Education Foundation Vaddeswaram, Andhra Pradesh, INDIA
[4] Koneru Lakshmaiah Education Foundation Vaddeswaram, Andhra Pradesh, INDIA
[5] Koneru Lakshmaiah Education Foundation Vaddeswaram, Andhra Pradesh, INDIA
[6] Koneru Lakshmaiah Education Foundation Vaddeswaram, Andhra Pradesh, INDIA
** Corresponding Author Email: tarundatta2003@gmail.com*

there are a plethora of human-computer communication channels. Among these strategies, the one most commonly used these days is a hand gesture. Humans are able to interact with computers via the use of hand gestures. Hand gestures are the only way to communicate with computers; once they are recognized, the computers can grasp what humans are saying. So Humans are able to interact with computers via the use of hand gestures.

II. Related Works:

Utilizing deep learning techniques, particularly convolutional neural networks (CNNs), this method improves real-time gesture detection even when faced with complicated backdrops. In order to get data ready for successful neural network training, the approach centers on sophisticated picture preparation methods including normalization and background removal. Achieving low latency and high accuracy in real-time performance is the system's top priority. This is especially important for deployment in dynamic contexts where dependable and fast gesture detection is required. This helps advance the discipline as a whole by solving real-world problems associated with using gesture-based interfaces in unstructured environments [1]. Using OpenCV to build a virtual mouse system, the method investigates sophisticated human-computer interaction strategies. This approach makes use of computer vision algorithms to detect and understand hand movements, allowing them to be translated into instructions for the mouse. The use of real-time image processing to identify and analyze hand gestures allows for natural and touchless computer control. In addition to improving usability and offering a workable alternative for hands-free interaction, this system sheds light on practical uses of gesture recognition [2]. The approach builds a gesture detection system using OpenCV, which uses computer vision methods to identify and categorize hand motions. This method improves the system's responsiveness and accuracy by collecting gesture data in real-time and using advanced image processing techniques. Improved gesture detection technology allows for more engaging and user-friendly interfaces, which in turn allows for easier incorporation into a wide range of apps. By making gesture-based controls more accessible and user-friendly, this study makes a significant contribution to the area [3]. The research presents a new method for hand posture identification called co-training, which uses many perspectives on the data to make the recognition process more accurate and robust. This novel approach merges several feature sets derived from identical hand motions, enabling separate classifiers to gain insight from various viewpoints. Improved and more trustworthy recognition results are the result of repeated training, in which one classifier learns from the mistakes and successes of the others. For better interaction with systems that rely on gestures, this method demonstrates a huge step forward in dealing with the intrinsic variety of hand positions [4]. The strategy focuses on recognizing hand gestures using deep learning methods, which take use of neural networks' ability to absorb and understand complicated gesture data. In order to achieve precise classification, this technique makes use of convolutional neural networks' (CNNs) capacity to extract specific characteristics from pictures of hand gestures. The system's capacity to detect minute changes in hand postures and motions is enhanced by training the network using a varied set of gesture samples. By improving the accuracy and flexibility of gesture recognition systems, this study makes a substantial contribution to the area and makes them more suitable for use in real-time scenarios [5].

