HAND GESTURE RECOGNITION USING CNN

Paluri Satya Sanjna Reddy
Masters in Computer Science
University of Central Missouri
Missouri, United States
sxp29251@ucmo.edu

Nandhini Chandrakala Raghukumar
Masters in Computer Science
University of Central Missouri
Missouri, United States
nxc63150@ucmo.edu

Sunil Mallik Addagiri
Masters in Computer Science
University of Central Missouri
Missouri, United States
sxa25990@ucmo.edu

***Abstract*—** Human gestures are nonverbal forms of communication that are essential in human-robot interactions. Which is also known as Sign Language Recognition. Vision-based Methods for detecting hand motion and supporting such interactions are critical. Hand gesture detection enables a user-friendly interaction between gadgets and users. Hand gestures can be utilized in a variety of professions, making them useful. capable of being used for communication and other purposes. The goal of sign language recognition (SLR) is to translate sign language into text or speech in order to improve communication between deaf and non-deaf individuals. This exercise has a big societal impact, yet it is nevertheless difficult due to the complexity and wide range of hand gestures. Existing SLR approaches characterize sign language movements with handcrafted features and then develop classification models based on those features. However, it is challenging to design dependable features that can respond to the wide range of hand motions. To address this issue, we offer a unique convolutional neural network (CNN) that automatically extracts discriminative spatial-temporal features from raw video streams without any prior knowledge, hence avoiding feature design. To improve performance, multi-channel video streams. In order to integrate color, depth, and trajectory information, color information, depth clues, and body joint locations are fed into the CNN. We verify the proposed model on a real-world dataset acquired with Microsoft Kinect and show how it outperforms existing approaches based on hand-crafted features. Hand gesture recognition is beneficial not only for persons who are deaf or disabled but also for the people who experienced a stroke, as they need to communicate with other people using Various common vital gestures such as the sign of eating, drinking, family, and others. This paper describes a method for recognizing different hand gesture signs. A Convolutional Neural Network (CNN)-based hand gesture is proposed. The method devised is assessed and compared. depending on measures such as execution time, accuracy, sensitivity, specificity, positive and negative predictive value, likelihood, and root mean square. The results reveal that utilizing CNN, testing accuracy is 99%. An efficient method for extracting different features and classifying data.

Keywords: Convolutional Neural Network, Deep Learning, Hand Gesture Recognition, Pooling layer, Activation function, Sign language recognition.

I. INTRODUCTION

Hand Gesture Recognition, is one of the most extensively utilized communication methods for the deaf, is represented through hand shapes, body movement, and even face expression. Because it is difficult to use information from hand forms and body movement trajectory collectively, sign language identification remains a difficult problem. This work offers an effective recognition model for translating sign language into text or speech in order to assist hearing impaired people in communicating with normal people using sign language. Direct contact has recently been the major mode of communication between the user and the machine. The communication channel is comprised of devices such as a mouse, keyboard, remote control, touch screen, and other methods of direct contact. Non-contact techniques of human-to-human communication, such as sound and physical motions, are more natural and intuitive.

Many academics are considering using non-contact communication methods to support human-computer interaction because of its flexibility and efficiency. The gesture is a non-contact human communication mechanism that is an integral aspect of the human language. Wearable data gloves have traditionally been used to record the angles and positions of each joint in a user's motion. The difficulty and cost of developing a wearable sensor have limited its broad application. Gesture recognition is the ability of a computer to understand gestures and implement

Link of github source code: https://github.com/SanjnaReddy27/Fall-2023-CS5720

commands based on those motions. Gesture recognition's major goal is to create a system that can identify and comprehend certain motions and transfer information from them [1].

Non-contact visual inspection-based gesture recognition systems are currently prevalent. This is owing to their low cost and user ease. A hand gesture is a kind of expressive communication that is utilized in the healthcare, entertainment, and education industries, as well as to aid people with special needs and the elderly. Hand tracking is required to accomplish hand gesture recognition, which entails performing several computer vision operations such as hand segmentation, detection, and tracking.

Within hearing impairment communication, sign language uses hand movements to transmit feelings or information. The fundamental issue is that an average person would readily misinterpret the meaning given. AI and computer vision advancements can be used to recognize and learn sign language [2]. Modern technology can assist a layperson in recognizing and comprehending sign language. This article discusses a method for recognizing hand motions that employs deep learning.

The fundamental technical problem in sign language recognition is establishing descriptors to express hand forms and motion trajectories. Hand-shape description, in particular, entails tracking hand regions in a video stream, segmenting hand-shape images from complicated backgrounds in each frame, and recognizing gestures. Motion trajectory is similarly connected to key point tracking and curve matching. Despite the fact that many studies have been conducted on these two concerns, it is still difficult to produce satisfactory results for SLR due to hand and body joint variation and occlusion. Furthermore, combining the hand-shape and trajectory features is a difficult problem. To overcome these issues, we created CNNs that organically incorporate hand forms, action trajectory, and facial expression. Instead of using commonly used color images as input to networks like [1, 2], we employ color images, depth photos, and body skeleton images all provided by Microsoft Kinect as input.

