Deep Learning Approach to convert American Sign Language to Text for Deaf and Mute

A Thesis Submitted In Partial Fulfillment of the Requirement for the Award of the Degree of

MASTER OF TECHNOLOGY In
COMPUTER SCIENCE AND ENGINEERING

Avtar Chandra
U2052012 - M.TECH (CSE)

Under the Supervision of:
Dr. SHIV PRAKASH

DEPARTMENT OF
ELECTRONICS & COMMUNICATION ENGINEERING
UNIVERSITY OF ALLAHABAD

Academic Year: 2021-2022

Department of Electronics and Communication Engineering JK Institute of Applied Physics and Technology, University of Allahabad

Date: _________

CANDIDATE’S DECLARATION

I hereby declare that the work presented in this report titled “**Deep Learning Approach to convert American Sign Language to Text for Deaf and Mute** ” is an authentic record of my own work carried out at **Department Of Electronics And Communication Engineering, JK Institute Of Applied Physics And Technology, University Of Allahabad** as required for the award of degree of **Master of Technology** in Electronics Engineering, submitted in the JK Institute of Applied Physics an Technology, under the supervision of **Dr. Shiv Prakash**, Department Of Electronics And Communication Engineering, JK Institute Of Applied Physics And Technology, University Of Allahabad.

It do not embrace any part of work, which has been submitted for the award of any available degree either in this Institute or in other University/Deemed University without proper quotation.

Date: ____________

Place: ____________

AVTAR CHANDRA

U2052012

Department of Electronics and Communication Engineering JK Institute of Applied Physicsand Technology, University of Allahabad

Ref.   Date: __________

CERTIFICATE FROM SUPERVISOR

This is to certify that the project "**Deep Learning Approach to convert American Sign Language to Text for Deaf and Mute**" submitted by **Avtar Chandra** (Enroll. no.: **2104754**), to the Department Of Electronics And Communication Engineering, JK Institute Of Applied Physics And Technology, University Of Allahabad, in partial fulfillment of the requirements for the award of the degree of **Master of Technology** in **Computer Science and Engineering** is an authentic work carried out at Department Of Electronics And Communication Engineering, JK Institute Of Applied Physics And Technology, University Of Allahabad by him under my supervision and guidance.

Dr. Shiv Prakash
(Supervisor)

Acknowledgement

No project reports ever grows full boom from one's personal head. This report owes its existence to help and guidance of many people and I am humbly thankful to all of them.

First and forever, I would like to express deepest sense of gratitude to project mentor, **Dr. SHIV PRAKASH**. He has been my mentor throughout the thesis process. I was privileged to experience a sustained enthusiastic and involved interest from his side. He has a source of constant inspiration throughout the period of this world. His inspiring guidance and innovative ideas were invaluable towards the completion of this project.

I am extremely grateful to **Prof. R.S. Yadav,** Head of Department of Electronics and Communication Engineering, JK Institute of Applied Physics and Technology for providing necessary facilities in order to completeproject work.

Last but not the least, I would like to thanks JK Institute staff members and institute, in general for extending a helping hand at every junction of need.

Date: ___________

Place: ___________

AVTAR CHANDRA

U2052012

Abstract:

Sign Language is a language within which we tend to create use of hand movements and gestures to communicate with other people who are Deaf and Mute. Using convolution neural network, we attempted to develop a real-time finger spelling technique based on “American Sign Language” (ASL). In this paper shows the sign language recognition of 26 alphabets hand gestures of American sign linguistic communication. This organized system contains various modules like pre-processing, training and testing as well, our method is providing **95.8%** accuracy for the 26 alphabets extraction, conditioning, training and testing of model and American sign to text conversion. In this project we have used Deep Learning, OpenCV and Tensor Flow to recognize face masks, we found our dataset performed better accuracy in respect to recognition

Dedicated to everyone who supported me.

Contents:

Introduction:

Gesture may be called to motion in any body part like hand and face. For recognition of gesture we are using computer vision and image processing. Gesture recognition allows system to grasp human’s actions associated additionally act’s as an interface in-between human and computer. This could enable genuine human interaction with computers without requiring direct physical touch with machinery. The Deaf and Mute communities sign with gestures.

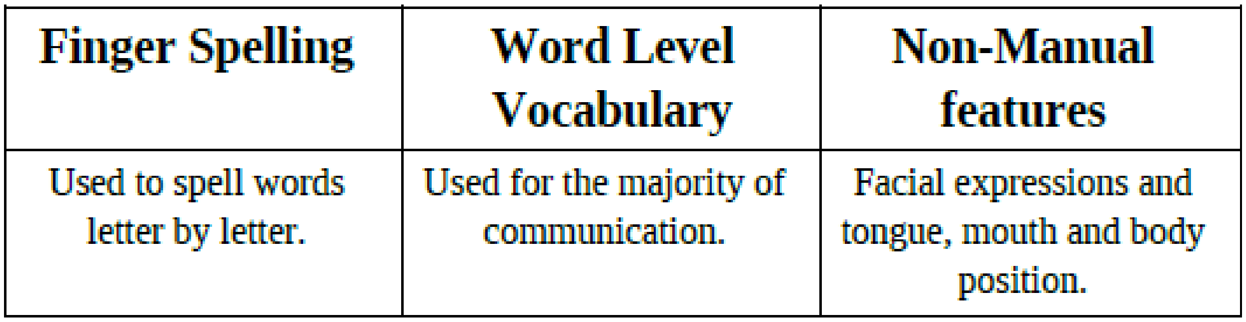
When it was hard to transmit audio or when writing was challenging, this group employed sign language when vision was still an option. The only mode of inter-person communication at the time was sign language. The deaf and dumb community across the world regularly use this, although in regional variants like ISL and ASL. Hand gestures, made with either one hand or both, can be used to communicate in sign language.

Since the single communication-related handicap Deaf and Mute persons have prevents them from using spoken languages, the only means of communication available to them is through ASL (American Sign Language), very extensively used sign language.

Objective:

American Sign Language is a perfectly structured language. American Sign Language is the visual language that deaf and dumb individuals use as their first language. Unlike audio samples that pass through sound, sign language uses body language and hand-to-hand communication to dynamically convey one's thoughts. Gestures might be clearly understood by everyone if a universal link that translated sign language into text existed. In order to enable Deaf and Mute people to communicate without really understanding one another's languages, study has been done on interface system based on vision.

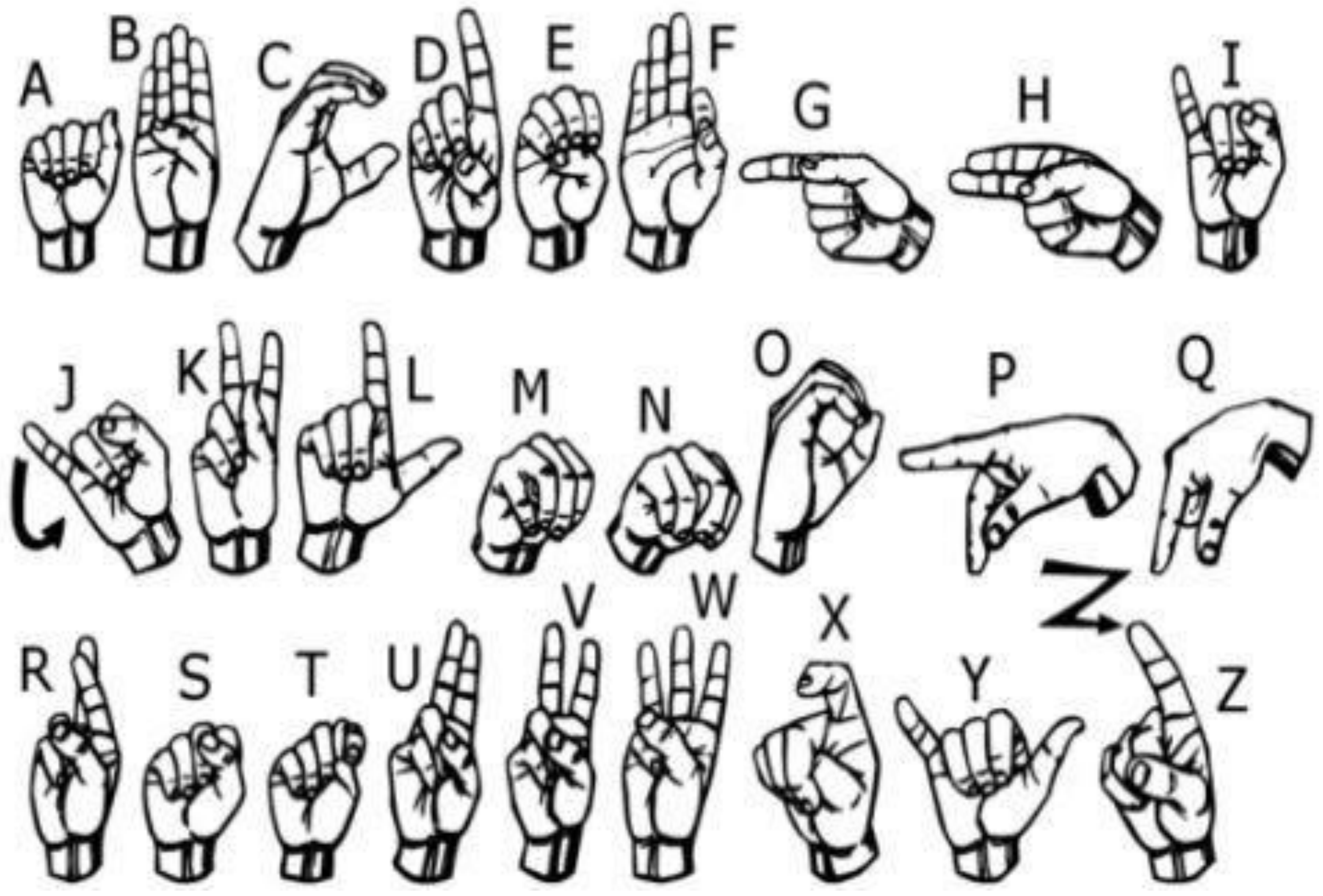
The aim in the project is to project a practical system that’s useful for help to people with hearing impairments and uses a very simple and effective method.



Finger Spelling	Word Level Vocabulary	Non-Manual features
Used to spell words letter by letter.	Used for the majority of communication.	Facial expressions and tongue, mouth and body position.