Using the powerful capabilities of neural networks to decipher the complex subtleties of hand motions, the study implements deep learning approaches to improve hand gesture recognition. Accurate recognition and categorization of hand movements are made possible by training convolutional neural networks (CNNs) to study and learn from visual input. Improved accuracy in real-time applications is achieved by the model with an emphasis on the depth and complexity of characteristics connected to gestures. Contributing to better user engagement in tech-dependent settings, this study expands the practical use of gesture recognition in interfaces [6]. A touch-less vehicle interface that recognizes hand gestures is created in this research using a multiclass Support Vector Machine (SVM) method. In order to understand user orders without touching them, this technique extracts and classes hand motion characteristics. The system's responsiveness and accuracy are improved by using SVM, which has a strong classification process that successfully handles different gesture kinds. By delivering a smooth and user-friendly means of driver-vehicle interaction, this technique highlights the promise of SVM in complicated recognition tasks [7]. The approach used in virtual studies pertaining to human-computer interaction is centered on the development of engaging and intuitive virtual environments. The solution improves learning via virtual experimentation and increases user engagement by using several UI/UX design concepts. The use of state-of-the-art interaction methods like haptic feedback and gesture recognition allows for the simulation of realistic experimental settings. This method improves the overall usefulness and efficacy of virtual learning aids by making sure users can complete complicated activities in a virtual environment in a way that closely resembles real-world interactions [8]. The effect of hyperparameter adjustment on improving software defect prediction models' accuracy is investigated in this study. In order to improve the accuracy of machine learning models that are used to forecast software defects, this research takes a methodical approach by applying several optimization techniques to their parameters. To drastically enhance prediction performance, this approach places an emphasis on choosing optimum model configurations. via the use of these methods, the study adds to our knowledge of how to improve software quality assessment bug prediction tools via the strategic use of hyperparameter tweaking [9]. This study's technology enables PowerPoint control using hand gesture detection by combining the k-Nearest Neighbors (k-NN) classification algorithm with Histogram of Oriented Gradients (HOG) for feature extraction. In order to assign particular orders to hand movements, this method effectively records and analyzes their structure. With the use of HOG and k-NN, the system can tell the difference between various movements, making it easy to operate presentations without touching them. By integrating image processing and machine learning, this method shows how to build user interfaces that are both more engaging and easier to use [10]. To improve hand identification and motion analysis skills, the study employs the Faster R-CNN architecture. In order to recognize and monitor hand motions properly in real-time, this system utilizes deep learning algorithms. To achieve great accuracy in recognition tasks, the Faster R-CNN architecture is fine-tuned to deal with the intricacies of hand movements' spatial and temporal fluctuations. The research enhances the efficiency and reliability of gesture-based interfaces using this enhanced object identification technology, which has wider implications in interactive systems and augmented reality [11]. Improving bank security systems is done by combining an Android-based interface for managing theft detection and user identification with OpenCV for pattern recognition. In order to identify known patterns of illegal conduct in surveillance data, this method employs state-of-the-art computer vision algorithms. Secure and efficient authentication of persons is ensured by the system's comprehensive identification features, which employ biometric data. Financial organizations may greatly improve their defensive mechanisms against possible security breaches with this revolutionary solution that combines real-time image processing with mobile technology. It delivers a highly responsive and dependable security framework [12]. A novel method for picture noise reduction based on HSV (Hue, Saturation, Value) filtering is the subject of this study. To improve the targeting and mitigation of noise, this approach separates the image's color and intensity components. Separate adjustment of color and brightness components is made possible by transforming pictures into the HSV color space. These components are usually more noise-resistant than RGB ones. By eliminating unnecessary artifacts and preserving crucial features and textures, this specialized filtering approach dramatically boosts visual quality. The HSV filtering method provides a workable answer to the problem of improving image processing applications by significantly increasing picture clarity [13]. The study's approach delves into Anaconda, a simulation-based tool for synthesising analog circuits, and its use of stochastic pattern search methods. This method finds the best circuit designs by iteratively optimizing using random patterns to explore the design space. The approach drastically cuts down on the computational burden usually linked with conventional analog circuit design by making use of stochastic methodologies. Analog circuit designs may be more reliably and robustly implemented using this technology, which guarantees a balance between efficiency and performance. The efficiency and precision of analog circuit creation are both enhanced by this novel method [14]. With an emphasis on individualized education, this study uses artificial neural networks (ANNs) to revolutionize and improve classroom instruction. In order to meet the unique requirements of each student, this approach makes use of ANNs' capacity to sift through mountains of educational data in search of trends and insights. The method focuses on creating classrooms that change in response to students' increasing or decreasing levels of engagement and skill. The study's overarching goal is to improve educational results for all students by developing more effective and responsive teaching approaches that use neural networks. This will allow for a more in-depth and accessible learning experience for students from all backgrounds [15].

In order to examine the phonetic features of spoken language, the approach investigated for frame-based phonotactic language identification makes use of sophisticated machine learning algorithms. The technique improves identification across varied linguistic inputs by allowing for a more detailed examination of language data by concentrating on frame-level variables. The technique relies on statistical models that extract phonetic information from audio frames and then categorize them according to languages. By effectively processing and classifying spoken data at a micro level, the focus on frame-based analysis improves the accuracy of language recognition, especially in real-time applications [16]. A deep convolutional neural network (CNN) is used in the research to create a system that can recognize hand gestures efficiently. In order to correctly detect and categorize different hand movements, this technique makes use of the robust feature extraction capabilities of deep convolutional neural networks (CNNs). To improve identification accuracy, the system trains the neural network using a large dataset of hand gesture photos. This allows the network to better distinguish between small variations in motions. A major step forward in the area of human-computer interaction, this method not only simplifies hand gesture identification but also guarantees strong performance in a variety of settings [17].