Kinect is a motion sensor that can offer both a color and a depth stream. The body joint locations can be acquired in real-time using the public Windows SDK. As a result, we chose the Kinect as the capture device for the sign words dataset. Color and depth changes at the pixel level provide useful information for distinguishing between distinct sign actions. Furthermore, the fluctuation of bodily joints in time dimension can show the path of sign actions. Using a variety of visual sources as input causes CNNs to pay attention to changes not only in color, but also in depth and trajectory.

It is worth noting that we may bypass the difficulties of tracking hands, segmenting hands from backgrounds, and constructing hand descriptors because CNNs can learn features automatically from raw data without prior knowledge [3]. CNNs have lately been used in video stream classification. CNNs have the potential to be time consuming. Training a CNN with million-scale in million videos takes several weeks or months. Fortunately, using CUDA for parallel processing, it is still possible to attain real-time efficiency. We propose using CNNs to extract spatial and temporal information from video streams in order to perform Sign Language Recognition (SLR). Existing SLR approaches use hand-crafted characteristics to represent sign language motion and then develop a classification model based on these features. CNNs, on the other hand, can automatically extract motion information from raw video data, removing the need to construct features. We create CNNs using various sorts of data as input. By conducting convolution and subsampling on consecutive video frames, this architecture incorporates color, depth, and trajectory information. On some sign words recorded by ourselves, experimental results show that 3D CNNs outperform Gaussian mixture models with Hidden Markov models (GMM-HMM) baselines significantly.

Stroke is a condition that affects the arteries that lead to and inside the brain. Stroke is the sixth leading cause of mortality and a primary cause of disability. A stroke happens when a blood artery carrying oxygen and nutrients to the brain becomes blocked or breaks. Certain security methods maintain privacy and protect a crucial aspect of the profile. People have collected to demand skilled and capable knowledge as a result of this information. Networks have a medical diagnosis system that allows users to tap into the knowledge and experiences of groups and individuals. This experiment demonstrates that hand gestures are an excellent way to transmit information and that they can be used to interpret a wide range of emotions and facts. Hand gesture recognition is a common challenge with a variety of solutions depending on the application.

We used a 2D convolution network to categorize 10 different motions from 20,000 photos in the Leap Motion Hand Gesture Recognition dataset without utilizing any additional devices. Our model was successful in categorizing ten hand gestures, with an accuracy rate of 99%. Hand gesture recognition has piqued the curiosity of researchers studying human-computer interaction.

Hand gesture detection is critical in many sectors of

human-computer interaction, including virtual reality, gaming, vehicle system control, and robotic control. As more sensors are added, there are more ways to identify hand gestures. We picked a convolutional neural network for this task because 2D CNNs are successful in image classification, and gesture recognition is an image classification problem.

The goal of the assignment is to improve the recognition of human hand poses in a Human Computer Interaction application, save time figuring, and improve client comfort with regard to the used human hand stances. The developers created a PC mouse control application. The program introduces great conduct in terms of time processing based on the provided calculation, hand cushion shading, and hand include.
The proposed hand postures to manage the system boost the user's comfort, and hand gestures can be used to detect and convey the signs as language via speech synthesizer.

II. Motivation

Gesture recognition is a rapidly expanding topic in image processing and artificial intelligence. Gesture recognition is a process that identifies and uses the motions or postures of human body parts to operate computers and other electronic devices. The most important reason for the development of gesture recognition is the ability to construct a simple communication line between humans and computers known as HCI (Human Computer Interaction). This research focuses on identifying hand positions and establishing a man-machine interface. The image's hand region is detected, and the number of active fingers is calculated. The input, which can be an image or a frame from a video, can be collected from a web camera or any other camera in this manner.

A large training database is usually necessary to develop a competent hand gesture recognition system, and diverse gestures should be modelled. We build a human gesture identification system based on a Convolution Neural Network (CNN) with minimal effort on modelling diverse motions, in which the skin color model is improved and the hand stance is calibrated to increase recognition accuracies.

There is no need to design sophisticated algorithms to extract and learn visual information when using a CNN to learn human gestures. Invariant features are allowed with little dislocation across the convolution and sub-sampling levels of a CNN. In this work, the principal axis of the hand is discovered to calibrate the image in order to lessen the effect of diverse hand postures of a hand gesture type on recognition accuracies. Calibrated images aid a CNN's ability to learn and recognize accurately.

III. Main Contributions and objectives

1. The primary purpose is to categorize hand motions into predetermined groups. Each gesture denotes a different instruction or action, such as "open hand," "close fist," or "thumbs up." CNNs are particularly good at learning hierarchical features, making them ideal for image classification applications.

2. Hand gesture detection in real-time or near-real-time is critical for many applications, including human-computer interaction, sign language translation, and virtual reality. CNNs are designed to handle image data efficiently, making them appropriate for real-time applications.