Table 1.1 Components of visual sign language

Gestures that we have used to train are as specified in the image below:



In an application that is portable and real-time, we are attempting to communicate words, characters, and phrases in American Sign Language with an understanding of symbols.

The major objectives for this project are:

- To create a model that, in comparison to currently existing models, would predict symbols with the greatest accuracy and in the shortest amount of time.

- To lower the cost and create a graphical user interface (GUI) programme that is simple to use and requires little upkeep in order to convert a sign to its matching text.

- To offer ideas depending on the present term in order to avoid having to translate the entire word, enhancing accuracy and speeding up sign to text conversion

.

- To lower the likelihood of spelling errors by proposing appropriate spellings for terms nearby in the English lexicon.

Literature Review:

There has been a lot of research and experiments carried over gesture recognition of hand in past decade.

With the aid of a literature research, we accomplished the fundamental steps in the identification of hand gestures as :-

- Acquisition of Data.
- Preprocessing of Data.
- Extraction of Features.
- Gesture classification.

Acquisition of Data:

The different procedures to acquire information about handgesture may be executed in following ways:

1. Using sensor devices

Electromechanical gadgets are used to deliver exact hand function (position) and configuration. Various glove based strategies may be used to abstract records .But it isn't user friendly and steeply-priced.

2. Vision based approach

The vision based approaches know a regular contact between computer and humans with no usage of any additional gadgets because they just need a digital camera. Computer cameras are used as input devices in vision-based tactics to examine the statistics of hands and fingers. We will talk about simulated vision system that are applied in hardware and/or software and that frequently serve to supplement natural creativity and foresight. Managing the considerable variation in the look of human hands caused by a large number of hand motions, a diversity of skin tones, as well as changing camera angles, scales, and shutter speeds, is the core challenge in vision-based hand detection.

Data-preprocessing and extracting Features for approach based on feature:

- In [1] the hand detection method combines colour detection by threshold with background removal. Because both the face and the hand have the same skin colour, we may utilise“ Adaboost face finder ” to tell them apart.

- We could also apply the Gaussian Blur filter in order to abstract the images needed for training. We can easily apply filters by usage of Open Computer Vision, too to be called as OpenCV and is described in [3].

- We can employ instrumented gloves, as suggested in [4], to extract the essential picture for training. In comparison to adding the filters to data extracted from the videos, this will help us minimise computation time for preprocessing and provide us with more succinct and accurate data.

- Using a colour segmentation method, we attempted to manually segment the image., however as stated in the study paper, the segmentation results we obtained did not match since skin colour and tone are highly depend on lightning circumstances. The resultant we achieved by segmentation were not so best fit. Also, many characters that are similar to each other, such as the "V" sign and the gesture for the number "2", needs to be trained for the project, so we decided to improve it in terms of better accuracy. Instead of segmenting the hand from any random background for a huge number of symbols, we retain the background of the hand a single colour so we don't have to separate it based on skin tone. This would allow us to get better outcomes.

Gesture classification:

- In [1] “ Hidden Markov Models (HMMs)” are to be used to classify gestures. This model focuses on the dynamical aspect of gestures. Tracing skin-colored patches corresponding to hands in body and facial space centered on the face of used is used to excerpt gestures from video picture sequences. The purpose is to identify two types of motions. Images are filtered using a quick lookup index table. After filtering, skin-colored pixels are collected as droplets. A blob is a statistical object based on position [x, y] and colorimetric [Y, U, V] of a skin-colored pixel to define a uniform area.

- In [2], an efficient and fast method to recognize static hand gestures is the Naive Bayes classifier. It is based on the classification of various movements based on geometric extracted from segmented visual data. As a result, unlike some of the other approaches, our method is not affected by skin tone. Each frame of a video with a static background is used to extract gestures. The initial step is to categories and name the item of interest before extracting geometric invariants from it. The next step is to do gesture categorization using K-nearest neighbor method combined with the distance weighting (KNNDW) to generate data appropriate for a locally weighted Naïve Bayes Classifier.

- According to the paper “ Human Hand Gesture Recognition Using a Convolution Neural Network by Hsien I Lin, Ming-Hsiang Hsu, and Wei-Kai Chen the graduates of the Taiwan National Taipei Institute Of Automation Technology ” creates a skin model, A picture's hands are extracted, and then a binary threshold is applied to the entire image. To centre the image, the threshold image is calibrated around the main axis. They train and forecast a convolutional neural network model using this image. They trained the model on more than seven hand motions and got around 95% accuracy for seven hand gestures.

HARDWARE AND SOFTWARE REQUIREMETS:

- Operating System: Windows
- Language used: python
- Platform used: Anaconda, Visual Studio
- Application used: Jupiter
- Library used: NumPy, MatPlotlib, OpenCV, Keras, etc.

Key Words and Definitions:

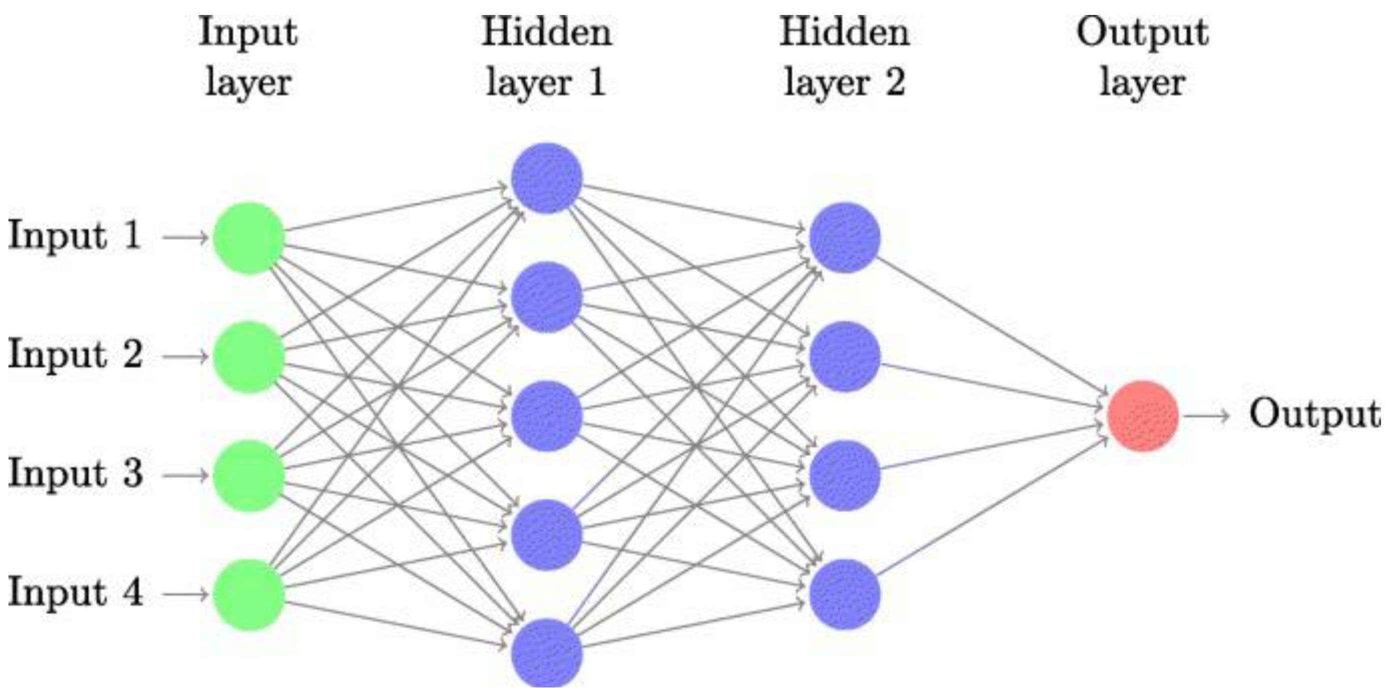
.

Extraction Feature and Representation

The 3D matrix representation of an image, where the dimensions of images are height, width, and depth, respectively (1 for Grayscale and 3 for RGB). Additionally, these values of image pixels are employed along with CNN to extract major updates.

Artificial Neural Networks

An artificial neural network is a collection of neurons connected in a way that resembles the way the human brain is organized. Information is sent from one neuron to another through every connection. Before transmitting the inputs to the buried layer of neurons, the first layer of neurons receives and processes them. The information is transmitted to the final output layer after being processed through multiple hidden levels.



Artificial Neural Network

They can learn and must be trained. There are various strategies of learning as:

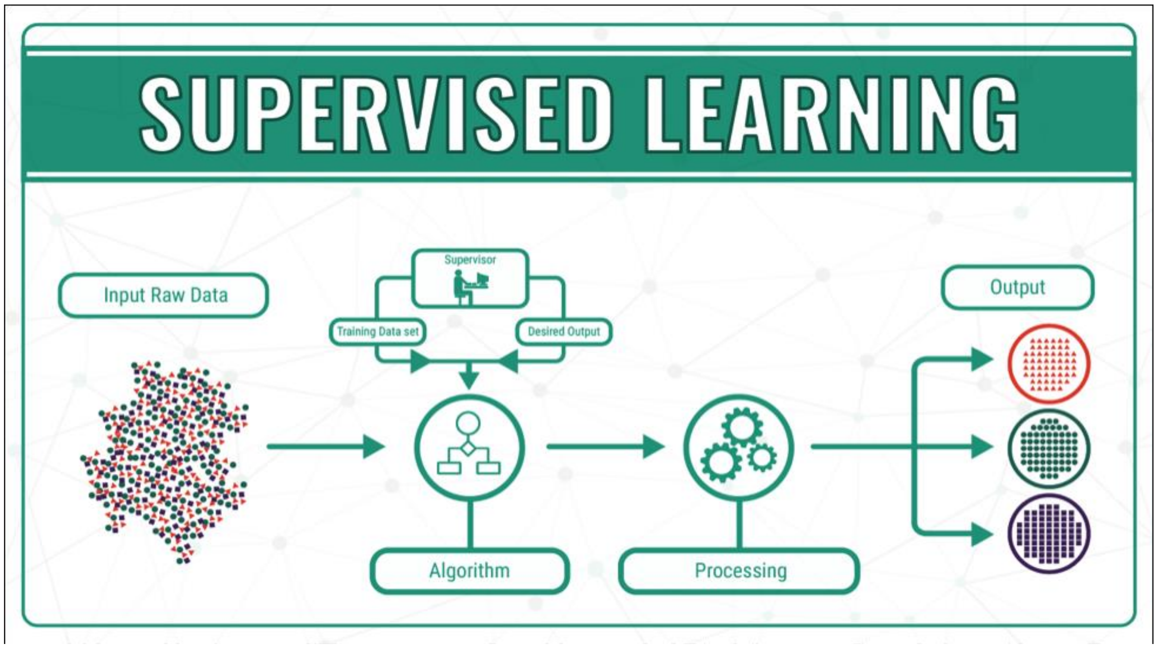
- Unsupervised Learning.
- Supervised Learning.
- Reinforcement Learning.

a) Unsupervised Learning:

Unsupervised learning is the process of using AI algorithms to search for patterns in datasets including data points that are neither classed nor labelled. Principal component and cluster analysis are two of the most often used techniques in unsupervised learning. The only need for an unsupervised learning method is to identify a replacement feature area by maximising some objective function or decreasing some loss function that captures the properties of the original space. Making a variance matrix is not unsupervised learning, but getting the eigenvectors of the variance matrix is since the algebra chemist decomposition procedure maximises the variance, and this is known as main part analysis.

