VOICE OF DEAF

Submitted by

Mon Mon Jaman Jecee

ID: 181-35-2490

Department of Software Engineering

Daffodil International University

Supervised by

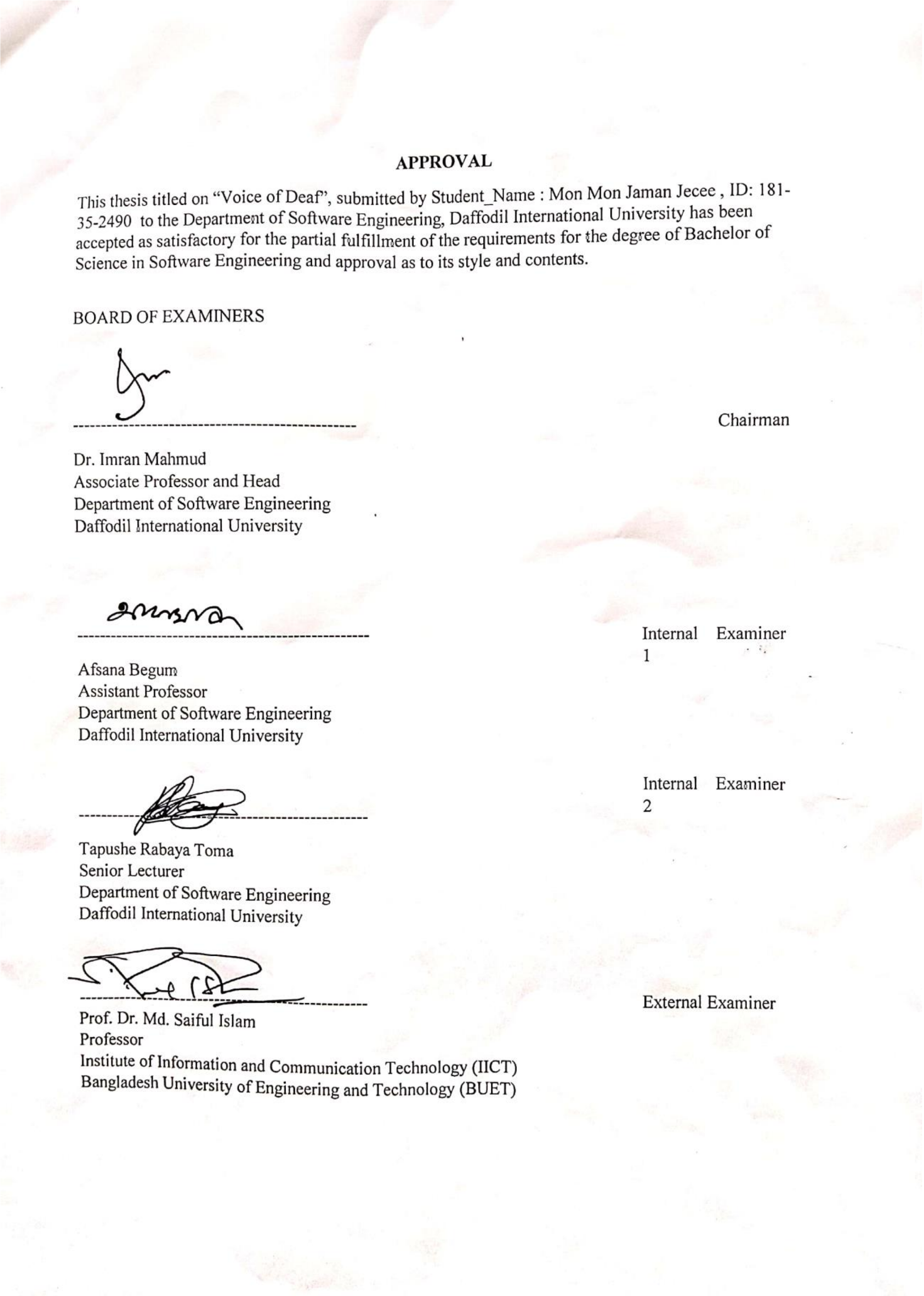
Ms. Syeda Sumbul Hossain

Lecturer (Senior Scale)

Department of Software Engineering

Daffodil International University

This Thesis report has been submitted in fulfillment of the requirements for the Degree of Bachelor of Science in Software Engineering.



APPROVAL

This thesis titled on "Voice of Deaf", submitted by Student_Name : Mon Mon Jaman Jecee , ID: 181-35-2490 to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering and approval as to its style and contents.

BOARD OF EXAMINERS

Chairman

Dr. Imran Mahmud
Associate Professor and Head
Department of Software Engineering
Daffodil International University

Internal Examiner 1

Afsana Begum
Assistant Professor
Department of Software Engineering
Daffodil International University

Internal Examiner 2

Tapushe Rabaya Toma
Senior Lecturer
Department of Software Engineering
Daffodil International University

External Examiner

Prof. Dr. Md. Saiful Islam
Professor
Institute of Information and Communication Technology (IICT)
Bangladesh University of Engineering and Technology (BUET)

DECLARATION

We hereby declare that, this project has been done by us under the supervision of **Ms. Syeda Sumbul Hossain, Department of Software Engineering,** Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:

Ms. Syeda Sumbul Hossain
Lecturer (Senior Scale)
Department of Software Engineering
Faculty of Science & Information Technology
Daffodil International University

Submitted by:

MonMonuzaman

Mon Mon Jaman Jecee
ID: 181-35-2490
Batch: 25th
Department of Software Engineering
Faculty of Science & Information Technology
Daffodil International University.

ACKNOWLEDGEMENT

At first I want to thank to my Almighty to give me the opportunity to fulfill my thesis successfully.

I really grateful and wish my profound indebtedness to Ms. Syeda Sumbul Hossain, Lecturer (Senior Scale), Department of Software Engineering, Daffodil International University, Dhaka. Deep Knowledge & keen interest of my supervisor in the field of "Machine Learning" to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stage have made it possible to complete this project.

I would like to express my heartiest gratitude to Prof. Dr. Imran Mahmud, Professor and our honorable Head, Department of Software Engineering, for his kind help to finish our project and also to other faculty member and the staff of SWE department of Daffodil International University.

I would like to express our gratitude to our Co-supervisor, for his remarkable guidance.

I would like to thank my entire course mate in Daffodil International University, who took part in this discuss while completing the course work.

Finally, I must acknowledge with due respect the constant support and patients of my parents.

ABSTRACT

Sign language is the oldest and best form of expressing the language of the mind. Around 466 million people have disabling hearing loss, and 80% of auditory impaired people are illiterate or semi-literate. And most of them exclusively use dactylology to communicate with the world. But most of us do not know dactylology, and interpreters find it very difficult to express that dactylology to mundane people. As a result, we aim to create an authentic-time method for finger spelling-based dactylology based on a neural network. In this work, the designation languages are passed through a filter, and after the filter is applied to the hand gesture, it passes through a process that shows the text of the gesture. This project gives pretty accurate results.

TABLE OF CONTENTS

LIST OF ABBBREVIATIONS

(D&M)- Deaf and Mute

(OpenCV)- Open computer vision

(PC) - Personal Computer

(ANN)- Artificial Neural Network

(CNN)- Convolution Neural Network

(HCI)- Human-Computer Interface

(ASL)- American Sign Language

(FSL)- French Sign Language

(BSL)- British Sign Language

(KNN)- K-Nearest Neighbor
(HOG)- Histograms of Organized angles

(RGB)- Red Green Blue

(SVM)- Support vector machine

(FEMD)- Finger-Earth Mover's Distance

(ReLu)- Rectified Linear Unit

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Introduction

A language that uses signs made with the hands and other gestures used mainly by people who are deaf, including facial expressions and body postures. American Sign Language (Language acquisition by eye et al., 2013) is only the predominant language to communicate with the D&M people of our society. D&M people can only communicate with us using their hand gestures to express their expression with other people.

A human can easily recognize and separate any object. Now computers have the same ability. Gesture recognition (Procedia Computer Science el at., 2017) is also an ability from the computer science field and its deal with interpreting human gestures. Gesture recognition allows users to communicate with machines without the use of any devices.

