SIGN LANGUAGE RECOGNITION USING MACHINE LEARNING ALGORITHM

V. Charitha	***19951A0544***
P. Karthik	***19951A0569***
N. Chaitanya Goud	***19951A0542***

SIGN LANGUAGE RECOGNITION USING MACHINE LEARNING ALGORITHM

A Project Report submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

in

Computer Science & Engineering

by

V. Charitha	**19951A0544**
P. Karthik	**19951A0569**
N. Chaitanya Goud	**19951A0542**

Department of Computer Science & Engineering

INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal, Hyderabad – 500 043, Telangana

May 2023

CERTIFICATE

This is to certify that the project report entitled **Sign language recognition using machine learning algorithm** submitted by **V. Charitha, P. Karthik, N. Chaitanya Goud** to the Institute of Aeronautical Engineering, Hyderabad in partial fulfillment of the requirements for the award of the Degree Bachelor of Technology in **Computer Science and Engineering** is a bonafide record of work carried out by him/her under my/our guidance and supervision. The contents of this report, in full or in parts, have not been submitted to any other Institute for the award of any Degree.

Supervisor **Head of the Department**

Mr. N. Rajasekhar
Assistant professor, CSE

Dr. C. Madhusudhana Rao
Professor and Head of dept CSE

Date:

APPROVAL SHEET

This project report entitled **Sign language recognition using machine learning algorithm** by **V. Charitha, P. Karthik, N. Chaitanya Goud** is approved for the award of the Degree Bachelor of Technology in **Computer Science and Engineering.**

Examiners **Supervisor (s)**

Principal

Date:

Place:

DECLARATION

I certify that

a. The work contained in this report is original and has been done by me under the guidance of my supervisor(s).

b. The work has not been submitted to any other Institute for any degree or diploma.

c. I have followed the guidelines provided by the Institute in preparing the report.

d. I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

e. Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the report and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

Place:	**Signature of the Student**
Date:	**Vedantam Charitha - 19951A0544**
	Pellakuru Karthik - 19951A0569
	Navapete Chaitanya Goud - 19951A0542

ACKNOWLEDGEMENT

I am very contempt with the work and research I have done on this project. The pride that comes from the completion of the work would always be unfinished without mentioning the people who guided us to make it possible. I am thankful to our institution management and respected **Sri M. Rajashekar Reddy, Chairman, IARE, Dundigal** for helping me with the necessary inputs that helped us complete this project work.

I would like to thank **Dr. L. V. Narasimha Prasad, Professor and Principal** who was very encouraging and a constant driving force for me, and **Dr. C. Madhusudhana Rao, Professor and Head, Department of CSE** who have been a helping hand in developing and improving our work.

I am especially thankful to our supervisor**, Mr. N. Rajasekhar, Assistant Professor**, for his extend support, guidance and suggestions that helped me in enhancing the work and get better results.

I would like to use this moment to thank everyone who has supported and assisted me to work hard and make this work to its current form.

ABSTRACT

Keywords: Convolutional Neural Network, Sign-Recognition, Hand Movements

The Sign Language Recognition Software intends to incorporate the use of sign language into material or speech to improve interpersonal interaction between deaf and hearing people. This topic has far-reaching repercussions, yet it remains incredibly difficult due to the complexities and wide range of hand movements. Current SLR techniques include palm-drafted characteristics for determining sign language change and creating models for classification using these characteristics. However, it is difficult to create trustworthy features that allow a wide range of movements. The use of the KNN (k-nearest neighbor) technique has shown significant issues such as the enchantment of dimensionalities, high dataset sizes, class inequalities, noise, and outliers. To address this issue, this research proposes a novel convolutional neural network (CNN) which is capable of extracting discriminative spatial-temporal aspects from raw video streams without the need for feature creation. Multi-channel video feeds (such as color data, depth cues, and body joint orientations) are transmitted into CNN as inputs to integrate color, depth, and movement data to increase performance. Using a real-world dataset obtained with Microsoft Kinect, this article assessed the suggested model and demonstrated its superiority over traditional techniques based on manual labor.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

CNN	Convolutional Neural Network
SLR	Sign Language Recognition
SVM	Support Vector Machine
ASL	American Sign Language
API	Application Programming Interface
ANN	Artificial Neural Networks
PCA	Principal Component Analysis
OO tools	Object Oriented Tools
UML	Unified Modeling Language
KNN	K-Nearest Neighbor
GMM-HMM	Gaussian Mixture Model-Hidden Markov Model
ISL	Indian Sign Language
GUI	Graphical User Interface

Chapter – 1

Introduction

1.1 Introduction

Sign language can be considered one of the most often utilized communication tools for those with hearing difficulties, communicated through variations in palm forms, body motions and additionally expressions on the face. Sign language recognition remains a very challenging task due to the difficulty in sharing information from hand shapes and body trajectories [1][2]. This study postulates a useful recognition model to aid deafness individuals in communicating with hearing individuals by translating sign language into written and speech. Practically, the biggest difficulty in sign language development of characteristics to represent the shape and trajectory of the hand. Hand form specification in case in point necessitates identifying palm areas in streams of footage and splitting palm shape imagery from composite views. Backdrop and beckon recognition issues for each frame. Trajectories are also concerned with key point tracking and curve coordinating. Although a lot of study has been done on these two areas so far, it is still challenging to produce adequate results for Sign Language Recognition with regard to the fluctuation as well as interference of hand and joint motions. Additionally, merging hand features and network features is a non-trivial challenge [2]. To address these adversities, this paper proposing CNNs to naturally incorporate hand forms, trajectories, and facial looks instead than utilizing the standard color image as the inputs of the network, use the color image, deep image and hand skeleton image as input at the same time all powered by Microsoft Kinect. It is a motion sensor. Using the Windows SDK, the positions of body joints can be retrieved in real time. Therefore, select Kinect as the acquisition medium for recording the character word dataset. Changing chroma and depth at an individual pixel level provides beneficial data to differentiate between distinct drawing procedures. These changes in the time dimension of the body joints can represent the trajectory of the character's movements. Utilizing a range of photo resources as input helps the CNN to focus not just on color changes but also on profundity and momentum. Notably, since CNNs possess the capacity to effortlessly acquire characteristics from unprocessed information without prior experience, by-passing the obstacles of watching hands, distinguishing hands from the backdrop, and constructing hand descriptions [3]. In recent years, CNNs have been applied to the classification of video streams. A possible drawback with CNNs is that they are time demanding. Training a CNN with millions of videos takes weeks or months. thankfully it is still conceivable for one to accomplish real-time efficiency. This research proposes to infer both temporal and geographical data from web cameras for sign language recognition (SLR) using Convolutional Neural Network. Existing SLR approaches employ handwritten characteristics to characterize the flow of sign language and build classification models on the basis of these features. CNNs, on the other hand, can collect information about motion using raw video footage automatically. Avoid design elements. This study creates a CNN that uses a variety of information as inputs. The design uses compression alongside video pixel sub sampling to incorporate colour, depth, and trajectory intelligence. Results from experiments show that for numerous character words, 3D CNN beats the Gaussian Mixture Model utilizing the Hidden Markov Model (GMM-HMM) framework. [4].

1.2 Existing System

Gestures are extracted and converted into text. It is now attempting “four-fold cross validation”, with the validation set Correspond to photographs of people who are not from the training set. SVM (support vector machine) is used. Uses a webcam Interactions in general.

1.2.1 Limitations

As this is an expanding subject, there are no limitations save that the technique is only suitable for static ASL numerical signs. The ASL recognizer system can't be considered flawless. The current method achieves just 65% accuracy.

1.3 Proposed Work

In this study, we will offer a reliable as well as effective method for identifying sign language. Instead of employing Data gloves for sign language detection, we would use image processing. The key advantage of employing processing of images against data gloves is the fact that the system does not need to be re-calibrated when a new user logs in. Furthermore, by employing a value for the threshold while converting the picture from Grayscale to Binary form, this approach may be employed in any backdrop and is not limited to Black or White backgrounds. Using multiple approaches and the CNN algorithm, we hope to translate the sign gesture to speech.

1.3.1 Advantages

Using CNN algorithm provides several perks, such as high accuracy in recognizing sign language, efficient processing of large datasets, robustness against lighting, hand shape and movement variations, scalability for both small and large-scale applications, and automation of the sign language recognition process. This technology has the potential to boost accessibility for persons who primarily utilize sign language for communication. Our proposed model coverts the signs into speech which can be audible to the user and understand it very effectively.

1.4 Software and Hardware Requirements

1.4.1 Software Requirements

Operating System	:	Windows
Language	:	Python

1.4.2 Hardware Requirements

Processor	:	I3
Speed	:	2.4GHz
RAM	:	1 GB
Hard Disk	:	500 GB

1.5 Modules Used

1.5.1 Tensorflow

TensorFlow is a free software library to download, and allowing differentiable computing and information flow for a number of activities. It constitutes a general-purpose math package that is additionally applied for artificial intelligence procedures such as neural networks. Google uses it for both research and product development. TensorFlow was built for internal Google usage by the Google Brain team. It was published beneath the Apache2.0 open-source agreement.

