E-ISSN 2224-2333
P-ISSN 2222-9930

Pakistan Journal of Engineering Technology and Science (PJETS)
Volume 12. Issue.01, PP. 37-52, December 2024
https://doi.org/10.22555/pjets.v12i2.1086

A Node Point Approach for Real-Time Hand Gesture Recognition Using Support Vector Machines (SVM)

Nida Khalil [1*], Maheen Danish [2], and Kanza Zafar [3]

[1]Faculty of Software Engineering, Sir Syed University of Engineering & Technology (SSUET), Karachi 75300, Pakistan

[2]Faculty of Computer Engineering, Sir Syed University of Engineering & Technology (SSUET), Karachi 75300, Pakistan

[3]Riphah School of Computing & Innovation, Riphah International University, Lahore, Pakistan

**Corresponding Author: nkhalil@ssuet.edu.pk*

Abstract:

Gesture recognition systems have become increasingly popular in recent years due to their ability to improve the user experience and accessibility in real-time. However, accurate and efficient hand gesture recognition remains challenging, especially in dynamic environments. This study presents real-time hand gestures and poses using computer vision. Fundamentally, the system can identify and categorize a wide range of hand movements through real-time analysis of the node connect points of the user's fingers. Every successive point is rendered in such a way that associations between several points have been oriented, and differences between each gesture may be classified. Apart from this, the machine learning model is to be trained on such joint points of the user's fingers using angle approximation in between them and gradient as per pixels per inch according to a hand gesture-capturing adapter like a camera. For this task to be done, we aim to use the Support Vector Machine (SVM) algorithm for recognizing the classifying parameters of live gestures. Because of its higher performance in high-dimensional feature spaces, SVM is well suited to the complicated and dynamic nature of hand gestures. After being trained on the dataset obtained by extracting the attributes of the node points, the SVM recognizes the complicated patterns associated with each motion. Our model achieved an accuracy of X%, whereas the benchmarked study reported an accuracy of Y%. To enhance the accuracy and resilience of the whole model, a support vector machine is incorporated into the classifier to maximize the classification function of the courier-based domain.

Keywords: *Support Vector Machine, Hand Gestures, Gesture Detection, Gesture Recognition, Machine Learning, Node Point, Computer vision.*

I. INTRODUCTION

For almost half a century, human-computer interaction technology has been a major area of scientific study due to the quick advancement of computer vision and intelligent learning. In contrast to other forms of information, such as written text, sounds, visual pictures, and brain electrical signals, gesture information is simpler to get and more likely to become widely used [16]. Thus, hand gesture control, which involves gaining control of the computer through the input of hand image data, has emerged as a significant area of study in human-computer interaction. Hand gestures can significantly reduce the learning curve for users and increase the number of application sites available as compared to standard input modes for managing computers. This study offers a novel method for quickly identifying hand movements utilizing the Support Vector Machine (SVM) algorithm and Node Point Approach. The primary goal is to accurately identify and categorize these hand gestures according to their intended meaning.

To redefine real-time hand gesture recognition, this work gives a novel Node Point Approach. Taking benefit of the natural dynamics and shape of hand anatomy, the node factor technique goals to seize the geometric relationships between the nodes' join factors of the fingers. The node point approach collects and analyzes these node points' spatial and temporal houses so that it will extract wealthy function representations that capture the essence of diverse hand gestures. The research incorporates Support Vector Machine (SVM) algorithms for gesture classification to further improve the Node point approach's discriminative power and generalization abilities [8]. The research has two goals: first, it will create a comprehensive Node point approach with an SVM framework that can recognize a variety of hand gestures in real-time; second, it will compare the performance of seven hand gestures of the suggested framework in the form of accuracy in gesture recognition. Support Vector Machine (SVM) is among the best suitable algorithms that are being used for CV tasks to be done as it recognizes the gestures, classifies them into discrete node points of fingers, and comprehensively integrates them again for final gesture recognition. The aim is to use seven gestures to be recognized, and for each gesture, a specific task is to be done. For gesture recognition, use the cv2 library of Python, which is commonly regarded as an open CV, by which it is suitable to define the parameters of hand gestures such as distance orientation and time frame allocation of gestures against the recognition adapter, which is a webcam or cell phone camera in our case [8]. Furthermore, the classification of gestures is done through keras, Media Pipe, and Tensor Flow packages of machine learning for gesture classification and training the model utilizing a dataset of sample images for each successive gesture [11].

We aim to provide several hundred images for each gesture so that the model can be trained by avoiding every possible error or anomaly and may produce efficient results. Every gesture has a specific duty assigned to it when it is recognized [10]. The Kivy framework in Python, SQLite for database administration, and an API gateway to enable SMS functionality and GPS integration are all needed in the creation of a mobile application for these activities [19]. Through seamless integration of cutting-edge gesture detection technology with useful functionality, our all-encompassing strategy promises to provide a mobile application that is both feature-rich and sturdy. Apart from the functional side of the project, each gesture is dealt with as the object parameters of the subsequent task to be done in a generic class of every feature. Among them, some of them are dependent on the prerequisites, and the rest are independent [2] [12].

A. Objectives

The study aims to address the challenges caused by a lack of productivity, accuracy, efficiency, and quality control that the courier sector may be experiencing. Depending on the industry model, our project might find use in multiple industries. When succeed in doing so, the courier industry will find it to be sufficiently profitable.

B. Problem Statement

The gap in the supplier industry is between supervisors and delivery agents. Supervisors are desperate for such a solution that they are fully updated regarding the work ethics of their agents so they might not commit fraud with them. Apart from this, agents may be reluctant to receive a real-time update from their supervisor to ensure their loyalty to the job. This requires a great deal of time to sort out such problems efficiently, and a time-saving solution is needed. Another problem is the courier industry, which lacks in the field of delivery. The delivery agent might deliver the parcel to an unconcerned person, or he may commit fraud with his supervisor, lack performance, and not maintain timely delivery.