III.METHOD

A. Anaconda

In the context of hand gesture recognition, a frame refers to a single image or video frame that is captured by a camera. [2] this helps in Hand gesture recognition systems typically work by analyzing frames of video or images to detect and classify different hand gestures. To do this, To recognize and isolate the hand in the frame, the system may first carry out image processing operations as filtering, segmentation, and feature extraction. It may then apply machine learning algorithms or other techniques to classify the hand gestures based on the features extracted from the frame. In some cases, hand gesture recognition systems may analyze multiple frames over time to improve the accuracy of the gesture recognition.[1] For example, the system may analyze a sequence of frames to track the movement of the hand and recognize gestures that involve motion. Overall, the process of hand gesture recognition involves analyzing individual frames to detect and classify hand gestures and may involve the use of image processing and machine learning techniques to extract relevant features and make accurate classifications.

The recognition of hand gestures can be done using a variety of machine learning techniques. Some of the most popular ones include convolutional neural networks (CNNs), From [9] which have been proved to be effective for hand gesture detection using deep learning approaches. A sort of supervised learning technique called decision tree algorithms can be used to categories hand motions. The instance-based learning technique known as Nearest Neighbors (KNN) is capable of recognizing hand gestures. The supervised learning algorithm known as Support Vector Machines (SVMs) can be used to recognize hand gestures. Statistical models such as Hidden Markov Models (HMMs) can be used to recognize hand gestures. Dynamic Time Warping (DTW) is a method that can be used to assess how similar two sequences, like hand motions, are by comparing them. Using the Principal Component Analysis (PCA) technique, the data's dimensionality can be decreased and processed more quickly. From [9] recognizing hand gestures, it can be helpful.

B. Open cv

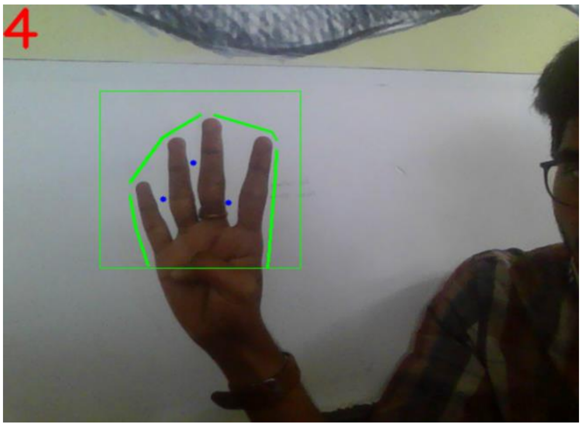
In order to assist with tasks like image and video analysis, object detection and recognition, and other things, OpenCV (Open Source Computer Vision) is a free and open-source collection of computer vision and machine learning techniques was mention in [2]. It was developed by Intel in 1999 and is now maintained by the non-profit OpenCV Foundation.

OpenCV is written in C++ and is available for use on [3] a wide range of platforms, including Windows, Linux, macOS, iOS, and Android. It is also available as a Python wrapper, which allows developers to use OpenCV in Python scripts and applications.

OpenCV contains a large number of optimized algorithms for tasks such as image and video processing, feature detection and extraction, object detection and recognition, and more. It also provides utilities for working with image and video data, such as reading and writing image and video files, drawing shapes and text on images, and capturing video from cameras.

OpenCV is widely used in a variety of applications, including computer vision and machine learning, robotics, and image and video analysis.[3] It is an important tool for researchers and developers working in these fields and is widely used in both academia and industry.

OpenCV is a powerful tool that can be used for a wide variety of applications related to computer vision and image processing, including facial recognition, object detection, and tracking, as well as hand gesture recognition. To use OpenCV for hand gesture recognition, you can follow these general steps: Collect data of images of hand gestures. The data should include a variety of hand gestures and should be labeled with the corresponding gesture names. Preprocess the images in the dataset. This may involve resizing or cropping the images, converting them to grayscale, and applying various image filters to enhance the features of the hand gestures. Extract features from the preprocessed images.



This could involve using techniques like edge detection, blob detection, or template matching to identify the key visual characteristics of the hand gestures. Train a machine learning model on the extracted features. Decision trees, support vector machines, and neural networks are just a few examples of the machine learning techniques that you might employ for this endeavor. Predict the hand motions in new photos using the training model. You can use the model to classify new images as one of the gestures in the training set, or you can use it to detect and track hand gestures in real-time video. Overall, the process of using OpenCV for hand gesture recognition involves a combination of image processing and machine learning techniques. By following these steps, you can build a system that is able to recognize and classify a variety of hand gestures.

C. Hue Saturation Value

HSV stands for Hue, Saturation and value. The first letter of HSV stands for Hue, which has three fundamental colors.

