Walchand College of Engineering, Sangli.

(An Autonomous Institute)

Department
Of
Computer Science and Engineering

TY CSE Mini Project-II
Report
On

Sign Language Recognition using Hand Gestures

Submitted by

Omkar Sharad Patil **(2019BTECS00104)**
Pratik Babaso Chougule **(2019BTECS00102)**
Pravin Santosh Lokhande **(2019BTECS00084)**

Under the Guidance
of

Mrs. A. T. Umrani
Guide
Computer Science & Eng. Dept,
WCE, Sangli.

Mr. Amey Tambe
Mentor (IOT Expert)
Director, SoftTech Data
Securities.

2021-2022

Walchand College of Engineering, Sangli
(An Autonomous Institute)

Department
Of
Computer Science and Engineering

CERTIFICATE

This is to certify that the Project Report entitled, **"SIGN LANGUAGE RECOGNITION USING HAND RECOGNITION"** submitted by Mr. Omkar Patil, Mr. Pratik Chougule, Mr. Pravin Lokhande to Walchand College of Engineering, Sangli, India, is a record of bonafide Project work of course *"Mini Project-2"* carried out by him/her under my/our supervision and guidance and is worthy of consideration for the award of the degree of Bachelor of Technology in Computer Science & Engineering of the Institute.

Mrs. A. T. Umrani	**Mr. Amey Tambe**	**Dr. M. A. Shah**
Guide	Mentor	Head Of Department
Computer Sci. & Eng. Dept,	Director, SoftTech Data Securities	Computer Sci.& Eng. Dept,
WCE, Sangli.		WCE, Sangli

Acknowledgement

We would like to express our special thanks of gratitude to our guide Mrs. A. T. Umrani mam as well as our HOD Dr. M. A. Shah who gave us the golden opportunity to do this wonderful project on the topic “Sign Language Recognition using Hand Gestures”, which also helped us in doing a lot of research and we came to know about so many new things. We specially want to thank our industry mentor Mr. Amey Tambe sir for his guidance from selecting the topic to finishing the project. We are thankful to them.

Secondly, we would also like to thank the teammates who worked together in finishing this project within limited time. Finally, thanks to all who supported the project.

Declaration

I hereby declare that work presented in this project report titled "**SIGN LANGUAGE RECOGNITION USING HAND GESTURE"** submitted by me in the partial fulfillment of the requirement of the award of the degree of **Bachelor of Technology (B. Tech)** Submitted in the **Department of Computer Science & Engineering, Walchand College of Engineering, Sangli**, is an authentic record of my project work carried out under the guidance of Mrs. A. T. Umrani and Mr. Amey Tambe sir.

Date:
29-11-2021

Place: Sangli

(Omkar Patil)
2019BTECS00104
(Pratik Chougule)
2019BTECS00102
(Pravin Lokhande)
2019BTECS00084

Table of Contents

1 Project title 6

2 Abstract 7

3 Introduction and Related work 8

4 Problem statement 8

5 Objectives 9

6 Methodology 10

7 Project Diagrams 11

8 Testing 12

9 Results and Conclusion 17

10 References 18

11 Annexure A: A meeting with Mentor 19

1 Project title

Sign Language Recognition using Hand Gestures

2 Abstract

Hand Gesture is one of the methods used in sign language for non-verbal communication. It is most used by deaf and hard-of-hearing people or people with mental disability who are facing problems to communicate among themselves or with normal people.

Various sign language recognition systems have been developed by many makers around the world. Our aim is to produce a model that can recognize hand gestures and signs.

We will train a model for the purpose of sign language conversion, i.e., a simple gesture recognizing model that will help people to experience effective communication.

3 Introduction and Related work

American sign language is a predominant sign language Since the only disability D&M people have been communication related and they cannot use spoken languages hence the only way for them to communicate is through sign language.

Communication is the process of exchange of thoughts and messages in various ways such as speech, signals, behavior, and visuals. Deaf and dumb(D&M) people make use of their hands to express different gestures to express their ideas with other people.

Gestures are the nonverbally exchanged messages and these gestures are understood with vision. This nonverbal communication of deaf and dumb people is called sign language.