C. Proposed Solution

The main goal in solving this problem is to get the best output of resources from a single mobile-based application in a centralized way. Such domains may include the supply industry, in which suppliers provide stocks and goods to the primary consumer and are accountable to their supervisor. They had to manage stocks and inventory, so supervisors were very concerned regarding the job responsibilities of agents so that they would not commit fraud. Under these circumstances, agents were required to report that they were on duty through multiple channels, none of which are typically approved by their supervisors.

In the following sections, we will present a thorough literature review, methodology with hand gestures setup, and results, culminating in conclusions and potential future directions. Through this research, we aspire to contribute to the ongoing efforts in real-time hand gesture recognition and ultimately positively impact in courier-based domain and for beneficial in other domains.

II. LITERATURE REVIEW

The main overview of HGR using radar sensors is the work described in [1]. Both deep learning-based HGR algorithms and multi-domain hand gesture data representation approaches are now in use for sign processing. HGR radars are divided into two categories: pulsed and continuous waves. Detailed information on the hardware and algorithms of each type is given. An assessment is also given on the ongoing developments in radar-based HCI, as well as the radar hardware and algorithms that are now accessible. They used radar to recognize gestures and produced devices and applications.

The authors in this study [2] have trained a model in which they use 32 Arabic alphabet sign classes, which can classify the Arabic sign language. Images can identify sign language by the pose of the hands. They suggested a structure, where they used two CNN models, and trained both models individually on the training set, they used the ArSL 2018 dataset. The primary contribution of this research is to resize 64-pixel images, convert grey-scale images to three-channel images, and then apply them. Then, the preprocessed image is introduced into two different models, which are ResNet50 and MobileNetV2. The results they obtained all of the data on the test set are accurate at about 97 percent after the application of many preprocessing and different techniques.

Using a live video feed as a source of data, this work [3] aims to identify human gestures from an undefined stream of data. The fact that there are different lighting conditions, backgrounds, and gesture positions within the same data stream makes this task even more complicated. To classify gestures captured from various angles and with different object sizes, this work presents an efficient deep-learning architecture. We created a synthetic real-world dataset in this work, comprising 4500 photos taken from individuals ranging in age from 10 to 50, to carry out the classification. The findings show that, with an accuracy of 99.63%, the method outperforms previous works in detecting gestures in situations where lighting is deteriorating and gesture positions are unclear.

The authors of [6] focus on providing an overview of attainable gesture strategies in research and outline their advantages and limitations in certain scenarios. The effectiveness of these methods, which center on laptop vision strategies for handling factors such as similarities and differences, classification algorithms, hand segmentation and drawbacks, quantity and types of gestures, dataset, detection range (distance), and type of camera used, is comparable.

A single six-axis patchable IMU mounted at the wrist through recurrent neural networks (RNN) is used in the described hand gesture detection system [9]. The IMU is made up of electronic parts with IC architecture mounted on stretchy, sticky interconnections that are serpentine-structured on the substrate. The suggested patchable IMU's soft form factors allow it to be pleasantly worn in intimate coming into contact with the human body while accommodating skin flaws. As a result, there is a minimum amount of signal distortion (also known as motion artifacts caused by vibration during motion.

The development of mathematical algorithms for human-computer interaction that can identify human gestures is the main objective of this research, according to the authors in [10]. Utilizing a keyboard, mouse, touch screen, and other devices are just a few of the interface options available to users of computers. There are certain limitations with each of these devices when it comes to incorporating more flexible technology into PCs. This describes how some operations, like turning pages or paging up and down a page, may be taught to be performed by hand gestures.

Using a live video feed as a source of data, this study [15] aims to identify human motions from an undefined stream of data. The fact that there are varied lighting conditions, backdrops, and gesture placements within the same data stream makes this process considerably more complicated. In this study, we have synthesized a real-world dataset of 4500 photos taken from individuals ranging in age from 10 to 50 to conduct the categorization. To address the complexity of the gesture identification process, a wide range of variables are taken into consideration when compiling the dataset. The findings show that, with an accuracy of 99.63%, our method exceeds previous research in detecting gestures in situations when illumination is decreasing and gesture placements are unclear.

Hand-gesture recognition is becoming an essential component of Human-Machine Interaction (HMI) due to the quick development of technology [12]. This paper presents a visual recognition-based hand-gesture recognition system; three different scenarios are covered in the research. Using 50 pictures that depict the thumb, index, middle, ring, and pinkie fingers, HGR is created in the first scenario. To enhance the size of the dataset, the second scenario adds 200 photos for the identical set of fingers as the first. The last example uses the fingers that were previously stated and counts motions up to 2000 photos. Several issues arise with real-time hand gesture recognition, such as Variations in illumination that can impact gesture recognition precision. The distance range between the hand and the camera is important—issues about GPU configuration.

III. ISSUES AND CHALLENGES

There are various issues and challenges we have faced so far while developing this project. These can be classified into a categorical listing that best describes them, following the domain they are associated with, and further, we intend to define every aspect within this classification. These are as follows:

A. CPU Centric Challenges

CPU Configuration: The clocking speed or frequency of the processor must be a result of a range between 9.0 and 9.6 Hz and higher. It may apply to at least a quad-core processor utilizing all of its logical cores' clocking frequency as a comprehensive value. It is better to use a processor with eight logical cores, which results in a higher clocking speed [24].

Hyper-Threading: Hyper-threading is essential for this project to be developed during the implementation phase and post-development within a working environment. It is the concept of the parallel working mechanism of the logical cores of the processor that works on the priority algorithm. This is used in the training of the machine learning model for this project which is based on x86 processor architecture [4].

B. GPU Centric Challenges

GPU Configuration: While dealing with such a huge dataset of three thousand sample images for each gesture, there are eight gestures on which this project is based, so it sums up to a total of twenty-four thousand sample image datasets. It is foremost needed that at least 2 Gigabytes of memory of the Graphical Processing Unit (GPU) along with the buffer graphics frequency be high-end, and the GPU buses must be synchronized with the CPU and primary memory (RAM).