1.5.2 Numpy

Numpy is a multifaceted array-processing package. It offers an outstanding durability multi-dimensional array component alongside additional equipment for dealing with these arrays. It's a must-have Python module for scientific computing.

It has many distinct characteristics, the most significant of which are:

- An essential “N-dimensional array” object
- Advanced (broadcasting) functionalities
- Tools for integrating “C/C++ “, and “Fortran code”
- Capabilities for “quadratic algebra”, “Fourier alter”, and “arbitrary numbers"

Aside from its readily apparent research programs, Numpy may also be utilized as a strong multi-faceted repository of common data. Numpy may provide any kind of information types, helping it to interact with a broad assortment of databases easily and rapidly.

1.5.3 Pandas

Pandas is a Python library that is freely available that provides outstanding durability data analysis and manipulation tools based on its key data organizing principles. Python was originally largely used for information munging and preprocessing. It provided simply a tiny contribution to the study of data. Pandas replied to this issue. No matter the origin of the information load, we may negotiate five main procedures in handling and analyzing data using Pandas: organize, change, framework, and analysis. Python with Pandas has been used in a wide range of industrial sectors, including educational and professional domains, plus financial services, economics, research, statistical analysis, and so on.

1.5.4 Matplotlib

Matplotlib, In a variety of visual media, a Python 2D plotting programme offers publication-quality graphics. and multi-platform dynamic settings. Matplotlib could be utilized by all four GUI resources furthermore, Jupyter Notebook, web-based programme servers, Python scripts, as well as the Python and IPython shells. Matplotlib aims for making straightforward tasks simple and difficult tasks manageable. You may generate plots, histograms that plus frequency spectrum bar charts, error charts, scatter diagrams, and much more with only a handful of lines of code. Examples can be found in the source charts and miniatures galleries. The pyplot package, especially when combined with IPython, gives users a MATLAB-like environment for simple graphing. There are two alternatives for manipulating line styles, font settings, axis attributes, and so forth for the power user: an Object-Based API or a group of MATLAB-compatible techniques.

1.5.5 Scikit – learn

Scikit-learn delivers a range of both unsupervised and supervised learning techniques leveraging a standard Python interface. It is available in multiple versions of Linux and is given under a liberal modified BSD license that promotes both academic as well as business usage. Python is a general-purpose, high-level, interpretive programming language. Python, invented by Guido van Rossum and subsequently initially released in 1991, stresses accessibility to code and makes considerable use of emptiness. Python offers a flexible type model and a distinct handling of memory mechanism. It comes with a big and comprehensive standard collection and implements a wide number of perspectives for programming, including important, working, administrative in nature, and object-oriented.

- Python is Interpreted - Python is processed by the interpreter as it is utilized. Your software doesn't require to be built before it may be run. Unlike PHP and PERL, your software program does not need to be built before executing.
- Python is interactive; users can sit in a Python moment and speak with the interpreter right immediately while developing programs.

Python appreciates the value of development speed. This requires having access to sophisticated structures that prevent laborious code duplication, as well as clear and short code. Maintainability is tied to each of these and may be a pointless metric, however it does give data about exactly what amount of code someone must scan, comprehend and/or absorb in order to rectify faults or alter behavior. Another area in which Python shines is its swift development the simplicity with which developers from different languages may acquire the fundamentals of Python, and the enormous standard library. Each of its tools are straightforward to get started with, which saving a lot of time. Furthermore, a few of them may subsequently be repaired and enhanced without any harm.

Chapter – 2

Literature Review

Zaid Omar et al. [5] stated that the steps of gesture recognition involve gathering data, processing it beforehand, segmenting it, identifying its features, and classifications. Whereas dynamic sign languages use video, which is a continuous stream of frames of pictures, Single frames of photos are used as the input for static gesture recognition. Vison-based methods differ from sensor-based methods primarily in the way that data is acquired. This section focuses on the methods and procedures utilised by studies on vision-based gesture recognition.

Rafiqul Zaman Khan et al. [6] used non-geometric characteristics and a multivariate Gaussian distribution to identify hand movements. Two strategies are used to segment the input hand image: HSV colour model-based segmentation based on skin tone and histogram noise removal techniques. Some operations are carried out in the interest of acquiring palm features, capture the hand's shape; The modified Direction Analysis Algorithm is employed to determine the direction of the hand movement in order to calculate the object's (the hand's) slope and trend. It is utilised in order to determine a linkage between statistical parameters from the data.

According to Sukanya Dessai et.al. [7] The capacity of a machine to understand human conduct and its importance may be applied in a number of contexts. The study of sign language recognition is one topic that has aroused interest. Deaf or dumb individuals use sign language to converse. Without the assistance of interpreters who can read sign language, these individuals are helpless to interact with the outside world. The techniques utilized in current studies on sign language recognition are reviewed in this report. The approaches are studied at the phases of gathering data, pre-processing, segmenting, obtaining features, and classifying it. The aforementioned literature review is conducted individually at each stage. Some of the most popular classification methods used for recognition include ANN, SVM and fuzzy inference systems.

Chapter – 3

Methodology

3.1 Block Diagram

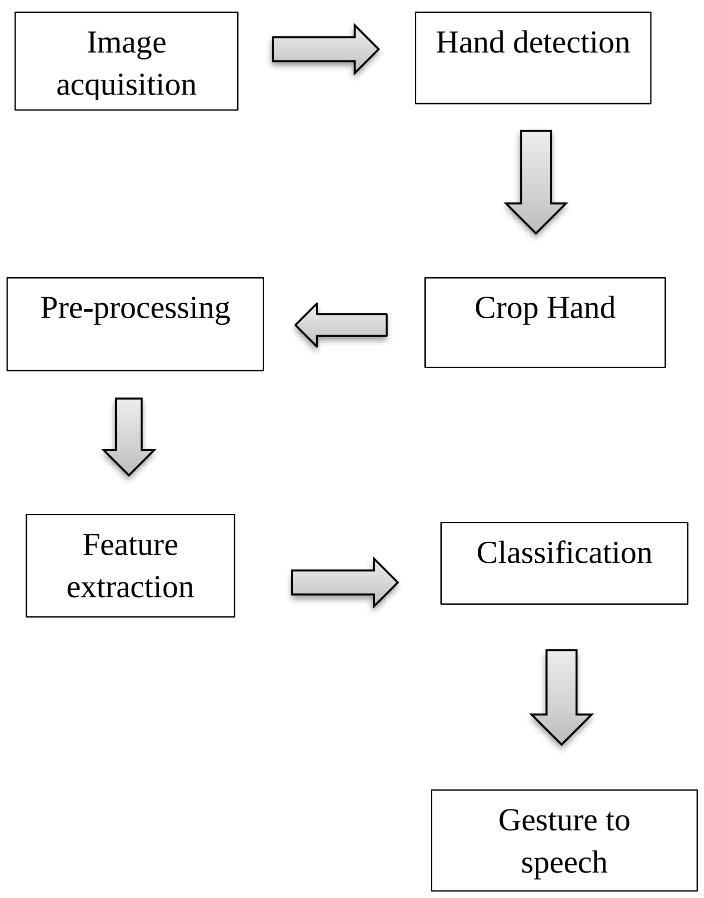


Figure 1: Block diagram of SLR (Sign Language Recognition)

3.2 Module 1

Image Acquisition

It's the initial and most important phase of the gesture recognition process. webcam connection is required in order to capture images using a Webcam. Several laptops now include an inbuilt camera system, which aids in the capture of photos for further processing. The embedded camera can identify hand gesture and position by capturing gestures.

3.3 Module 2

Image Pre-processing

This phase involves retouching, filtration, adjusting intensity and tone and many other operations are included in image pre-processing. Picture enhancing, picture clipping, and picture thresholding are employed in this procedure. Images that are captured are in RGB format. Hence, the primary task is to transform RGB photos to binary information, and then crop the image to get rid of the undesired parts. [8][9].

3.4 Module 3

Feature Extractions

This is an essential process in building the dataset for hand gesture interpretation. To properly and quickly define the numerous visual principles of the letters of the manual alphabet, both the local and global visual properties can be obtained for word image similarity characterization [9]. Feature Extraction can be acquired by using PCA (principal component analysis). The unwanted components which does not help to the wanted elements are removed for dimension reduction [10].

3.5 Module 4

Classification and Gesture to speech

In this phase, collecting and labelling pictures is the process of classifying them for recognition. The category of the provided input is also predicted by classifiers. Based on their ability to learn, classifiers can be categorized. All the audio recordings of the collected datasets of gestures are stored in the system. when it recognizes the gesture, it gives the output audio.

3.6 UML Diagrams

3.6.1 Detailed Design

The abbreviation UML stands for Unified Modelling Language. Simply said, UML is a current method for software planning and development. It is, in fact, a few of the most extensively used business procedure modelling methodologies. It is founded on graphical representations of application components. "A photograph is like a thousand words," as the old adage goes. By using visual representations, we may be able to comprehend possible flaws or issues with technology or business processes better. UML arose from the turmoil that surrounded the creation of software and documentation. There were various approaches to represent and record software systems in the 1990s. As a consequence of the requirement for a more consistent manner to graphically depict those systems, three Rational Softwares, software engineers created the UML in 1994-1996. It was eventually approved as the norm in 1997 and has stayed so ever since, with only minor modifications.