4 Problem statement

To create a platform which helps to recognize the sign language with the help of hand gestures made by the users using CNN.

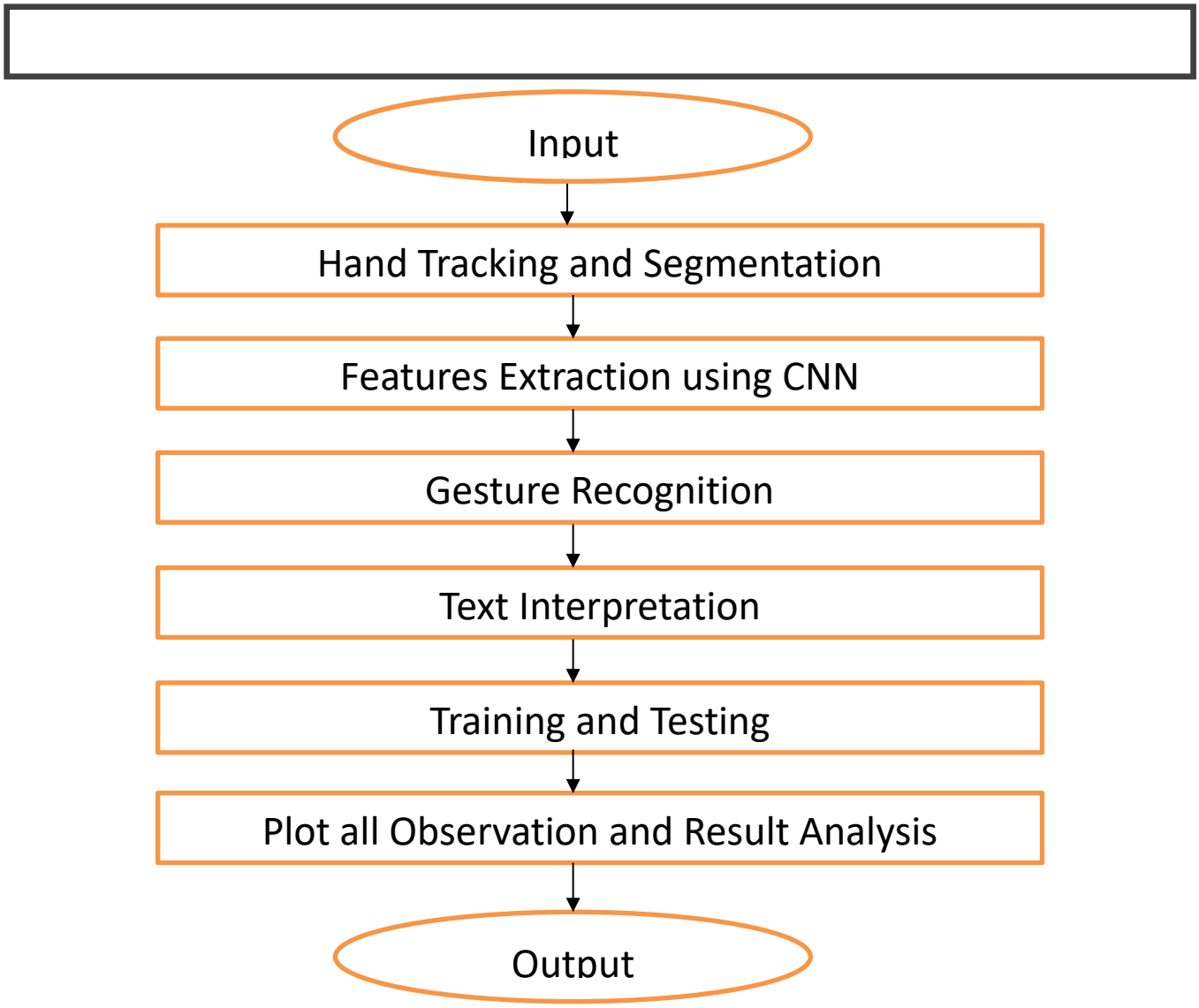
5 Objectives

- **To study the concept of Machine Learning (Convolutional Neural Network Technology), OpenCV, and Python.**
- **To collect images from Kaggle Dataset.**
- **To develop a CNN model for Hand Sign Language Recognition.**
- **To train and test CNN model.**
- **To link the program on the web.**