C. RAM Related Challenges:

Primary Memory (RAM): The magnitude of primary memory (RAM) for this project to be developed is at least eight gigabytes, along with a bus match of a minimum frequency of 1300 megahertz. The minimum generation required is DDR3 SDRAM. (Double Data 3: Synchronous Dynamic Random Access Memory) [17].

Operating System Environment for Development: Since this project is developed mostly for Microsoft Windows 10 Enterprise and Home versions, it is needed to find the scope for those external resources such as libraries, packages, application programming interfaces, virtual environments, and development kits that are compatible with this operating system and lie under the Open Source Software Licensing Community for the Windows 10 operating system [10].

Operating System Environment for Deployment: The project's goal is to be deployed on the Android operating system since its fixes must be compatible with the Android kernel, which is derived from the Linux Long-Term Supported Kernel (LTS). Known as Android Common Kernels (ACKs), LTS kernels sometimes combine with Android-specific fixes [21].

D. Firmware-Related Challenges

Network Configuration: Several components in the network are crucial for this project since it is based on machine learning technology. The uploading speed should be a minimum of 15-20 megabytes per sec [18]. Bandwidth is another major component; there must be at least 1.5 Megabytes/sec of bandwidth required for this project during the implementation phase and post-deployment as well [7].

Port Forwarding and Port Sharing: Ports are the gateways to any node system for data sharing with the server. For a given scenario, if a port is assigned to any specific node, it is assigned a fixed bandwidth, and this is to be kept in mind that this port is not shared with another pathway to avoid squeezed bandwidth, resulting in resistance in the way of data transfer for this project. Port sharing using network switching affects in the form of lessening the data flow streamline and increased pinging in terms of network packet transmission through network layers [20].

E. Challenges Related to Camera and Sensors of Cellular Device

Aperture Time and Frames: The number of frames per second is the ability of the camera to capture one-second frames in either quantity since it must satisfy the formulation of 2n, where n = 3, 4, 5, 6, 7, 8,... [21] Since the elements of set values for ‘’n” start with 3, it is obvious that there must be at least 8 frames per second captured within one second. These values vary from device to device, as nowadays, for most cellular devices; the value is at least 32 frames per second as per the default settings. Whatever the aperture speed of the desired device on which the project is to be deployed, the fact remains that its frame speed must be synchronized with the speed of the frames on which the machine-learning model is trained. The model in this project needs to be responsive in all situations as long as the parameters do not go beyond the frame speed limitations for the Android device and model as well as because the external third-party resource from which the model was trained leaves its configuration footmarks on the machine learning model indefinitely. For this reason, we trained the model at a rate of seven frames per second. To conduct identification, classification, and grouping according to the distinct pattern for gestures, each set of seven frames from the camera sensor will be compared with the next seven sample photos from the dataset.

Delay Time Coordination to Response Time: This process has an upright association with aperture time in terms of frames per second. It is responsible for the mechanism that the amount of time required for the machine learning model to recognize each gesture and decide to discriminate between all of the eight gestures is regarded as the delay time [22]. Apart from this, the response is the decision made by the model for the recognition of gestures and to produce the best possible judgment by parameters passed for each specific gesture in terms of critical points that differentiate between them against the proposed pattern initialized for them [23].

IV. METHODOLOGY

With the use of computer vision techniques and the Support Vector Machine (SVM) algorithm for precise classification, this research focuses on developing a real-time hand gesture detection system. Real-time analysis of the node connect points of the user's fingers is the primary emphasis to recognize and classify a wide variety of hand actions. We used PyCharm for the experiment of the hand gesture mechanism; it is an ideal Integrated Development Environment (IDE) for effective Python 3, web, and data science development. Scientific libraries and tools like TensorFlow, tensor flow lite, OpenCV, and Numpy can be integrated with PyCharm. This section explains the main characteristics of this research as well as the training process for machine learning models.

A. Training of Machine Learning Model

Approach Mechanism: One of the most important aspects of this research is model training. It is stated that the more precise the model, the more certain the workflow procedure will be. There are numerous approaches for the training of models as we grow in the research to be classified into chunks and building blocks. The criteria, that the group, has adopted so far are defensive at first, followed by a proactive approach. The steps are carried out sequentially in general; otherwise, they may be carried out concurrently. By adopting this, we intend to reduce the risk of ending up at some point in the project where the components end up in a dry-run virtual manner, negating the exception handling [13]. These various steps are carried out just to face a smaller proportion of limitations regarding the working environment and associated parameters. This approach is carried out to be more factual, and to maintain the operational environment of the project, it is to be more practical in the utmost manner. Approaches for the training of models as we grow in the project to be classified into chunks and building blocks, the criteria, as the group, has adopted so far are defensive at first, followed by a proactive approach.

The proposed approach is being taken to such an extent that the user may not face limitations and difficulties while operating the project and is realistic; otherwise, there would be no point of reference for the use of machine learning technology. This is why machine learning is not that well implemented in the projects up to a far extent in our society, though it is the talk of the town among enthusiasts, scholars, students, and professionals. By defensive approach, we mean that it is to be kept in mind that the foremost primary goal of the system must be that gesture projection limitations must be dealt with perfection. There are several limitations to the projection of gestures against the system. It is to be made possible in nearly all circumstances that the system must recognize the gesture using providing a tendency to the project model so that it recognizes the adjacent and similar pattern in the model. Gesture orientation is a major difficulty since varied orientations of gestures might make it difficult for the system to understand them correctly, which can result in misinterpretations. Another challenge is variations in light intensity, which might make projected movements less visible and possibly make it harder for the system to recognize human input [13]. There is a chance that the system will recognize inadvertent gestures because of false pattern correction, which could cause mistakes or confusion. Furthermore, the system has to deal with issues about the color and intensity of movements, necessitating a high level of color accuracy for dependable identification. These weaknesses highlight the necessity of ongoing developments in gesture projection technology to increase its dependability and practicality in a variety of real-world settings.