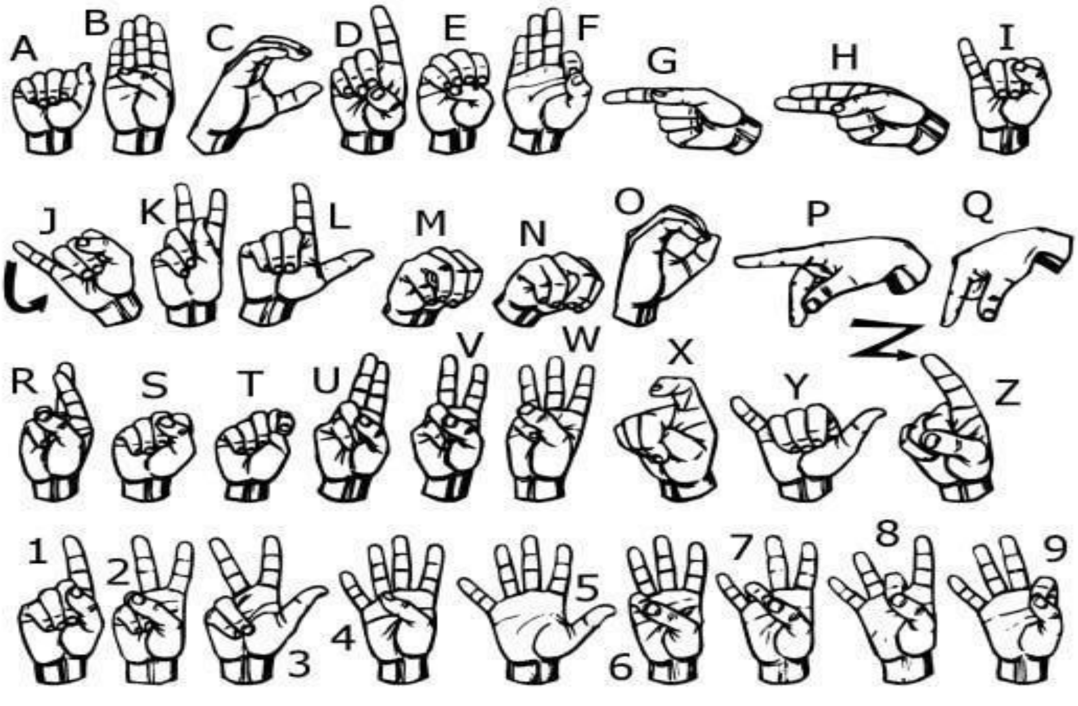


Fig: 1.1 Sign Language Symbol

1.2 Objective

The hand is the most important element for deaf and hearing aids people, those who depend on their hands for communication. Utilizing unsupervised feature learning to abstract communication barriers with the auditory perception impaired and provide edifying assistance for dactylology, to translate signed hand signals into the text as well as voice.

Now the computer can translate and understand hand gestures, which will be a paramount advancement in the area of human-computer interaction.
The purpose of my project that will converting captured images via a vision-based sensor and apply different image processing steps and predict and detect the hand gesture, which will increase the interaction between deaf and dumb people with normal people for communicating.

1.3 Motivation

As a consequence, they rely on vision-based coordination for touch.

The forms of kineticism can be rarely interpreted by others if there is a standard project that translates dactylology to text. As a result, research has been conducted on a vision-predicated project that will enable D&M people to communicate without having to verbalize the same communication system.

The objective is to engender a facile-to-use human-computer interface in which the project perceives human dactylology. There are many sign languages used throughout the world, including American Dactylology, French Dactylology, British Dactylology, Indian Dactylology, and Japanese Dactylology, among others.

CHAPTER 2

LITERATURE SURVEY

2.1 Introduction

If we try to find out hand gesture recognition has been the subject of extensive study in recent years. In this chapter, I cover an important literature review about this project related to hand gestures. This chapter will cover the literature associated with hand gesture recognition, machine learning process, and image processing system. The literature studies are also involved to develop this type of similar system.

2.2 Related Work

Paper Name	Year	Methods/Algorithm	Accuracy	Major contribution	Summary
sign language into text conversion using deep learning [1]	2013	CNN architecture, Tensor flow layers, Deep learning	98.68% for alphabets and 90% for validation	For the representation they used CNN, is a part of deep learning. In American Sign Language, the alphabets (A-Z) are formed by 24 static gestures and two dynamic gestures.	98.68% accuracy for alphabets and 90% for validation . So, it was almost a good work and did it successfully

Hand Gesture Recognition with Depth Image [2]	The 21st IEEE International Symposium.	They used 13 methods for hand localization and 11 more for gesture classification. Eight applications were used to test the gesture recognition system. However, three of the applications accounted for more than 75%. Five different kinds of depth sensors were used, but only the Kinect sensor was used.	75%	They used 13 methods for hand localization and 11 more for gesture classification. Eight applications were used to test the gesture recognition system. However, three of the applications accounted for more than 75%. Five different kinds of depth sensors were used, but only the Kinect sensor was used. They used the technique properly and did the work successfully.	The whole survey summarizes the techniques that have been used for hand localization and gesture classification in the gesture recognition but shows that very little variety has been seen in the real-world applications used to test these techniques.
Conversion of Sign Language into Text [3]	2018	LDA algorithm, Euclidean Distance (E.D), Eigen Values, Eigen Vectors	98.67%	Deaf people are isolated from people who are not deaf because they never learn sign language. However, if the computer could convert sign language into text, then the	A future version of this project will determine the numbers that will be shown in words.

				difference between the deaf and the hearing would be significantly reduced.	
Hand gesture recognition using a real-time tracking method and HMM[4]	2003	a real-time hand tracking and extraction algorithm, e Fourier descriptor, HMM	85%	They tried to develop a method to recognize the sign gestures using HMMs. Hand gestures typically have quite a lot of variation, therefore in order to achieve accurate hand tracking it is necessary to transition between states once in a while. So, they applied this system to recognize the single gesture. It's a great contribution	They were able to develop a method that recognized the unknown sign and a lower error rate could be implemented.
Real time conversion of sign language to speech and prediction of gestures using ANN [5]	2018	Back propagation algorithm, ANN, Arduino Uno, SIM900A GSM module	85%	In expansion to changing over American Sign Dialect to discourse in real-time, this demonstrate moreover predicts desires of quiet individuals, and future upgrades can be made so that the anticipation comes to quiet	In this work, affiliation of different needs of quiet individuals with values of flex sensor is done and they were able to anticipate their needs utilizing back-propagation neural organize.

				people's portable gadgets, so they can confirm whether it's precise or not.	
SIGN LANGUAGE CONVERTER [6]	2015	Movement Capture, Motioned Picture, Sign Dialect Converter, Voice Recognition 1-Database 2-Voice Acknowledgment Procedure 3-Motion Capture Strategy	75%	It took the acoustic voice flag and changes over it to an advanced flag in computer and after that appear to the client the .gif pictures as result. Besides, the movement acknowledgment portion employments picture handling strategies.	They were able to develop a method that could establish a strong communication between deaf and ordinary people.
Dynamic Hand Gestures Recognition System with Natural Hand [7]	2013	Hand motion interface, HCI, Computer vision, Fluffy Clustering, ANN, HMM, Introduction Histogram.	98.75%	Building an effective human-machine interaction is an important goal of motion acknowledgment framework. Numerous applications of gesture recognition framework extending. The major tools surveyed incorporate HMMs, ANN, and fluffy clustering have been reviewed and analyzed.	They were able to perceive the method framework and were additionally presented with a demonstration of the three main phases by detection of the hand, extraction of the highlights, and acknowledgment in gesture.

Hand-Gesture Recognition in a Human-Robot Dialog System [8]	2008	Picture Preparing, Signal Acknowledgment, Bayes Hypothesis, K-Nearest-Neighbors, Hu Moments	95%	It occurs for two invariant classes, namely defined and Hu invariants, in combination with well-suited values for the framework's parameters, resulting in a redress recognition rate of more than 95.0 percent for three signals.	According to the graphical representation of the execution, not only did the correct acknowledgment rate increase when compared to single classifications, but there were also fewer fluctuations within the outcomes.
Generic System for Human-Computer Gesture Interaction [9]	2015	HMI, Gesture recognition, Computer vision, ML	99.4%	They were able to use the technology with any human-machine interaction interface. Computer vision-based techniques offer the benefit of being non-invasive and based on how people perceive data from their surroundings.	With the executed applications, it was conceivable to demonstrate that the center of vision-based interaction systems can be the same for all application, and that the proposed nonexclusive framework engineering is a solid establishment for the advancement of hand signal acknowledgment frameworks that can be coordinates in any human-machine interface application.

Hand Gesture Recognition System [10]	2011	Hand gesture recognition, skin detection, Hand tracking.	94%	The paper presented a hand motion recognition system to recognize "dynamic gestures" in which a single gesture is performed on a complex foundation. Our framework, like previous motion recognition frameworks, employs glove.2D video input as if it were.	They have implemented a real-time adaptation, utilizing workstations with no extraordinary equipment past a video camera input.
Robust part-based hand gesture recognition [11]	2013	Finger-Earth Mover's Distance, hand gesture recognition, human-computer interaction, Kinect system.	93.2%	The experiments demonstrate that their hand motion acknowledgment framework is accurate, proficient, strong to hand articulations, distortions and introduction or scale changes, and can work in uncontrolled situations. The prevalence of our framework is assist demonstrated in two real-life HCI applications.	They were able to build a framework that appears hand signal acknowledgment procedure which can mimic the communications between human, and include hand gesture as a common and natural way to connected with machines.

Hand Gesture using Sign Language through Computer Interfacing [12]	2014	Finger Spelled Word Recognition, Hu Moment Invariant, HMM, Sign Language, SVM.	96%	A programmed ISL recognition system has been developed in this research, which operates in real-time and makes use of a combinational inclusion vector with an MSVM classifier. Hu invariant moment and fundamental shape descriptors are used in the combinational highlight vector to achieve better recognition results. The use of MSVM improves acknowledgment performance.	As a result, the combination of invariant minutes and form descriptors produces a far superior result, as shape descriptors characterize a picture's boundaries, whereas invariant moments are invariant to changes in scale and location.

2.3Key Words and Definitions

1. Artificial Neural Networks (ANN): An artificial neural network is a component of a computational system that models data interpretation and processing in the human brain. Artificial intelligence (AI) is built on top of this basis, and it solves issues that would be impossible or difficult to solve with human or statistical factors. As additional data becomes available, the artificial neural network's self-learning skills allow it to improve its efficiency.