3.6.2 Goals

The points that follow are the main objectives for the UML's development:

1. Give clients a full, usable graphical modelling language that enables them to produce and exchange valuable models.
2. Offer methods for specialization and extendibility to expand the underlying principles.
3. Lack the use of specialized languages in creation processes and computers.

4. Develop an official framework towards understanding the modelling language.

5. Boost the broadening of consumer demand for OO tools.

6. Encourage the usage of higher-level concepts for advancement including partnerships that applications, designs, and modules.

7. Implement best practices.

3.6.3 User Use Case Diagram

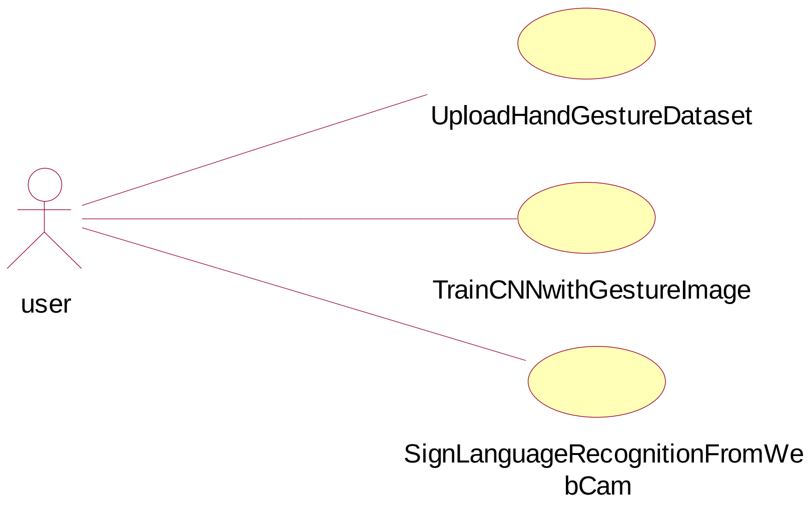


Figure 2: User use-case diagram of SLR

3.6.4 Class Diagram

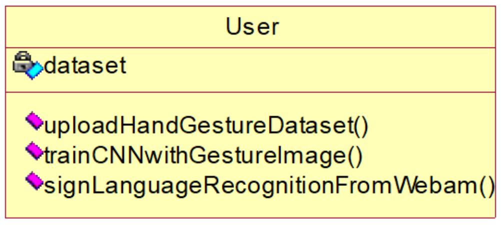

User
dataset
uploadHandGestureDataset()
trainCNNwithGestureImage()
signLanguageRecognitionFromWebam()

Figure 3: Class diagram of SLR

3.6.5 Sequence Diagram

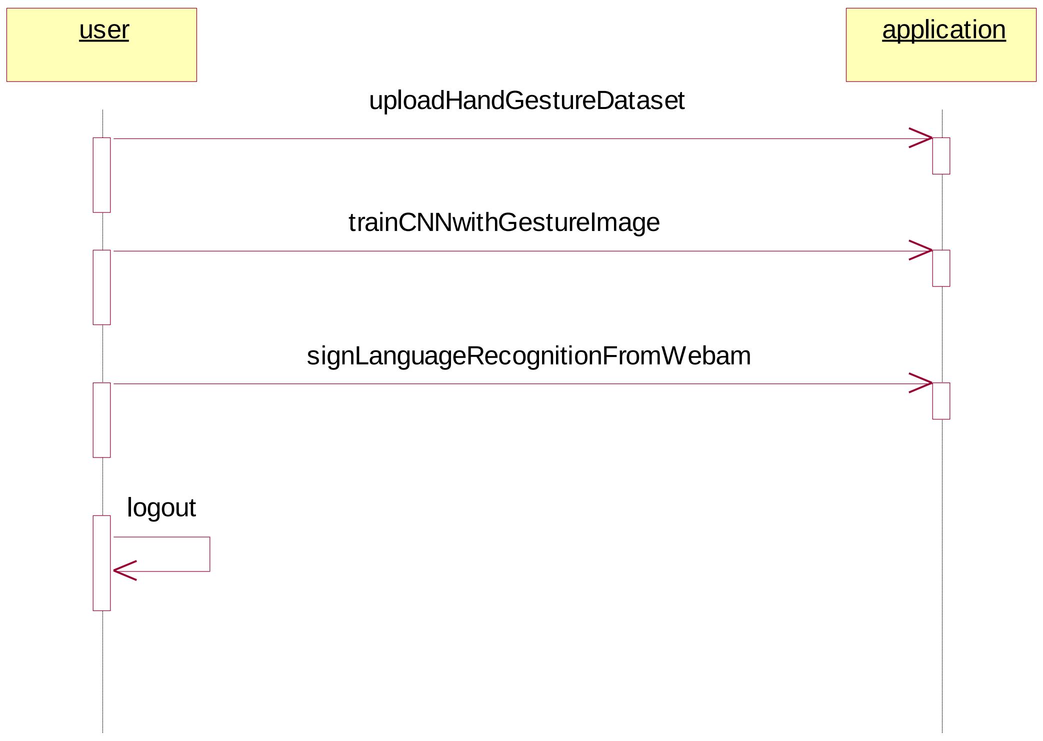


Figure 4: Sequence diagram of SLR

3.6.6 Collaboration Diagram

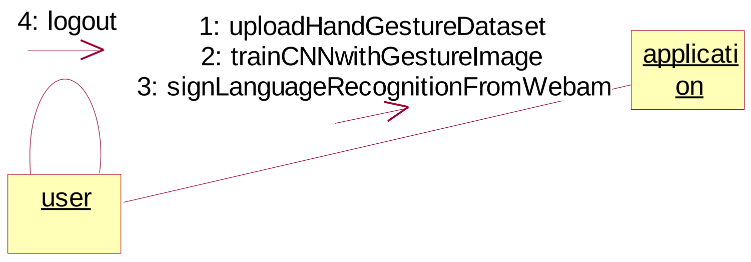

4: logout
1: uploadHandGestureDataset
2: trainCNNwithGestureImage
3: signLanguageRecognitionFromWebam
applicati
on
user