In the follow-up of the defensive approach, we measured the proactive approach, which meant strengthening the model behavior as per the systems' proposed interaction with the users and environment. These include the mechanism of reinforcing the gestures' paradigms so that they could be more definite and crystal clear in their respective mechanisms. To add context, dealing with the complexities, such as; Limited light intensity, Aperture orientation of gestures, Angular measure in between the components of gestures, Mechanism of axes in gesture canvas, Matrix formation and orientation, Co-ordinate system (i.e., Abscissa) and ordinate plane geometry of canvas. Users' skin tone complexity, System delay-response coordination, etc.

Model Development: The model is developed based on several parameters. These parameters encircle the components of the project in regard with deliverables being kept in consideration. We have classified the parameters into two types; generic and hyperparameter.

- ***Generic Parameter:*** Generic parameters define the fundamental orientation of gestures in terms of definition, discrimination, and initialization. These are discussed below:
  - **Gesture Saturation:** It is the process of determining the number of pixels per inch within a projected gesture in its original geometry [16] as shown in Figure 1. The orientation of the gesture is not disturbed using the extension in dimensions, but the missing pixel blocks are classified.

Figure 1: Gesture Saturation

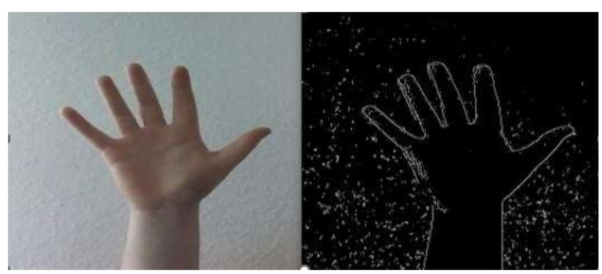

Figure 2: Edge Detection

- **Edge Detection:** The external boundaries of gesture are based upon the three-dimensional axes' matrices as shown in Figure 2. This is to be measured to determine whether the frame of the gesture is capable to such an extent that it uses all its projections while execution on its axes and the limitations in between axes by using image and object distance to be within the working extension in terms of distance [5].
- **Dilation of Gesture:** Dilation as shown in Figure 3 is the process where the gesture is transformed into an edged image, followed by the spreading of magnified pixel transformation using increasing the axes magnitudes so that the axes of rotation are not disturbed.

- **Grey Scale:** Grey scaling is the process that is a prerequisite for the gesture axes vector mechanism [15]. It enables the classifier to acknowledge the gesture using the units of saturation present in he projected gesture as shown in Figure 4.

Figure 3: Edge Dilation

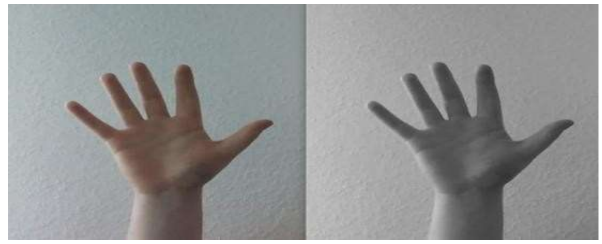

Figure 4: Grey Scale

- **Blurring of Gesture:** Blurring of intangible objects within the gesture projection image is carried out by negating the unwanted objects within the aperture of the gesture as shown in Figure 5. This is a concurrent and recursive process by classifying all the time within the gesture projection until the execution timed out [1]

- **Hyperparameters:** Hyperparameters refer to external configuration settings that are employed to regulate a machine learning model's learning process. Before the start of the training process, they are set and have not learned from the data. Hyperparameters are the secondary parameters which entirely rely on generic parameters. They are responsible for the working mechanism of the model. These include:
  - **Iterations:** Iterations are defined as any formal and rational loop structure that is achieved under a certain condition. It forms the basis for the mechanism of the model in such a way that it has several axes of matrices for all the gestures of the system. It is a real-time-based recursive process where several generic parameters are trained on a specific gesture at a time [17] [4].

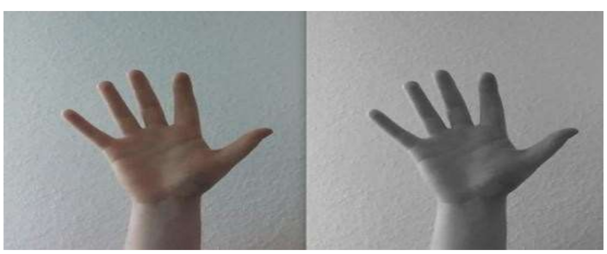

Figure 5: Blurring of Gesture

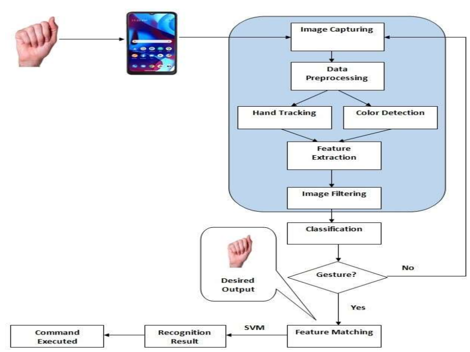

Figure 6: Representation of Model Classification

- **Iterations:** Iterations are defined as any formal and rational loop structure that is achieved under a certain condition. It forms the basis for the mechanism of the model in such a way that it has several axes of matrices for all the gestures of the system. It is a real-time-based recursive process where several generic parameters are trained on a specific gesture at a time [17] [4].
- **Epochs:** Epochs Fare is regarded as the number of passes gone through a dataset of each gesture from the model that is yet in the training phase. For each significant gesture, we have used three thousand sample images to minimize the risk of constraints in operating epochs, which paved the way for the dataset to interact with the parameters for training the model in every possible circumstance in terms of the use case scenario [16].

- **Batches:** Batches are the clusters from the dataset of each gesture that set boundaries for the number of sample images that are trained at a given time in a single iteration.