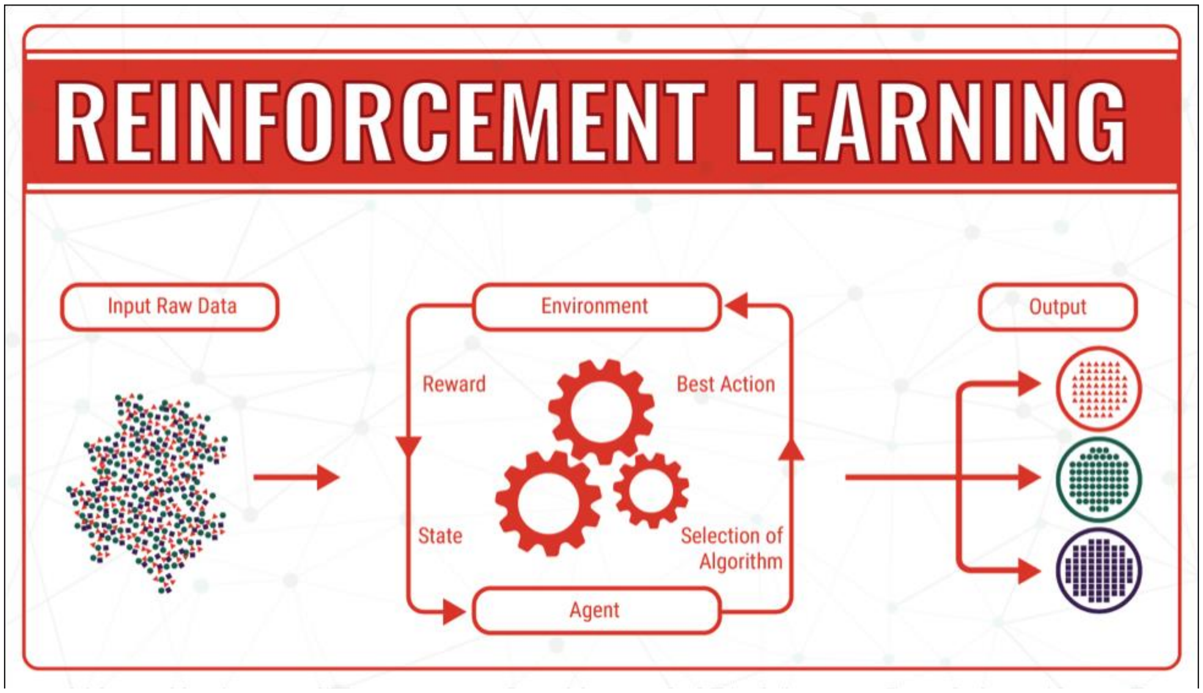
b) Supervised Learning:

Supervised learning refers to the machine learning problem of knowledge to carry out an input-to-output translation supported by example input-output efforts. It develops a functionality using labelled training data, which is a collection of coaching instances. In supervised learning, each example might be a combination of a predicted output value and an input item (often a vector) also known as the super ordinate signal. Before developing an inferred function, a supervised learning algorithm analyses the training data. This function may then be utilised to map new samples. The algorithm can be left in the best case scenario so that the category labels for unforeseen circumstances are correctly confirmed. For this, the teaching algorithm must extremely "reasonably" generalize from the training data to items that have not yet been observed.



c) Reinforcement Learning:

A subset of machine learning called reinforcement learning (RL) is concerned with how software engineers should function in their environment to take full advantage of the idea of cumulative reward. One of the three main machine learning paradigms, together with supervised and unsupervised learning, is reinforcement learning. The difference reinforcement learning and supervised learning in that input/output pairings are not required to be tagged before being presented, and suboptimal behaviors do not need to be explicitly modified. In its place, the primary goal is to strike a stability between exploitation and exploration (of uncharted region) (as of present knowledge).

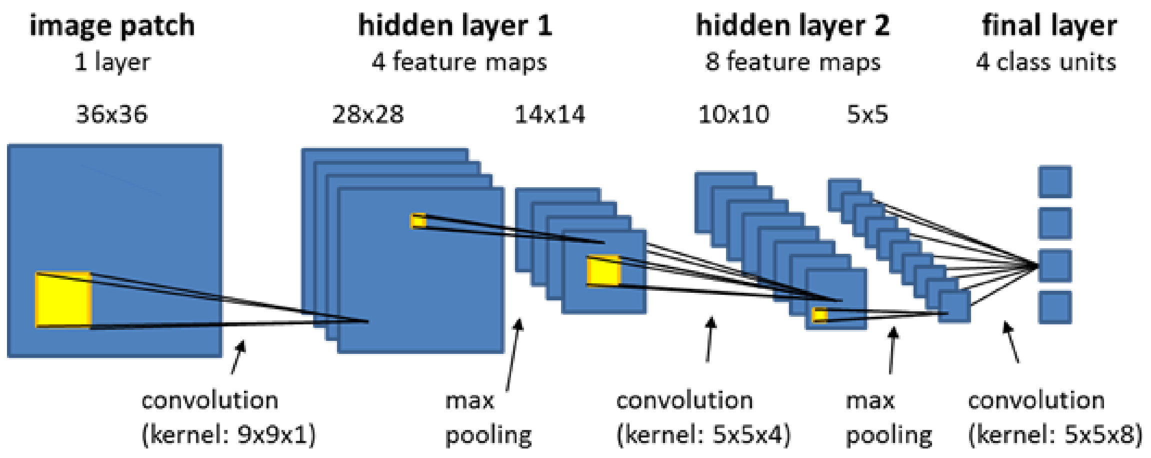


Convolution Neural Network

A Convolutional Neural Network (also called as ConvNet/CNN) is a Deep Learning approach that could takein input as an image, and assign priorities to the various components and objects in the image (learnable weights and biases), and discriminate between them. Other classification techniques need significantly more pre-processing than a Convolutional Network. ConvNets are capable of learning these filters and properties, whereas filters in fundamental approaches must be hand-engineered.

The organisation of the Visual Cortex affected the building of a ConvNet, which is akin to the linking network of neurons in the human brain. Individual neurons only respond to stimuli in this narrow area of the visual field known as the Receptive Field.

The whole visual field is enclosed by a series of such fields that overlap. CNN layers feature neurons organised in three dimensions: width, height, and depth, in contrast to traditional neural networks. Rather than being completely coupled to every neuron, the neurons of an extraordinarily layer can only be connected to a very small percentage of the layer (window size) before it. Because the CNN design may compress the whole image to a single vector of class scores, the final output layer would likewise have dimensions (number of classes).



Convolutional neural network

There are four steps in CNN:

1. Convolution.
2. Subsampling.
3. Activation.
4. Full Connectedness.

A. **Convolution Layer:**

The convolutional layer is the first layer of a convolutional network. The fully-connected layer is the last layer, even though convolutional layers, additional pooling layers or convolutional layers, might come afterwards. The CNN becomes more complicated with each layer, detecting more areas of the picture. The first layers emphasize fundamental features such as colours and borders. The bigger features or forms of the item are first recognized when the visual data moves through the CNN layers, and eventually the desired object is recognised.

As a result, the input will contain three components that correspond to the RGB values in an image: width, depth, and depth. Furthermore, we have a feature extraction tool, also known as a kernel or filter that will check the activation functions of the picture and determine whether or not the characteristic is there. Convolution is the term used to describe this process.

The feature detector is a weighted two-dimensional (2-D) array that represents a piece of the image. This dot product is then sent into a series of outputs. The filter then proceeds by a stride and continues the action until the kernel has covered the entire image.

B. Pooling Layer:

The pooling layer is every other constructing block of CNN. Its distinctive feature is a gradual reduction in the illustration's spatial length to cut down on the number of factors and internal computations. Every characteristic map is treated separately by the pooling layer. Max pooling, in which the majority of a location is picked as its representation, is the most common place strategy used in pooling. For instance, the most value is used to replace a 2x2 position in the following diagram.

a) Max Pooling:

In max pooling, we use a window size as of [2x2] as an example and only use the maximum of four values. To properly obtain the activation matrix's unique Size, close this window and continue the current operation.

b) Average Pooling :

We need the average of all the data in a window for average pooling:

c) Fully Connected Layer:

Inside Convolution layer, neuron are linked simplest to a nearby region, at the same time as in a completely linked region, will join the all of the input to neurons.

