IJNRD.ORG ISSN : 2456-4184

INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG

An International Open Access, Peer-reviewed, Refereed Journal

HAND GESTURES RECOGNITION USING CNN

Muvva Pratap Kumar

Department of Computer Science and Engineering
Vignan's Foundation for Science, Technology & Research (Deemed to be University), Vadlamudi, Guntur, Andhra Pradesh

Manyam Teja

Department of Computer Science and Engineering
Vignan's Foundation for Science, Technology & Research (Deemed to be University), Vadlamudi, Guntur, Andhra Pradesh

Maruri Sivaramalingareddy

Department of Computer Scienceand Engineering
Vignan's Foundation for Science, Technology & Research (Deemedto be University), Vadlamudi, Guntur, Andhra Pradesh

Dr. Susanta kumar Satpathy

Department of Computer Science and EngineeringVignan's Foundation for Science, Technology & Research (Deemed to be University), Vadlamudi, Guntur, Andhra Pradesh

Abstract:

Communication aids that are apparent to the eye include hand signals and nonverbal clues. Deaf and dumb persons, for example, will be allowed to freely speak to others without disabilities because to this effort. Convolutional neural networks, also known as CNNs, are a viable method for recognizing hand gestures because they have shown great success in recognition of images tests. In this research, we propose a CNN-based hand gesture detection model that performs well in real-time and with high accuracy. The correctness of the paper's conclusion was 94%.

Because of its uses in interaction between humans and computers, spoken language interpretation, and robotics, hand gesture recognition is a rapidly expanding topic of study. The most common and crucial method of communication in our world today are hand gestures.

Index Terms: **Deep learning; Convolution Neural Networks; Hand Gesture Recognition.**

Introduction

Recently, robotics and artificial intelligence have been employed to assist people with disabilities become more independent. Increasing the enjoyment by allowing people to execute a larger range of tasks, daily duties more effectively is the major goal in this scenario. For many application domains, including sign language recognition (SLR), the ability to recognize hand motions has been viewed as a benefit. In sign languages, complex hand signs are used, but even small various meanings can be expressed using hand gestures. The conclusion is that over the span of ten years, numerous vision-based, evolving hand motion detection strategies have been developed.

Similarly, it was discovered that combining CNNs that had been taught with two independent channels of unmodified and geographically cropped video frames generated the best results when categorizing enormous volumes of footage.

The importance of giving CNNs an extensive array of training examples has been stressed by many authors. They have suggested data augmentation tactics to stop CNNs from overfitting while utilizing samples with a modest amount of variability. Before being sorted into 1000 categories, both training and test pictures were altered, horizontally turned, and RGB jittered. Simonian and Zisserman trained CNNs for video-based human behavior recognition using same spatial augmentation techniques for every frame of the video. However, all techniques for data augmentation were geographically restricted variances. We present a technique for recognizing hand gestures in this research that employs convolutional neural networks with two dimensions for prediction and learning and hand component extraction from photos. To prevent overfitting and improve the generalization of the gesture classifier, we provided a potent spatiotemporal data augmentation strategy. The complexity of sign language's grammar, the variety of symbols produced by different signers, and the speed and fluidity of signing all make detection challenging. Conventional methods for sign language proof of identity, such as rule-based or template-matching techniques, have limitations when addressing these issues. However, recent advances in computer vision and deep learning have increased the precision of gesture detection.

CNN LAYERS

A.DATA PREPROCESSING

The raw photos used in this investigation are converted into grayscale versions. The maximum value of the grey level range is used to equalize the grey levels of the input images. Low-resolution images are used to speed up training without significantly lowering recognition rates. 64 * 64 pixels is the new size of the photos.

Pre-processing refers to all the modifications performed to the initial information before submitting it to a machine learning or neural network algorithm. A convolutional neural network, for instance, will typically produce inferior classification results when trained on raw pictures (Pal & Sudeep, 2016). Pre-processing is also helpful for accelerating training (for instance, cantering and scaling methods).

B.CNN MODEL DESCRIPTION

The total number of neurons and layers of convolution in a CNN's architectural design are carefully chosen to ensure optimal performance. The total number of neurons or layers of convolution could chosen according to no established guidelines. In this article, we have offered a CNN design that maximizes recognition accuracy. It was known as SLRNet-8. Our suggested SLRNet-8 additionally includes a layer that is completely connected, three layers for pooling, and six layers of convolution, as well as the layers for input-output. Figure 1 depicts the key planned ASL recognition in its early stages.