  If the light intensity is lower in a section of a series of sample images for a specific gesture, then it must follow the same instructions using creating insights from them for all the other sample images in the dataset. Figure 6 shows the representation of model classification, firstly the user initiates the process by performing a certain hand gesture in the context of gesture recognition systems [14]. To improve its clarity and have it ready for analysis, the gesture is preprocessed once it is made. Noise reduction and normalization techniques may be used during this preprocessing stage to guarantee proper interpretation. An extraction of features from the hand motion occurs after preprocessing. These functions as distinct qualities that set one gesture apart from another, enabling accurate identification. The system then determines if the detected hand type corresponds with predetermined gesture patterns. The system then determines if the detected hand type corresponds with predetermined gesture patterns. The system performs the appropriate action or output that corresponds to the identified gesture if a match is found. To ensure a smooth and responsive user contact, the system reroutes the process if the gesture is not recognized. This all-encompassing method guarantees dependable and efficient gesture-based interactions across a range of applications, including user interfaces and gaming.

  The critical analysis would include determining how various hyperparameters (kernel type, width, and regularization parameter) affect the overall performance.

B. System Features

The functions of the product are regarded as the features, which are termed the absolute deliverables of the product which include; E-mail verification of parcel deliverables to the user entity, Global Positioning System (GPS) location token generation as per live location and transmitting it to the user and/or admin panel, Video track recording of the meeting, Inventory management update to the admin panel, Audio track recording of the meeting between client and delivery concerned personal, SMS update to the admin and user entities and the Fundamental operations performed regarding deliverables to the admin panel, such as the number of parcels in the queue waiting to be catered, the classification of parcels in terms of urgency level, time sensitivity, etc.

C. Gestures Representation & User Interaction with Application

Gestures are a natural and intuitive means of communication in the field of human-computer interaction. They provide a rich tapestry of cues that can be used to bridge the gap between humans and machines because they capture complex movements, expressions, and intentions. Hand gestures are found to be an effective means of intuitive and effective communication in this study. Of these gestures, seven fundamental ones shown in Table 1 stand out as particularly pertinent and accommodating. This research provides insightful information about the categorization, representation, and uses of these gestures through a thorough analysis of their communicative purpose. It is the proportion of accurately predicted points (predicts) to all predictions. Its value is in the interval of 0 to 1. We can predict the accuracy of every gesture using the accuracy formula above.

$$\textbf{Accuracy: } \frac{\textit{Number of Correct Predictions}}{\textit{Total Predications}}$$

The process of utilizing hand gesture recognition to create a rider/user sign-up and login system for a courier-based application has several stages, as shown in Figure 7 and Figure 8. Establish a conventional user registration procedure where users input their name, email address, and password. Incorporate a step in the registration procedure where users have to validate themselves using a series of predefined hand gestures. Pay attention to the relevant details that emerged from these moves and securely preserve them in the database.

Table 1: Representation of Hand Gestures

Gestures	Meaning Of Gesture	Features
	OK	SMS Update
	Peace	Video recording
	Thumbs Up	Verification to admin panel before email update
	Thumbs Down	Email update
	Call Me	Audio trackrecording
	Stop	Inventory real-time update
	Live Long	GPS Location track

Figure 7: Representation of Sign up Page

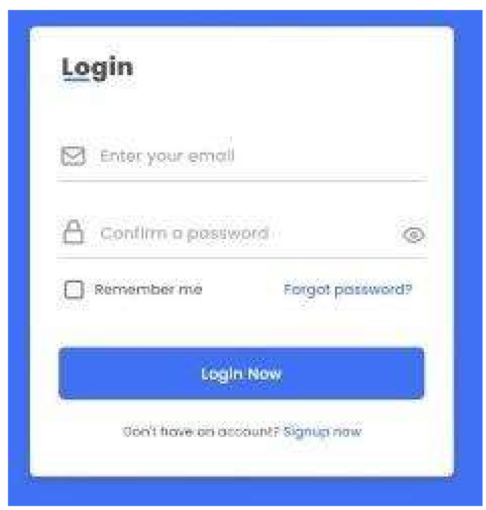

Figure 8: Representation of Login Page

The application's main screen, which allows the rider/user to operate the features on his panel, is described in Figure 9.

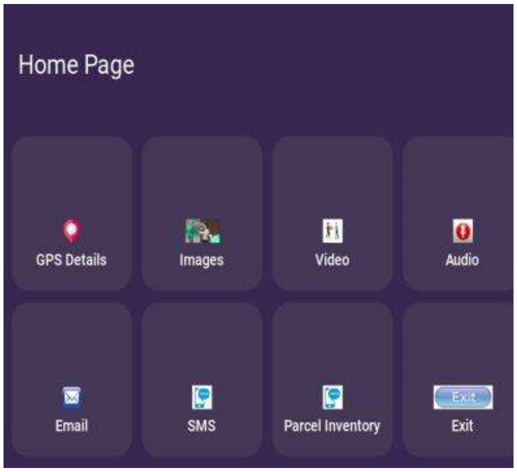

Figure 9: Representation of Home Page

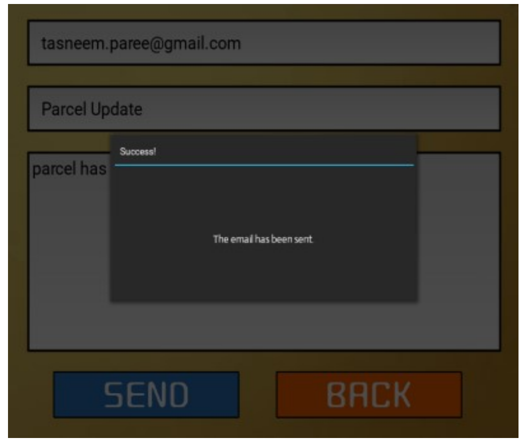

Figure 10: Representation of Email verification

Email verification is a crucial step in the courier-based application, as shown in Figure 10. Several key operations in the package delivery ecosystem rely on this verification procedure. The architecture of the system makes sure that every activity is closely linked to an authenticated email, encouraging responsibility and openness in the courier service. The parcel delivery procedure is made more efficient and trustworthy overall by the program, which develops a dependable and traceable communication route by utilizing email verification for these crucial operations. The administrator shown in Figure 11 has received the email verification status. This implies that managing the email verification procedure is the responsibility of the administrator. The application has several other features as shown in Figure 9, to improve the user experience and offer a wide range of options to users.