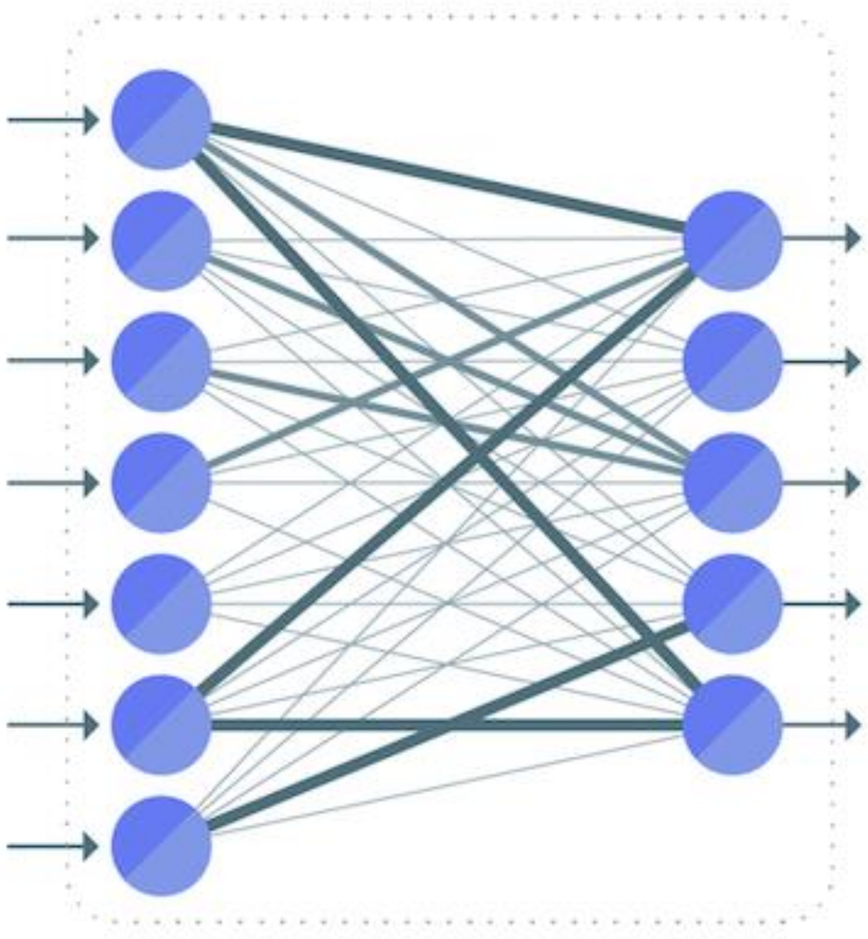
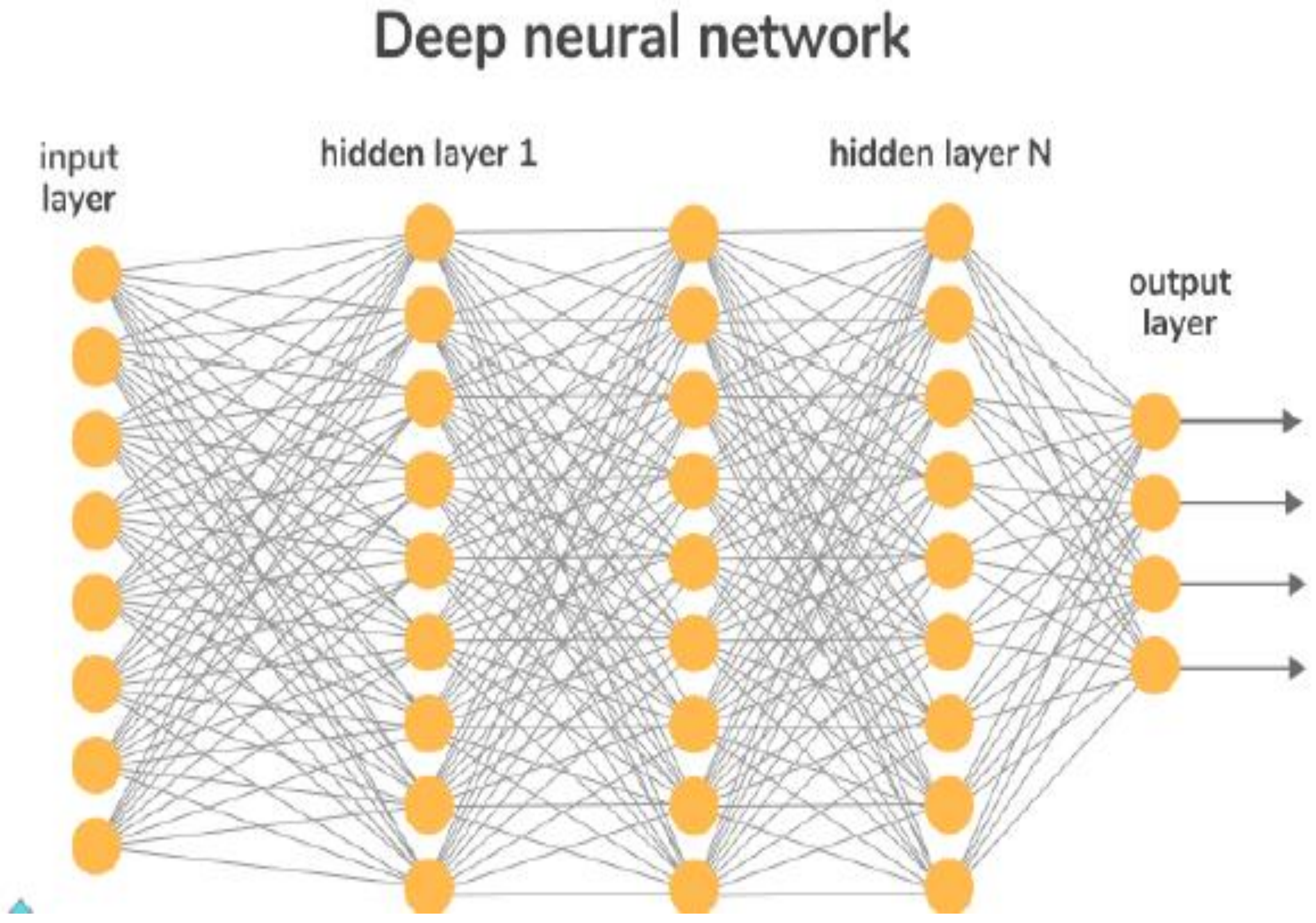


Figure: **Fully connected layer**

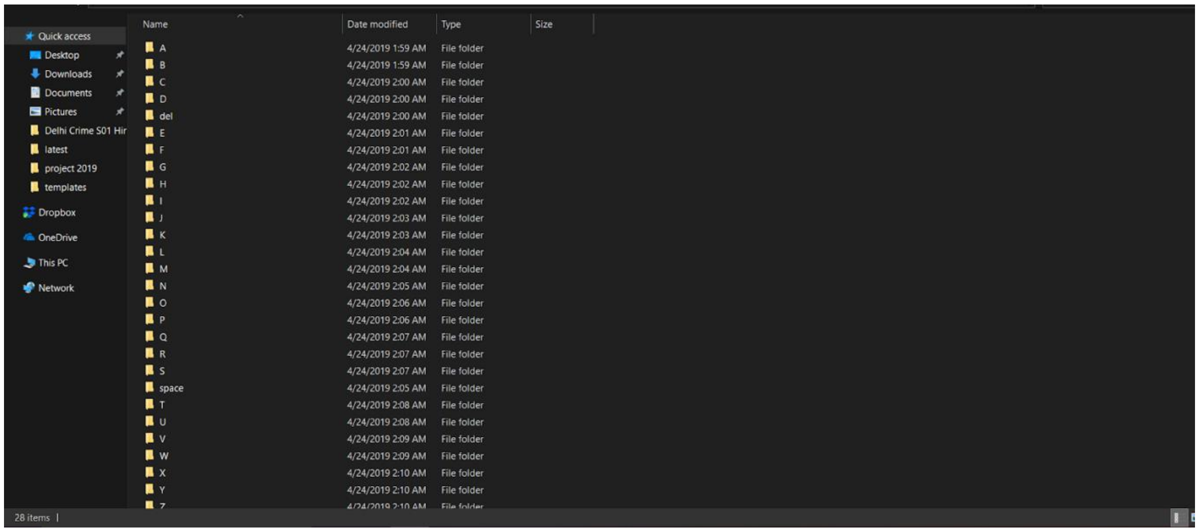
D) Final Output Layer:

After gathering input from a completely connected layer, we will merge it with the very last layer of neurons [where count equals the total number of classes], on the path to predicting every image to be in a unique class.



SOME IMAGES OF THE AMERICAN SIGN LANGUAGE DATASET:

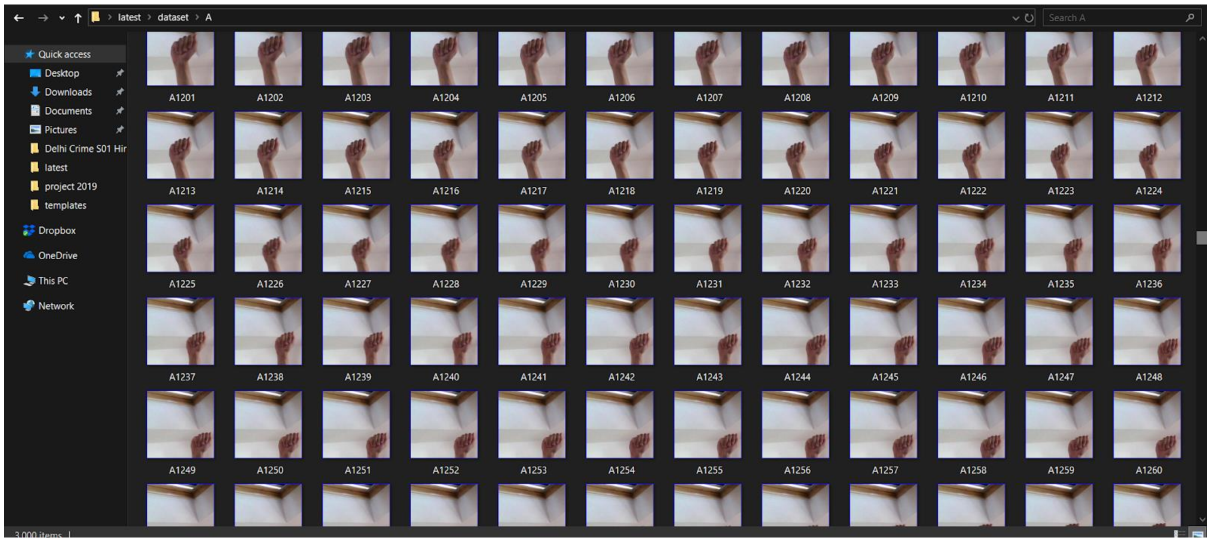
Image of dataset generated:



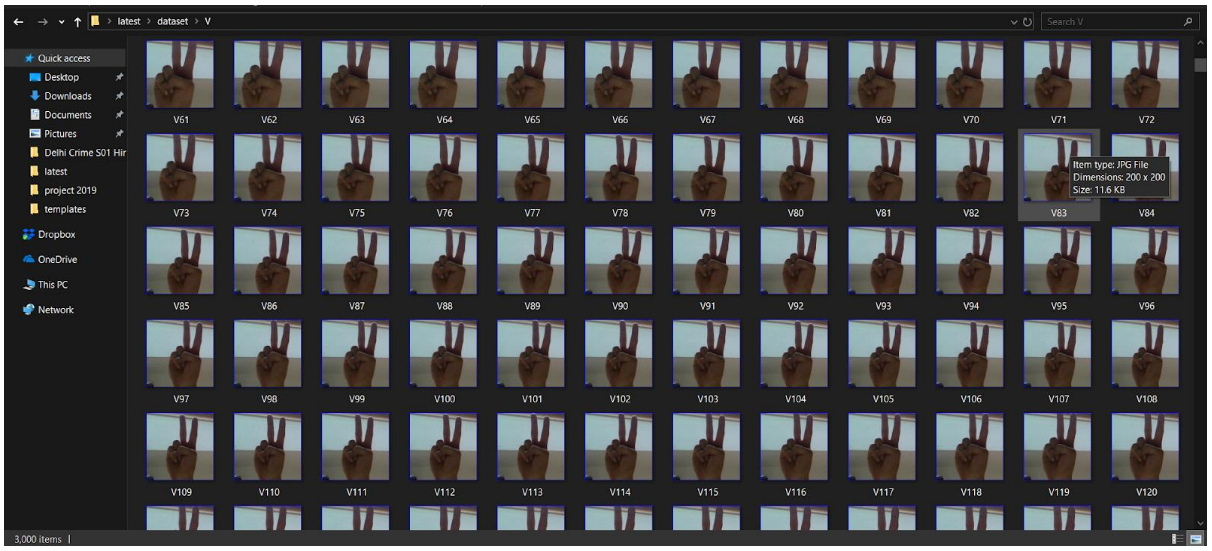
Name	Date modified	Type	Size
A	4/24/2019 1:59 AM	File folder	
B	4/24/2019 1:59 AM	File folder	
C	4/24/2019 2:00 AM	File folder	
D	4/24/2019 2:00 AM	File folder	
del	4/24/2019 2:00 AM	File folder	
E	4/24/2019 2:01 AM	File folder	
F	4/24/2019 2:01 AM	File folder	
G	4/24/2019 2:02 AM	File folder	
H	4/24/2019 2:02 AM	File folder	
I	4/24/2019 2:02 AM	File folder	
J	4/24/2019 2:03 AM	File folder	
K	4/24/2019 2:03 AM	File folder	
L	4/24/2019 2:04 AM	File folder	
M	4/24/2019 2:04 AM	File folder	
N	4/24/2019 2:05 AM	File folder	
O	4/24/2019 2:06 AM	File folder	
P	4/24/2019 2:06 AM	File folder	
Q	4/24/2019 2:07 AM	File folder	
R	4/24/2019 2:07 AM	File folder	
S	4/24/2019 2:07 AM	File folder	
space	4/24/2019 2:05 AM	File folder	
T	4/24/2019 2:08 AM	File folder	
U	4/24/2019 2:08 AM	File folder	
V	4/24/2019 2:09 AM	File folder	
W	4/24/2019 2:09 AM	File folder	
X	4/24/2019 2:10 AM	File folder	
Y	4/24/2019 2:10 AM	File folder	
Z	4/24/2019 2:10 AM	File folder	

28 items

Images of dataset of letter “A” :



Images of dataset of letter “V” :



Required Libraries:

TensorFlow

Tensor Flow helps to have some working knowledge of linear algebra and vector calculus in order to comprehend tensors. Tensors are already represented in Tensor Flow as multidimensional data arrays, but more background information is required to fully understand tensors and their applications in deep learning and machine learning.

Keras

A Python-based open-source neural network library called Keras was developed. TensorFlow, Microsoft Cognitive Toolkit, Theano, or PlaidML may all be run on top of it. It is focused on being user-friendly, modular, and extendible in order to enable quick experiments with the help of deep neural networks. It has been developed as a component of ONEIROS ("Open ended Neuro Electronic Intelligent Robot Operating System") research project and François Chollet, a Google developer, is primarily responsible for creating and maintaining it. The XCeption deep neural network model was also created by Chollet.

NumPy

NumPy is module for Python. The call is an acronym for "Numeric/Numerical Python". Numpy is pronounced /ˈnʌmpaɪ/ (NUMpy) or much less often / ˈnʌmpi (NUM-pee)). It is an extension module for Python language, ordinarily written in C. This makes certain that the pre-compiled mathematical and numerical features and functionality of Numpy assure notable execution speed.

OpenCV

OpenCV is an acronym for Open Source Computer Vision Library that is an open supply BSD- certified library that consists of numerous masses of pc imaginative and prescient algorithms. The report describes the so-referred to as OpenCV 2.x API, that is basically a C++ API, as contrary to the C-primarily based totally OpenCV 1.x API. OpenCV has a modular structure, because of this that the bundle consists of numerous shared or static libraries.

Matplotlib

Python's Matplotlib package allows users to create 2D graphs of array data. Although it was developed to mimic the MATLAB graphics commands, it may be used in a Pythonic, object-oriented manner and is independent of MATLAB. Despite being primarily written in pure Python, Matplotlib heavily utilises NumPy and other extension codes to improve performance even for enormous arrays.

Hunspell (Autocorrect feature)

Hunspell propose a Python package, is used to suggest viable alternatives for each (incorrect) input word.. The user can pick from a list of words that correspond to the current keyword to be added to the current sentence. This decreases spelling errors and aids in word prediction for difficult words.

In order to install hunspell package in python use: pip install hunspell.

OS

OS module of Python offers tools to communicate with the operating system. OS is included in Pythons modern application modules. This module gives a transportable manner of the use of running gadget based functionality. The *os* and *os.path* modules consist of many features to have interaction with the document gadget. The Python OS module provides the ability to configure the interface between the user and the operating device. OS module provides several necessary OS characteristics that may be utilised to conduct out OS primarily dependent entirely duties and get relevant data about the functioning device. This module provides a portable method of using running gadget-based features. The Python OS module will be used to create paintings using documents and folders.

Methodology and Materials:

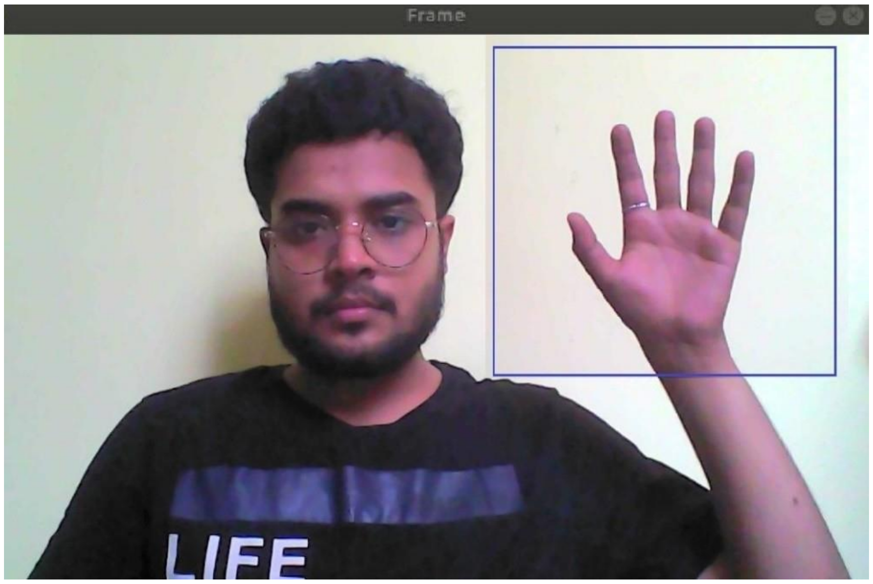
The American Sign Language to text system has been built on the concept of computer vision. Every alphabet signs or gesture are formed using their own hands that eliminates the requirement for any artificial device or equipment for engagement.

Data Set Generation

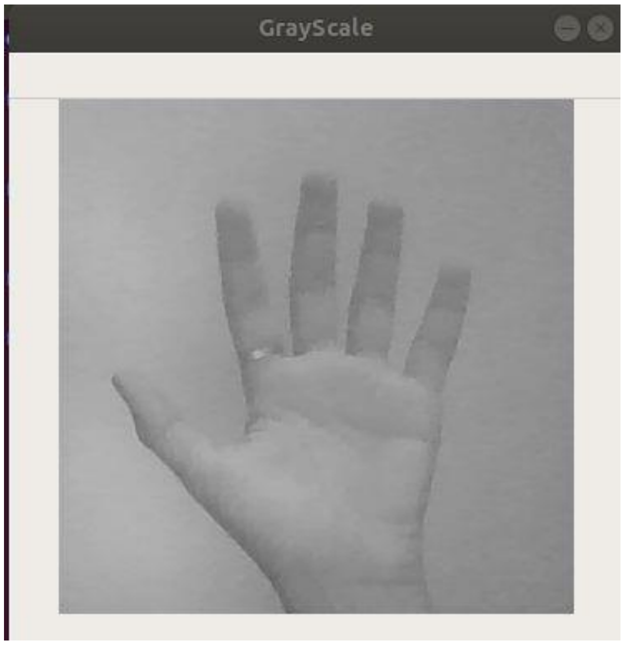
We looked for pre-built datasets for the project, but none of them had the raw photos we need. The only datasets that we could discover were in form of RGB values. We made the decision to build our own Dataset as a result. The steps we utilized to create our dataset are as follows:

- The Open Computer Vision (OpenCV) library was utilised to construct our dataset. The first step was to take roughly 200 images for testing and about 800 pictures of each ASL symbol for training.