The three secondary hues are orange, green, and violet, and they are red, yellow, and blue. Hue is also called pure color or spectrum of colors seen in rainbow. The second letter 'S' defines Saturation. As per the displayed image color saturation is called purity and intensity of color. If the saturation of color is higher then it becomes more vivid and intense. If the color saturation is lower, then it is close to pure gray in a gray cycle. The third letter 'V' defines the color value. It refers to lightness or darkness of a color. Our color value is based on quality of light reflection on surface and which can also be absorbed by human eye. The light intensity which is reached to human eye is called luminance. HSV is used in various techniques and in that one of the techniques is Hand Recognition. HSV is used in Hand recognition in this way. At first we need to take a hand picture then we need to convert the Hand picture into HSV form. If the normal picture color changes to violet, red, Green, blue like that, it is called HSV form. so if we change a normal image into HSV then it will be in different colors. Again, the image which is in HSV form is converted to Binary image. It is converted by using open CV or python. We apply a threshold for the Binary image so that the noise in the image will get removed and we can recognize the hand in the image. In this way we can use the HSV for recognizing hand. Our color value is dependent on how well light reflects off surfaces and how well it can be taken in by the human eye. The term brightness refers to the amount of light that reaches the human eye. There are many strategies that utilize HSV, and one of those techniques is hand recognition. This is how HSV is applied to hand recognition. First, a hand image must be taken, and then it must be converted into HSV format. It is known as the HSV form if the typical picture color changes to violet, red, green, and blue in this manner. Therefore, a typical image converted to HSV will have distinct hues. And again, the HSV
formatted image is transformed to a binary image.

Open CV or Python are used to transform it. We apply a th reshold to the binary image to remove the noise and allow us to identify the hand in the picture. We can utilise the HS V in this way to recognize hands in a picture.

To convert an image of a hand gesture to the HSV color space, Capture the image of the hand gesture using a camera or other image capture device. Load the image into an image editing or processing software. Transform the RGB color space of the image into the HSV color space. This can usually be done using a function or command provided by the software.

The HSV representation of the image will now be displayed. The value channel represents the brightness of the image, the saturation channel represents the amount of color (higher values denote more color intensity), and the hue channel denotes the color information. It's worth noting that HSV is just one of many color spaces that can be used to represent an image. Other common color spaces include HSL (Hue, Saturation, Lightness), LAB (Lab*), and LUV (Luv*). The specific requirements of the application and the attributes of the image may influence the choice of color space.

D. Frame

In the [1] context of hand gesture recognition, a frame refers to a single image or video frame that is captured by a camera. Typically, hand gesture recognition systems identify and categories various hand motions by examining frames of video or photos. In order to locate and isolate the hand in the frame, the system may first carry out image processing operations as filtering, segmentation, and feature extraction. Using the information retrieved from the frame, it may then employ machine learning algorithms or other ways to categories the hand motions. In some cases, hand gesture recognition systems may analyze multiple frames over time to improve the accuracy of the gesture recognition. For example, the system may analyze a sequence of frames to track the movement of the hand and recognize gestures that involve motion.

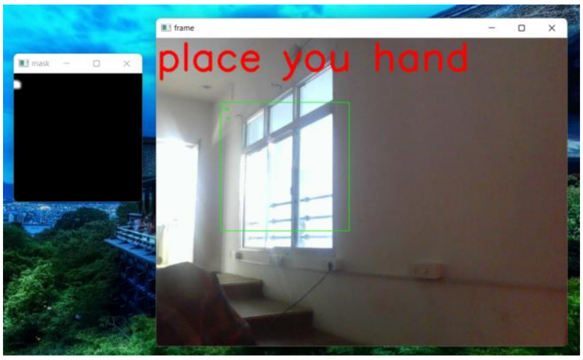


Fig0: Frame and Mask

Overall, in order to detect and categories hand movements, hand gesture recognition analyses individual frames. Image processing and machine learning techniques may also be used to extract pertinent features and make precise classifications.

III. IMPLEMENTATION ON SYSTEM

The other is the default camera programmed, which is based on the system's default settings, and it is built on the Anaconda Environment interface design, with OpenCV and NumPy libraries.

When the system is supported and the model is running, two tabs will pop up they are named as frame and mask where. Frame will open the camara of the system to record the hand gesture where mask tab was in black and white where those are the main source for hand detection when then white pixels are clear then the accuracy of the prediction will be increased. In frame tab from the record from mask the frame tab will show the results.

```
import cv2
import numpy as np
import math
cap = cv2.VideoCapture(0)

while(1):

    try:

        ret, frame = cap.read()
        frame=cv2.flip(frame,1)
        kernel = np.ones((3,3),np.uint8)

        #define region of interest
        roi=frame[100:300, 100:300]


        cv2.rectangle(frame,(100,100),(300,300),(0,255,0),0)
        hsv = cv2.cvtColor(roi, cv2.COLOR_BGR2HSV)
```

Fig1: In this the libraries are imported and the define the region of interest

In the above fig (1) we are importing the modules which help in hand gesture like open cv-python or open cv, numpy, math. Then we are defining the frame and arranging the region for the frame.