3. Hand gesture recognition systems should be able to recognize motions from a variety of users, taking into account differences in hand shapes, sizes, and looks. CNNs are capable of learning generalizable features that aid in user-independent recognition.

4. Hand gesture recognition systems should be easily expandable or customizable to new motions or commands. CNNs with transfer learning skills can help new gestures be integrated without lengthy retraining.

5. A large amount of labelled data is required to train a hand gesture recognition system. Data augmentation and efficient data utilization can assist CNNs generalize well even with little training samples.

6. Combining information from several modalities, such as depth data from depth sensors or RGB data from cameras, can improve hand gesture recognition accuracy and robustness. CNNs can be constructed to properly handle multi-modal input data.

IV. RELATED WORK

Gesture is a type of body language that individuals use to express their emotions and thoughts. Physical connotations may exist for the various gestures of the five fingers and palm. Hand gesture recognition is a complex system made up of gesture modelling, gesture analysis and recognition, and machine learning. In prior work on gesture modelling, a real-time semantic level American Sign Language detection system was built using the Hidden Markov Model (HMM). A gesture can also be represented as an HMM state sequence.

They used a Finite State Machine (FSM) model to

recognize human motions in employed a Time Delay Neural Network (TDNN) to match motion trajectories and train gesture representations. Feature extraction is vital in a human gesture recognition system since it provides information on the shape, attitude, and texture of a gesture. For example, the training characteristics for the gesture model were fingertips and hand contour. However, because non-geometric elements such as color, silhouette, and texture are unstable, the different light conditions have a significant impact on gesture identification.

Using gesture semantic analysis is appropriate for recognizing a sequence of gestures while doing a complex activity, but it is insufficient for correctly recognizing gestures in a simple continuous motion. FSM was used by Jo, Kuno, and Shirai to solve a task-level recognition problem in which a task was represented by a state transition diagram and each state represented a possible gesture. For gesture recognition, several researchers employed a rule-based system. Culter and Turk created a set of guidelines for identifying waving, jumping, and marching movements. Deep learning has been widely used in a variety of applications in recent years. CNN, in particular, is an excellent tool for image-based learning. for example, used a CNN to recognize open and closed hands. The goal of static hand gesture recognition is to categorize the given hand gesture data represented by various attributes into a finite number of gesture classes.

The goal of static hand gesture recognition is to categorize the given hand gesture data represented by various attributes into a finite number of gesture classes. The fundamental goal of this endeavor is to investigate the efficacy of two feature extraction methods, namely, hand contour and complex moments, in solving the hand gesture identification problem by identifying the primary benefits and drawbacks of each method. The back-propagation learning algorithm is used to construct an artificial neural network for classification. The suggested system includes a recognition algorithm that can recognize a set of six static hand gestures: open, close, cut, paste, maximize, and minimize. The image of a hand motion is processed in three stages: pre-processing, feature extraction, and classification. Some operations are performed during the preprocessing stage used to separate the hand gesture from its context and prepare the hand gesture image for feature extraction. The hand contour is employed as a feature in the first technique to tackle scale and translation issues (in some circumstances).

However, the complex moments algorithm is utilized to define the hand motion and treat the rotation. In addition to the scale and translation issues. Back-propagation learning is an algorithm. In the multi-layer neural network classifier. The findings indicate that hand contour technique has a recognition performance of (71.30%), but complicated moments have a greater performance. (86.90%) recognition rate performance.

A Pattern Recognition model for dynamic hand gesture recognition is proposed, which blends CNN with a weighted fuzzy min-max neural network. Based on the target's motion information, the model also performs feature extraction, feature analysis, and spatiotemporal template data encoding. The classifier's efficiency is boosted by combining a feature analysis technique with a weighted fuzzy min-max neural network. The results suggest that the proposed implementation can lessen the influence caused by feature point spatial and temporal fluctuation.

Artificial intelligence is closing the gap between human and machine capabilities. Researchers are working in a variety of sectors to achieve amazing things. Computer Vision is one such field. The goal of computer vision is to teach machines to see and experience the world in the same manner that humans do, and to utilize this knowledge to perform extraordinary tasks such as image recognition, image analysis and classification, video recognition, media recreation, natural language processing, and so on.

Convolutional Neural Network (CNN) is one such method that has played a significant part in the breakthroughs of computer vision and deep learning. CNN is a multi-layer neural network with a distinct design that is utilized for deep learning. The authors share their research on the process and methods of sign language recognition using Deep Learning in this paper. For image recognition, 3D CNN was utilized through Kinect sensor. The 3D CNN method was discovered. The most effective and accurate method was discovered to be 91.23% [6]. CNN is used to identify Indian sign language gestures. Selfie mode was utilized to capture the image a continuous sign language film in which a hearing-impaired person The sign language recognizer mobile would be operated by a person independently. Because the datasets were not available for mobile use, the scientists produced datasets with five subjects performing 200 signs in five distinct viewing angles against a variety of background situations. Various CNN architectures were constructed and tested, and the best recognition rate obtained on the dataset was 92.88%.