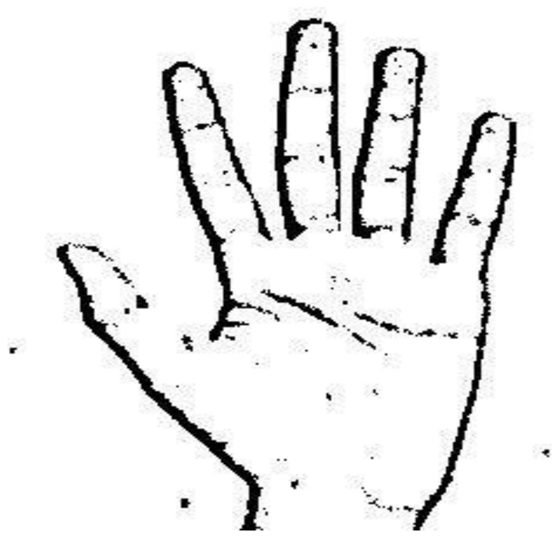
- We start by taking a snapshot of each frame that the webcam on our computer displays. A blue enclosing square designates the region of interest (ROI) for each frame.



- We extracted our RGB, Region Of Interest (ROI) from this entire image and transformed it to a grayscale-image, as depicted below.



• To extract several elements from the final image, we lastly employed a gaussian blur filter. This is how the image appears following the use of Gaussian blur.



GESTURE CLASSIFICATION:

The approach implemented in project:

Our method uses 2 layer of the algorithm in order to forecast the last gesture of the user.

Algorithm Layer “1”:

1. To create processed images following feature extraction, apply the Gaussian blur filter and threshold to OpenCV captured frames.

2. When generating a term, a character is printed and considered. if it is found in more than 60 frames of this processed image, which has been given to the CNN model for prediction.

3. Spaces in-between words are to be considered by using the blank symbol.

Algorithm Layer “2”:

1. We detect a variety of symbol sets that provide similar consequences when detected.

2. Then we apply specific classifier to sets to classify between them.

Layer 1:

CNN Model:

1. First Convolutional Layer:

The resolution of the original image is 128 x 128 pixels. It was previously processed with 32 filter weights in the first convolutional layer(3x3 pixels each). The result is a 126 x 126 pixel image, one for each filter weight.

2. First Pooling Layer:

Images have been down sampled by the use of 2 x 2 max pooling. That is, itstores the largest value in the square of a 2 x 2 array. So the image is abridged to 63 x 63 pixels.

3. 2nd Convolution Layer:

The first full layer's output (63 x 63) is now fed into the second convolutional layer. It should be processed using 32 filter weights in the second convolutional layer. (3 x 3 each pixel). The result is a 60x60 pixel image.

4. Second Pooling Layer:

The resulting image is down sampled again by the usage of (2 by 2) pool and down sampled to a 30x30 image resolution.

5. First Densely Connected Layer:

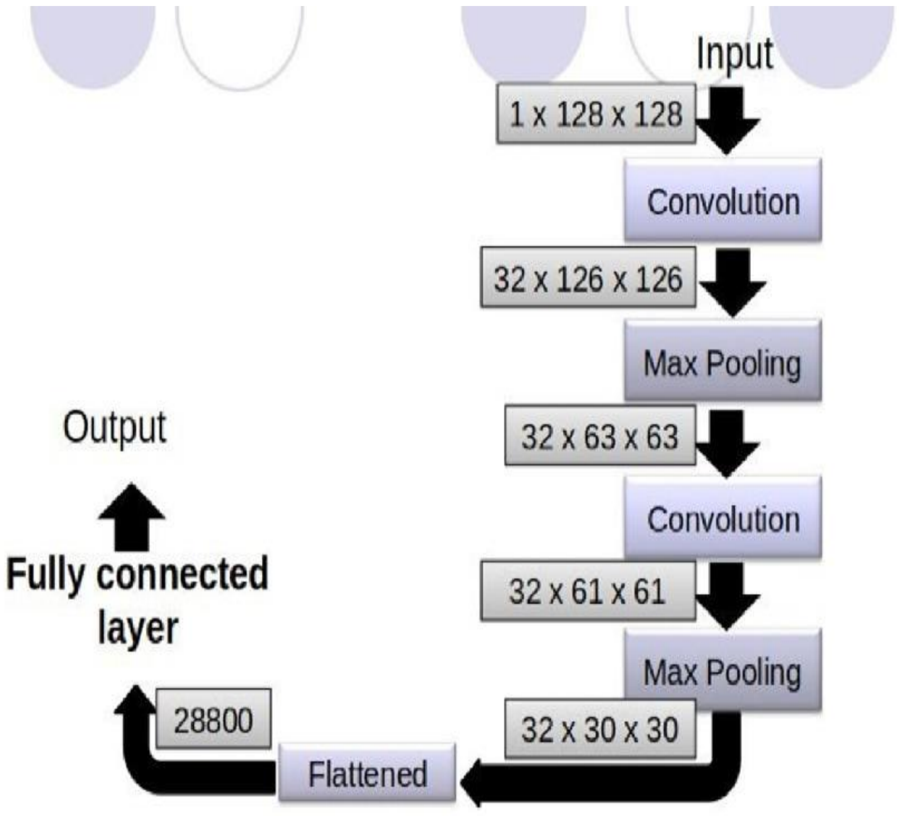
These images are fed into a layer with 128 fully connected neurons, and the output of this layer is transformed into an array. at every step of 30 x 30 x 32 = 28800 values. This layer's output is sent into a second tightly connected layer. We have used a dropout layer of 0.5 to avoid over fitting.

6. **Second Densely Connected Layer:**

96 neurons and result from very first densely linked layer have to be used as the intake for the completely connected layer.

7. **Final layer:**

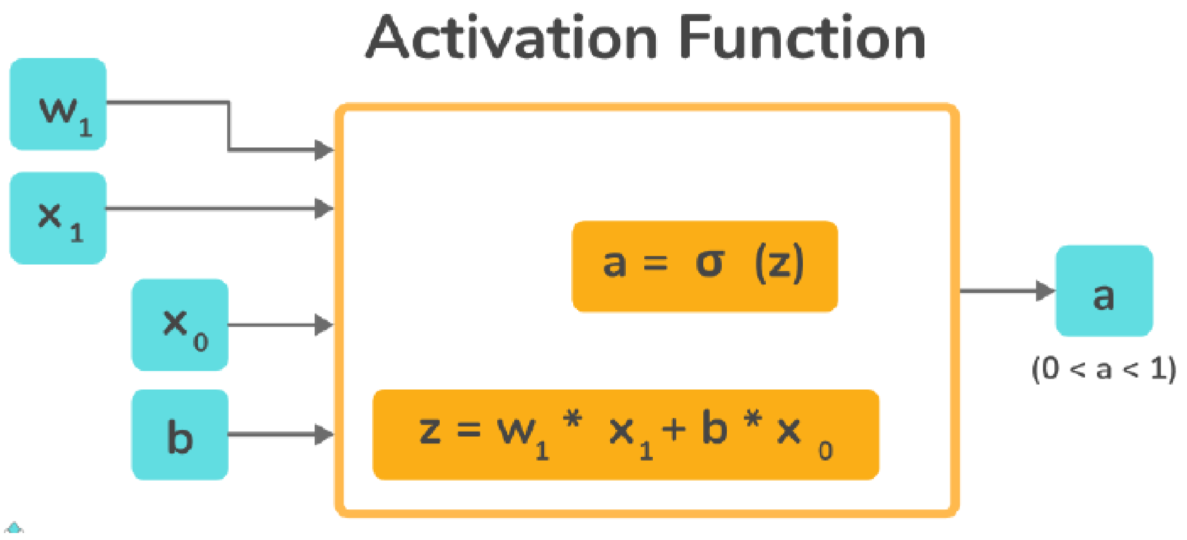
The output of the closely connected second layer is utilised as input to the final layer, which contains the same number of neurons as the number of classes classified. (blank and alphabets symbols).



Activation Function:

Rectified Linear Unit, or ReLu has been engaged in each layer (fully connected neurons along with convolution).

For every pixel input, ReLu determines max[x, 0]. This increases the nonlinearity of the formula and forms it simpler to understand its excessive complicated characteristics. By cutting down on computation time, it helps to solve the vanishing gradient problem and speed up training.



Pooling Layer:

With the reLU activation function, We use Max pooling with a pool size of on the input image (2, 2). The number of parameters is reduced, which decreases computing costs and prevents overfitting.

Dropout Layers:

The issue of over fitting, in which the weight's of the network are so tuned to the exercise instances that the network performs poorly when given fresh examples after training. This layer "drops out" a random set of activation in the same layer by initialising them as 0.. Even if certain activations are missing, the network must be capable of providing the correct classification or result for a single sample [5].

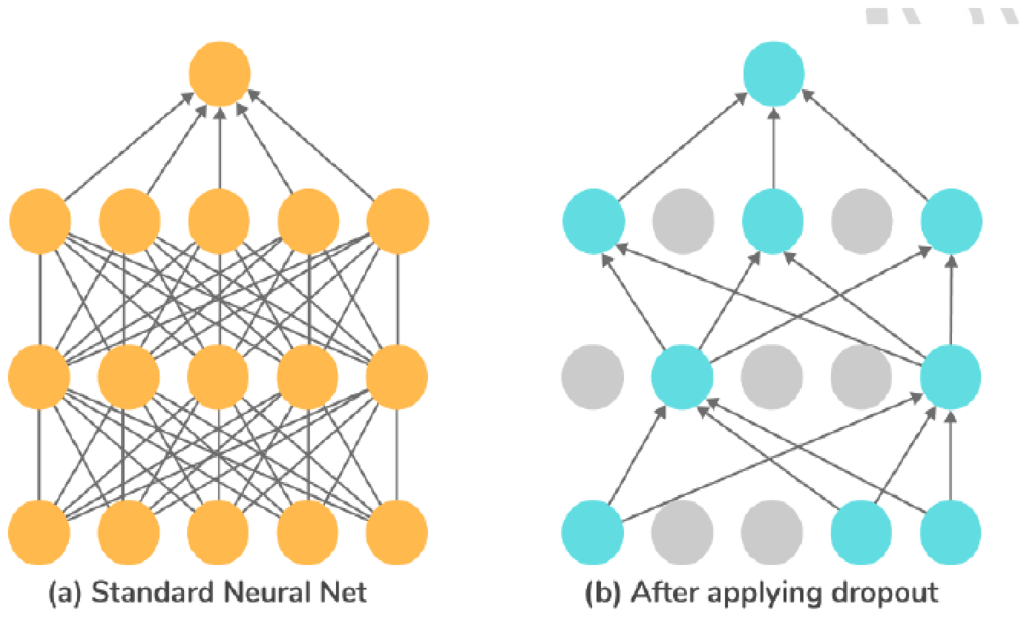
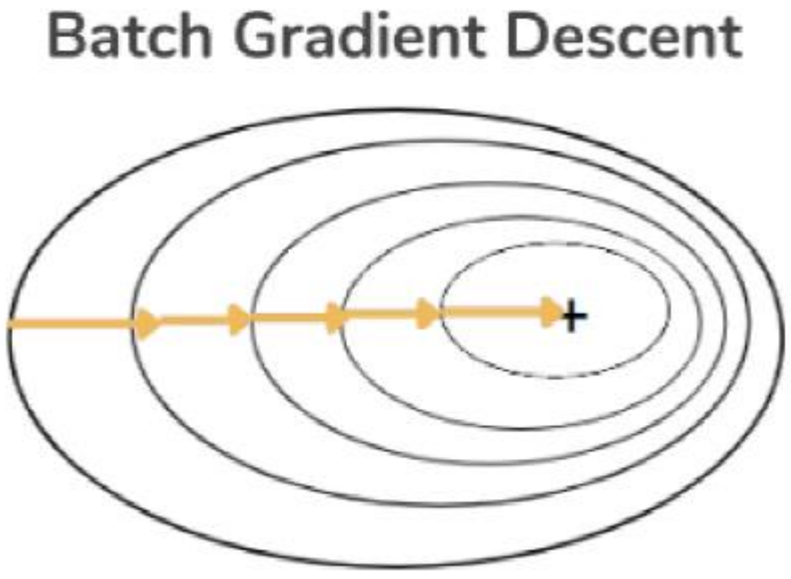