Input Layer

In this work, grayscale images are created from the original color photos. The highest value in the range is used to normalize the input pictures' grey levels. Low-resolution images allow for quicker training without having a significant influence on recognition rates. The pictures have been downsized into 64*64 pixels.

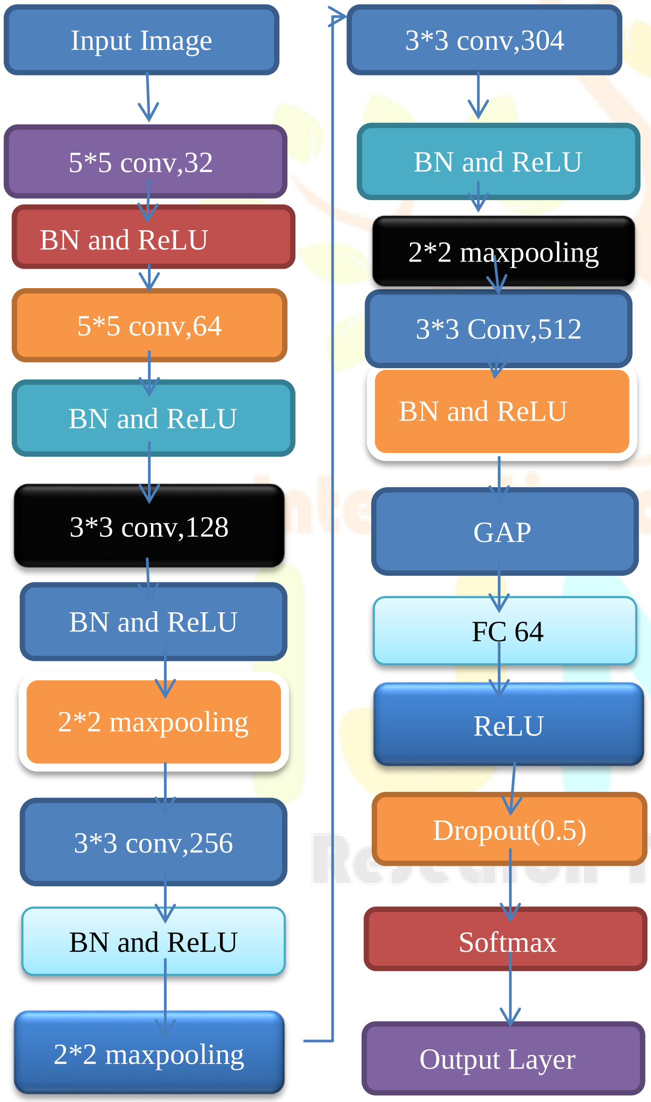
CONVOLUTION LAYER

As the essential CNN building block, Convolution serves as the first layer in the process of characterizing input data. The forward and backward propagation used by the kernels in the convolution layer allows them to identify the key features in the input data. The filters of dimensions 33 and 55 are moved over the input data matrix to execute this operation in our investigation. The element-wise matrix is executed at each shifting.

The nonlinear element of a CNN model's performance may be influenced by the amount of kernels utilized therein. The number of kernels in the layer of convolution can be chosen without following any predetermined rules. In this study, we investigated different kernel combinations, that ranged from 32 to 512, in various steps sizes, and finally selected the one that maximized accuracy. A convolution layer comes before the group level of normalization (BN) [17], which is accountable for speeding training and minimizing internal covariate shift..

ACTIVATION FUNCTION

Activation functions in a CNN design determine which nodes in the network must be fired at a particular time. Using an activation function for ReLU, we have preserved each positive value constant while converting every negative value to zero. The model's learning phase led to the selection of ReLU. ReLUs often train more quickly than their counterparts (soft plus, tanh, and sigmoid), which helps lessen the risk of the gradient disappearing. ReLU(y) = max (0, y), where y represents an input to a neuron, is the formula for the ReLU function



POOLING LAYER

After convolution, the geographical aspect of the picture being processed utilizes a pooling layer to decrease. In the area it is used between two convolution layers. Without using pooling or maximum pooling, making use of FC following the Convo layer will be computationally costly, which is what we do not want. Therefore, the only technique for decreasing the source image's spatial dimension is maximum pooling. In the aforementioned illustration, only one deeper cut with a stride of two employed maximum pooling. The four dimensions of the input are visible have been reduced to only two. There are no parameters in the pooling layer, but there are two hyperparameters: filter (F) and stride (S). Generally speaking, if the input dimensions are W1, H1, and D1, then

W_2= (W_1−F)/S+1

H_2 = (H_1−F)/S+1

D_2 = D_1

W_2, H_2, andD_2 stand for the output's height, width, and depth.