The multi-class SVM and k-NN classifiers are used to monitor seven gestures for post-stroke patients' residential rehabilitation. The gestures were demonstrated on seventeen young people. The k-fold cross validation approach was used to evaluate the results. The accuracy of the multi-class SVM and KNN classifiers was 97.29% and

97.71%, respectively. Because the datasets were not available for mobile use, the scientists produced datasets with five subjects performing 200 signs in five distinct viewing angles against a variety of background situations. Various CNN architectures were constructed and tested, and the best recognition rate obtained on the dataset was 92.88%.

The authors utilized a webcam to track a region of interest (ROI), which was hand motions. To monitor the discovered ROI, the kernelized correlation filters (KCF) technique is utilized. The image is enlarged and fed into deep CNN to recognize numerous hand gestures. Two deep CNN architectures are constructed, each adapted from AlexNet and VGGNet. This tracking procedure is performed indefinitely, and movements are identified until the hand is out of camera range. The training data set achieved a recognition percentage of 99.90%, whereas the test data set achieved a recognition rate of 95.61%. To assist disabled people, CNN and back propagation technologies are employed to recognize gestures. The system can understand and identify photos, which is extremely useful in a variety of situations.

The Adapted Deep Convolutional Neural Network (ADCNN) is proposed in this paper to recognize hand movements. Data augmentation is used to expand the amount of the dataset and improve the resilience of deep learning. In the presence of RELU and Soft-max, the pictures are fed into ADCNN, and L2 regularization is employed to remove overfitting. This strategy has been shown to be effective for recognizing hand movements. The model is first trained using 3750 photos with various variations in characteristics such as rotation, translation, scale, lighting, and noise. ADCNN had a 99.73% accuracy compared to baseline CNN, a 4% improvement over the baseline CNN model (95.73%).

V. PROPOSED FRAMEWORK

The CNN or convolutional neural networks are the most commonly used algorithms for image classification problems. An image classifier takes a photograph or video as an input and classifies it into one of the possible categories that it was trained to identify. The convolutional layer performs the hard computing processes and is the foundation of CNN. The parameters of the convolutional layer include a series of learnable filters. During the forward pass, we pass each filter across the height and width of the input volume, calculating the dot product between the filter's entries and the input at all places. When the filter is moved throughout the input volume, a 2D activation map is generated that shows the filter's responses at each spatial position.

CNN is a multi-layer neural network with a distinct design that is utilized for deep learning. A CNN architecture is made up of three layers: the Convolutional Layer, the Pooling Layer, and the Fully-Connected Layer. CNN is commonly used for object recognition, scene recognition, image detection, extraction, and segmentation. CNN has been widely used in recent years due to the three factors listed below: (1) CNN eliminates the need for feature extraction using image processing tools because it can directly learn the image data; (2) it is very good at recognizing outcomes and can be quickly re-trained for different recognition purposes; and (3) CNN can be constructed on an existing network.

The convolutional layer performs the hard computing processes and is the foundation of CNN. The parameters of the convolutional layer include a series of learnable filters. During the forward pass, we pass each filter across the height and width of the input volume, calculating the dot product between the filter's entries and the input at all places. When the filter is moved throughout the input volume, a 2D activation map is generated that shows the filter's responses at each spatial position. Each convolutional layer now has its own set of filters. Each layer will generate a separate activation map, which will be stacked with depth dimensions to construct the output volume. Depth, stride, zero padding and speed are three hyperparameters that influence the size of the output volume.

Depth is the number of filters that the user want to utilize. Each filter will try to learn something new from the input. The stride with which we slide filter is referred to as the stride. When stride is set to 1, filters move one pixel at a time; when set to 2, they jump two pixels at a time. It is sometimes helpful to pad input volume with zeros around the border. This is simply a zero-padding hyper parameter. Controlling the spatial size of output volumes is possible with zero padding.

The number of neurons "fit" is calculated as (W-F+2P)/S+1, where W is the input volume size, F is the receptive field size of the Convolution Layer neurons, S is the stride with which they are applied, and P is the amount of zero padding utilized on the border. Let's say we have a 7x7 input and a 3x3 filter with stride 1 and pad 0, and we obtain a 5x5 output. Pooling Layer is a frequent practice in convolutional neural network architecture to have a pooling layer between convolution layers. The pooling layer's main objective is to reduce spatial size in order to reduce hyperparameters and therefore network computation. This also addresses the overfitting issue. These layers operate independently on each depth slice of

the input and resizes them spatially using MAX operation. It is uncommon to pad the input of Pooling layers with zero padding. It is important to note that pooling units can also execute functions such as average pooling and L2-norm pooling. However, max-pooling has been shown to be the most effective in practice. Neurons in fully connected layers are totally connected to all activations in the prior layer. The activations can be calculated using matrix multiplication and a bias offset. Converting a fully connected layer to a convolutional layer stage; It is worth noting that the neurons in a convolutional layer are only connected to a local region in the input, and many neurons in a convolutional volume share parameters.