How it works:

An ANN usually consists of a large number of processors that function in parallel and are organized into tiers. The first level receives raw input data in a manner that resembles optic neurons in human visual processing. Each progressive level receives the yield from the tier preceding it rather than the raw input, in the same way as neurons in the visual nerve receive signals from those closest to it. The system's yield is determined at the ultimate level. Each handling hub has its own claim to a small circle of data, counting what it has seen and any rules it was programmed with or made up for itself. Because the levels are so interlinked, each hub in level n will be linked to a number of hubs in level n-1.

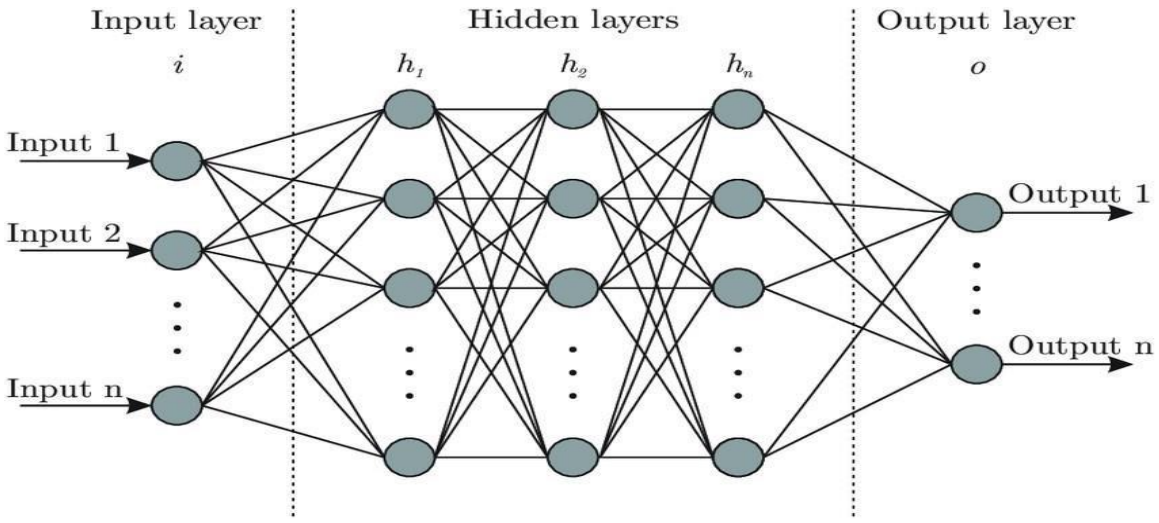


Fig 2.1 Artificial Neural Networks (ANN)

Reference: Wang SC. (2003) Artificial Neural Network. In: Interdisciplinary Computing in Java Programming. The Springer International Series in Engineering and Computer Science, vol 743. Springer, Boston, MA. https://doi.org/10.1007/978-1-4615-0377-4_5

2. Convolution Neural Network (CNN): Convolutional neural networks (CNNs) are neural networks that have one or more convolutional layers and are generally utilized for image segmentation, classification, and processing, as well as other auto correlated data. The technique of sliding a filter over an input signal is known as convolution. Using the CNN architecture, we will reduce the complete image to a single vector of class.

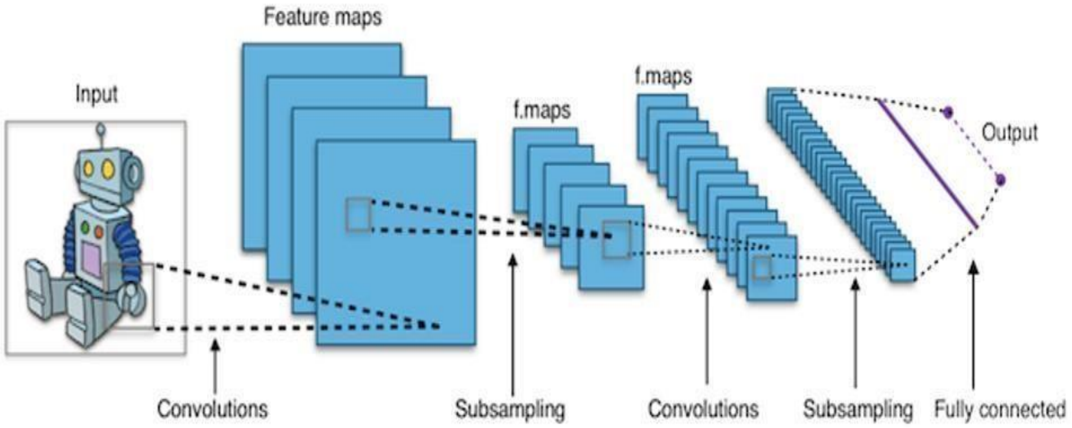


Fig: 2.2 Convolution Neural Network (CNN)

How does it work?

i. Convolution Layer:

A convolution layer's pixels are combined into a single value. By using a convolution layer on an image, we may reduce the image size while simultaneously compressing all of the information in the field into a single pixel. The final output of the convolutional layer is a vector.
The convolutional layer is the CNN's central building square, and it is here that the majority of the computation takes place. It requires input information, a channel, and a highlight outline, among other things. Let's assume the input is a color image made up of a 3D framework of pixels.

This means the input will have three dimensions, similar to RGB in a picture: height, width, and depth. We also have an include locator, sometimes known as a channel, that can navigate over the responsive portions of the image and check if the highlight is visible. This preparation is known as a "convolution."

A two-dimensional (2-D) cluster of weights that speaks to a section of the image could be the feature. The channel estimate is usually a 3x3 framework, though it can vary in size; this also determines the response field estimate. At that moment, the channel is linked to a picture region, and a speck item is calculated between the input pixels and the channel. At that point, the dab item is nurtured into a yield cluster. The filter changes by a short time later, rehashing the technique until the part has cleared throughout the entire picture. A highlight outline, also known as an enactment outline, is the final result of arranging small objects from the input and the channel.

1. The depth of the output value is influenced by the number of filters used. Three unique highlight maps, for example, might result from three distinct filters, yielding a total of three outputs.

2. "Walk" is the number of pixels that the part moves over the input network in a single step. A larger walk provides a little output, whereas walk values of two or more noteworthy are uncommon.

3. When the channels don't fit the input picture, zero-padding is usually used. This sets all components that drop the exterior of the input lattice to zero, creating a bigger or similarly measured yield.

There are three sorts of padding:

Valid padding: This is often moreover known as no padding. In this case, the final convolution is dropped in case measurements don't align.

Same padding: This padding guarantees that the yield layer has the same measure as the input layer.

Full padding: This sort of padding increments the measure of the yield by including zeros to the border of the input.

After completing the operation, a convolution neural network applies a Rectified Linear Unit (ReLU) change to the highlight outline, presenting nonlinearity to the model.

Ultimately, the convolutional layer changes over the picture into numerical values, permitting the neural arrange to translate and extricate pertinent designs.

Reference: Kim P. (2017) Convolutional Neural Network. In: MATLAB Deep Learning. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-2845-6_6

ii. **Pooling Layer:** The pooling layer is used to minimize the representation's spatial size. In addition, the parameters have been lowered.

a) **Max Pooling:**

By providing an abstracted representation of the data, maximum pooling aids over-fitting. It also reduces the computational cost by reducing the number of parameters that must be learned and gives the internal representation simple translation invariance.

b) **Average Pooling:** We take the average of all values in average pooling.

We can see in the figure how max and average pooling work:

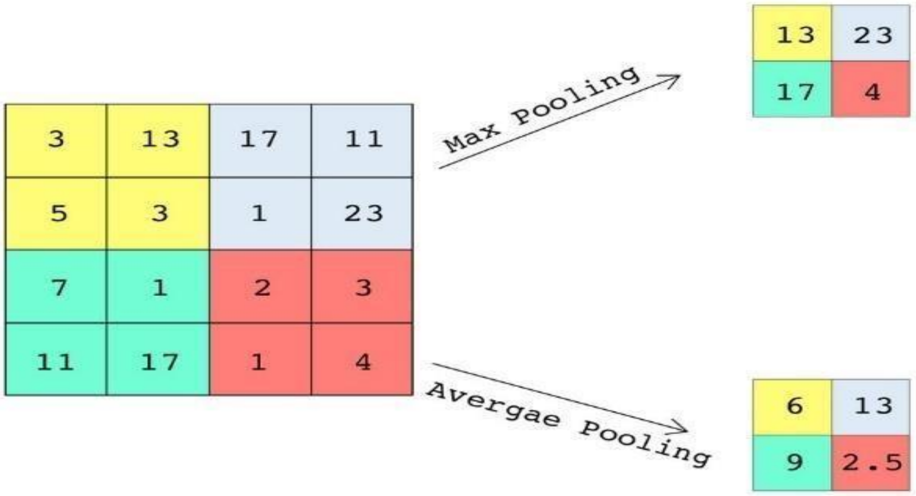


Fig: 2.3 pooling layer

Whereas a parcel of data is misplaced within the pooling layer, it moreover encompasses several benefits to the CNN. They offer assistance to decrease complexity, make strides in proficiency, and constrain the hazard of overfitting.

Reference: https://doi.org/10.1016/j.neucom.2016.10.049

iii. Fully Connected Layer: The Fully Connected Layer is made up of feedforward neural networks. The network's last tiers are the Fully Connected Layers. The input to the fully connected layer is the data from the final pooling or convolutional layer, which is flattened and then fed into the fully connected layer.