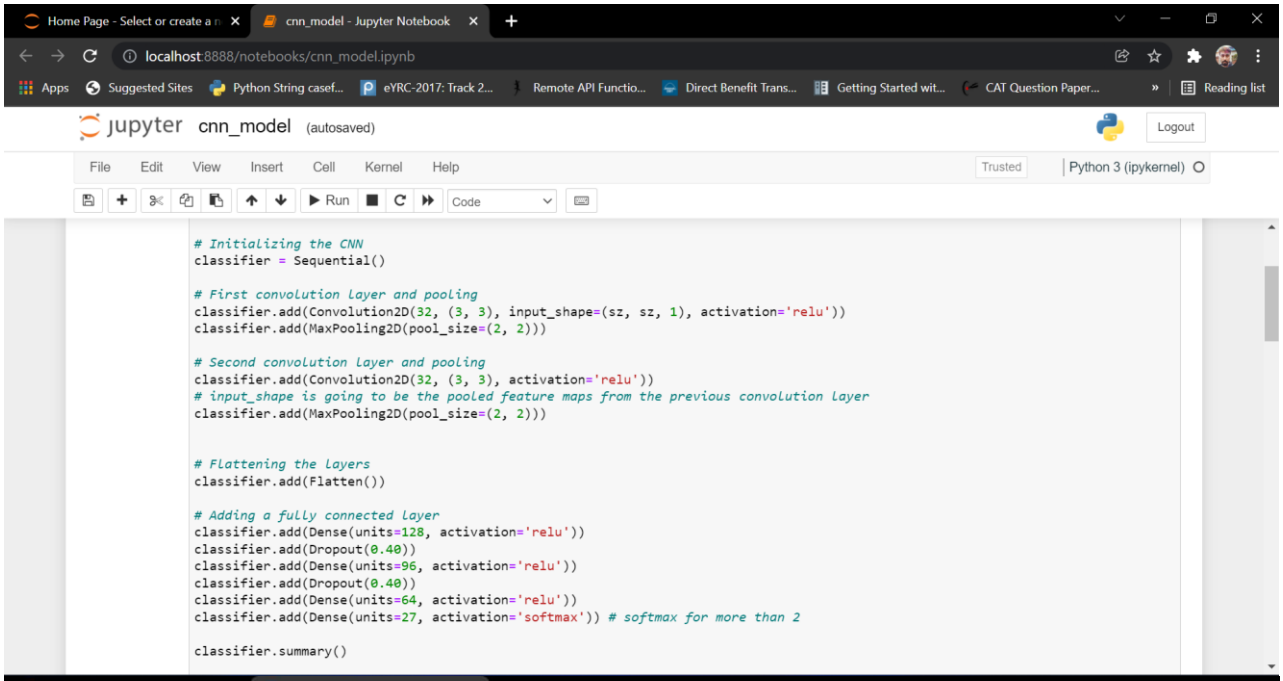
6 Methodology

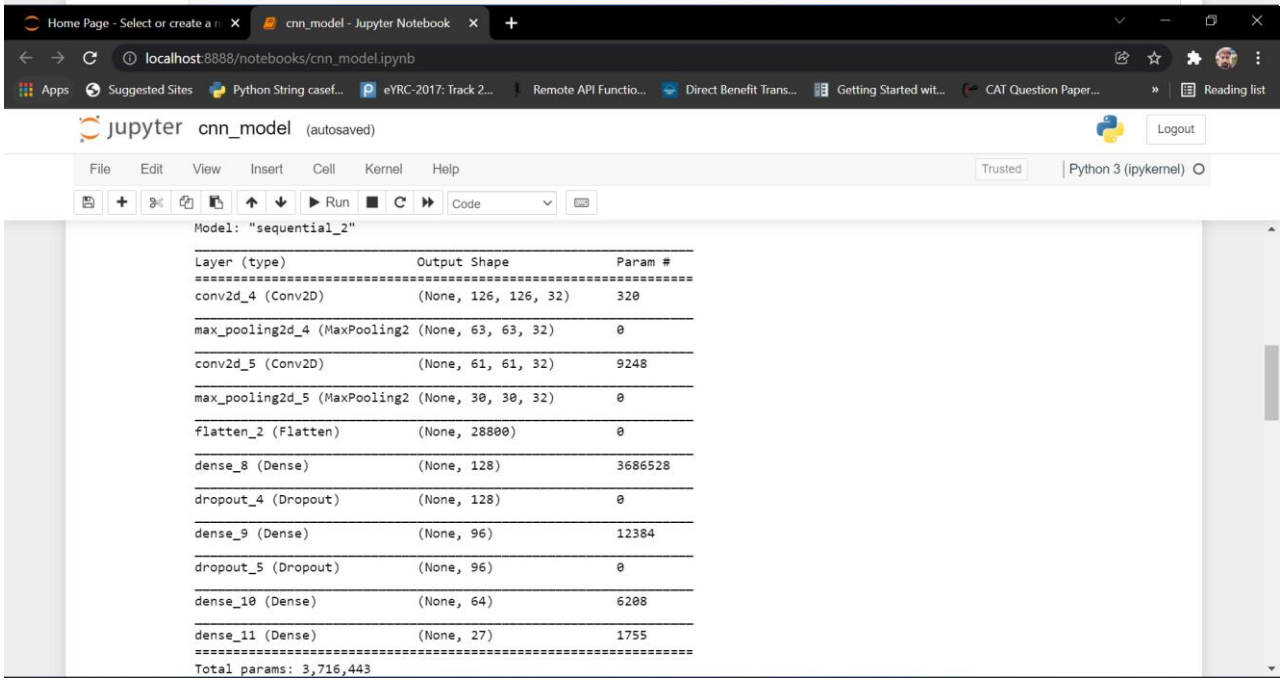
- **We will first create a window to take input image from our camera (web cam).**
- **After tracking the hands, next step is to segment the hands from the background.**
- **We then extract several general features and establish the relationship between features and classes.**
- **After extracting the features of the input characters, we search its features in the Dataset from Kaggle and consider the most similar features as the result.**
- **We will then train the model to achieve precise output.**
- **We will then test and validate the trained model.**
- **Installing the model on the website.**