FULLY CONNECTED LAYER

Fully linked layers contain neurons, weights, and biases. Through this, neurons in a particular layer are connected to cells in a different layer. It is used to teach people how to divide images into several groups.

SOFTMAX/LOGISTIC LAYER

CNN's SoftMax, or operating layer, is the top layer. It is situated at the FC layer's base. While linearity is used for multi-classification, SoftMax is used for binary categorization.

DROPOUT LAYER

A dropout is a normalization technique that, with a certain probability, randomly sets input elements to zero. The overfitting problem appears when an algorithm's training accuracy is disproportionately more accurate than its evaluation. In The dropout layer following the FC layer in CNN models enables the prevention of the over-fitting issue and improves performance by arbitrarily lowering activation to zero throughout the training phase [20], [21]. A likelihood of dropouts of 0.5 was used in this experiment.

OUTPUT LAYER

At this layer, the classification model's output, or the projection of a class with a particular probability, is obtained. The chosen class should have the highest likelihood, as was previously stated. For each category, we list the entire number of cells located in each resulting layer. The SoftMax function calculates the odds of each class in a classification issue with many classes, with the wished-for class having the highest likelihood. For I = 1, 2, and 3, e(X)i = exj / exk, while xi denotes the values provided K is the total number of classes transmitted to every maximum level of flexibility from the previous FC layer.

Literature Review

Sh. M. Autee and Vitthal K. Bhosale [1] presented a technique in which the angle and peak computation method is used to identify the features of hand gestures using MATLAB, and the recognized gesture is then converted into speech using a built-in command in MATLAB. Sangeetha. In contrast to American sign language, which only uses one hand to make a gesture, R.K., Valliammai. V., and Padmavathi. S. [2] have developed a method based on Indian hand sign language. Without any additional hardware requirements for the consumer, their system is constructed using MATLAB. Runtime live images are taken, followed by the extraction of image frames and application of image processing using the HIS model, before features are extracted using the distance transform approach. For the majority of the hand symptoms and outcomes produced by this technique are determined to be satisfactory.

The Harris method is used in Anchal Sood and Anju Mishra's [3] recommended system for recognizing signs, which obtains the feature after the photo pre-processing stage and saves it in the matrix Nx2. Then, this matrix is used to match the database image. The system has a number of limitations. An exceedingly pale brown to relatively dark brown causes errors. Background because they fall outside of the acceptable range for skin segmentation. However, the outcomes are productive.

Tejaswini A. Jawake, Prashant G. Ahire, Kshitija B. Tilekar, and Pramod B. Warale A real-time video was used as the input for the [4] system, which uses MATLAB for hand gesture detection. Following the image processing step, the correlation-based a technique was used for mapping, and finally, the Google TTS API was used to generate the sounds. According to the system's recommendations, the system produces an effective result.

A system built around the hardware approach has been proposed by Mrs. Neela Harish and Dr. S. Poonguzhali [5]. The system is made up of hardware known as a data glove, which includes the sensory part made up of flex detectors, an accelerometer, and a PIC microcontroller that handles the system's input and output. The outcomes are effective and fulfilling. A method based on hand recognition of actions has been developed by Sonal Kumar and Suman K. Mitra [6] using the background removal approach for processing images and the direct Fourier transformation (DTE) algorithm for picture extraction using MATLAB.

.

I. Methodology:

In order to distinguish dynamic hand motions, I used a CNN classifier. For tasks like image identification and data pixel processing, a CNN, a particular type of deep learning it uses network design. Although there are numerous types of neural networks accessible, CNNs are the preferred network design for deep learning to help in object recognition and differentiation. Because precise object identification is so important in scenarios such as recognizing faces and self-driving cars, they are perfect for machine vision (CV) tasks.

INSIDE CONVOLUTION NEURAL NETWORK:

Deep learning techniques heavily rely on artificial neural systems (ANNs). Recurrent neural networks (RNN), a type of ANN, accepts input in the form of time series data or sequence data. It is appropriate for the processing of natural languages (NLP) applications, speech recognition, translation of languages, and captions for images.