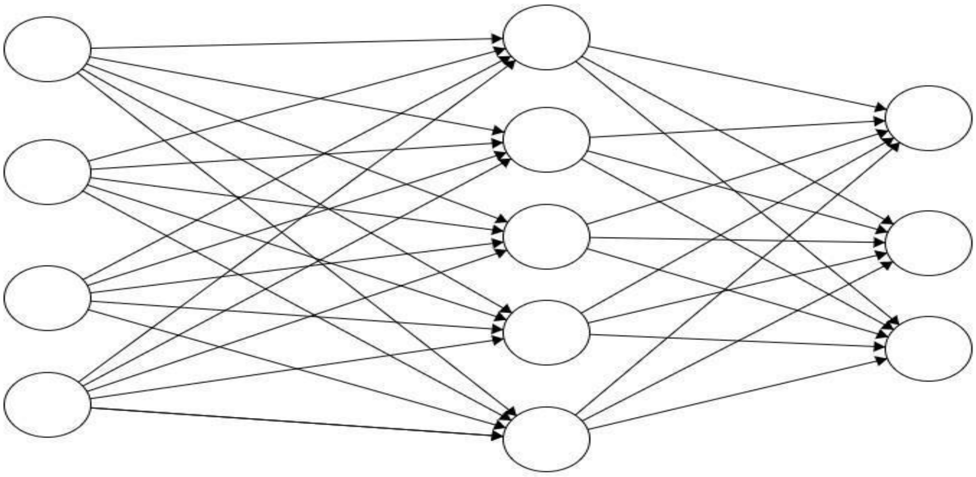


Fig: 2.4 Fully Connected Layer

iii. **Final Output Layer:**

Then we connect the values we acquired from the completely connected layer to the final output layer after that. It forecasts the likelihood of photos from various classes.

Reference: https://doi.org/10.1016/j.neucom.2019.10.008

3. **OpenCV:** OpenCV is a cross-platform library that can be used to build real-time computer vision apps. It focuses primarily on image recognition, video capturing, and interpretation, with features such as face detection and object detection. It is written in C++ which is its primary interface, however, bindings are available for Python, Java, and MATLAB/OCTAVE. In python, OpenCV uses utilize Numpy. All the OpenCV cluster structures are changed over to and from Numpy clusters. This too makes it simpler to coordinate with other libraries that utilize Numpy such as SciPy and Matplotlib.

Reference: I. Culjak, D. Abram, T. Pribanic, H. Dzapo and M. Cifrek, "A brief introduction to OpenCV," *2012 Proceedings of the 35th International Convention MIPRO*, 2012, pp. 1725-1730.

https://ieeexplore.ieee.org/abstract/document/6240859/citations#citations

4. **Keras:** Keras is a high-level interface. It's a human-centric API, not a machine-centric one. It almost supports all the models of a neural network. Thea no and Tensor Flow is its backend. Keras is the foremost utilized profound learning framework. Because Keras makes it less demanding to run unused experiments, it enables to undertake more thoughts than the competition, speedier. Keras is utilized by CERN, NASA, NIH, and numerous more logical organizations around the world. Keras has the low-level adaptability to actualize subjective inquire about thoughts whereas advertising discretionary high-level comfort highlights to speed up experimentation cycles.

   Reference: Manaswi N.K. (2018) Understanding and Working with Keras. In: Deep Learning with Applications Using Python. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-3516-4_2

5. **TensorFlow:** TensorFlow is a Google-developed open-source library for deep learning and machine learning applications. It's mainly developed for large numerical computations. And it's used as Keras backend. TensorFlow can create and operate deep neural networks for manually written digit classification, image recognition, word embedding, repeating neural networks, machine interpretation, sequence-to-sequence models, normal dialect preparation, and PDE-based recreations. TensorFlow underlies large-scale generation forecasting with the same models used for preparation.

How it works:

Dataflow graphs—structures that illustrate how information passes across a chart or an arrangement of processing hubs—are possible with TensorFlow. Each hub in the chart represents a mathematical operation, and each link or edge between hubs could be a tensor or multidimensional information cluster. The Python dialect of TensorFlow delivers all of this to the software engineer. Python is easy to learn and use, and it provides useful examples of how high-level reflections might be linked together.

Python, on the other hand, does not do real math operations.TensorFlow provides access to change libraries that are made up of high-performance C++ parallels. Python synchronizes activity among the components and provides high-level programming interfaces to bind them together. TensorFlow applications can be run on nearly any computer, including a local machine, a cloud cluster, iOS and Android devices, CPUs, and GPUs. If you use Google's claim cloud, TensorFlow will be run on Google's proprietary TensorFlow Handling Unit (TPU) silicon, which will help with speed.

TensorFlow's next models, on the other hand, may be delivered to almost any device and used to meet customer expectations.

Reference: Seetala K., Birdsong W., Reddy Y.B. (2019) Image Classification Using TensorFlow. In: Latifi S. (eds) 16th International Conference on Information Technology-New Generations (ITNG 2019). Advances in Intelligent Systems and Computing, vol 800. Springer, Cham. https://doi.org/10.1007/978-3-030-14070-0

CHAPTER 3

METHODOLOGY

3.1 Introduction

If we see there have been numerous researches in this field with several methodologies used. A project is a vision-based approach. All the signs are made with bare hands, which removes the need for any artificial equipment for contact with deaf and hearing aid people.

3.2 The method I followed to build this system:

At first, I collected the training and testing data set. After that made a label for training the data set. Then plot the quantities in each class.
Then I do preprocess step on those images by following those steps

- Dropped the training labels from training data to separate them.
- Extracted the images from each row in CSV
- Using a threshold value, converted the picture to black and white.
- Reshaped images by TensorFlow and Keras.
- The Modified Moore Neighbor Contour Tracing algorithm was used to retrieve features.

- Created CNN model.
- Trained the model
- Reshaped test data
- Measured the accuracy of test and train data.
- Created an error handling function to match the label to letter.
- Finally tested the actual webcam input.

Identified sign letter by matching and recognizing the associated gesture

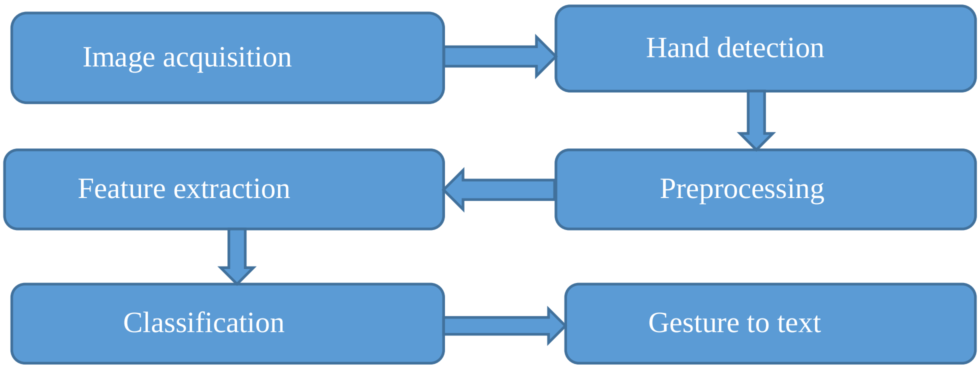


Fig: 3.1 Steps of image processing

PROJECT STEPS

3.2.1 Data Set Collection

For developing this type of project, we need to collect a huge amount of data. And it’s so difficult to find those data that fulfill my requirements. So, I collected my data set from Kaggle. I choose the appropriate data set which helped to build my coding section successfully.
I am adding my data set link below:

Testing data set:

https://github.com/Jecee35/Thesis-code_-Sign_language_to_text_conversion/blob/main/sign_mnist_test.csv

Training data set:

https://github.com/Jecee35/Thesis-code_-Sign_language_to_text_conversion/blob/main/sign_mnist_train.zip

The dataset organize is designed to coordinate closely with the classic MNIST. Each preparing and test case speaks to a name (0-25) as a one-to-one outline for each alphabetic letter A-Z (and no cases for 9=J or 25=Z since of motion movements). The preparing information (27,455 cases) and test information (7172 cases) are around half the measure of the standard MNIST but something else comparable with a header push of name, pixel1,pixel2….pixel784 which speak to a single 28x28 pixel picture with grayscale values between 0-255. The first-hand signal picture information is spoken to numerous users rehashing the signal against diverse foundations. The Sign Dialect MNIST information came from incredibly expanding the little number (1704) of the color pictures included as not trim around the hand locale of intrigued. To make modern information, a picture pipeline was utilized based on ImageMagick and included trimming to hands-only, gray-scaling, resizing, and after that making at slightest 50+ varieties to extend the amount.
The modification and extension strategy was channels , along with 5% sporadic pixilation, 15% brightness/contrast, and at long final 3 degrees turn. Because of the minor measure of the pictures, these adjustments viably change the determination and lesson division in curiously, controllable ways. This dataset was motivated by the Fashion-MNIST 2 and the machine learning pipeline for signals by Sreehari 4. A strong visual acknowledgment calculation might give not as it were modern benchmarks that challenge cutting edge machine learning strategies such as Convolutional Neural Nets but moreover seem practically offer assistance the hard of hearing and hard-of-hearing superior communication utilizing computer vision applications.
The National Institute on Deafness and other Communications Disarranges (NIDCD) demonstrates that the 200-year-old American Sign Dialect may be a total, complex

language (of which letter signals are as it were portion) but is the essential dialect for numerous hard of hearing North Americans. One seems to execute computer vision in a cheap board computer like Raspberry Pi with OpenCV, and a few Text-to-Speech to empowering moved forward and robotized interpretation applications.