Figure 5: Collaboration diagram of SLR

3.7 Algorithm

Step 1: Begin

Step 2: Upload Hand Gesture Dataset

Step 3: Use Gesture Images to Train Convolution Neural Network

Step 4: Upload a Test Image and Identify a Gesture

Step 5: Transform the Recognized Gesture into a Letter or Word and a Speech

Step 6: Complete

3.8 Sample code

```
from tkinter import messagebox
from tkinter import *
from tkinter import simpledialog
import tkinter
from tkinter import filedialog
from tkinter.filedialog import askopenfilename
import cv2
import random
import numpy as np
from keras.utils.np_utils import to_categorical
from keras.layers import  MaxPooling2D
from keras.layers import Dense, Dropout, Activation, Flatten
from keras.layers import Convolution2D
from keras.models import Sequential
from keras.models import model_from_json
import pickle
import os
import imutils
from gtts import gTTS
from playsound import playsound
import os
from threading import Thread
main = tkinter.Tk()
main.title("Sign Language Recognition to Text & Voice using CNN Advance")
main.geometry("1300x1200")
global filename
global classifier
bg = None
playcount = 0

#names = ['Palm','I','Fist','Fist Moved','Thumbs up','Index','OK','Palm Moved','C','Down']
names = ['C','Thumbs Down','Fist','I','Ok','Palm','Thumbs up']
```

```
def getID(name):
  index = 0
  for i in range(len(names)):
    if names[i] == name:
      index = i
      break
  return index
bgModel = cv2.createBackgroundSubtractorMOG2(0, 50)
def deleteDirectory():
  filelist = [ f for f in os.listdir('play') if f.endswith(".mp3") ]
  for f in filelist:
    os.remove(os.path.join('play', f))
def play(playcount,gesture):
  class PlayThread(Thread):
    def __init__(self,playcount,gesture):
      Thread.__init__(self)
      self.gesture = gesture
      self.playcount = playcount
    def run(self):
      t1 = gTTS(text=self.gesture, lang='en', slow=False)
      t1.save("play/"+str(self.playcount)+".mp3")
      playsound("play/"+str(self.playcount)+".mp3")
  newthread = PlayThread(playcount,gesture)
  newthread.start()
def remove_background(frame):
  fgmask = bgModel.apply(frame, learningRate=0)
  kernel = np.ones((3, 3), np.uint8)
  fgmask = cv2.erode(fgmask, kernel, iterations=1)
  res = cv2.bitwise_and(frame, frame, mask=fgmask)
  return res
```

```
 def uploadDataset():
    global filename
    global labels
    labels = []
    filename = filedialog.askdirectory(initialdir=".")
    pathlabel.config(text=filename)
    text.delete('1.0', END)
    text.insert(END,filename+" loaded\n\n");
 def trainCNN():
    global classifier
    text.delete('1.0', END)
    X_train = np.load('model1/X.txt.npy')
    Y_train = np.load('model1/Y.txt.npy')
    text.insert(END,"CNN is training on total images : "+str(len(X_train))+"\n")
    if os.path.exists('model1/model.json'):
       with open('model1/model.json', "r") as json_file:
          loaded_model_json = json_file.read()
          classifier = model_from_json(loaded_model_json)
       classifier.load_weights("model1/model_weights.h5")
       classifier._make_predict_function()
       print(classifier.summary())
       f = open('model1/history.pckl', 'rb')
       data = pickle.load(f)
       f.close()
       acc = data['accuracy']
       accuracy = acc[9] * 100
       text.insert(END,"CNN Hand Gesture Training Model Prediction Accuracy ="+str(accuracy))
else:
   classifier = Sequential()
   classifier.add(Convolution2D(32, 3, 3, input_shape = (64, 64, 3), activation = 'relu'))
   classifier.add(MaxPooling2D(pool_size = (2, 2)))
   classifier.add(Convolution2D(32, 3, 3, activation = 'relu'))
   classifier.add(MaxPooling2D(pool_size = (2, 2)))
   classifier.add(Flatten())
```

```
        classifier.add(Dense(output_dim = 256, activation = 'relu'))
        classifier.add(Dense(output_dim = 5, activation = 'softmax'))
        print(classifier.summary())
        classifier.compile(optimizer = 'adam', loss = 'categorical_crossentropy', metrics =
  ['accuracy'])
        hist = classifier.fit(X_train, Y_train, batch_size=16, epochs=10, shuffle=True,
  verbose=2)
        classifier.save_weights('model1/model_weights.h5')
        model_json = classifier.to_json()
        with open("model1/model.json", "w") as json_file:
           json_file.write(model_json)
        f = open('model1/history.pckl', 'wb')
        pickle.dump(hist.history, f)
        f.close()
        f = open('model1/history.pckl', 'rb')
        data = pickle.load(f)
        f.close()
        acc = data['accuracy']
        accuracy = acc[9] * 100
        text.insert(END,"CNN Hand Gesture Training Model Prediction Accuracy =
  "+str(accuracy))
  def run_avg(image, aWeight):
     global bg
     if bg is None:
        bg = image.copy().astype("float")
        return
     cv2.accumulateWeighted(image, bg, aWeight)

def segment(image, threshold=25):
   global bg
   diff = cv2.absdiff(bg.astype("uint8"), image)
   thresholded = cv2.threshold(diff, threshold, 255, cv2.THRESH_BINARY)[1]
   ( cnts, _) = cv2.findContours(thresholded.copy(), cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
```

```
    if len(cnts) == 0:
        return
      else:
        segmented = max(cnts, key=cv2.contourArea)
        return (thresholded, segmented)
    def webcamPredict():
      global playcount
      oldresult = 'none'
      count = 0
      fgbg2 = cv2.createBackgroundSubtractorKNN();
      aWeight = 0.5
      camera = cv2.VideoCapture(0)
      top, right, bottom, left = 10, 350, 325, 690
      num_frames = 0
      while(True):
        (grabbed, frame) = camera.read()
        frame = imutils.resize(frame, width=700)
        frame = cv2.flip(frame, 1)
        clone = frame.copy()
        (height, width) = frame.shape[:2]
        roi = frame[top:bottom, right:left]
        gray = cv2.cvtColor(roi, cv2.COLOR_BGR2GRAY)
        gray = cv2.GaussianBlur(gray, (41, 41), 0)
        if num_frames < 30:
          run_avg(gray, aWeight)
  else:
    temp = gray
    hand = segment(gray)
    if hand is not None:
      (thresholded, segmented) = hand
      cv2.drawContours(clone, [segmented + (right, top)], -1, (0, 0, 255))
      #cv2.imwrite("test.jpg",temp)
      #cv2.imshow("Thesholded", temp)
```

```
            #ret, thresh = cv2.threshold(temp, 150, 255, cv2.THRESH_BINARY +
cv2.THRESH_OTSU)
            #thresh = cv2.resize(thresh, (64, 64))
            #thresh = np.array(thresh)
            #img = np.stack((thresh,)*3, axis=-1)
            roi = frame[top:bottom, right:left]
            roi = fgbg2.apply(roi);
            cv2.imwrite("test.jpg",roi)
            #cv2.imwrite("newDataset/Fist/"+str(count)+".png",roi)
            #count = count + 1
            #print(count)
            img = cv2.imread("test.jpg")
            img = cv2.resize(img, (64, 64))
            img = img.reshape(1, 64, 64, 3)
            img = np.array(img, dtype='float32')
            img /= 255
            predict = classifier.predict(img)
            value = np.amax(predict)
            cl = np.argmax(predict)
            result = names[np.argmax(predict)]
            if value >= 0.99:
               print(str(value)+" "+str(result))
               cv2.putText(clone, 'Gesture Recognize as : '+str(result), (10,
25),  cv2.FONT_HERSHEY_SIMPLEX,0.5, (0, 255, 255), 2)
               if oldresult != result:
                  play(playcount,result)
                  oldresult = result
                  playcount = playcount + 1
            else:
               cv2.putText(clone, '', (10, 25),  cv2.FONT_HERSHEY_SIMPLEX,0.5, (0, 255,
255), 2)
            cv2.imshow("video frame", roi)
      cv2.rectangle(clone, (left, top), (right, bottom), (0,255,0), 2)
      num_frames += 1
```

```
        cv2.imshow("Video Feed", clone)
        keypress = cv2.waitKey(1) & 0xFF
        if keypress == ord("q"):
            break
    camera.release()
    cv2.destroyAllWindows()
font = ('times', 16, 'bold')
title = Label(main, text='Sign Language Recognition to Text & Voice using CNN
Advance',anchor=W, justify=CENTER)
title.config(bg='yellow4', fg='white')
title.config(font=font)
title.config(height=3, width=120)
title.place(x=0,y=5)
font1 = ('times', 13, 'bold')
upload = Button(main, text="Upload Hand Gesture Dataset", command=uploadDataset)
upload.place(x=50,y=100)
upload.config(font=font1)
pathlabel = Label(main)
pathlabel.config(bg='yellow4', fg='white')
pathlabel.config(font=font1)
pathlabel.place(x=50,y=150)
markovButton = Button(main, text="Train CNN with Gesture Images",
command=trainCNN)
markovButton.place(x=50,y=200)
markovButton.config(font=font1)
predictButton = Button(main, text="Sign Language Recognition from Webcam",
command=webcamPredict)
predictButton.place(x=50,y=250)
predictButton.config(font=font1)
```

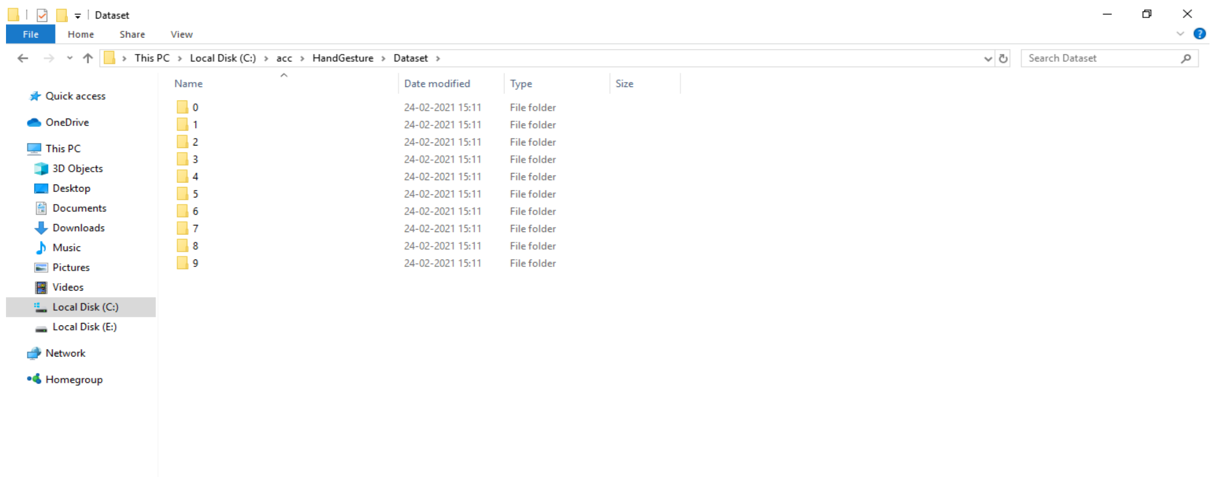
```
font1 = ('times', 12, 'bold')
text=Text(main,height=15,width=78)
scroll=Scrollbar(text)
text.configure(yscrollcommand=scroll.set)
text.place(x=450,y=100)


text.config(font=font1)
deleteDirectory()
main.config(bg='magenta3')
main.mainloop()
```

Chapter – 4

RESULTS AND DISCUSSION

4.1 Results

In this study, we are using CNN to recognize hand gesture motions. To teach CNN, we are utilizing the picture seen in the screen photos below.