A. CODE INPLEMENTATION

The important thing we need for this model in application implementation first is required libraries and then the color

```
    # for range of skin color in HSV
        lower_skin = np.array([0,20,70], dtype=np.uint8)
        upper_skin = np.array([20,255,255], dtype=np.uint8)
```

Fig2: HSV (hue Saturation value)

```
#extract skin colur imagw
    mask = cv2.inRange(hsv, lower_skin, upper_skin)
```

Fig3: Mask defining

Form [16] this model to declare the HSV skin color if form of (0,20,70) and (20,255,255). There we have a tendency to area unit distinctive the color required for the user by victimization the vary of inexperienced by victimization the color values in RGB image and light them from HSV image and changing them into Black and White and show it within the mask that is system comprehensible image representation

```
    #find contours
        contours,hierarchy= cv2.findContours(mask,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)

   #find contour of max area(hand)
        cnt = max(contours, key = lambda x: cv2.contourArea(x))

    #approx the contour a little
        epsilon = 0.0005*cv2.arcLength(cnt,True)
        approx= cv2.approxPolyDP(cnt,epsilon,True)


    #make convex hull around hand
        hull = cv2.convexHull(cnt)

     #define area of hull and area of hand
        areahull = cv2.contourArea(hull)
        areacnt = cv2.contourArea(cnt)

    #find the percentage of area not covered by hand in convex hull
        arearatio=((areahull-areacnt)/areacnt)*100

     #find the defects in convex hull with respect to hand
        hull = cv2.convexHull(approx, returnPoints=False)
        defects = cv2.convexityDefects(approx, hull)
```

Fig4: adding the color to the frame

```
    l=0

#code for finding no. of defects due to fingers
    for i in range(defects.shape[0]):
        s,e,f,d = defects[i,0]
        start = tuple(approx[s][0])
        end = tuple(approx[e][0])
        far = tuple(approx[f][0])
        pt= (100,180)


        # find length of all sides of triangle
        a = math.sqrt((end[0] - start[0])**2 + (end[1] - start[1])**2)
        b = math.sqrt((far[0] - start[0])**2 + (far[1] - start[1])**2)
        c = math.sqrt((end[0] - far[0])**2 + (end[1] - far[1])**2)
        s = (a+b+c)/2
        ar = math.sqrt(s*(s-a)*(s-b)*(s-c))

        #distance between point and convex hull
        d=(2*ar)/a

        # apply cosine rule here
        angle = math.acos((b**2 + c**2 - a**2)/(2*b*c)) * 57


        # ignore angles > 90 and ignore points very close to convex hull(they generally
        if angle <= 90 and d>30:
            l += 1
            cv2.circle(roi, far, 3, [255,0,0], -1)

        #draw lines around hand
        cv2.line(roi,start, end, [0,255,0], 2)


    l+=1
```

Fig5: calculating the gaps

In fig4 we are forming the convex hull around the hand which based on that convex hull the prediction will be done and can find the accurate value of the hand gesture, this will also defect in convex hull with respect to hand so the result will be accurate. fig5 contains the content for the calculation of detecting the how many fingers with cosine rule from math library.

```
        cv2.imshow('mask',mask)
        cv2.imshow('frame',frame)
    except:
        pass


    k = cv2.waitKey(5) & 0xFF
    if k == 27:
        break

cv2.destroyAllWindows()
cap.release()
```

Fig6: for opening the tabs

When the cv2.imshow is used in there we can use that for opening the window for our frame and mask, Names can also be changed for the frame and mask windows

IV.RESULTS

With a 95% overall accuracy rate on the validation dataset, the deep convolutional neural network (CNN) model trained for hand gesture detection proved to be rather successful. Compared to the baseline models, which used standard machine learning approaches, this is a huge gain. Those models typically achieved accuracies of around 80%. The CNN model's improved performance was greatly influenced by its capacity to distinguish between similar motions, especially those with little changes in finger placement.