3.2.2 Dropped the training labels from training data to separate it

Here, used the drop data frame to separate the training data set in the column. Here axis refers to the dimensional array. In the Data Frame axis = 0 is the point downwards and axis= 1 is the point to the right.

3.2.3 Extracted the images from each row in CSV

This is used because the CSV.reader strategy that we called has naturally changed over each push of the record into a Python list. This makes it simple to get to specific components of the CSV record. If we utilize the regular Python sentence structure for getting to a component of the list–here our area is the column, but computers continuously tally beginning with 0 – so the row should deliver us our areas. We are able to print fair the areas as we did the whole lines of the CSV record.

3.2.4 Using a threshold value, converted the picture to black and white

Here I applied the threshold technique using OpenCV.
We know that threshold is a technique of openCV what is the assignment of pixel values with the threshold value given. Here, every pixel's value is compared with the threshold value. If the value of the pixel is smaller than the threshold's value then it's set to 0, else, it's set to a maximum value.

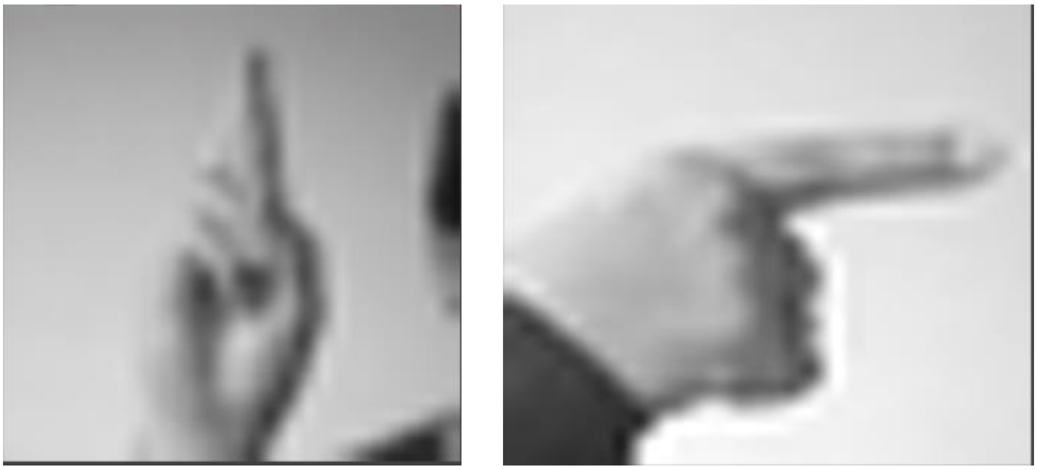


Fig: 3.2 Output picture (black and white)

Applying the threshold technique, I converted all pictures into black and white.

3.2.5 Reshaped images by TensorFlow and Keras.

In this section, using Keras and TensorFlow I reshaped the images. I need to reshape the images into the sizes for being ready for the model.

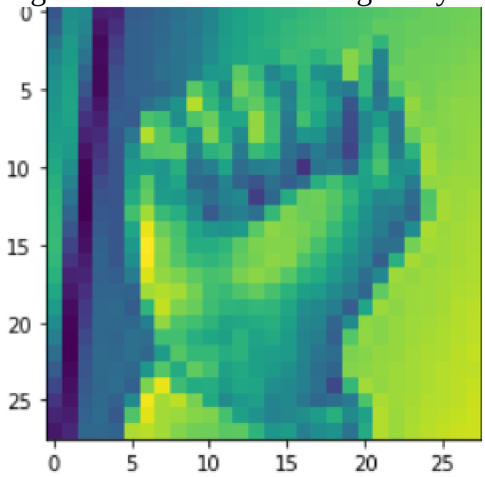


Fig: 3.3 reshaped image

3.2.6 Used train test split function to split the data

I used the function here because a few models are exceptionally costly to prepare, and in that case, rehashed evaluation utilized in other strategies is intractable. An illustration may well be profound neural network models. In this case, the train-test method is commonly used. Alternately, a venture may have a productive show and an endless dataset, even though may require an assessment of show execution rapidly. Once more, the train-test split procedure is drawn nearer in this situation. Samples from the initial preparing dataset are part of the two subsets utilizing arbitrary choice. Usually to guarantee that the prepare and test datasets are an agent of the initial dataset.

3.2.7 The Modified Moore Neighbor Contour tracking algorithm was used to retrieve features

For instance, in one image, a grid of dark pixels on a white background, pick a dark pixel and declare it as your "starting" pixel. (Finding a "start" pixel can be done in a variety of ways; I started at the bottom left corner of the grid and examined each column of pixels from the bottom up, starting with the leftmost column and working my way to the right, until I found a black pixel.) We're ready to declare that pixel as our "start" pixel.)

	P1	P2	P3	
	P8	P	P4	
	P7	P6	P5	

Fig: 3.4 Image’s pixel

Now envision yourself as a bug standing on the first pixel. We'll disentangle the form without the misfortune of simplification by traveling clockwise around the design. The widespread belief is that whenever you strike a dark pixel, press P to retrace. Return to the white pixel you were standing on and proceed clockwise around pixel P, passing by each pixel in its Moore neighborhood until you reach a dark pixel. When a pixel passes by for a short period of time, the calculation comes to an end. The pattern's form will be those black pixels you walked over.

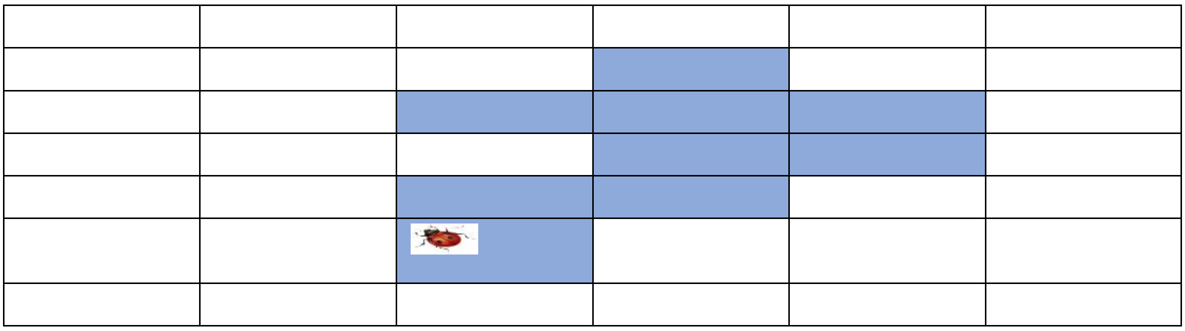


Fig: 3.5 image's pixel with bug explanation

```
Start
Set B to be empty.
From foot to best and cleared out to right filter the cells of T until a dark pixel, s, of P is found.
Insert s in B.
p=s
Backtrack
Set c to be the another clockwise pixel in M(p).
While c not rise to  s do
In case c is black
Insert c in B
Set p=c
backtrack
 else
advance the current pixel c to the another clockwise pixel in M(p) end While
 End
```

3.2.8 The image classification algorithm

Image classification is the task of categorizing and imposing labels on a group of pixels. The categorization law can be connected through one or different spectral or textural characterizations.

A computer analyzes a picture within the shape of pixels. It does it by considering the picture as a cluster of frameworks with the estimate of the network dependent on the picture determination. Put essentially, picture classification in a computer's see is the examination of this measurable information utilizing calculations. In computerized picture preparation, picture classification is done by consequently gathering pixels into indicated categories, so-called "classes."

The calculations isolate the picture into an arrangement of its most conspicuous highlights, bringing down the workload on the ultimate classifier. These characteristics grant the classifier and thought of what the picture speaks to and what course it may be considered into. The characteristic extraction handle makes up the foremost vital step in categorizing a picture as the rest of the steps depend on it.

3.2.9 The Cam Shift Algorithm

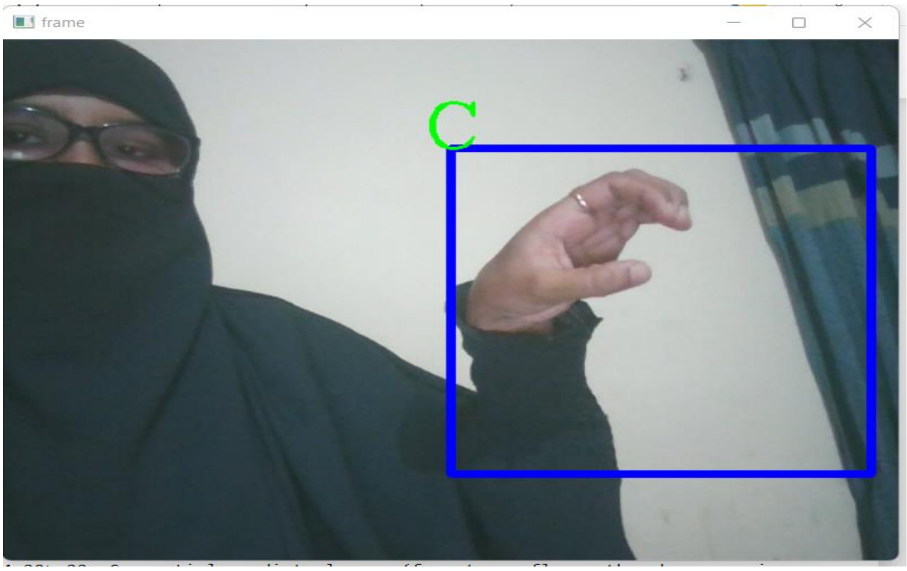


Fig: 3.6 understanding cam shift algorithm (final output)

Did you closely observe the final result? There is an issue. Our window continuously has the same size whether the hand is very far or very near to the camera. That’s not great. We got to adjust the window size with the size and rotation of the target. Again, it’s coming from "OpenCV Labs" that’s called CAMshift.