Name	Date modified	Type	Size
0	24-02-2021 15:11	File folder	
1	24-02-2021 15:11	File folder	
2	24-02-2021 15:11	File folder	
3	24-02-2021 15:11	File folder	
4	24-02-2021 15:11	File folder	
5	24-02-2021 15:11	File folder	
6	24-02-2021 15:11	File folder	
7	24-02-2021 15:11	File folder	
8	24-02-2021 15:11	File folder	
9	24-02-2021 15:11	File folder	

Figure 6: Datasets

In the screenshot above, we can see that we have ten different types of hand motion shots, and to view those images, simply navigate to any folder.

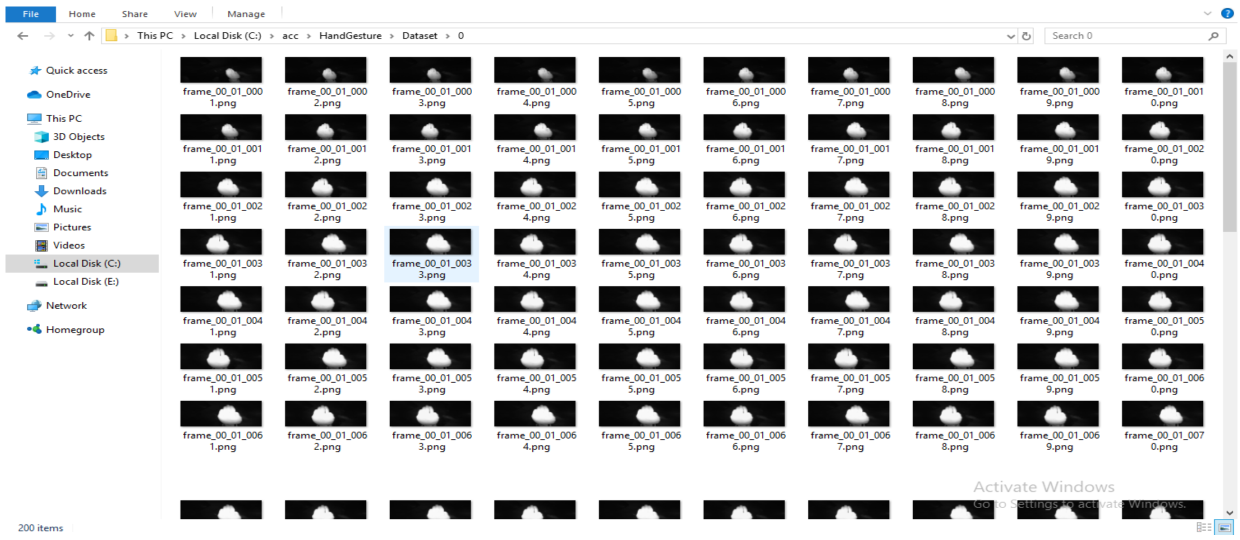


Figure 7: Images of different hand gestures

The photos on the top screen are from the 0 folder, and you may explore images from other directories as well.

Click the run button twice to launch the project.bat file to get the display seen below

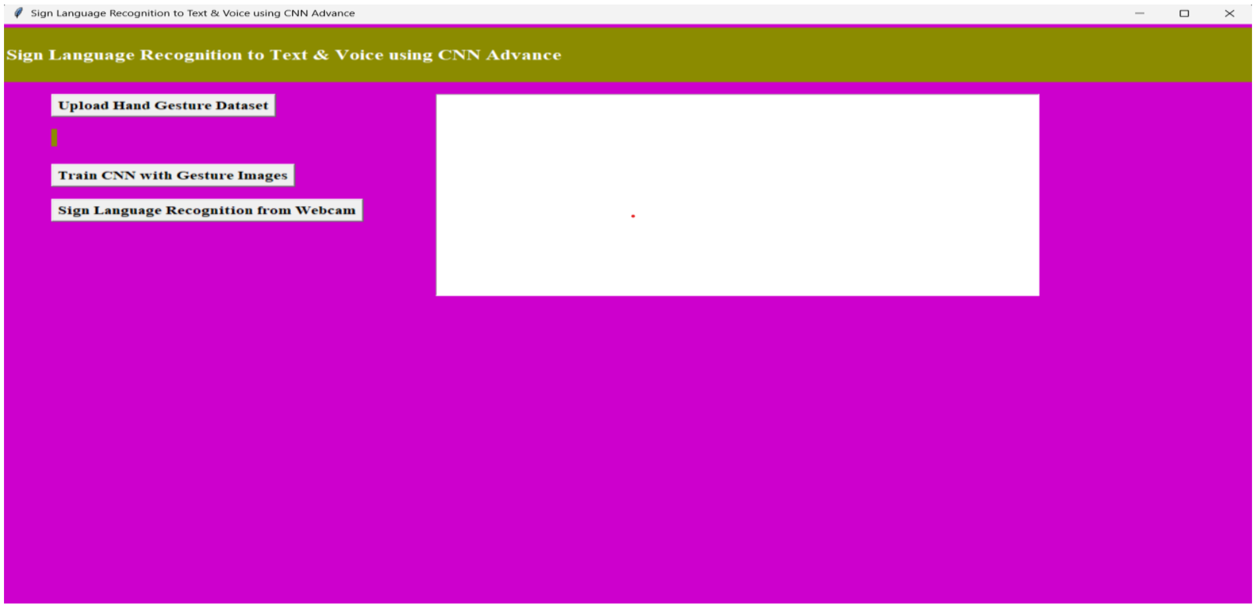
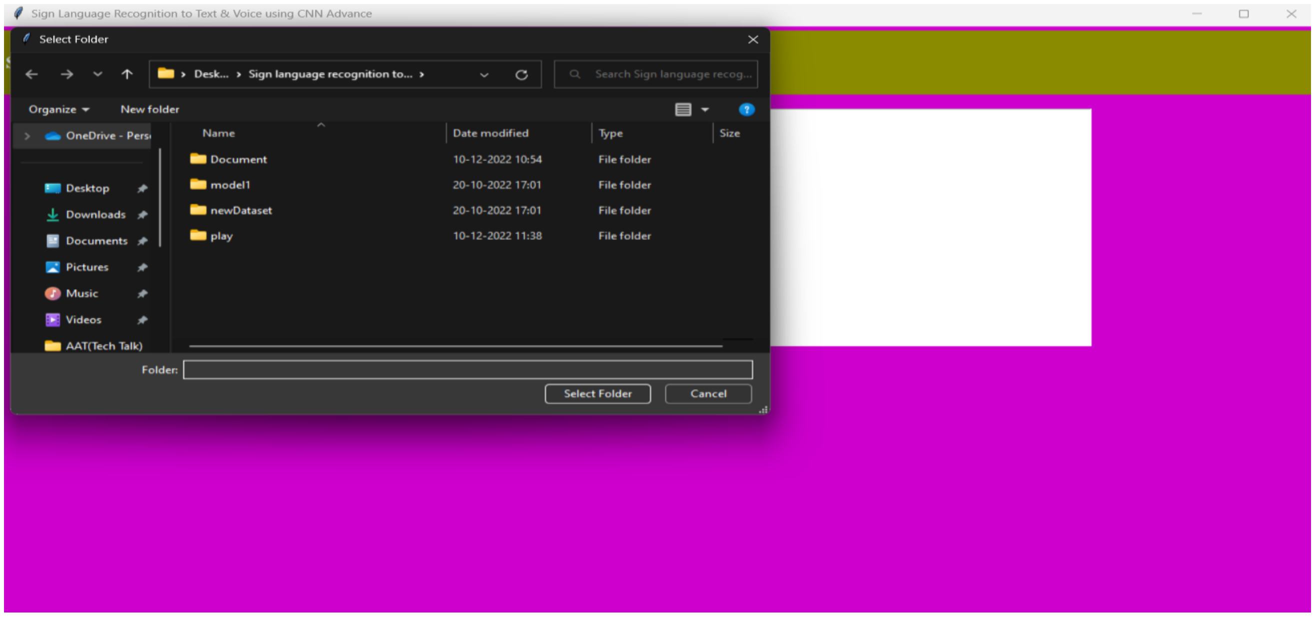

Sign Language Recognition to Text & Voice using CNN Advance
Sign Language Recognition to Text & Voice using CNN Advance
Upload Hand Gesture Dataset
Train CNN with Gesture Images
Sign Language Recognition from Webcam