Figure: **Solving the problem of overfitting using dropou**

Optimizer:

As a result of the loss function's output, Adam optimizer has been used in order to update the model. Adam combines the benefits of the “root mean square propagation”, extensions of two stochastic gradient descent algorithms (RMSProp) and adaptive gradient algorithm (ADA GRAD).



Layer 2:

In order to get as close as we can to accurately identifying the symbol displayed, to identify and prediction symbols that are increasingly alike each other, we use two layers of algorithms. The following symbols were unclear or showed incorrectly during our testing, we are providing various symbols also:

a. For U: D & R.
b. For S: M & N.
c. For D: R & U.
d. For I: T, D, K & I.

We are making 3 different classifiers to classify this sets, in order to handgrip above occurring case's:

1. (D, R, U)
2. (T, K, D, I)
3. (S, M, N)

Implementation of Finger-spelling sentence formation:

We print the character and add it to the current line once the number of detected characters exceeds a particular value and no additional characters are near to the threshold (We recorded the value as 50 and the difference threshold as 20 in our code.).

Else, clean the present vocabulary with number of occurrences of the current character to dodge the possibility of predicting an invalid character.

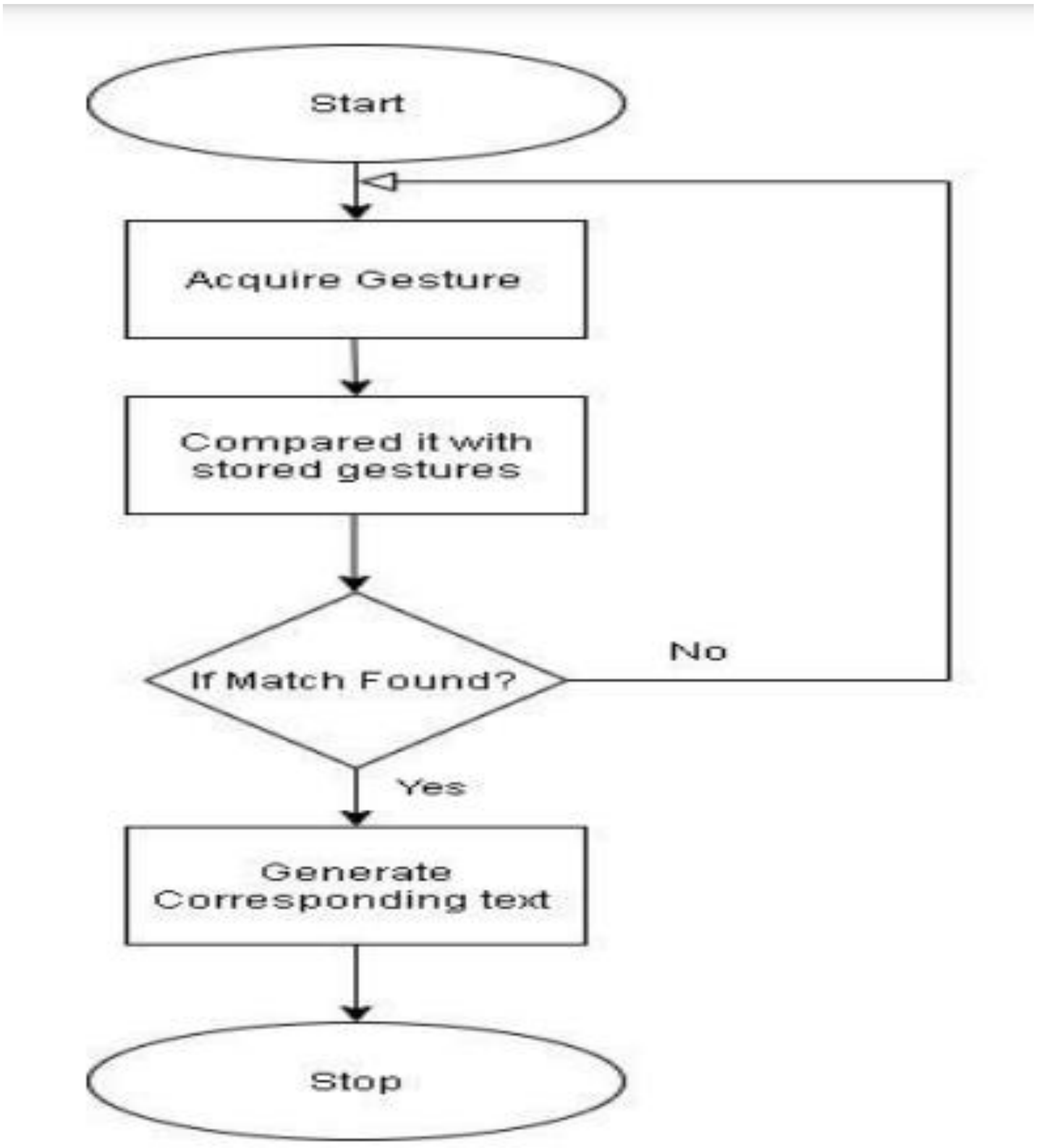
Whenever the total of detected blank's (normal background) surpasses a definite value, if present buffer is empty, no blanks are sensed..

Otherwise, it will print one space to predict the finish of a word, and the current word will be added to the sentencebelow.

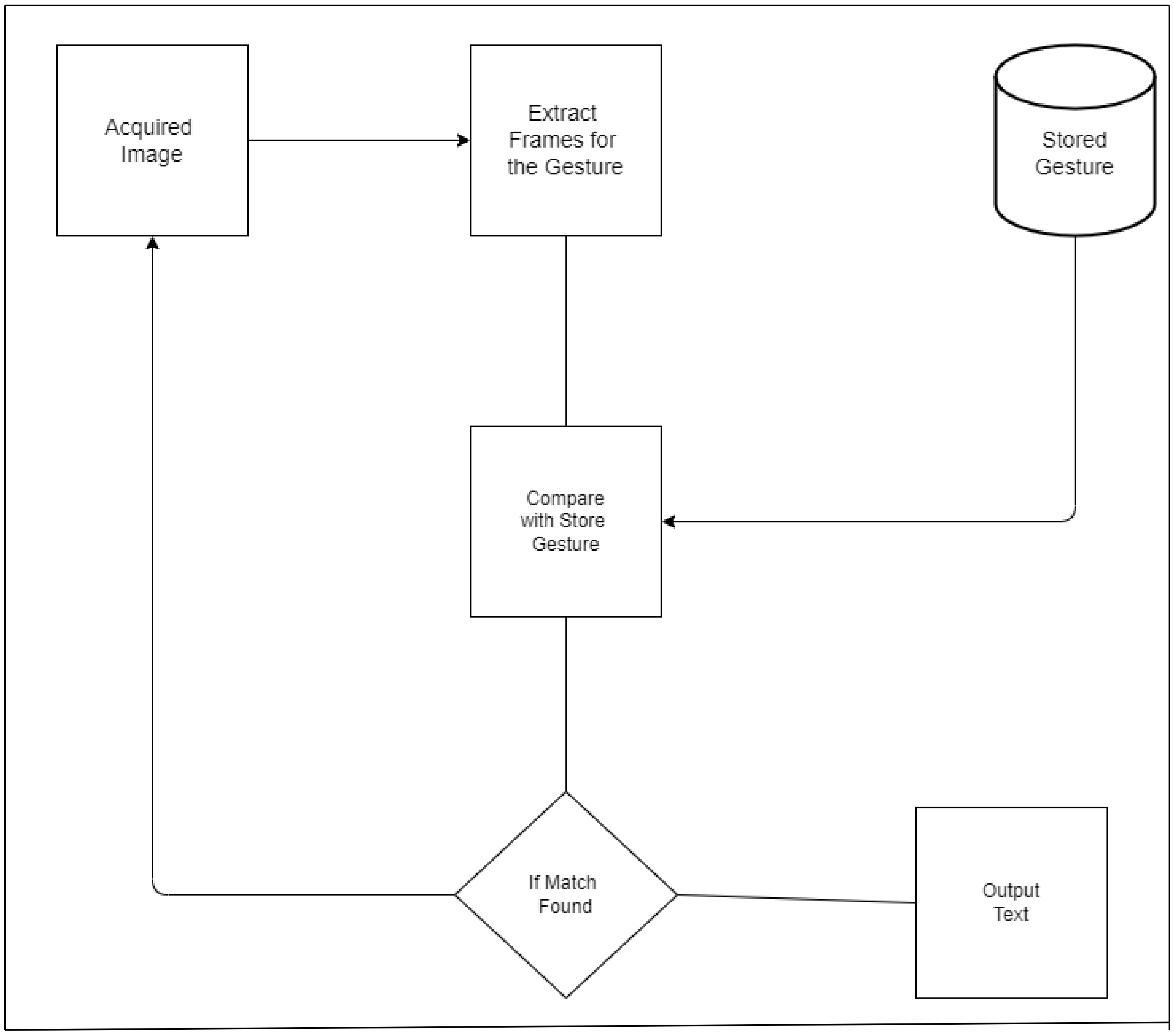
Auto correction Feature:

In Python libraries for each (incorrect) input word, Hunspell suggests appropriate replacements, and the user can add phrase to the current sentence by selection. From a list of terms that match the current word. It decreases spelling errors and aids in prediction for difficult words.