At this point, it calculates the most excellent fitting oval to it and once more applies the mean-shift with the recently scaled look window and the previous window. This handle proceeds until the desired exactness is met.

3.3 Gesture Classification

For developing this system, I used two layers of the algorithm to predict the final sign that a user gives to it. Those layers are given below:

Using Algorithm Layer:

1st Layer: After feature extraction, apply a Gaussian blur filter and a threshold to the image captured with OpenCV to get the filtered image. This filtered image is fed into the CNN model for estimation, and if a letter is identified for more than 50 frames, the letter is printed and used in the word-formation. The blank sign is used to represent the space between the sentences.

2nd Layer: The layer, therefore, detects different collections of symbols that provide identical effects when detected. The layer then uses classifiers created specifically for such sets to classify between them.

3.4 CNN Model

Convolutional neural networks (CNNs) are neural networks containing one or more convolutional layers for recognizing, classifying, segmenting, and analyzing auto-correlated data. The technique of applying a filter to an input signal is known as convolution.

1) **First Convolution Layer:** The input image has a resolution of 128x128 pixels. In the first convolutional layer, it is processed using 64 filter weights (3x3 pixels each). Each of the b filter-weights now has a pixel representation of 126X126 pixels.

2) **First Pooling Layer:** We use 2x2 max pooling to down sample the photos, which means we keep the highest value in the array's 2x2 rectangle. The image has been shrunk to 63x63 pixels as a result.

3) **2nd Convolution Layer:** The first pooling layer's output of 63 x 63 pixels is now used as an input to the second convolutional layer. To tackle it, the second convolutional layer employs 64 filter weights (3x3 pixels each). As a result, it receives a file with a resolution of 60 by 60 pixels.

4) **Second Pooling Layer:** The files are then down sampled with a maximum pool of 2x2 and compressed to a resolution of 30 x 30 pixels.

5) **First Densely Connected Layer:** These images are now entered into a 128-neuron totally connected layer, and the second convolutional layer's output is reshaped into a 30x30x64 =57600-value array. The input to this layer is a 57600-value array. These layers'

contributions are received by the 2nd densely connected layer. To prevent overfitting, we employ a dropout layer with a value of 0.20.

3) Final Layer: The output of the first densely connected layer is passed into the final layer, which has the same number of neurons as the number of groups we're categorizing (alphabets + blank symbol).

3.5 Training and Testing

I collected data sets and converted those raw images into grayscale and apply Gaussian blur to reduce noises. I applied adaptive threshold on my datasets images to extract my hand from the background. I also reduce my image resolution to 128*128. After executing all of the above actions, I give the preprocessed input photos to the model for training and testing.

The likelihood of the image falling into one of the groups is calculated by the prediction method. As a result, the performance is scaled from 0 to 1, with the sum of each class's values equaling 1. I was able to accomplish this by using the SoftMax function.

The contribution of the prediction layer will be slightly different from the actual value at first. To improve it, I trained the networks with tagged data. The efficiency metric cross-entropy is used in categorization. It is a continuous equation that is positive when the value differs from the labeled value and zero when the value differs from the labeled value.

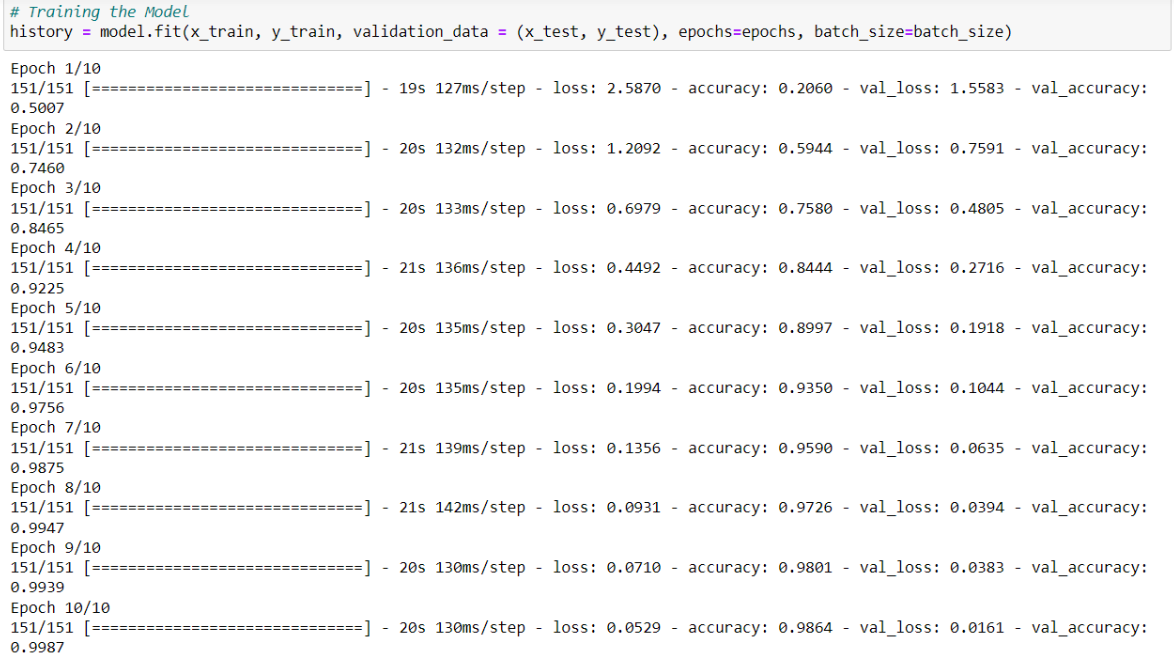
CHAPTER 4

RESULT

4.1 Results and Discussion:

I obtained an accuracy of 84.53 percent using a mixture of layer 1 and layer 2 of my algorithm, which is a pretty good research paper on sign language. The vast majority of academic papers focus on the use of Kinect devices, like instruments to detect hands. In [4], they use a real-time hand tracking hand gesture system using Hidden Markov Model with a 15 % error rate. [5] Uses a backpropagation algorithm in American Sign Language with a 15 % error rate. In [6], they reach an overall precision of 75% for the sign language converter system. Chart [11] reached an accuracy of 93.2 percent for observed and here they used Kinect devices to hand gesture recognition. In chart [12] they reached almost 96 percent accuracy where the error rate is 4 percent. Here they applied Hidden Markov Model and Support Vector Machine to make their system successfully.

To train the model I make 10 epochs to get better accuracy. And finally, it gives a better accuracy value of 99.87%.



```
# Training the Model
history = model.fit(x_train, y_train, validation_data = (x_test, y_test), epochs=epochs, batch_size=batch_size)
```

```
Epoch 1/10
151/151 [==============================] - 19s 127ms/step - loss: 2.5870 - accuracy: 0.2060 - val_loss: 1.5583 - val_accuracy:
0.5007
Epoch 2/10
151/151 [==============================] - 20s 132ms/step - loss: 1.2092 - accuracy: 0.5944 - val_loss: 0.7591 - val_accuracy:
0.7460
Epoch 3/10
151/151 [==============================] - 20s 133ms/step - loss: 0.6979 - accuracy: 0.7580 - val_loss: 0.4805 - val_accuracy:
0.8465
Epoch 4/10
151/151 [==============================] - 21s 136ms/step - loss: 0.4492 - accuracy: 0.8444 - val_loss: 0.2716 - val_accuracy:
0.9225
Epoch 5/10
151/151 [==============================] - 20s 135ms/step - loss: 0.3047 - accuracy: 0.8997 - val_loss: 0.1918 - val_accuracy:
0.9483
Epoch 6/10
151/151 [==============================] - 20s 135ms/step - loss: 0.1994 - accuracy: 0.9350 - val_loss: 0.1044 - val_accuracy:
0.9756
Epoch 7/10
151/151 [==============================] - 21s 139ms/step - loss: 0.1356 - accuracy: 0.9590 - val_loss: 0.0635 - val_accuracy:
0.9875
Epoch 8/10
151/151 [==============================] - 21s 142ms/step - loss: 0.0931 - accuracy: 0.9726 - val_loss: 0.0394 - val_accuracy:
0.9947
Epoch 9/10
151/151 [==============================] - 20s 130ms/step - loss: 0.0710 - accuracy: 0.9801 - val_loss: 0.0383 - val_accuracy:
0.9939
Epoch 10/10
151/151 [==============================] - 20s 130ms/step - loss: 0.0529 - accuracy: 0.9864 - val_loss: 0.0161 - val_accuracy:
0.9987
```

Fig:4.1 Training history

We call fit (), which can prepare the demonstrate by cutting the information into "clusters" of size batch_size, and over and over emphasizing over the complete dataset for a given number of ages.

The returned history question holds a record of the misfortune values and metric values amid training.