Because the neurons in both levels continue to compute dot products, their functional form is similar. As a result, it is able to switch between fully connected and convolutional layers. A convolutional neural network is made up of three layers: convolutional, pooling, and fully connected. In practice, the RELU activation function is also written as a layer.

A few convolutional-RELU layers are typically stacked, followed by pooling layers, and this pattern is repeated until the image has been spatially merged to a tiny size. The output is held by the final fully-connected layer.

The typical CNN architecture follows this pattern:
INPUT -> [[CONV -> RELU]]*N -> POOL?]*M -> [FC -> RELU]*K -> FC,

where * denotes repetition and POOL? denotes an optional pooling layer. Furthermore, N >= 0 (and N = 3), M >= 0, K >= 0 (and K 3). Figure 1 depicts a simple CNN architecture that accepts m*n*1 input. This input is routed via various layers, including Convolutional, Pooling, and ReLu, before reaching the fully connected layer, where the gesture in the image is identified.

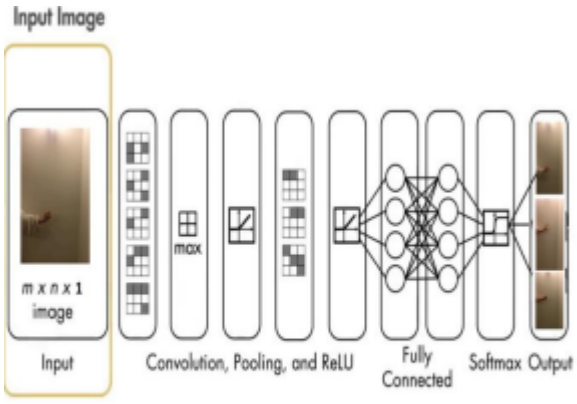

Fig. 1 Convolutional Neural Networks architecture

A. Input from Hand Gestures

Hand gestures constitute a total of 140 motions for twenty participants as an input to the gesture detection system examined and compared in this study. Figure 2 depicts three examples of twenty persons demonstrating seven 2-D and 3-D universal common hand gestures using three distinct mobile cameras, backgrounds, illumination, hand position, and hand form. The iPhone8 was used to record the first gesture, the Samsung Galaxy S10 was used to record the second gesture, and the iPhone8 was used to record the last gesture. The first background is light blue, the second is softly flowery, and the third is simple. The first example has less illumination than the second and third examples. The position of the hand, as well as the contour of the hands, change slightly. A first-hand gesture is for a young woman in her late twenties, another for a young woman in her mid-thirties, and the last for an elderly guy in his seventies. They are recorded across short distances and employed in the experimental activity of the study.

B. Specification of a Computing Platform

The experiment was carried out using a Dell desktop C2544404 with a processor generation Intel ® CoreTM i7-6700 CPU @ 3.40 GHz, memory type DDR4 16 GB, speed 2133 MHz, 512 GB storage hard drive, and a 24 inch (60.9 cm) display. Full HD (1920 1200) Ultrasharp IPS Panel display with Adobe RGB color space and touch. The operating system was Windows 10 (64 bits), and the system was built with the MATLAB R20187bV programming language.

C. Implementation of a Convolutional Neural Network

CNN is an essential component of deep learning since it is utilized to train data without the usage of any image processing software. For each movie, a new directory is established in this experiment. A total of 140 videos are read to generate 24,698 picture frames. CNN depicts the procedure for converting the image frame from RGB to grey and resizing it to 227227 from the original image size. Each video includes a varied number of frames ranging from 3394 to 3670. The image data is divided into two datasets: training and testing. The amount of training frames is 2485, which is 70% of the total. The CNN structure is composed of seven layers, each with the following capability and size: Image-Input-Layer input size [227,227,1], Convolution2-DLayer filter size [5,20], testing. The amount of training frames is 2485, which is ReLU-Layer (Rectified Linear Unit), MaxPooling2-DLayer pool size [2,2], Fully-Connected-Layer input and output sizes [auto], Soft-max-Layer and Classification-Output-Layer output sizes [auto]. CNN hyperparameters are generated within the training options function.

The epochs parameter is set to 50 epochs.

Figure depicts the system implementation processes using the framework model. The hand motions displayed in are recorded using distinct high-quality mobile cameras with two resolutions: HD and 4k. Each recording lasts between 2 and 10 seconds, and the video resolution varies. The false positive rate is measured by specificity. The PPV and NPV are percentages of positive and negative results in diagnostic and statistical tests that also describe genuine positive and true negative outcomes. The LR+ and LR are well-known diagnostic accuracy measures.