Figure 8: Interface obtained

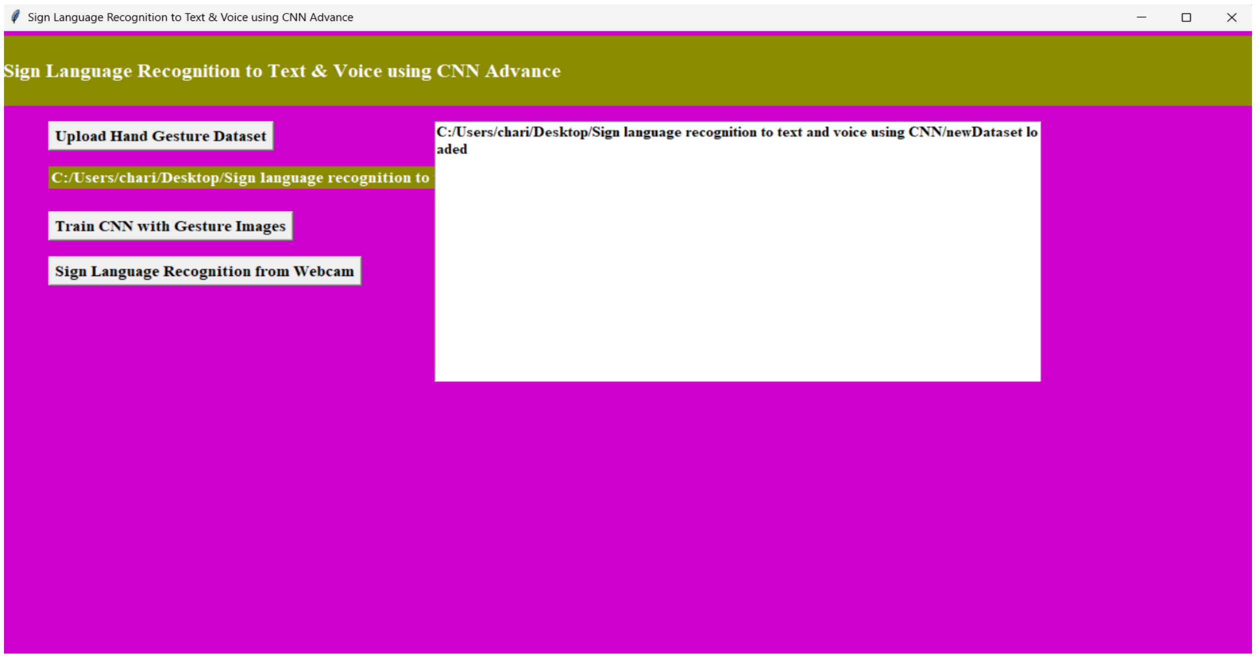
Click the 'Upload Hand Gesture Dataset' button to upload the dataset to the screen shown below.


Sign Language Recognition to Text & Voice using CNN Advance
Select Folder
Desk... › Sign language recognition to... ›
Search Sign language recog...
Organize
New folder
OneDrive - Pers
Desktop
Downloads
Documents
Pictures
Music
Videos
AAT(Tech Talk)
Folder:
Select Folder
Cancel

Name	Date modified	Type	Size
Document	10-12-2022 10:54	File folder	
model1	20-10-2022 17:01	File folder	
newDataset	20-10-2022 17:01	File folder	
play	10-12-2022 11:38	File folder	

Figure 9: Uploading dataset

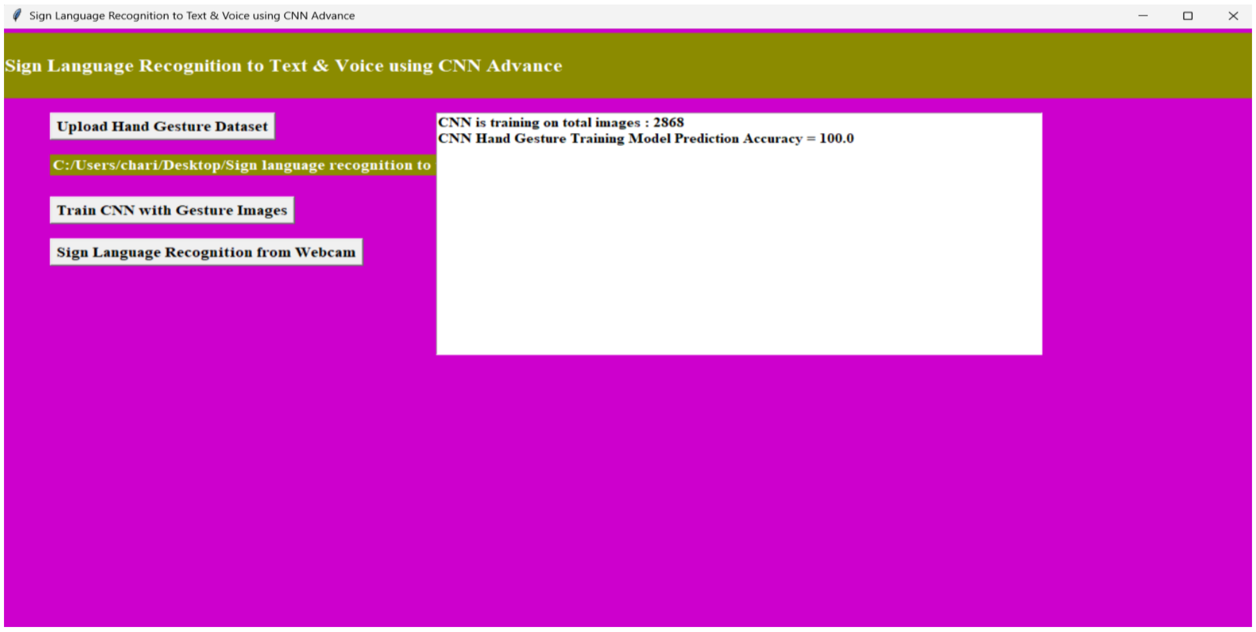
Select and upload the 'Dataset' folder in the above page, then click the 'Select Folder' button to load the dataset and see the screen below.


Sign Language Recognition to Text & Voice using CNN Advance
Sign Language Recognition to Text & Voice using CNN Advance
Upload Hand Gesture Dataset
C:/Users/chari/Desktop/Sign language recognition to
Train CNN with Gesture Images
Sign Language Recognition from Webcam

```
C:/Users/chari/Desktop/Sign language recognition to text and voice using CNN/newDataset lo
aded
```

Figure 10: Training the CNN

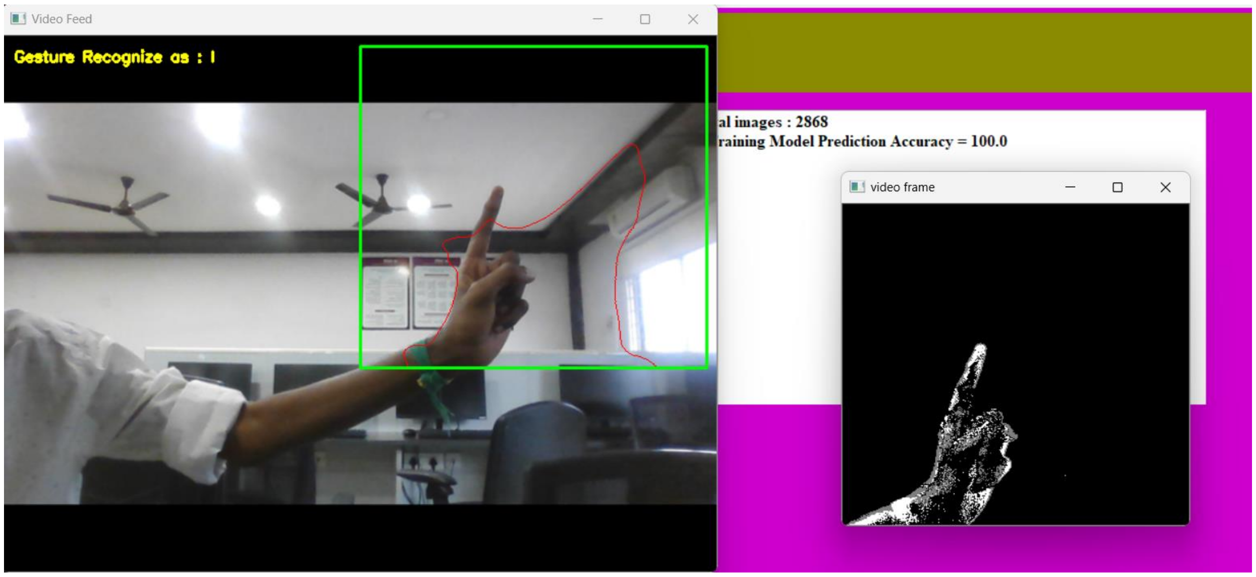
After loading the dataset, click the 'Train CNN using Gesture Images' button to train the CNN model and view the screen below.