The administrative dashboard, seen in Figure 12, is a consolidated interface that offers a clear, up-to-date summary of important data indicators and activities associated with administrative activity. A form will open when the administrator hits the "Supervisor" button as shown in Figure 13. The administrator may be able to enter or manage supervisor-related data on this form, including roles, permissions, and other pertinent information.

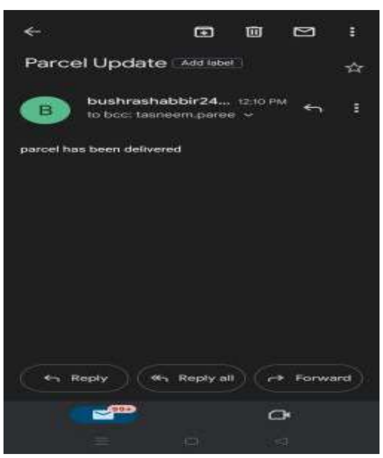

Figure 11: Representation of Email Status

Figure 12: Representation of Admin Dashboard

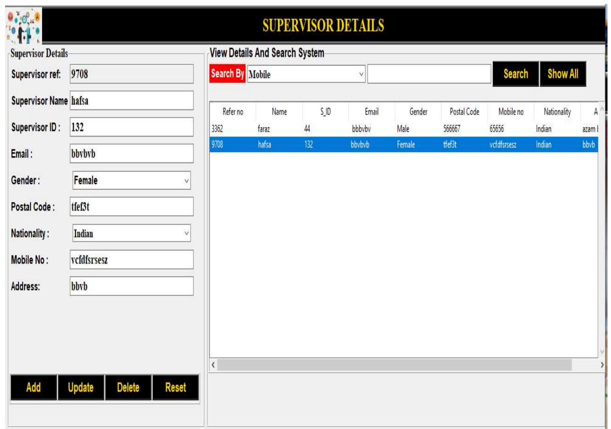

Figure 13: Representation of Supervisor form

V. RESULTS

The model detects gesture orientation quite well. Since their dislocation them is rightly associated with the inference formation within the axes of the matrix of each gesture, in general, all of the gestures are identified, classified, clustered, and acknowledged, followed by the discrete success rate of detection of each significant gesture in terms of percentage efficiency. Detection is done for certain goals to be achieved. Gestures in a trained model adhere to the hierarchy that was created within the model. To address the complexity of the gesture identification process, a wide range of variables are taken into consideration when compiling the dataset. We used 500 visuals to illustrate the seven hand motions. Though the model has been trained as per the proposed pattern of classification and has been recognized by the system, the concurrency factor remains in its place as per the fundamental component of the image processing domain of deep learning of cognitive neural networks.

Node points, such as fingers, the palm centroid, or knuckles, function as distinguishing features on the hand. Through the use of depth sensors or computer vision, these points are monitored in real time, allowing for dynamic analysis of hand configurations during gesture execution. The unique benefit of the node point method is its capacity to record movement trajectories and spatial interactions, converting them into feature vectors for gesture identification. The system's sensitivity to minute fluctuations in hand motions is improved by this fine-grained feature extraction approach, which also makes it easier to generate a large dataset for the SVM's training. The SVM picks up the complex patterns connected to every move after being trained on the dataset created from the node points' extracted characteristics. In real-time recognition, the SVM quickly classifies the hand motion being made by effectively mapping the current configuration of observed node points to the learnt patterns. A reliable real-time hand gesture identification solution is offered by the Node Point Approach and SVM combined. SVMs are great, but it's crucial to remember that they can be computationally expensive for large datasets and that careful tweaking is required to find the right kernel and hyperparameters. However, SVMs are a suitable and often preferable alternative for classification problems in many applications due to their advantages, including their resilience against overfitting, ability to handle non-linear interactions, and efficacy in high-dimensional fields. The metrics of gestures are oriented into forms of axes of the canvas of the gestures, namely the x, y, and z axes of orientation of gestures. For reference, screen shots of such axes of gestures are attached for each. The aggregate of values for x and y is determined, and the average is calculated, which results in a comprehensive accuracy test for each gesture. The z axes are nondeterministic since they are none other than the dislocation in the x-y coordinate plane, which has no absolute maximums and minimums for the z axes.

A. Call me:

The corresponding "Call me" action is regarded as an audio track recording, as illustrated in Figure 14, indicating that the system successfully identified the action to be performed in the courier-based application.

B. Live Long:

The corresponding "Live Long" action is regarded as a GPS Location track, as illustrated in Figure 15, indicating that the system successfully identified the action to be performed in the courier-based application.

Figure 14: Representation of Call me action

Figure 15: Representation of Live Long Action

C. OK Action: The corresponding "OK" action is regarded as a SMS Update, as illustrated in Figure 16, indicating that the system successfully identified the action to be performed in the courier-based application.

Figure 16: Representation of OK action

Figure 17: Representation of Peace action

D. Peace Action:

The corresponding "Peace" action is regarded as a video recording, as illustrated in Figure 17, indicating that the system successfully identified the action to be performed in the courier-based application.

E. Stop Action:

The corresponding "Stop" action is regarded as an inventory real-time update, as illustrated in Figure 18, indicating that the system successfully identified the action to be performed in the courier-based application.

F. Thumbs Down:

The corresponding "Thumbs down" action is regarded as an email update, as illustrated in Figure 19, indicating that the system successfully identified the action to be performed in the courier-based application.

G. Thumbs Up:

The corresponding "Thumbs up" action is regarded as Verification to the admin panel before the email update, as illustrated in Figure 20, indicating that the system successfully identified the action to be performed in the courier-based application.