Fig. 2 Three examples of seven universal hand gestures for three hands

VI. DATA DESCRIPTION

Convolutional neural networks (CNNs) are deep learning technologies that are ideal for computer vision tasks. They can learn to extract features directly from raw photos as well as do classification [Siv+12]. They are comparable to neural networks in that they have neurons, weights, and biases, and they have one or more fully connected layers, as do neural networks with many layers, but they are quicker to train because they have fewer parameters.

A significant advantage of employing convolutional neural networks for computer vision tasks is that each layer learns new picture information. These characteristics can be utilized to train the classifier.

Convolutional neural networks are mostly deep learning models inspired by the way our cornea operates through the alternating of convolutional and pooling layers. They are trained feature detectors, thus they are quite adaptive. This is why they achieve the maximum accuracy in picture detection: they can learn low-level features from training data, like HOG and SIFT do. A convolutional neural network is composed of four different layers [Shoi+16] which are: Convolutional layer, Pooling layer, Non-linear layer, Fully-connected Layer. Convolutional layer: a collection of filters that slide across the image. They will be activated if they discover the same pattern in it. Pooling Layer: The goal of this layer is to lower the size of the space, the parameters, and the net calculations. There are several functions that can be utilized, but max pooling is the most frequent.

Non-linear Layer: In the design of a convolutional neural network, there are non-linear functions such as rectified linear units (RELU), Identity, Tanh, and Arctan that are used to introduce non-linearity into the neural network, making training faster and more accurate.

Fully-connected Layer: the neurons in this type of layer connect to every neuron in another layer like in neural networks.

VII. RESULT

Dealing with the visual background and noise that is typically present in regions of interest, such as the hand region, is one of the issues in gesture identification. The use of neural networks for morphological operations, along with a polygonal approximation, produced good results in terms of distinguishing the hand region from the background and removing noise. This phase is critical because it removes visual objects that are irrelevant to the classification approach, allowing the convolutional neural network to extract the most significant gesture characteristics via their convolution and pooling layers and, as a result, increasing network accuracy.

Hand gesture detection is critical for providing a natural HCI competence. The most important parts of gesture recognition are now understood to be detection, segmentation, and tracking. In this experiment, a system for hand gesture recognition was built utilizing the CNN technique for feature extraction and classification. Within short distances, different mobile cameras, backdrops, illumination, hand position, and hand form are captured.

Experiments were carried out to compare the performance of the CNN method's training and testing. The results demonstrated that training outperforms testing in terms of accuracy.

High Accuracy in Gesture Classification: CNNs are well-known for their capacity to learn hierarchical features from data automatically. This correlates to great accuracy in distinguishing diverse motions in hand gesture recognition.
Robustness to Variability: CNNs are built to be resistant to changes in input data, such as changes in lighting, hand orientation, and background clutter. A well-trained model should perform effectively in a variety of settings.
Adaptability to New Gestures: Depending on the training approach, CNNs can be trained to adapt to new gestures with little retraining. For example, transfer learning algorithms enable the model to use knowledge obtained from one set of motions to recognize new ones.
User Independence: A well-designed CNN should be able to recognize different users' hand motions while allowing

for differences in hand shapes, sizes, and appearances. The model should be able to generalize well across a wide range of users.
Precise Hand Region Localization:
CNNs may be trained not just to recognize gestures but also to precisely locate the hand region in an image. This is necessary for correctly interpreting and recognizing gestures.

Adaptability to New Gestures: Depending on the training approach, CNNs can be trained to adapt to new gestures with little retraining. For example, transfer learning algorithms enable the model to use knowledge obtained from one set of motions to recognize new ones.
Training accuracy is accomplished by running a model on the training data and determining the algorithm's correctness. It was discovered that the execution time of training and testing is the same. When compared to testing, the accuracy result of training is 100%. Sensitivity in training is slightly higher than in testing. In training, the specificity is 100%, however in testing, it is 0.9989. Testing has a lower PPV and NPV than training. The best LR+ and LR values are saved for future reference.

RMS recognizes the importance of both training and testing.

A. Cross entropy loss function training

We utilized CNN for training, which consists of several layers with varying features. We chose 20 epochs since they are large enough to offer consistent results, and a higher number of epochs may result in a longer computation time. We picked a batch size of 64 since small batch sizes perform well in general and can provide lesser generalization error.
We used Adam optimizer since it is efficient and performs well on noisy tasks. The learning rate utilized for optimization was 0.001, because it aids in achieving more steady training while not being too tiny to result in failure to train. We used cross entropy loss as the loss function since our problem is a multi-class classification problem

B. Experimentation: Following training, we predicted the class of test images and compared them to their true classes, evaluating our model based on the number of successful and failed predictions.

C. Metrics: The major criterion utilized to assess the performance of our model was validation accuracy. It is generated automatically during the training and validation of our model. The percentage of correct classifications over all categories is referred to as validation accuracy. In addition to validation accuracy, we evaluated our model using testing accuracy and a confusion matrix. After we evaluated our model with the testing set, we calculated correct predictions over all predictions to generate testing accuracy. The confusion matrix gives us an overview of our prediction outcomes, which allows us to improve our model.