Sign Language Recognition to Text & Voice using CNN Advance
Sign Language Recognition to Text & Voice using CNN Advance
Upload Hand Gesture Dataset
C:/Users/chari/Desktop/Sign language recognition to
Train CNN with Gesture Images
Sign Language Recognition from Webcam

```
CNN is training on total images : 2868
CNN Hand Gesture Training Model Prediction Accuracy = 100.0
```

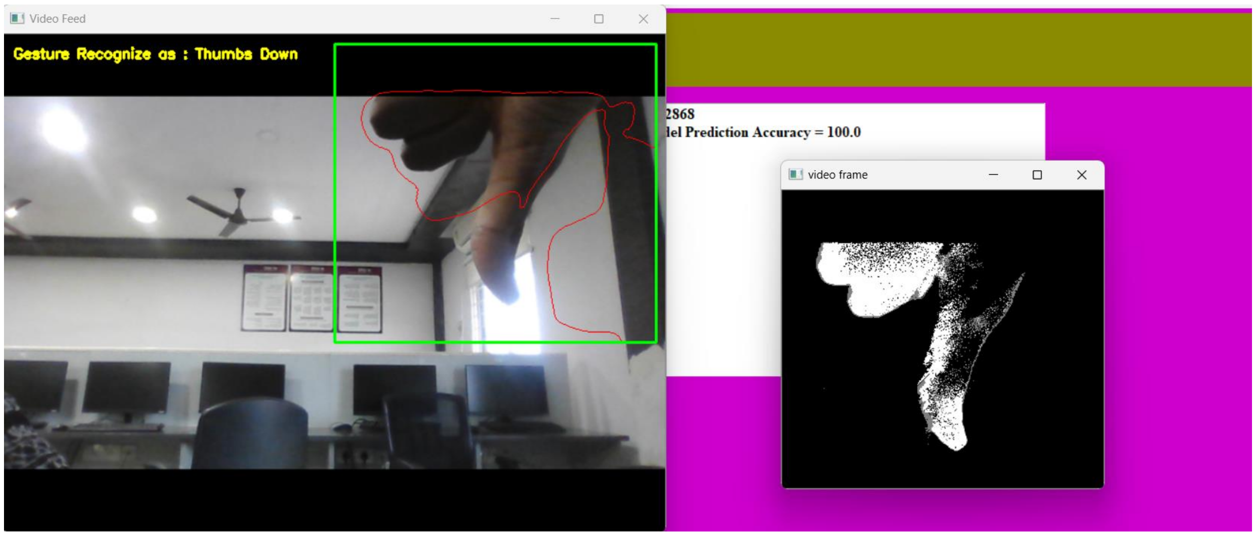
Figure 11: Result obtained after training CNN

The CNN model shown here was trained on 2868 pictures and the model's forecast accuracy was 100%.



```
al images : 2868
raining Model Prediction Accuracy = 100.0
```

Figure 12: Gesture recognized as I



```
2868
lel Prediction Accuracy = 100.0
```

Figure 13: Gesture recognized as Thumbs Down

We obtained an accuracy of 100% using CNN algorithm but due to continuous image capturing it takes few seconds to give the accurate result.

4.2 Result Comparison

After training the CNN Click on "Sign Language Recognition from Web Cam." Then your webcam opens, and it starts searching for gestures. Make a sign language gesture with your hand now. The CNN starts analysing the hand movements and searches for the perfect match with the provided datasets. It displays the gesture's output in the form of of text and speech, as illustrated in the below Fig.14 and Fig.15.

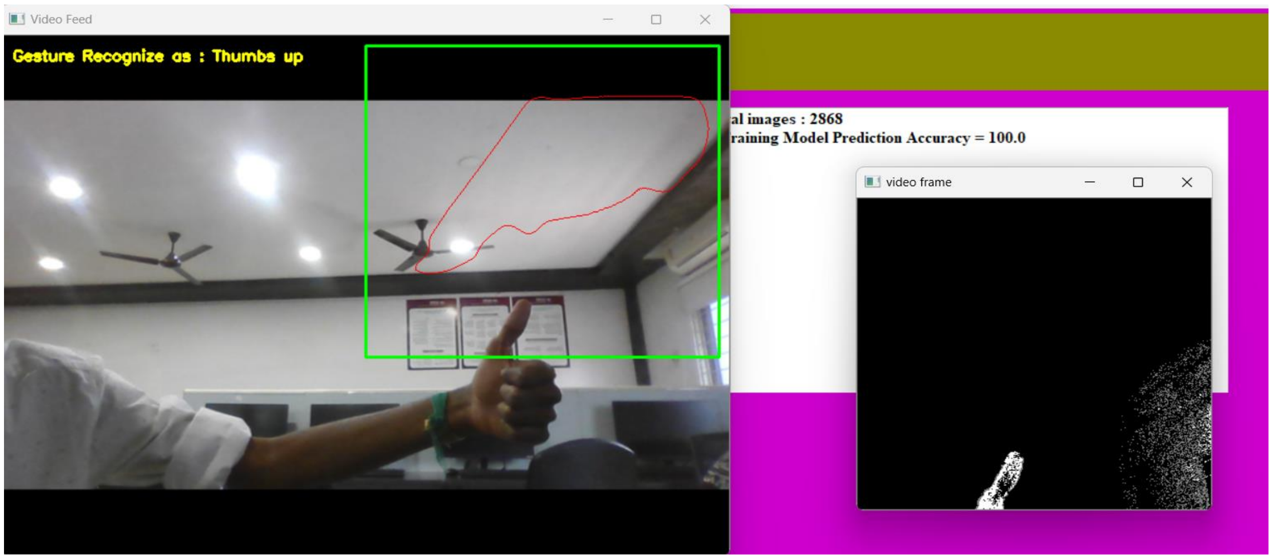

Video Feed
Gesture Recognize as : Thumbs up
al images : 2868
raining Model Prediction Accuracy = 100.0
video frame

Figure14. Gesture recognized as Thumbs up

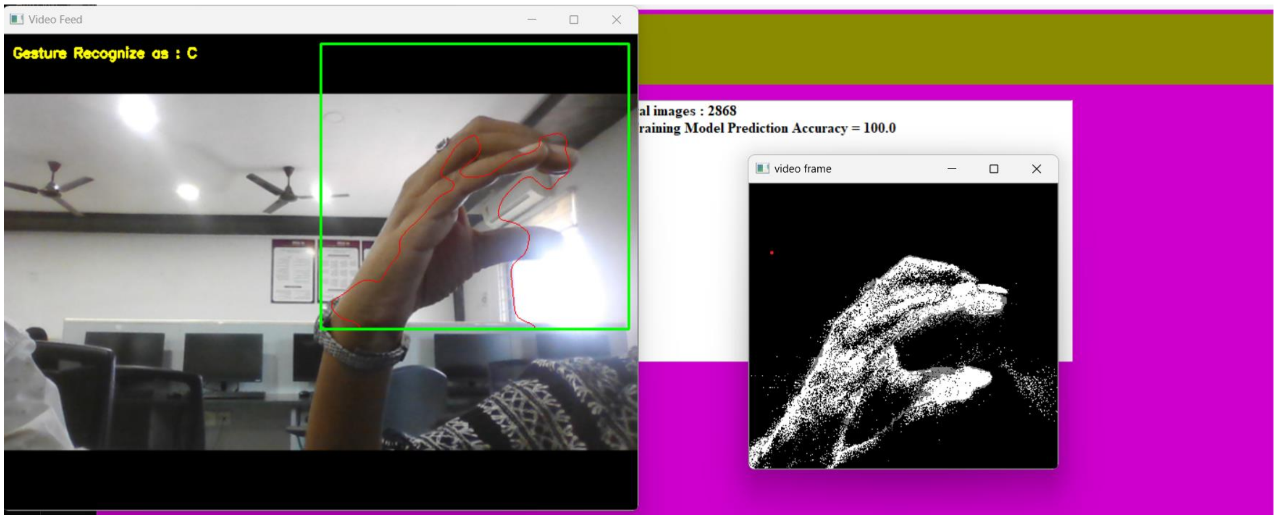


Figure 15. Gesture recognized as C

Both hand gestures are recognised with a perfect accuracy of 100 using CNN algorithm but due to continuous image capturing it takes few seconds to give the accurate result. Accuracy can be calculated by number of successful attempts divided by number of gestures X 100[11]. Table 1 represents the accuracy acquired by various algorithms.

TABLE 1: ACCURACY COMPARISON

ALGORITHM	**ACCURACY**
SVM [12]	95.3
KNN [13]	95
PROPOSED CNN	100

Chapter – 5

Conclusions and Future Scope of Study

5.1 Conclusion

This proposed work created a CNN model to acknowledge the sign phraseology. The approach obtains and retrieves both temporal and spatial information by employing 3D convolutions. Convolution and subsampling are carried out individually after the created deep architecture gathers various types of information from nearby input frames. Information from each channel is combined to create the final feature representation. These feature representations are classified using a multilayer perceptron classifier. For the similar datasets, this paper compare and contrast CNN and GMM-HMM. The results of the experiments show how successful the proposed approach is.

5.2 Future Scope

Future studies might concentrate on phrase creation and the identification of dynamic ISL gestures. The system's portability will allow it to be expanded to include smartphones and smart home appliances. In order to attain interaction and usefulness, hand gesture research must outperform present performance in terms of robustness and speed. Extracting characteristics that would recognize each sign regardless of source, color, and lighting conditions requires more attention. For ISL word and sentence level recognition, one can create a model. For this, a system that can detect changes in temporal space will be required. This can bridge the communication gap for people who are deaf or dumb by developing a comprehensive solution.

References

[1] Available Online at http://junikhyatjournal.in/no_1_Online_21/60.pdf

[2] Available Online at http://213.55.95.56/bitstream/handle/123456789/30421/Biruk%20Mengiste.pdf?sequence=1

[3] Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick ́ Haffner, “Gradient-based learning applied to document recognition,” Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.