Another synthetic neural network is CNN is capable of locating both series and picture data of importance. It is therefore very useful for picture-based applications including object classification, pattern recognition, and image identification. A CNN uses linear algebraic concepts like to determine, use matrix multiplication patterns in an image. CNNs may also classify audio and signal data.

The connections between the framework of an individual's A CNN's framework is comparable to the human brain. Similar to the brain, CNNs have billions of neurons, but they are structured differently. In reality, a CNN's cell arrangement is similar to the frontal lobe of the brain of a person, which processes visual stimuli. This arrangement eliminates the issue with traditional neural network partially image processing, which requires delivering users pictures in low-resolution pieces. It makes sure that every square inch of the eye is covered. Compared to earlier networks CNN works better with picture, speech, or audio signal inputs.

Thorough education, a layer for convergence, an additional one for accumulation, and CNN is composed of three layers, each of which is a fully connected (FC) layer. Convolutional layer is the first layer, and FC layer is the last layer.

At the layer of convolution to the FC layer, the CNN's level of granularity increases. Because of the increasing complexity, CNN can discriminate among ever-larger and more complicated pieces of visual data until it can correctly identify the entire item.

Most of simulations occur in the layer referred to as the con, and this is the main part of a CNN. An additional layer of convolution could come after the initial layer. During the convolution process, the center or filters in this layer travels across the picture's receptive regions to detect whether an element is present.

The majority of computations happen in the layers of convolution, known as the core component of a CNN. Additional layer of convolution could be added after the first layer. During the convolution process, the kernel of computing or filtering in that layer traverses the picture's receptive regions to assess the presence of a feature.

Over several repetitions, the kernel exhaustively scans the entire image.

The completely linked layer (FC): In CNN's FC layer, pictures are categorized based on the traits that were retrieved from the layers above. Fully connected in this context means that all inputs and activation units forms the preceding layer are connected to all inputs from the one above.

All of CNN's tiers are interconnected because doing otherwise would create a network that is far too thick, which CNN does not have. More losses would result, the computation would be highly expensive, and the output quality would suffer.

An explanation of how CNN works is provided below. A CNN may contain numerous layers, so each layer instructs the CNN how to detect different aspects of the processing an image. Each image receives An algorithm, or kernel, programmed to provide a result that gets better and more accurate with each layer. It's possible that the filters start out as basic characteristics at lower layers.

Each layer of the filter adds to its complexity by examining and detecting characteristics that specifically mirror the input item. As a result, the final product of each layer, also referred to as the convolved image, serves as the input for the layer that follows. The CNN may distinguish an illustration or object on the last layer, called a layer of FC. Convolution involves applying a number of filters on the input image. Each one of the filters does its task by casting light on a particular region of the image, then sending its outcomes to the processing on the layer underneath it. The methods involve carrying out a very large number, if not an infinite number, of layers because every coating has the capacity to recognize various traits.

The primary flaw of conventional neural networks is that they're unable to scale. A classical NN might yield useful results for smaller images with fewer colour channels. However, as an image's dimensions and level of detail increase, so does the need for resources and processing power, requiring a larger, more expensive NN.

Additionally, as time goes on, overfitting, where the neural network (NN) tries to learn excessive details from the data used for training, becomes a problem. Additionally, it can ultimately come to understand how the data's noise influences the way it performs on test datasets. In the final analysis, the NN fails to distinguish the item itself from the traits or trends in the data set.