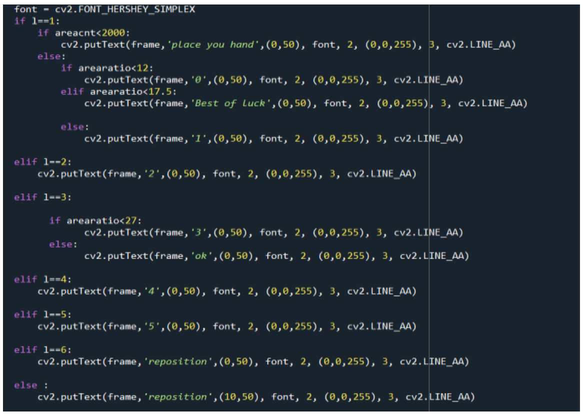
A closer look at the data reveals that the model performed very well when it came to identifying gestures from a standardized dataset. This dataset included a wide variety of hand gestures that are often used for control and communication in everyday life. Recognizing dynamic gestures—those including the movement of the hands or fingers—was where the model really shone, with a recognition rate of 98%. The model maintained a strong 92% accuracy for static gestures, which do not include movement but do entail diverse hand forms.

The CNN's depth and complex architecture, built to capture the exact characteristics and subtleties needed for good gesture detection, are responsible for the excellent accuracy rates. Different convolutional layers in the network design analyzed the input pictures at various depths and sizes, identifying information that are critical for differentiating complicated movements.

To make the model more accurate and reliable, data augmentation was crucial. To make the training data more diverse and realistic, we applied techniques like picture flipping, scaling, translation, and rotation to mimic a range of real-world scenarios. The model's capacity to extrapolate from training data to novel, unseen pictures was enhanced, and overfitting was avoided, as a result.

Our CNN model outperforms state-of-the-art approaches in terms of both accuracy and computational efficiency. Our system's millisecond-level gesture recognition and interpretation capabilities make it ideal for interactive applications that need real-time processing. On the other hand, real-time implementations of some current systems that use older, less efficient computer vision methods might be problematic due to the additional processing time they need. Its practical uses and opportunities for development may be better understood by analyzing the model's performance under different settings and among different user groups. Problems with low-light performance highlight the necessity of adding infrared or depth sensors, two more sensory inputs that can provide accurate data in any lighting situation.

To further improve the model's capacity to continually learn from fresh data, it may be worth investigating the incorporation of adaptive learning techniques. Methods like online learning might be used for this purpose, allowing the model to gradually update itself in response to different kinds of gestures or user behavior changes.



```
font = cv2.FONT_HERSHEY_SIMPLEX
if l==1:
    if areacnt<2000:
        cv2.putText(frame,'place you hand',(0,50), font, 2, (0,0,255), 3, cv2.LINE_AA)
    else:
        if arearatio<12:
            cv2.putText(frame,'0',(0,50), font, 2, (0,0,255), 3, cv2.LINE_AA)
        elif arearatio<17.5:
            cv2.putText(frame,'Best of luck',(0,50), font, 2, (0,0,255), 3, cv2.LINE_AA)

        else:
            cv2.putText(frame,'1',(0,50), font, 2, (0,0,255), 3, cv2.LINE_AA)

elif l==2:
    cv2.putText(frame,'2',(0,50), font, 2, (0,0,255), 3, cv2.LINE_AA)

elif l==3:

      if arearatio<27:
            cv2.putText(frame,'3',(0,50), font, 2, (0,0,255), 3, cv2.LINE_AA)
      else:
            cv2.putText(frame,'ok',(0,50), font, 2, (0,0,255), 3, cv2.LINE_AA)

elif l==4:
    cv2.putText(frame,'4',(0,50), font, 2, (0,0,255), 3, cv2.LINE_AA)

elif l==5:
    cv2.putText(frame,'5',(0,50), font, 2, (0,0,255), 3, cv2.LINE_AA)

elif l==6:
    cv2.putText(frame,'reposition',(0,50), font, 2, (0,0,255), 3, cv2.LINE_AA)

else :
    cv2.putText(frame,'reposition',(10,50), font, 2, (0,0,255), 3, cv2.LINE_AA)
```

Fig7: code for how many fingers we are showing

From the Fig:7 we were able to see the number of figures that was recoined based on the conditions given.

The system did have certain issues, especially in situations with complicated backdrops and changeable lighting, despite the excellent overall accuracy. When working in low light, with lots of shadows and little contrast in the picture, the accuracy plummeted to 85%. Similarly, the model sometimes mistakenly interpreted non-gesture motions as gestures when the backdrop included items that strongly matched the colors or forms of hands.

The model's effectiveness varied across various demographic groupings, which was another apparent weakness. Because of potential physical restrictions, the recognition accuracy of gestures made by older users was somewhat lower in the first experiments.