[4] Available Online at https://tongtianta.site/oss/paper_pdf/9794926e-56d7-11e9-a763-00163e08bb86.pdf

[5] Cheok, M. J., Omar, Z., & Jaward, M. H. (2017, August 8). A review of hand gesture and sign language recognition techniques. International Journal of Machine Learning and Cybernetics, 10(1), 131–153. https://doi.org/10.1007/s13042-017-0705-5

[6] Zaman Khan, R. (2012, July 31). Hand Gesture Recognition: A Literature Review. International Journal of Artificial Intelligence & Applications, 3(4), 161–174. https://doi.org/10.5121/ijaia.2012.3412

[7] Sukanya Dessai, & Siddhi Naik. (n.d.). Literature Review on Indian Sign Language Recognition System. International Research Journal of Engineering and Technology (IRJET), Volume: 09(Issue: 07). https://www.irjet.net/archives/V9/i7/IRJET-V9I7447.pdf

[8] X. Jiang and W. Ahmad, "Hand Gesture Detection Based Real-Time American Sign Language Letters Recognition using Support Vector Machine," 2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), Fukuoka,Japan,2019,pp.380-385, doi:10.1109/DASC/PiCom/CBDCom/CyberSciTech.2019.00078.

[9] Radha S. Shirbhate, Vedant D. Shinde, Sanam A. Metkari, Pooja U. Borkar, & Mayuri A. Khandge. (2020, March). Sign language Recognition Using Machine Learning Algorithm. International Research Journal of Engineering and Technology (IRJET) , Volume: 07(Issue:03).https://www.irjet.net/archives/V7/i3/IRJET-V7I3418.pdf

[10] K. Amrutha and P. Prabu, "ML Based Sign Language Recognition System," 2021 International Conference on Innovative Trends in Information Technology (ICITIIT), Kottayam, India, 2021, pp. 1-6, doi: 10.1109/ICITIIT51526.2021.9399594.

[11] J. Singha and A. Karen Das, "Recognition of Indian Sign Language in Live Video," 2013, arXiv preprint arXiv:1306.1301.

[12] S. C. Agrawal, A. S. Jalal, and C. Bhatnagar, "Redundancy removal for isolated gesture in Indian sign language and recognition using multi-class support vector machine," Int. J. Comput. Vis. Robot., vol. 4, no. 1–2, pp. 23–38, 2014, doi: 10.1504/IJCVR.2014.059361.

[13] A. Kumar Sahoo, M. Sharma, and R. Pal, "Indian sign language recognition using neural networks and KNN classifiers Image processing" View project IoT and 5G View project vol. 9, no. 8, 2014.

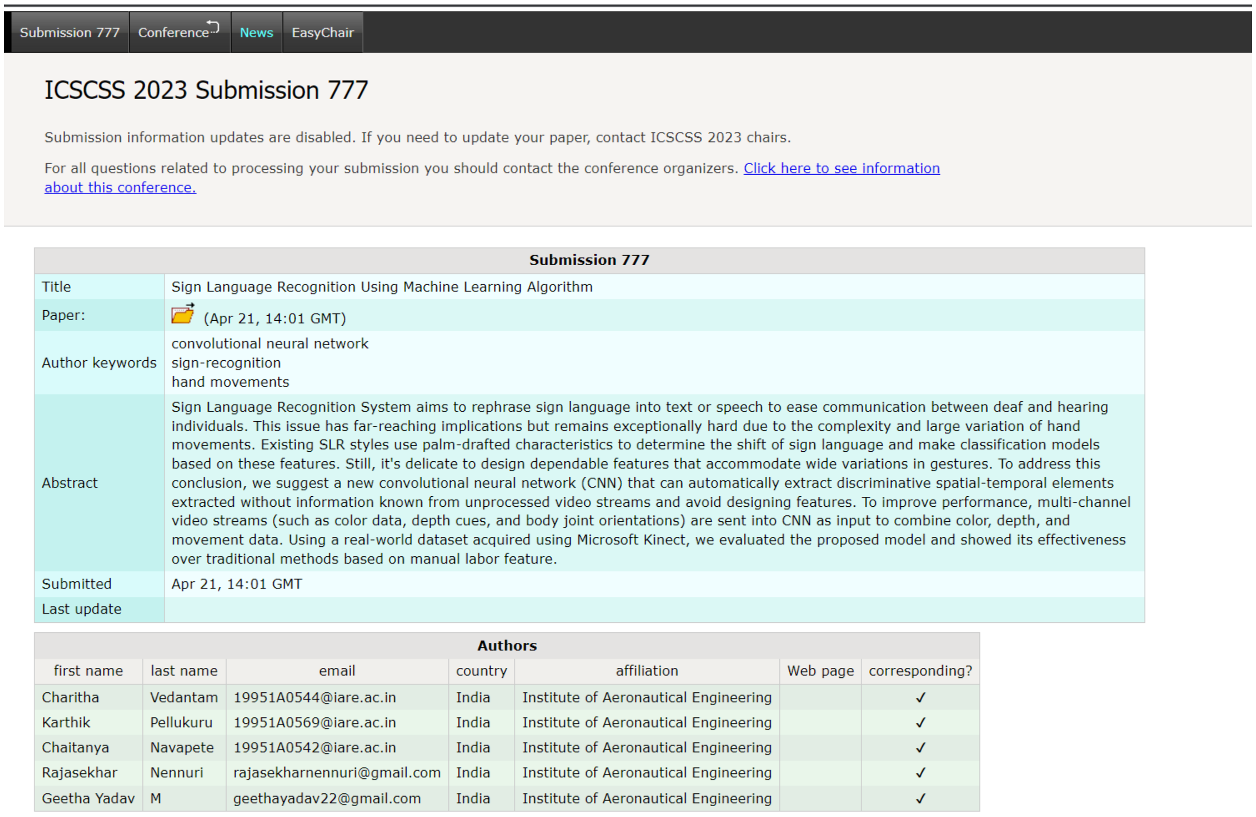
List of Publications

I Conference

1. Sign Language Recognition Using Machine Learning Algorithm

N. Rajasekhar, M. Geetha Yadav, V. Charitha, P. Karthik, N. Chaitanya Goud

International conference on sustainable computing and smart systems



ICSCSS 2023 Submission 777

Submission information updates are disabled. If you need to update your paper, contact ICSCSS 2023 chairs.

For all questions related to processing your submission you should contact the conference organizers. Click here to see information about this conference.

Submission 777	
Title	Sign Language Recognition Using Machine Learning Algorithm
Paper:	(Apr 21, 14:01 GMT)
Author keywords	convolutional neural network sign-recognition hand movements
Abstract	Sign Language Recognition System aims to rephrase sign language into text or speech to ease communication between deaf and hearing individuals. This issue has far-reaching implications but remains exceptionally hard due to the complexity and large variation of hand movements. Existing SLR styles use palm-drafted characteristics to determine the shift of sign language and make classification models based on these features. Still, it's delicate to design dependable features that accommodate wide variations in gestures. To address this conclusion, we suggest a new convolutional neural network (CNN) that can automatically extract discriminative spatial-temporal elements extracted without information known from unprocessed video streams and avoid designing features. To improve performance, multi-channel video streams (such as color data, depth cues, and body joint orientations) are sent into CNN as input to combine color, depth, and movement data. Using a real-world dataset acquired using Microsoft Kinect, we evaluated the proposed model and showed its effectiveness over traditional methods based on manual labor feature.
Submitted	Apr 21, 14:01 GMT
Last update	

Authors

first name	last name	email	country	affiliation	Web page	corresponding?
Charitha	Vedantam	19951A0544@iare.ac.in	India	Institute of Aeronautical Engineering		✓
Karthik	Pellukuru	19951A0569@iare.ac.in	India	Institute of Aeronautical Engineering		✓
Chaitanya	Navapete	19951A0542@iare.ac.in	India	Institute of Aeronautical Engineering		✓
Rajasekhar	Nennuri	rajasekharnennuri@gmail.com	India	Institute of Aeronautical Engineering		✓
Geetha Yadav	M	geethayadav22@gmail.com	India	Institute of Aeronautical Engineering		✓

REQUISITION FOR PLAGIARISM CHECK

1	Name of the student	V. Charitha P. Karthik N. Chaitanya Goud			
2	Email ID and Phone Number	19951A0544@iare.ac.in; 7386045627 19951A0569@iare.ac.in; 9390165953 19951A0542@iare.ac.in; 9666333979			
3	Roll Number	19951A0544 19951A0569 19951A0542			
4	Date of submission				
5	Name of the Guide	Mr. N. Rajasekhar			
6	Title of the project work / research article	Sign language recognition using machine learning algorithm			
7	Department	Computer Science and Engineering			
8	Details of the payment				
9	No. of times submitted	**First / Second / Third** (First time – Free; Second time – Rs 200/-; Third – Rs 500/-; There after multiple of third)			
10	Similarity Content (%) (up to 25% acceptable)	**1st**	**2nd**	**3rd**	**4th**

For R & D Centre Use

Date of plagiarism check	
Similarity report percentage	
R&D staff Name and Signature	

I / We hereby declare that, the above mentioned research work is original & it doesn't contain any plagiarized contents. The similarity index of this research work is……………..
Justification for similarity index:
……………………………………………………………...……………………………
……………………………………………………………………………………………
……………………………………………………………………………………………
……………………………………………………………………………………………
……………………………………………………………………………………………

Signature of Student **Signature of the Guide**