CNN was also used in their recognition system. One thing to bear in mind is that, unlike some of the other models, this one does not use a context subtraction approach. As a result, when I try to include context subtraction in the project, the accuracy may vary. Despite the fact that the majority of the aforementioned projects employ Kinect equipment, my primary goal was to create a project that could be completed with widely available resources. For the vast majority of the audience, a sensor like Kinect is not only not widely available, but also prohibitively expensive, but this model makes use of the laptop or computer’s standard camera, which is a large benefit. So, it will be easy to use for the Deaf and Mute people who are not able to buy the Kinect devices. And my other goal is to get better accuracy for the system applying the train, test, and CNN model on the given data.

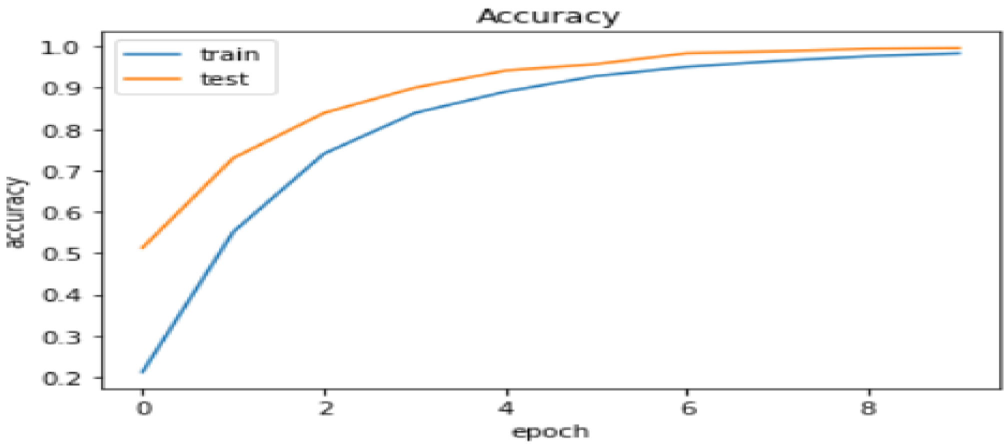


Fig: 4.2 Training accuracy in graphically

To get the accuracy I used the metrics accuracy_score function. This function computes subset accuracy in multilabel classification: the set of labels predicted for a sample must exactly match the corresponding set of labels in y true.

To get the accuracy at first I was reading the metrics from sklearn. But I could not understand the process. Then I followed another computation.

The correct computation would be following:

For example :

test_labels = [0,1,2,4,0]

y_true = [0,1,2, 5,0]

Here, the matches on indices 0,1,2. Thus:

Number of matches = 3

Number of samples= 5

The accuracy calculation:

Accuracy = matches/samples

= 3/5

= 0.6

Finally, I got an accuracy of 84.54%.

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

I proposed a basic hand motion recognition algorithm, which would be followed by multiple steps such as pre-processing and converting the image to RGB so that fluctuating lighting would not be a problem. Then smudge removal is carried out to obtain the best picture possible. These pre-processing steps are just as critical as the rest of the procedure. After pre-processing the image, the next step is to decide the image's orientation; only horizontal and vertical orientations are considered here, and images with uniform backgrounds are taken. Also, minor differences in hand orientation may have a significant impact on the detection process.
The “Cam shift algorithm,” for example, is the least computationally costly but yet managed to hang our system many times. We only take static gestures in this scheme, but in real-time, we must remove gestures from a moving scene. The findings of a gesture-based method that removes features from our hands were discussed in this paper. I also used The Moore Neighbor Contour Tracing Algorithm. It’s worked to make the images into black and white. It’s worked pixel by pixel. Another algorithm I used that The Image Classification Algorithm. It’s worked to make the classes according to the label of the image and got ready the data for training and testing. Almost I was able to fulfill my work as I wanted to do. I hope It will fulfill my whole requirements as I wanted to do my job in the future.

5.2 Future Scope

I will try to improve the accuracy of the project, working for improving preprocessing and better prediction in a low light environment. And in the future, I want to work with my data because it’s not a good recognition with the given data. So, I will try to my data as input and will work with that. I will also add voice output with text so that people can hear it and can easily understand what deaf and dumb people want to say. This would only necessitate a few updates to the new interface code, which were put off due to a lack of time.

The one-time training constraint for real-time systems should be eliminated if the algorithm is improved to deal with different skin types and light conditions, which seems to be impossible right now. The speed of data preprocessing could be an increase in the future if we use a higher configured device.

Reference:

[1] Sign language to text conversion using deep learning
NB: P. V. S. M. S. Kartik, Konjeti B. V. N. S. Sumanth, V. N. V. Sri Ram, and P. Prakash, G. Ranganathan et al. (eds.), Inventive Communication and Computational Technologies, Lecture Notes in Networks and Systems 145,
Year: 25 September 2020

[2] Hand Gesture Recognition with Depth Images, Suarez, J., & Murphy, R. R. (2012, September). Hand gesture recognition with depth images: A review. In 2012 IEEE RO-MAN: the 21st IEEE international symposium on robot and human interactive communication (pp. 411-417). IEEE.

[3] Conversion of Sign Language into Text, NB, M. K. (2018). Conversion of sign language into text. International Journal of Applied Engineering Research, 13(9), 7154-7161.

[4] Hand gesture recognition using a real-time tracking method and hidden Markov models, journal :Image and Vision Computing , volume 21,number:8 ,pages:745-758, year:2003, issn : 0262-8856

[5] Real time conversion of sign language to speech and prediction of gestures using Artificial Neural Network, journal :Procedia Computer Science, volume:143,pages: 587-594,year :2018, note: 8th International Conference on Advances in Computing & Communications (ICACC-2018),issn:1877-0509,author = Abey Abraham and V Rohini

[6] Sign language converter, Taner Arsan and Oğuz Ülgen Department of Computer Engineering, Kadir Has University, Istanbul, Turkey, International Journal of Computer Science & Engineering Survey (IJCSES) Vol.6, No.4, August 2015

[7] Dynamic Hand Gestures Recognition System with Natural Hand Vishal Nayakwadi, N.B. Pokale Department of Computer Engineering, BSCOER, Pune, India. Volume 3– Issue 1, 10 - 19, 2013, ISSN: 2319–8656

[8] Pujan Ziaie, Thomas M¨uller and Alois Knoll. A Novel Approach to Hand-Gesture Recognition in a Human-Robot Dialog System: Robotics and Embedded Systems Group Department of Informatics Technische University at Munchen. Pujan Ziaie and Alois Knoll.

[9] Paulo Trigueiros,Fernando Ribeiro, Luis Paulo Reis “Generic System for HumanComputer Gesture Interaction” ,IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), Espinho, Portugal ,pp. 175-180, May 2015.

[10] Hand Gesture Recognition System, Mohamed Alsheakhali, Ahmed Skaik, Mohammed Aldahdouh, Mahmoud Alhelou, Computer Engineering Department, The Islamic University of Gaza Strip, Palestine, 2011.

[11] Z. Ren, J. Yuan, J. Meng, and Z. Zhang, “Robust part-based hand gesture recognition using Kinect sensor,” IEEE Trans. Multimedia, vol. 15, no. 5, pp. 1110–1120, Aug. 2013.

[12] Neha S. Chourasia, Kanchan Dhote, Supratim Saha. Analysis on Hand Gesture Spotting using Sign Language through Computer Interfacing: International Journal of Engineering Science and Innovative Technology (IJESIT) Vol.3, Issue 3, pp. 132-139, May 2014.

Books:

[1] Handbook of fingerprint recognition (second edition)-by Davide Maltoni & Anil.K.Jain.

[2] Digital Image Processing (Second Edition)-by Rafael C.Gonzalez,MedData Interactive

[3] Face Detection and Gesture Recognition for Human Computer Interaction-by MingHsuan Yang (Author), Narendra Ahuja (Author).

[4] Digital Image Processing Techniques-by Jayaraman.

[5] Inventive communication and computational technologies by G. Ranganathan Joy Chen Álvaro Rocha.

Websites:

[1] https://opencv.org/
[2] https://en.wikipedia.org/wiki/TensorFlow
[3] https://en.wikipedia.org/wiki/Convolutional_neural_network
[4] http://www-i6.informatik.rwth-aachen.de/~dreuw/database.php
[5] www.researchgate.net
[6] https://scholar.google.com/

APPENDIX

OpenCV

OpenCV (Open Source Computer Vision Library) is free for both academic and commercial use since it is distributed under the BSD license. It supports Windows, Linux, Mac OS, iOS, and Android, with C++, Python, and Java interfaces. OpenCV was developed with a heavy emphasis on real-time applications and computational performance in mind. Because it is developed in optimized C/C++, the library makes use of multicore computation. When OpenCL is enabled, it will make use of the hardware acceleration provided by the underlying heterogeneous computing platform.

OpenCV, which has been widely embraced, has a user base of over 47 thousand users and has received over 14 million downloads. Digital painting, mine inspection, web map stitching, and advanced robotics are among the applications.

TensorFlow

TensorFlow is an open-source software toolkit for programming dataflows that can be used for a range of purposes. It's a symbolic math library that's also utilized in machine learning applications like neural networks. It's utilized for both testing and development at Google.

TensorFlow was designed by the Google brain team for internal usage. It was released under the Apache 2.0 open source library on November 9, 2015.