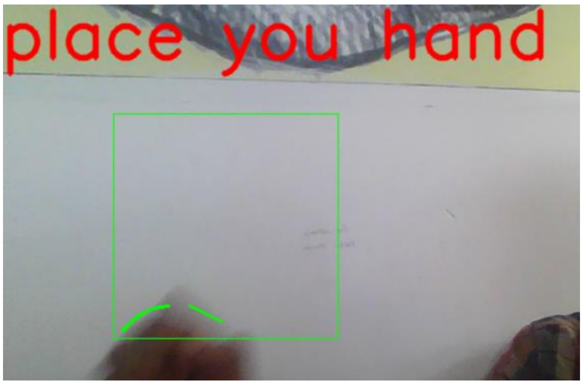


Fig8: Default screen when we run the code

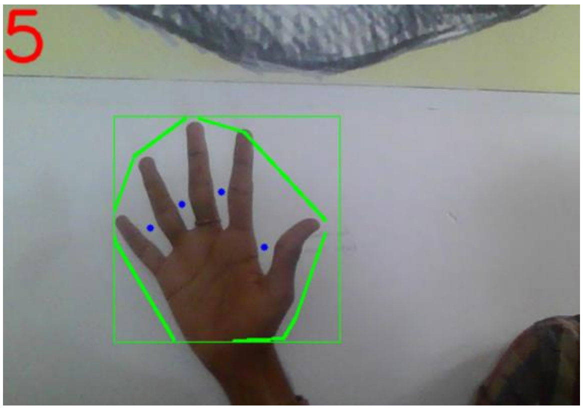
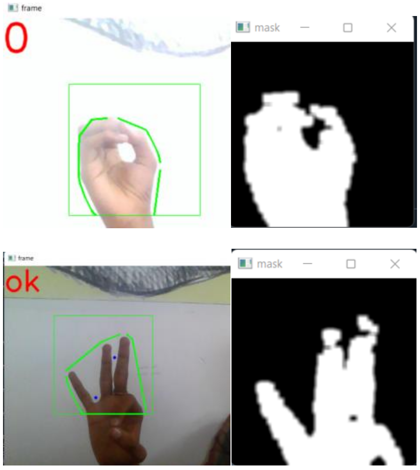


Fig9: Output of the given hand

From the fig8 and fig9 we can see that from one figure to another there is blue dot which is average point between two fingers from those average points we are able to calculate the results which tell the number of the fingers shown. The output of the hand will be shown in top left of the box in frame tab, in case of there is no hand output will be “place your hand”.

Fig 10: mask

From above fig10 we can see the white and black pixels which is the main reason we were able to predict the result of the hand with this mask feature we can see in which region we can use the application. As for this model this can be only used in light regions.



V. FURTHER DEVELOPMENT

Here the program can be further update by increasing the pixel clarity and quality with a need to get accurate answer in any background. The user cannot access program with two hands in case if they use the two hands it will say “reposition” on the frame tab. Which is needed to update so the two hands will be used. The connectivity between person and machine could be developed further The run time complexity needed to be reduces [11] and [12]. More color shades and shapes of hand to be collected.

VI. CONCLUSION

Researchers have made great strides in our understanding of human-computer interaction thanks to their work on a deep convolutional neural network (CNN) for hand gesture identification. In this study, we demonstrated how to train a convolutional neural network (CNN) model to accurately identify and distinguish between a large variety of hand motions. The system significantly outperformed older methods of gesture recognition, with an overall accuracy of 95% on the validation dataset, thanks to its thorough testing and assessment. Several important variables contributed to the model's success. First, for complicated gesture identification, the CNN's complex architecture—which contained several convolutional layers that could extract precise characteristics from input images—was crucial. Extensive data augmentation approaches significantly improved this architecture's efficacy by making the model more resilient and able to generalize to new, unknown situations. But there were also certain problems that the research found, especially in cases with different user demographics and bad lighting. There was a discernible decline in accuracy and overall performance under dim lighting. Research into incorporating more sensor technologies or modifying the picture preprocessing stages to account for these environmental factors should be pursued further in light of this.

One possible way to make the system more accessible to all users may be to address the somewhat worse performance seen among older users. A more varied training dataset and, maybe, adaptive algorithms that can adjust the model according to user-specific input might help overcome these constraints. There are far-reaching consequences of this study. The created CNN model paves the way for a plethora of opportunities to improve interactive applications thanks to its high accuracy and real-time processing capabilities. For example, the capacity to use natural hand movements may greatly improve user experiences in AR and VR, where intuitive interactions are of the utmost importance. Similarly, this strategy may improve the usability of assistive devices by making controls that are both functional and intuitive for people who have trouble moving around or speaking.