D. Outcomes: The results demonstrated that the model works well. Our model's validation accuracy was 0.9999, its validation loss was 0.00003, and its test accuracy was 0.999875.

E. Deliberations: Our model performs virtually flawlessly overall; however, because our validation loss stopped reducing after several epochs, it is quite likely that overfitting occurred during training. Overfitting is likely because the dataset is small and half of our dataset was produced by flipping the other half of our dataset. To avoid this in the future, we can train the model with a larger sample size and a more diversified dataset. The model projected a 'thumb' as a 'fist', which was incorrect. It makes sense because a 'thumb' is a 'fist' with the thumb pointing out. It is quite difficult for our model to distinguish between them, and some incorrect predictions are likely in this circumstance.
This study investigates the opportunities and limitations in

hand gesture detection. It also investigates the impact of data augmentation on deep learning. We may conclude from this study that CNN is a data-driven methodology, and data augmentation has a significant impact on deep learning. Although the technology can effectively recognize gestures, some extension is still possible. For example, using knowledge-driven methodologies such as Belief Rule Base (BRB).

As a result, gesture recognition can be conducted more precisely. More gestures can be added to the list of recognized gestures. The backdrop was thought to be less complicated. CNNs were utilized to recognize vision-based hand movements. The proposed network eliminates the need for pre-processing illumination variation, rotation, and hand region segmentation. The depth thresholding technique made hand region segmentation easier even in the presence of human noise and complicated backdrops. Furthermore, the suggested technique accurately distinguishes the majority of closely comparable gesture postures, which improves recognition performance. The proposed technique is also employed in real time recognition of ASL gesture poses. Hand Gesture Identification Input from an RGB Sensor.

Several strategies for recognizing hand motions in vision-based environments have been developed. Some studies employed hand-crafted features followed by classification, while others used convolutional neural network techniques that combined feature extraction and classification in a single network. The results of a recent survey on these strategies are shown below. A CNN network based on feature fusion was presented. The CNN structure is fused at the final fully linked layers to generate a final gesture class. A unique two-model hierarchical architecture for real-time hand gesture recognition systems is presented in this study. The suggested architecture offers resource efficiency, early detections, and single-time activations, all of which are important for real-time gesture recognition applications.
The suggested method is tested on two dynamic hand gesture datasets and yields similar results for both. We have proposed using a new metric, Levenshtein accuracy, for real-time evaluation because it can quantify misclassifications, multiple detections, and missing detections all at the same time.
Furthermore, we used weighted-averaging on the class probabilities over time, which enhances overall performance while also allowing early identification of motions. Using the difference between the top two average class probabilities as a confidence measure, we obtained single-time activation per gesture. However, as a future effort, we would like to investigate more on statistical hypothesis testing for the confidence measure of single-time activations.
Convolutional neural networks were created primarily to derive conclusions from visual input such as photos and movies.

The features are retrieved and trained to train the model, which results in higher recognition accuracy than traditional Machine Learning techniques. CNNs have extensive applications in signal processing, robotics, medical imaging, data analysis, business intelligence, and other fields. Learning using CNNs from an unchanged and smaller dataset produces remarkably better outcomes.
The Image Database provides images of many hand signs. These photos are several repeats of images collected from different users. The resolution of the photos can vary. There are several datasets available for American Sign Language.

Image Pre-processing: Training raw photos as is may result in poor results. As a result, simple image processing techniques can be used to obtain maximum accuracy. Image processing technologies like RGB to grey conversion shorten training time and power usage. The noise in the photos can be removed.

Image Augmentation: In the case of a small database, data augmentation comes in handy. Image augmentation is accomplished using a variety of procedures, including mirroring (flipping the image horizontally). Cropping is the process of removing a section of an image. Color shifting - For RGB datasets, the pixel values can be changed by rotating, shearing, or local warping.

CNN Training & Training Alternatives: The project makes advantage of deep learning. Before training the database with any CNN architecture, the training options are appropriately established. Maximum batch size, number of epochs, and learning rate are the training options.

Image Acquisition: To acquire the image to be recognized, any camera, even a laptop webcam, can be utilized. Because the acquired image will eventually be reduced to the CNN's input size. As a result, a high-resolution camera is not required. Display Output: The recognized sign can is displayed in text format or can be also conveyed with audio description.

A five-fold cross validation is used to assess the classification's performance. The collection is partitioned into five subsets, each with 40 sample photos from a different gesture class. The classifier is trained using any four subsets and tested with the remaining one subset. The experiment is performed five times in a row until each

subset is used for development and testing. This shows the classification result based on the average accuracy, precision, recall, and F1-score values. For determining the values of the performance measures, the macro averaging approach was used. The table depicts the proposed CNN model's improved recognition capacity.

VIII. REFERENCES

[1] A. Kojima, M. Izumi, T. Tamura, and K. Fukunaga, "Generating natural language description of human behavior from video images," in Int. Conf. Pattern Recog., vol. 4. IEEE, 2000, pp. 728–731.