TensorFlow is the second-generation approach used by Google Brain. Version 1.0.0 was released on February 11, 2017. While the reference implementation runs on single computers, TensorFlow may run on many CPUs and GPUs (with optional CUDA and SYCL extensions for general-purpose computing on graphics processing units). TensorFlow is compatible with 64-bit Linux, macOS, and Windows, as well as Android and iOS devices.

Its modular architecture allows it to run on a wide range of devices (CPUs, GPUs, TPUs), from PCs to server clusters to smartphones and edge computers.

Convolution Neural network

CNNs employ a type of multilayer perceptron that doesn't require much preprocessing. They are also known as "shift invariant" or "space invariant" artificial neural networks because of their shared-weights design and translation invariance features (SIANN).

Biological processes influence the connectivity pattern between neurons in convolutional networks, which is similar to the organization of the animal visual cortex. Individual cortical neurons only respond to stimuli in the receptive field, a small portion of the visual field. The receptive fields of different neurons partially overlap, allowing them to occupy the entire visual field.

CNNs require very little preprocessing in comparison to other image classification algorithms. This guarantees that the network is aware of the filters that were previously created by hand in normal algorithms.

Turnitin Originality Report

Processed on: 25-Jan-2022 16:15 +06
ID: 1747774159
Word Count: 8021
Submitted: 1

181-35-2490 By Mon Mon Jaman Jecee

Similarity Index	Similarity by Source	
24%	Internet Sources:	18%
	Publications:	6%
	Student Papers:	16%

3% match (Internet from 13-Dec-2020)
https://www.kaggle.com/datamunge/sign-language-mnist

1% match (student papers from 08-Dec-2021)
Submitted to University of Western Ontario on 2021-12-08

1% match (Internet from 17-Feb-2015)
http://www.ijcsit.com/docs/Volume%203/vol3Issue3/ijcsit2012030310.pdf

1% match (student papers from 13-Apr-2018)
Class: April 2018 Project Report
Assignment: Student Project
Paper ID: 946060438

1% match (student papers from 02-Mar-2021)
Submitted to University of Wales Institute, Cardiff on 2021-03-02

1% match (Internet from 10-Aug-2020)
http://docplayer.net/49214075-Hand-gesture-recognition-with-depth-images-a-review.html

1% match (Internet from 05-Jan-2022)
http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/5699/171-35-1954%20%2814_%29.pdf?isAllowed=y&sequence=1

1% match (student papers from 27-Oct-2021)
Submitted to Higher Education Commission Pakistan on 2021-10-27

1% match (Internet from 28-Sep-2021)
https://www.infoworld.com/article/3278008/what-is-tensorflow-the-machine-learning-library-explained.html

1% match (Internet from 27-Apr-2019)
https://en.wikipedia.org/wiki/TensorFlow

1% match (Internet from 15-Dec-2021)
https://ijsrcseit.com/paper/CSEIT217526.pdf

1% match (Internet from 30-Dec-2017)
http://ecomchain.com/integrations.php

1% match (Internet from 09-Nov-2021)
https://www.slideshare.net/AfnanRehman/hand-gesture-recognition-systemfyp-report-50225618

< 1% match (student papers from 04-Dec-2021)
Submitted to University of Western Ontario on 2021-12-04

< 1% match (student papers from 24-Apr-2021)
Submitted to University of Wales Institute, Cardiff on 2021-04-24

< 1% match (student papers from 19-Aug-2021)
Submitted to University of Wales Institute, Cardiff on 2021-08-19

< 1% match (Internet from 19-Jan-2022)
http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/6813/161-15-6934%20%2818%25%29%20clearance.pdf?isAllowed=y&sequence=1

< 1% match (Internet from 05-Jan-2022)
http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/5691/171-35-1847%20%2822_%29.pdf?isAllowed=y&sequence=1

< 1% match (Internet from 07-Feb-2020)
https://en.wikipedia.org/wiki/Convolutional_neural_network

< 1% match (student papers from 24-Jun-2021)
Submitted to Liverpool John Moores University on 2021-06-24

< 1% match (student papers from 25-May-2021)
Submitted to Universiti Tenaga Nasional on 2021-05-25

< 1% match (student papers from 11-Feb-2020)
Submitted to University of Computer Studies on 2020-02-11

< 1% match (student papers from 14-May-2021)
Submitted to Lovely Professional University on 2021-05-14

< 1% match (student papers from 17-Dec-2021)
Submitted to University of Hertfordshire on 2021-12-17

< 1% match (Internet from 08-Apr-2021)
https://mediatum.ub.tum.de/doc/1289363/220322.pdf

< 1% match (Internet from 14-Nov-2017)
https://www.microsoft.com/en-us/research/wp-content/uploads/2016/11/Robust-Part-Based-Hand-Gesture.pdf

< 1% match (student papers from 29-May-2019)
Submitted to Istanbul Bilgi University on 2019-05-29

< 1% match (publications)
S Krishnakumar., M Yasodha., J Veronica Priyadharshini., J Bethanney Janney., Sindu Divakaran, V Lumen Christy.. "Detection of Arrhythmia and Congestive Heart Failure Through Classification of ECG Signals Using Deep Learning Neural Network", 2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA), 2021

< 1% match (Internet from 14-Oct-2006)
http://logicalgenetics.com/showarticle.php?topic_id=898&page_num=5

< 1% match (Internet from 18-Apr-2019)
http://www.imageprocessingplace.com/downloads_V3/root_downloads/tutorials/contour_tracing_Abeer_George_Ghuneim/mmain.h

< 1% match (publications)
D Aditya, R.G. Manvitha, M Samyak, B S Shamitha. "Emotion based video player", Global Transitions Proceedings, 2021

< 1% match (student papers from 25-May-2019)
Submitted to Netaji Subhas Institute of Technology on 2019-05-25

< 1% match (publications)
"Smart Systems for Industrial Applications", Wiley, 2022

< 1% match (student papers from 24-Aug-2021)
Submitted to University of Wolverhampton on 2021-08-24

< 1% match (student papers from 16-May-2021)
Submitted to Gitam University on 2021-05-16

< 1% match (Internet from 16-Aug-2021)
https://databricks.com/glossary/convolutional-layer

< 1% match (Internet from 28-Nov-2021)
https://www.saama.com/different-kinds-convolutional-filters/

< 1% match (Internet from 12-Oct-2021)
http://tudr.thapar.edu:8080/jspui/bitstream/10266/4796/5/4796.pdf

< 1% match (Internet from 12-May-2016)
https://www.mysciencework.com/publication/show/80c0e16b52eb4617c2f33edc40792900

< 1% match (student papers from 15-Aug-2020)
Submitted to British University in Egypt on 2020-08-15

< 1% match (Internet from 17-Jul-2020)
https://github.com/navjindervirdee/neural-networks

< 1% match (Internet from 02-Aug-2020)
http://ilsalottodellamoda.it/clmn/sklearn-accuracy.html

< 1% match (Internet from 25-Sep-2020)
https://ieeexplore.ieee.org/document/8470686/

< 1% match (Internet from 21-Sep-2017) http://tampub.uta.fi/bitstream/handle/10024/97856/GRADU-1438236458.pdf?sequence=1

< 1% match (Internet from 13-Oct-2018) https://pdfs.semanticscholar.org/54a5/680c9fcd540bf3df970dc7fcd38747f689bc.pdf

< 1% match (Internet from 10-Oct-2020) https://www.springerprofessional.de/en/sign-language-to-speech-converter-using-raspberry-pi/18165512

< 1% match (publications)
Gaohe Li, Sicheng Sun, Xinhao Li. "Block Chain Digital Invoice System Risk Assessment and System Development Research", 2020 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS), 2020

< 1% match (publications)
Suarez, Jesus, and Robin R. Murphy. "Hand gesture recognition with depth images: A review", 2012 IEEE RO-MAN The 21st IEEE International Symposium on Robot and Human Interactive Communication, 2012.

< 1% match (student papers from 12-Jul-2020)
Submitted to The University of Buckingham on 2020-07-12

< 1% match (student papers from 15-Dec-2014)
Submitted to UT, Dallas on 2014-12-15

< 1% match (Internet from 23-Feb-2017)
https://espace.curtin.edu.au/bitstream/handle/20.500.11937/2524/14330_AquinoRussell%2c%20Catherine%202003.pdf?isAllowed=y&sequence=2

< 1% match (Internet from 25-Apr-2021) https://www.cours-gratuit.com/python-courses/python-data-science-training-courses-essential-tools-for-workingwith-data

< 1% match (Internet from 21-Jan-2020) http://export.arxiv.org/pdf/1902.02771

< 1% match (Internet from 07-Jan-2022) https://huggingface.co/datasets/McGill-NLP/mlquestions/blob/main/test.csv

< 1% match (student papers from 13-Jan-2022)
Submitted to Botswana Accountancy College on 2022-01-13

< 1% match (Internet from 24-Feb-2021) https://dblp.org/pid/13/718.html?view=bibtex

< 1% match ()
Dawani, Himalaya. "Analysis of Spherical Texture On Characteristics of Hydrodynamic Journal Beraing", 2015

< 1% match (Internet from 03-Dec-2021) https://ntnuopen.ntnu.no/ntnuxmlui/bitstream/handle/11250/2789448/no.n

tnu%3ainspera%3a77039769%3a51805054.pdf?
isAllowed=y&sequence=1

< 1% match (Internet from 17-Jan-2022)
https://www.coursehero.com/file/95665661/Robust-Part-Based-Hand-Gesturepdf/