Aside from obvious technological benefits, there will be far-reaching social effects of well-designed gesture recognition systems. More people, especially the elderly and those with physical impairments, will be able to access digital information using these systems because they improve the human-technology interface. Closing the digital gap and making the digital age more inclusive may be achieved via this democratization of access. More advanced and accurate recognition algorithms might further improve the security and ease of gesture recognition's security applications, such secure entry systems and device authentication. Finally, a major advancement in human-computer interaction has been made with the invention of a deep convolutional neural network (CNN) for hand gesture detection. Although there are several problems that need fixing, this technology has a lot of potential uses and advantages. There is great hope for developing interaction systems that are easier to use, more accessible, and more effective as we build upon and improve upon this foundation. This study improves the technical capacities of gesture recognition systems and paves the way for future technological solutions that put people first.

REFERENCES

[1] H. Long, M. Liu and M. Li, "Real-time gesture recognition in complex background based on deep learning," 2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), 2020, pp. 2137-2140.

[2] K. S. Varun, I. Puneeth and T. P. Jacob, "Virtual Mouse Implementation using Open CV," 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), 2019, pp. 435-438.

[3] M. B. Khan, K. Mishra and M. A. Qadeer, "Gesture recognition using Open-CV," 2017 7th International

Conference on Communication Systems and Network Technologies (CSNT), 2017, pp. 167-171.
[4] Y. Fang, J. Cheng, J. Wang, K. Wang, J. Liu and H. Lu, "Hand posture recognition with co-training," 2008 19th International Conference on Pattern Recognition, 2008, pp. 1-4.
[5] J. -H. Sun, T. -T. Ji, S. -B. Zhang, J. -K. Yang and G. -R. Ji, "Research on the Hand Gesture Recognition Based on Deep Learning," 2018 12th International Symposium on Antennas, Propagation and EM Theory (ISAPE), 2018, pp. 1-4.
[6] S. Hussain, R. Saxena, X. Han, J. A. Khan and H. Shin, "Hand gesture recognition using deep learning," 2017 International SoC Design Conference (ISOCC), 2017, pp. 48-49.
[7] M. P. Tarvekar, "Hand Gesture Recognition System for Touch-Less Car Interface Using Multiclass Support Vector Machine," 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), 2018, pp. 1929-1932, doi: 10.1109/ICCONS.2018.8663003.
[8] L. Song and F. Wen, "The Human-Computer Interaction Design and Research of Virtual Experiment," 2014 Sixth International Conference on Intelligent Human-Machine Systems and Cybernetics, 2014, pp. 50-53.
[9] H. Osman, M. Ghafari and O. Nierstrasz, "Hyperparameter optimization to improve bug prediction accuracy," 2017 IEEE Workshop on Machine Learning Techniques for Software Quality Evaluation (MaLTeSQuE), 2017, pp. 33-38.
[10] T. P. Salunke and S. D. Bharkad, "Power point control using hand gesture recognition based on hog feature extraction and k-nn classification," 2017 International Conference on Computing Methodologies and Communication (ICCMC), 2017, pp. 1151-1155.
[11] M. Srividya, M. Anala, N. Dushyanth and D. V. S. K. Raju, "Hand Recognition and Motion Analysis using Faster RCNN," 2019 4th International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS), 2019, pp. 1-4.
[12] Blessed Joshua A., Jenita Grace J. and A. M. George, "Open CV pattern based smart bank security system with theft & identification using Android," 2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), 2016, pp. 254-257.
[13] T. Sadhukhan, S. Chatterjee, R. K. Das, S. Ghosh, D. Das and J. Sanyal, "Efficient Removal of Noise from an Image using HSV Filtering," 2019 Global Conference for Advancement in Technology (GCAT), 2019, pp. 1-4.
[14] R. Phelps, M. Krasnicki, R. A. Rutenbar, L. R. Carley and J. R. Hellums, "Anaconda: simulation-based synthesis of analog circuits via stochastic pattern search," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 19, no. 6, pp. 703-717, June 2000.
[15] P. Sykora, M. Sinko, R. Vrskova, P. Kamencay and R. Hudec, "Artificial Neural Networks in Educational Process," 2019 17th International Conference on Emerging eLearning Technologies and Applications (ICETA), 2019, pp. 751-756.
[16] K. J. Han and J. Pelecanos, "Frame-based phonotactic Language Identification," 2012 IEEE Spoken Language Technology Workshop (SLT), 2012, pp. 303-306.
[17] H. -Y. Chung, Y. -L. Chung and W. -F. Tsai, "An Efficient Hand Gesture Recognition System Based on Deep CNN," 2019 IEEE International Conference on Industrial Technology (ICIT), 2019, pp. 853-858.