Figure 18: Representation of Stop action

Figure 19: Representation of Thumbs Down

Figure 20: Representation of Thumbs Up

To minimize ambiguity and guarantee a seamless identification and classification process within the limitations of the live canvas orientation, especially when using the mobile device camera, the study strategically employs the support vector machine (SVM) algorithm to cluster gestures based on their node points. SVM is the best option for this task as it can draw distinct decision boundaries between several classes, which should result in accurate and effective hand motion detection. Table 2 shows the system's performance concerning different hand gestures, including the proportion of correctly identified gestures relative to the total number of occurrences. Among the different gestures analyzed, OK with 89%, Peace with 78%, thumbs up with 80% and thumbs down with 77% consistently show the highest accuracy in the system's performance evaluation as shown in Figure 21. The system's remarkable accuracy suggests that it is very good at recognizing and categorizing these specific hand gestures. Due to the distinct and easily observable features associated with these gestures, the robust performance of the thumbs-down and thumbs-up gestures suggests a precise and dependable recognition mechanism.

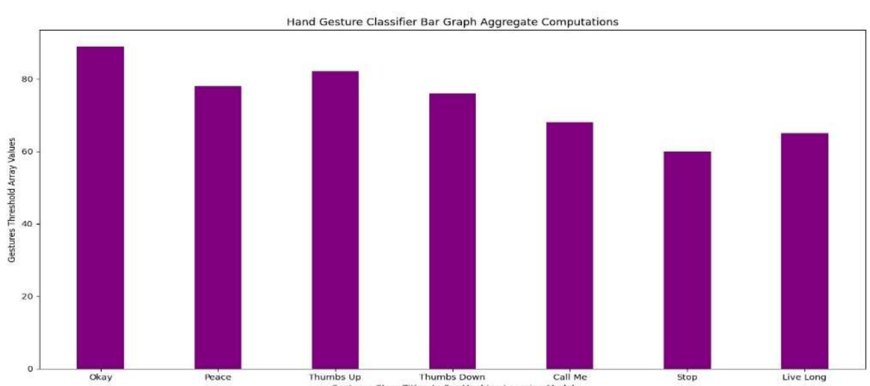

Figure 21: Representation of Bar Chart of Hand Gesture Classifier Accuracy

Table 2: Gesture's Accuracy

Gestures	Meaning Of Gesture	Accuracy
	OK	89%
	Peace	78%
	Thumbs Up	80%
	Thumbs Down	77%
	Call Me	65%
	Stop	59%
	Live Long	63%

In Figure 22, the confusion matrix explains the hand motions themselves (real labels) are represented in the rows. Here, we have eight distinct gestures: "CM" (Call Me), "St." (Stop), "OK" (Okay), "P" (Peace), "TU" (Thumbs Up), "TD" (Thumbs Down), and "LL" (Live Long). The hand gesture classifier's anticipated labels are shown in the columns. The right predictions are found in the diagonal cells (highlighted in yellow), where the expected label coincides with the actual label. The diagonal cells' yellow intensity represents the likelihood or degree of confidence in the right forecast. Higher confidence is indicated by darker yellow (values closer to 1). Misclassifications are shown by off-diagonal cells. An off-diagonal cell might be present, for instance, if the classifier predicted "OK," but the gesture that was made was really "P." These cells' values represent the percentage of incorrect classifications. This confusion matrix aids in assessing the hand gesture classifier's performance throughout a range of gestures. The setup of the system is found by comparing the SVM-generated expectations with the user's genuine movements. According to the discoveries, the framework consistently shows high exactness for specific motions like Alright, Peace, Thumbs Up, and Thumbs Down. Within the perplexity network, misclassifications are caused by off-diagonal components, while the number of substantial forecasts for each motion is determined by diagonal components. It appears that the SVM can precisely recognize signals, especially for the profoundly precise categories of OK, peace, thumbs up, and thumbs down.

VI. CONCLUSION

Understanding hand motions is important for computer interfaces. These interfaces allow computers to interact with living creatures, like humans. A new method, studying the connection points of a user's fingers, offers a solid base. It helps to accurately recognize and sort different hand signals. It uses gradient measurements and angle guessing. To avoid confusion and to recognize accurately within live- mobile device's came-ra limitations, support vector machines (SVMs) are used. They smartly join movements by their connection points. Looking closely at how the system works, it does a great work recognizing different hand signs. It's especially good at picking out gestures like OK, Peace, Thumbs Up, and Thumbs Down with high accuracy rates. This shows that the system is good at figuring out what these signs mean. It makes it easier for users to interact with, the system works well and these success points could guide future improvements.

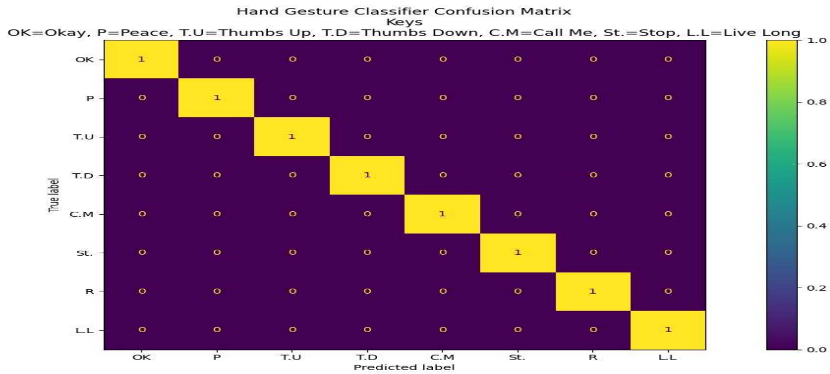

Figure 22: Confusion Matrix Obtained from SVM Technique

There are many possible uses for this technology, such as accessibility improvements and touchless device control. Further exploration of the fine points of gesture recognition systems including but not restricted to OK, peace, thumbs up and thumbs down signs can be anticipated shortly to understand how well they integrate with daily interactions. This will lead us closer to a time when basic and efficient communication with gadgets will be standard. This will further ensure that it broadens its user base across cultures and preferences for inclusion while improving the system's functioning capabilities and customer experience. Again, research into these fields may help us create useful tools which enhance efficiency, safety operations and overall user satisfaction within courier-based businesses by matching their unique demands and challenges. In addition, the hand gestures-based recognition model could be applied to various other use cases such as those that pertain to commercial or educational contexts.