[2] C. J. Cohen, F. Morelli, and K. A. Scott, "A surveillance system for the recognition of intent within individuals and crowds," in IEEE. Conf. Technol. for Homeland Secur. IEEE, 2008, pp. 559–565.

[3] S. Mitra and T. Acharya, "Gesture recognition: A survey," IEEE Trans. Syst., Man, Cybern. C, vol. 37, no. 3, pp. 311–324, 2007.

[4] C. Vogler and D. Metaxas, "ASL recognition based on a coupling between hmms and 3d motion analysis," in Int. Conf. Computer Vision. IEEE, 1998, pp. 363–369.

[5] C. F. Bond Jr, A. Omar, A. Mahmoud, and R. N. Bonser, "Lie detection across cultures," J. nonverbal behav., vol. 14, no. 3, pp. 189–204, 1990.

[6] H. I. Lin, C. H. Cheng, and W. K. Chen, "Learning a pick-and-place robot task from human demonstration," in Proc. Int. Conf. Automat. Control. IEEE, 2013, pp. 312–317.

[7] T. Starner, J. Weaver, and A. Pentland, "Real-time american sign language recognition using desk and wearable computer based video," IEEE Trans. Pattern Anal. Mach. Intell., vol. 20, no. 12, pp. 1371– 1375, 1998.

[8] J. Davis and M. Shah, "Visual gesture recognition," in IEE Proc. Vision, Image and Signal Process., vol. 141, no. 2. IET, 1994, pp. 101–106.

[9] M.-H. Yang and N. Ahuja, "Recognizing hand gestures using motion trajectories," in Face Detection and Gesture Recognition for Human Computer Interaction. Springer, 2001, pp. 53–81.

[10] K. Oka, Y. Sato, and H. Koike, "Real-time tracking of multiple fingertips and gesture recognition for augmented desk interface systems," in IEEE Int. Proc. Automat. Face and Gesture Recog. IEEE, 2002, pp. 492-434.

[11] A. A. Argyros and M. I. A. Lourakis, "Vision-based interpretation of hand gestures for remote control of a computer mouse," in Computer Vision in Human-Computer Interaction. Springer, 2006, pp. 40–51.

[12] K.-H. Jo, Y. Kuno, and Y. Shirai, "Manipulative hand gesture recognition using task knowledge for human computer interaction," in IEEE Int. Conf. Automat. Face and Gesture Recog. IEEE, 1998, pp. 468– 473.

[13] R. Cutler and M. Turk, "View-based interpretation of real-time optical flow for gesture recognition," in IEEE Int. Conf. and Workshops on Automat. Face and Gesture Recog. IEEE Computer Society, 1998, pp. 416–416.

[14] S. J. Nowlan and J. C. Platt, "A convolutional neural network hand tracker," Advances in Neural Inf. Process. Systems, pp. 901–908, 1995.

[15] M. K. Hu, "Visual pattern recognition by moment invariants," IRE Trans. on Information Theory, vol. 8, no. 2, pp. 179–187, 1962.

[16] Fang, L.; Liang, N.; Kang, W.; Wang, Z.; Feng, D.D. Real-time hand posture recognition using hand geometric features and fisher vector. *Signal Process. Image Commun.* **2020**, *82*, 115729.

[17] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," The Journal of Machine Learning Research, vol. 15, no. 1, pp. 1929–1958, 2014.

[18] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in Proceedings of the fourteenth international conference on artificial intelligence and statistics, 2011, pp. 315–323.

[19] R. A. Dunne and N. A. Campbell, "On the pairing of the softmax activation and cross-entropy penalty functions and the derivation of the softmax activation function," in Proc. 8th Aust. Conf. on the Neural Networks, Melbourne, vol. 181. Citeseer, 1997, p. 185.

[20] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in Proceedings of COMPSTAT'2010. Springer, 2010, pp. 177– 186.

[21] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," arXiv preprint arXiv:1712.04621, 2017.

[22] R. U. Islam, M. S. Hossain, and K. Andersson, "A novel anomaly detection algorithm for sensor data under

uncertainty," Soft Computing, vol. 22, no. 5, pp. 1623–1639, 2018.

[23] M. S. Hossain, S. Rahaman, A.-L. Kor, K. Andersson, and C. Pattinson, "A belief rule based expert system for datacenter pue prediction under uncertainty," IEEE Transactions on Sustainable Computing, vol. 2, no. 2, pp. 140–153, 2017.

[24] M. S. Hossain, F. Ahmed, K. Andersson et al., "A belief rule based expert system to assess tuberculosis under uncertainty," Journal of medical systems, vol. 41, no. 3, p. 43, 2017.

[25] M. S. Hossain, P.-O. Zander, M. S. Kamal, and L. Chowdhury, "Beliefrule-based expert systems for evaluation of e-government: a case study," Expert Systems, vol. 32, no. 5, pp. 563–577, 2015.