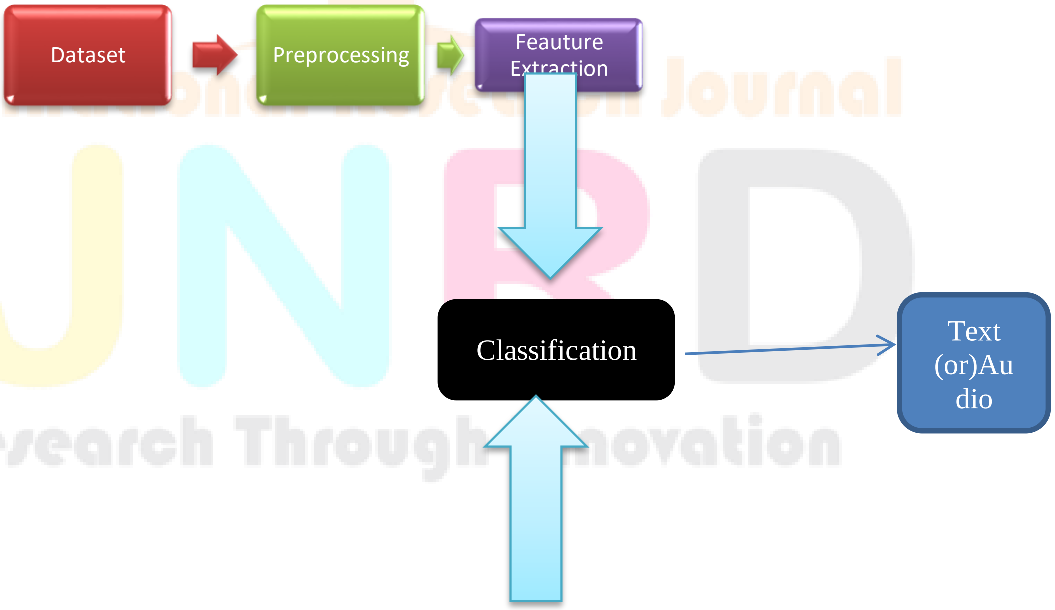
Contrarily, a CNN makes use of parameter sharing. Every layer and node are interconnected. As the level filters are applied to the image, a CNN likewise has a corresponding weight; this process is known as parameter sharing. As a result, processing time for the complete CNN system is lower than for a NN system.

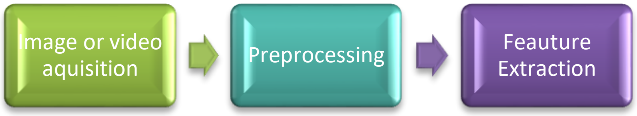
Benefits of using CNN for deep learning:

"Deep learning" is a subset of artificial intelligence that uses neural network models with a minimum of three distinct levels. A network with multiple layers can generate more accurate results than one with a single layer. RNNs and depending on the application, deep learning uses either or both CNNs. Because they yield extraordinarily precise results, CNNs are especially useful for picture recognition, classification, and computer vision, or CV, applications. This is especially true when there is plenty of data at play. The CNN continuously picks up the details of the object as it moves through its many layers. This immediate (and deep) training (feature engineering) eliminates the need for laborious feature extraction. On top of pre-existing networks, CNNs can be created and trained for new recognition tasks. These benefits bring up new opportunities for utilizing CNNs in actual-world contexts without raising costs the complexity or expense of processing.

As was already said, since CNNs use parameter sharing, they are more computationally efficient than ordinary NNs. The models run on any device, including smartphones, and are simple to deploy.

In order to do this, we pre-processed the dataset first by handling missing values, transforming categorical data into numerical information, and scaling the data. Each of the three artificial intelligence models was then trained using the initial training data after we divided the set of data into testing and training sets.





Architecture:

Data collection:

We collected datasets from up to 26 letters in the alphabet of English, each letter takes off nearly 100 images to recognize well and it able to predict the output based on the data.

Training Dataset:

A set of data used to develop a machine learning model is called a training dataset. It is often a subset of a bigger dataset that includes instances labelled with the right outcomes that train the model to make precise predictions.

Using the data set used for training, the parameters of the model are changed during the learning stage with the goal of minimizing the difference between the expected and actual outcomes of the training dataset. The accuracy and performance of the model can be evaluated after it has been trained using a new set of data, called a validating or test dataset.

The performance of the model can be greatly influenced by the quantity and caliber of the training data. Larger data sets can produce a model that is more accurate, while less information can result in overfitting, where a model is highly tuned to the data used for training and does poorly on fresh data. A training dataset with exact and representative samples can also help to prevent biases and errors in the model's predictions.

Feature Extraction:

Choosing and extracting relevant traits from a data set to feed a machine-learning system is referred to as feature extraction. Finding important patterns or links in the information and presenting them in a way that an algorithm can understand them are required.

A feature is a quantifiable attribute or characteristic of the data that may be used to separate one class or category from another in the context of machine learning. The color, texture, and shape of objects in an image collection, for instance, are examples of features.