REFERENCES

[1] Ahmed, S., Kallu,K.D.,Ahmed,S.,& Cho,S.H, "Hand Gestures Recognition Using Radar Sensors for Human-Computer-Interaction:AReview",Remote Sensing,13(3),527, 2021.

[2] Alnuaim, A., Zakariah, M., Hatamleh, W. A., Tarazi, H., Tripathi, V., & Amoatey,E.T., "Human-Computer Interaction with Hand Gesture Recognition Using ResNet and MobileNet", Computational Intelligence and Neuroscience, 2022.

[3] Mohammadi, Z., Akhavanpour, A., Rastgoo, R., & Sabokrou, M. "Diverse hand gesture recognition dataset", Multimedia Tools and Applications, 1-23, 2023.

[4] Mujahid, A., Awan, M. J., Yasin, A., Mohammed, M. A., Damaševičius, R., Maskeliūnas, R., & Abdulkareem, K. H., "Real-time hand gesture recognition based on deep learning YOLOv3 model", Applied Sciences, 11(9), 4164, 2021.

[5] Elmagrouni, I., Ettaoufik,A.,Aouad,S.,&Maizate,A.,"Approach for Improving UserInterface Based on Gesture Recognition", E3SWeb of Conferences,297,01030, 2021.

[6] Oudah, M., Al-Naji, A., & Chahl, J., "Hand Gesture Recognition Based on Computer Vision: A Review of Techniques", Journal of Imaging,6(8),73, 2020.

[7] Jain, R., Jain,M., Jain, R.,& Madan,S.,"Human Computer Interaction – Hand Gesture Recognition". Advanced Journal of GraduateResearch,11(1),1–9, 2021.

[8] Almasre, M. A., & Al-Nuaim, H., "A comparison of Arabic sign language dynamic gesture recognition models", Heliyon,6(3), e03554, 2020.

[9] Valarezo Añazco, E., Han, S. J., Kim, K., Lopez, P. R., Kim, T.-S., &Lee, S., "Hand Gesture Recognition Using Single Patchable Six-Axis Inertial Measurement Unit via Recurrent Neural Networks", Sensors,21(4),1404, 2021.

[10] B., A., Krishi, K., M., M., Daaniyaal, M., & H. S., A., "Hand gesture recognition using machine learning algorithms", Computer Science and InformationTechnologies, 1(3), 116–120, 2020.

[11] Haria, A., Subramanian, A., Asokkumar, N., Poddar, S., & Nayak, J. S., "Hand Gesture Recognition for Human Computer Interaction", Procedia Computer Science, 115,367–374, 2017.

[12] Mahmood, N. T., Jabbar, M. S., & Abdalrazak, M. "A Real-Time Hand Gesture Recognition Based on Media-Pipe and Support Vector Machine". In International Conference of Reliable Information and Communication Technology (pp. 263-273). Cham: Springer Nature Switzerland, 2023.

[13] Parvathy, P., Subramaniam, K., Prasanna Venkatesan, G. K. D., Karthikaikumar, P., Varghese, J., & Jayasankar, T. (2021). Development of hand gesture recognition system using machine learning. Journal of Ambient Intelligence and Humanized Computing, 12, 6793-6800.

[14] Sen, A., Mishra, T. K., & Dash, R. "Deep Learning-Based Hand Gesture Recognition System and Design of a Human–Machine Interface", Neural Processing Letters, 55(9), 12569-12596, 2023.

[15] Aggarwal, A., Bhutani, N., Kapur, R., Dhand, G., & Sheoran, K., "Real-time hand gesture recognition using multiple deep learning architectures", Signal, Image and Video Processing, 17(8), 3963-3971, 2023.

[16] Li, G., Tang, H., Sun, Y., Kong, J., Jiang, G., Jiang, D., & Liu, H., "Hand gesture recognition based on convolution neural network". Cluster Computing, 22, 2719-2729, 2019.

[17] Bucad, M., & Ramos. Proud Hands: An Android-Based Hand Gesture Recognition and Conversion System Using Image Processing, Image Segmentation and Feature Extraction. International Journal of Current Innovations in Advanced Research, 1(6), 44–56, 2018.

[18] Tsai TH, Huang CC, Zhang KL., "Design of hand gesture recognition system for human-computer interaction", Multimedia Tools and Applications, 79(9): 5989-6007,2020.

[19] Sahoo, J. P., Ari, S., & Ghosh, D. K. "Hand gesture recognition using DWT and F-ratio based feature descriptor", IET Image Processing, 12(10), 1780–1787, 2018.

[20] Nuzzi, C., Pasinetti, S., Lancini, M., Docchio, F., & Sansoni, G., "Deep Learning Based Machine Vision: First Steps Towards a Hand Gesture Recognition Set Up for Collaborative Robots". 2018 Workshop on Metrology for Industry 4.0 and IoT, 2018.

[21] Eid, A., & Schwenker, F., "Visual Static Hand Gesture Recognition Using Convolutional Neural Network", Algorithms, 16(8), 361, 2023.

[22] Liu, H., & Wang, L., "Gesture recognition for human-robot collaboration: A review", International Journal of Industrial Ergonomics, 68, 355–367, 2018.

[23] Jain, R., Jain, M., Jain, R., & Madan, S., "Human Computer Interaction – Hand Gesture Recognition", Advanced Journal of Graduate Research, 11(1), 1–9, 2021.

[24] Alnuaim, A., Zakariah, M., Hatamleh, W. A., Tarazi, H., Tripathi, V., & Amoatey, E. T., "Human-Computer Interaction with Hand Gesture Recognition Using ResNet and MobileNet". Computational Intelligence and Neuroscience, 2022, e8777355, 2022.