Flowchart:


Start
Acquire Gesture
Compared it with stored gestures
If Match Found?
No
Yes
Generate Corresponding text
Stop

Data Flow Diagram:



Data Flow Diagram

Training and Testing:

To get rid of extra noise, convert the ‘RGB’ input image to gray-scale and add a Gaussian blur. Resize the image to 128 by 128 and use an adaptive threshold to remove the hand from the background.

The pre-processed input photos are then given to the model for training and testing once we have completed all of the aforementioned procedures.The possibility that an image corresponds to one of the classes is estimated at the prediction level. In order for all values in each class to add up to 1, the output is standardised between 0 and 1. The softmax feature helped us do this.

Convert the RGB input image to grayscale, then add a Gaussian blur to remove additional noise. To get the hand out of the background, resize the image to 128 by 128 and apply an adaptive threshold.

Once all of the aforementioned steps have been carried out, the pre-processed input photographs are then supplied to model for the purpose of training as well as testing.

At the guess/prediction level, the likelihood that an image belongs to one of the classes is estimated. The result is standardized between 0 and 1 so that all values in each class total up to 1. This was made possible via the softmax feature.

Challenges Occurred:

We ran across various issues along the process. A lack of data was the first issue we encountered. Because operating with only square pictures was much more practical in Keras, we intended to deal with raw photographs, primarily square images. We chose to develop our own dataset for it because we were unable to find an existing one. The second challenge was deciding which filter to apply to our photographs in order to extract the necessary attributes to input into the CNN model. After testing with a variety of filters, including binary threshold, canny edge detection, and others, we ultimately opted on the gaussian blur filter. Many other disputes have been faced related to precision of model we have taught in past phases and have been evolved by subsequent increase the size of input image and dataset.

Results:

Using only layer 1 of our algorithm, we were receiving the precision of **91.3 percent** in the model, and when we combined both the layers ‘1’ and ‘2’, we could accomplish an accuracy of **95.8 percent**. This accuracy surpasses that of the vast majority of American Sign Language research articles that have just been released. The vast majority of research articles focus on the use of kinect-like gadgets for possible hand detection. A flemish sign language recognition system with a 2.5 percent mistake rate is created in [7] using kinect and convolutional neural networks. [8] Constructs a recognition model and obtains a defect rate of 10.90% applying hidden markov model (HMM) classifier and a dictionary of 30 word’s. They accomplished an average accuracy of 86% for 41 static motions in Japanese sign language in [9]. For signers who had already been seen, Map [10] obtained an accuracy of 99.99 percent utilising depth sensors, and 83.60 percent and 85.39 percent for new signers.
For their recognition system, they also utilised CNN. It should be noted that unlike some of the models listed above, our model does not use a background subtraction approach. As a result, if we try to apply background subtraction to our project, the accuracy may vary.

The bulk of the projects mentioned earlier need the use of Kinect devices, but our main objective was to design a project that could be utilised using readily accessible resources. The sensor similar to Kinect now is no longer most effective isn’t effortlessly to be had however is also high priced for maximum of target market to shop for and our version makes use of ordinary webcam of the computer as a result it is beyond plus point.

APPLICATIONS:

- The National Deaf Association (NAD) estimates that there are 18 million deaf people within India. Therefore, benefit to significant portion of community can be benefitted by this project of the disabled by giving them the means to use sign language to interact with the outside world. This will result in the removal of the intermediary, who mostly serves as a translator. The model's simplicity makes it suitable for implementation in mobile applications, which is what we intend to do in the future.

- Deaf human beings do now no longer available with alternatives for speaking and listening to person, and all the options have most important flaws that Interpreters are not normally existing, and additionally should be costly. Our venture as cited earlier than is pretty within your means and calls for minimum quantity of management. Hence is pretty high quality in phrases of fee decrease.

- Using pens and paper is not a good option because it takes a long time and is uncomfortable and messy for both hearing and deaf people.

- The concept of this project can also be applied to the Deaf and Mute community in a variety of settings, such as airport (in security checks or communication while boarding or in aircraft), school & college (for educational purpose), doctor offices (to properly understand the illness) and community facility organisations and jury.

Conclusion Drawn:

In this work, an efficient real time and vision based ASL alphabet recognition system for Deaf & Mute users was developed. On our dataset, we ended up with a final precision of 95.8%. We have successfully increased the accuracy of our prediction after building algorithm with two layers by predicting and confirming symbols which are quite like to each other.

The goal of this project is to create a practical system that is useful for those who struggle with hearing and who generally rely on the highly straightforward and efficient approach of sign language. This technique may be used to translate between text and sign language as well. A gesture recognition system was released for text conversion. It records the indications and displays them as in textual form on the screen.

If symbols are presented properly, and noise is avoided along with the lighting is right, gestures can nearly always recognise them in this way.

This system offers high reliability and fast response.

Scope In Future:

Future development will focus on creating a mobile application for a system that makes it possible for everyone to communicate with deaf individuals.

By utilising new approaches, we intended to expand this concept for words and sentences. The entire concept of this study was intended to be applied to smart phones as well. Implementing the image processing technologies is the difficult part of putting this concept into a mobile phone.

1. Dumb individuals must rely on an interpreter or another form of visual communication since they typically lack the ability to communicate normally with others. This effort can assist reduce reliance on the interpreter because they are no longer constantly available.

2. To fully comprehend the context and tone of the input speech, the system may be expanded to add knowledge of body language and facial expressions as well.

3. In order to increase its reach a mobile or web based approach could help well.

4. Integrating computer vision with a hand gesture detection system to create a two-way communication system.

References:

[1] T. Yang, Y. Xu, and “A. , Hidden Markov Modelfor Gesture Recognition”, CMU-RI-TR-94 10, Robotics Institute, CarnegieMellon Univ.,Pittsburgh,PA, May 1994.

[2] Pujan Ziaie, Thomas M ¨uller , Mary Ellen Foster ,and Alois Knoll“A Na ¨ıve Bayes Munich,Dept. of Informatics VI, Robotics and Embedded Systems,Boltzmannstr. 3, DE-85748 Garching, Germany.

[3] https://docs.opencv.org/2.4/doc/tutorials/imgproc/gausi
an_median_blur_b
ilateral_filter/gausian_median_blur_bilateral_filter.html

[4] Mohammed Waleed Kalous, Machine recognition ofAuslan signs using PowerGloves: Towards large- lexicon recognition of sign language.

[5] aeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks-Part-2/

[6] http://www-i6.informatik.rwth-aachen.de/~dreuw/database.php

[7] Pigou L., Dieleman S., Kindermans PJ., Schrauwen B.(2015) Sign Language Recognition Using ConvolutionalNeural Networks. In: Agapito L., Bronstein M., Rother C.(eds) Computer Vision - ECCV 2014 Workshops. ECCV2014. Lecture Notes in Computer Science, vol 8925. Springer, Cham

[8] Zaki, M.M., Shaheen, S.I.: Sign language recognition using a combination of new vision based features. Pattern Recognition Letters 32(4),572–577 (2011)

[9] N. Mukai, N.Harada and Y.Chang,"Japanese Fingerspelling Recognition based on Classification Tree and Machine Learning," *2017 Nicograph International (NicoInt)*, Kyoto, Japan, 2017, pp.19-24. doi:10.1109/NICOInt.2017.9

[10] Byeongkeun Kang , Subarna Tripathi , Truong Q. Nguyen ”Real-time sign language fingerspelling recognitionusing convolutional neural networks from depth map” 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)

[11] https://opencv.org/

[12] https://en.wikipedia.org/wiki/TensorFlow

[13]https://en.wikipedia.org/wiki/Convolutional_neural_network

APPENDIX:

Open-CV

OpenCV is available for free for both commercial and academic usasge as it was published beneath the BSD licence. It is compatible with Linux, Windows, MacOS, ioS, and Android and offers interfaces in Python, C++, and Java.

Real-time applications and IT efficiency were given top priority when developing OpenCV. Multi-core processing can be advantageous for libraries written in C++ or optimised C. When OpenCL is enabled, OpenCV may benefit from hardware acceleration of the underlying heterogeneous computing platform.

More than 50,000 people use OpenCV, which has been downloaded more than 18 million times, according to estimates. OpenCV has been widely adopted by people all around the world. Applications include advanced robotics, mine surveying, web map stitching, and interactive art.

Convolution Neural Network

CNNs are also called spatial / shift invariant (SIANN) ANN, that is based on a weight-sharing architecture and a invariance of translation function. CNNs use several multi-layered perceptron’s that have been designed to fit with minimal preprocessing.

In that it resembles how an animal's visual cortex is set up, the organisation of the connections by biological processes has inspired neurons in convolution networks.

A small area of the optical field, is where individual cortical neurons respond to stimuli. Multiple neurons can span the full visual field because their receptive fields partially overlap. CNNs require a lot less preprocessing than other image classification techniques. This indicates that the network picks up previously created filters manually using conventional methods.

Its flexible structure allows for a clean distribution of computing on many platforms (CPU, GPU, and TPU) and from computing systems to server farms, mobile devices and edge devices.

Tensorflow

An open source application package called Tensor Flow is used to programming dataflow via a variety of tasks. It is a representative maths library, to be used for more various machine learnings presentations like neural network. Used mainly on researches and productions at Google's research labs.

TensorFlow has been developed by brain team of Google for Googles interior usage.

Tensor flow's supple design helps to simply deployment of computations through diversity of platform (CPU's, TPU's, GPU's), & from desktop to the cluster of server's to mobiles and edged devices.

Tensor flow has been Google Brains 2nd gen. system. Version 1.0.0 that has been published on 11th February 2017. Whereas the reference implementations run on sole device. Tensor Flow has been capable of running on multiple CPU's and GPU's (along with the optional SYCL and CUDA extension for the general purpose computing on GPU's). Tensor Flow is currently accessible on 64-bit mac OS, Windows, Linux, and various mobiles supported by iOS and Android.

Thanking you,

AVTAR CHANDRA