For feature extraction, a variety of techniques can be utilized, including statistical techniques, signal processing, and algorithms for machine learning themselves. Common approaches include the analysis of principal components (PCA), linear discriminant evaluation (LDA), and convolutional neural networks (CNNs).

The goal of feature extraction is to reduce the complexity of the data set while keeping the most important information. By selecting and changing the most relevant data, feature extraction can improve the precision and efficacy of algorithms for machine learning while also making them easier to understand and analyze.

Data Pre-processing: Preparing raw data for analysis and modelling is known as data pre-processing, and it is an essential stage in machine learning. We carried out many data pre-treatment operations in our code, such as data cleaning and normalization.

Classification:

Using a collection of training data comprised of labelled examples, In machine learning, classification is an example of supervised learning, with the aim of predicting the classification or grouping of an input.

When classifying data, a set of features or attributes that characterize its properties are used to represent the incoming data. To find patterns or correlations between the features and the appropriate output classes, the algorithm then learns from the labeled training data.

Alternative methods for categorization include support vector regression, neural networks, decision trees, logistic regression, and k-nearest neighbours. The chosen algorithm is influenced by the specific problem at hand as well as the characteristics of the data.

There are many uses for classification, including sentiment analysis, spam filtering, fraud detection, and medical diagnosis. It is a crucial and extensively used technique in artificial intelligence and machine learning, and it is crucial for many real-world applications.

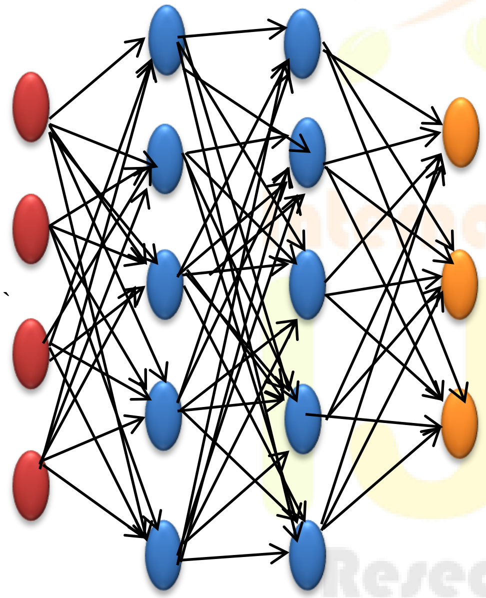
Image or Video acquisition:

Using cameras or other devices, visual data is captured through the process of image or video acquisition. This information may take the shape of still photos, video frames, or other visual files.

To acquire an image, light that bounces off of objects must normally be captured and transformed into a digital signal that a computer can understand. Cameras that use lenses to focus light onto a digital sensor or film are often used for this. Computer vision techniques can then be used to process and analyze the collected images.

Similar to image acquisition, video acquisition records a series of images at a high frame rate as opposed to only one image. A video that depicts motion and changes over time can be made using this series of photos.

Many different applications, including surveillance, security, entertainment, academic research, and medical imaging, use image and video acquisition. It is crucial to properly regulate acquisition factors like lighting, focus, and exposure because the caliber of the acquired photos and videos can significantly affect how well computer vision algorithms work.

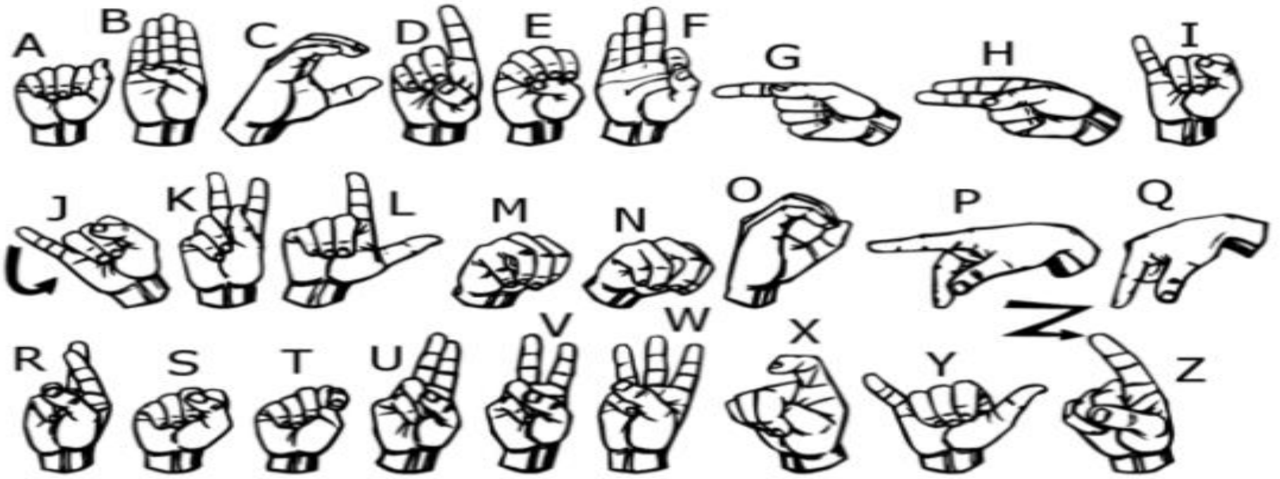


A. DATASET:

In order to assess the model, I collected 2600 photos of 26 hand movements using a webcam. 50x50 pixels make up each image. By deleting colour skin cells from the colour image, black and white skin cells are produced. These black-and-white photos are scaled down to 50 x 50 pixels.

Images related to each hand gesture are arranged into a separate folder. There are entries for each image inside each folder in a text file. The text file contains a reference to one of the finger movements shown in the image. Using spatial data augmentation methods, I was able to add an additional 4,000 pictures to this dataset. This section goes into more depth about the method.

Each hand gesture's associated images are organized into a different folder. A text file contains information for each image contained in each folder. The text file contains a reference for one of the fingers movements shown in the image. In addition to this dataset, I also collected an additional 4000 pictures utilizing spatial-temporal data augmentation techniques. Section goes into further depth on the method.



CLASSIFIER:

The network has six 2D layers of convolution, with a max-pooling operator coming after each layer. Fig. 2 shows the amount of space at each layer, the sizes of the convolution kernels, and the pooling operators. The sixth convolution layer's output is sent into a nine-layer, completely linked network. Each layer contains 512 hidden neurons, with the exception of the final output layer, which has nine neurons (one for every one of the nine hand movements). A function with a sigmoid activation is used in the output layer. The Tanh activation feature is used on the remaining eight levels.

B. BATCH NORMALIZATION:

A recent technique called batch normalization (BN) [20] normalizes every single set of data across each layer.

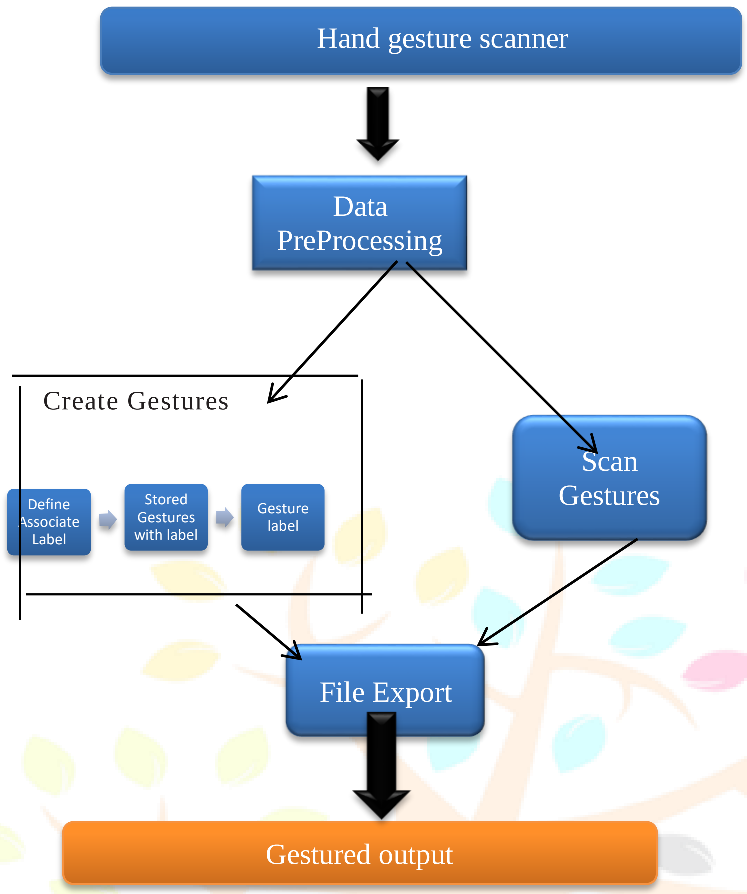
C. TRAIINING:

Network variables must be optimized as part of a CNN's training process in order to lower the cost curve for the dataset. I decided on mean squared error as the cost function.

To optimize, we applied stochastic gradient descent. I changed the network's parameters each time using the nesterov accelerating gradient. I initialized the weightings of the 2D layers of convolution using random samples. The following subsections provide more information on these words:

If the cost curve did not improve by more than 10% in the prior 40 epochs, I reduced the learning rate by a factor of 2 and restored the training speed to 0:005 for fine-tuning. Once the rate of learning has slowed down by a minimum of four occasions, or if the amount exceeds four of a CNN's training process in order to lower the cost curve for the dataset. I decided on mean squared error as the cost function.

WORKING:


Hand gesture scanner
Data PreProcessing
Create Gestures
Define Associate Label
Stored Gestures with label
Gesture label
Scan Gestures
File Export
Gestured output

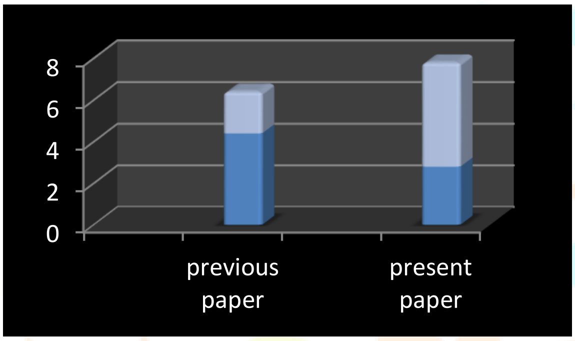
We have collected the datasets initially, and through the webcam access the hand will be captured by the camera the web camera separates the images from the background separately, and the data is pre-processed. To create a gesture it was divided into three sets one is to define the label and the next one is of standard gesture with a label and the final is the gesture label which is off combination of the before labels and the next part is off scanning the gestures. The scanning gesture will be compared to the datasets that we have gathered, and it generates text for the hand motion that can also be converted into voice after being provided as text so that the software can speak the type of output it produced.

Discussions:

With the use of references from other publications and the addition of a few new features, we were able to complete the project with greater precision than the others' and with greater accuracy than their work. The project we have done is well advanced and we implemented it with some new features and it's completely different from the existing ones.

Results:

We have done the project with more accuracy and compared to the existing project we have added some of new features.



CONCLUSSION:

With the help of 2D convolutional neural networks, we have created a potent technique for recognizing dynamic hand motions. To prevent overfitting, the suggested classifier augments the data with spatiotemporal information. We have demonstrated through careful analysis that integrating low- and excellent quality sub-networks significantly increases classification accuracy. We've also demonstrated how essential the recommended data augmentation technique is for achieving better performance. The validation accuracy of our proposed solution for the dataset was 98.2%. In the future, we will investigate robust classifiers that can classify higher-level dynamic gestures, such as occupations and movement contexts, as well as more flexible hyper-parameter selection for CNNs.

References:

1. Neil Buckley, Lewis Sherrett, Emanuele Lindo Secco.A CNN sign language recognition system with single & double-handed gestures.
2. Varsha M, Chitra S Nair. Indian Sign Language Gesture Recognition Using Deep Convolutional Neural Network.
3. Md. Jahangir Hossein, Md. Sabbir Ejaz Recognition of Bengali Sign Language using Novel Deep Convolutional Neural Network.
4. May and Kumar, Aman Bhatia, Piyush Gupta, Bickey Kumar Shah, Khushi Jha SIGN LANGUAGE ALPHABET RECOGNITION USING CONVOLUTION NEURAL NETWORK.
5. Dardina Tasmere, Bashir Ahmed. Hand Gesture Recognition for Bangla Sign Language Using Deep Convolution Neural Network.
6. Meenu Gupta, Gopal Singh, Akash Yadav, CNN Based Speech and Text Translation Using Sign Language.
7. Md. Nafis Saiful, Md. Nafis Saiful, Abdulla Al Isam, Hamim Ahmed Moon, Real-Time Sign Language Detection.
8. Aditya Rathi, Sumanta Pasari, Sarita Sheoran, Live Sign Language Recognition Using Convolution Neural Networks
9. Anupriya, sudhashu, Nidhi, Leena. sign language recognition system using deep learning