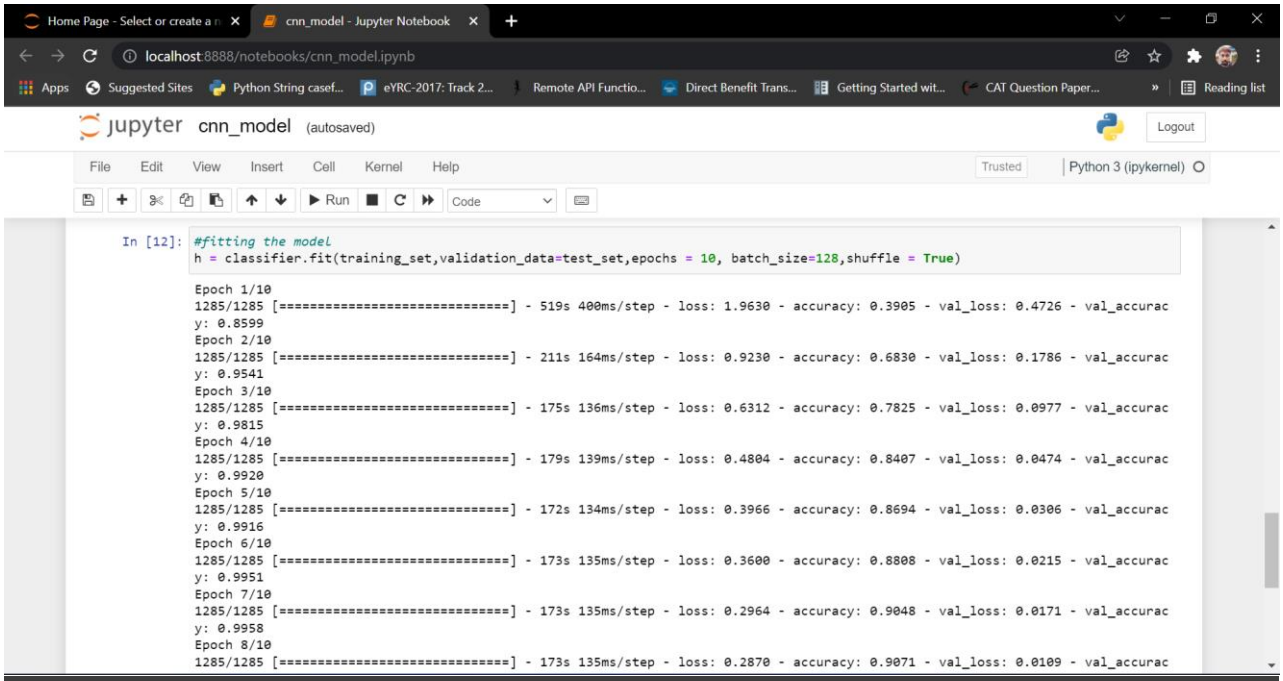
7. Project diagrams

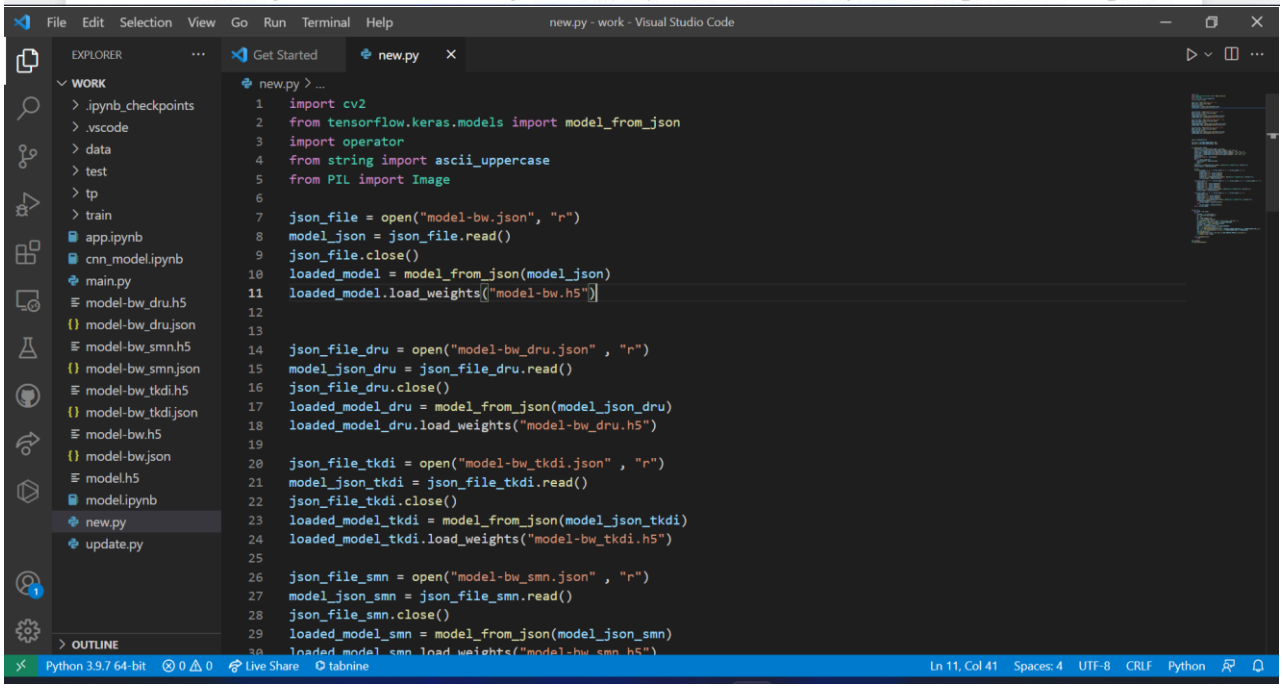


8. Testing


Home Page - Select or create a n
cnn_model - Jupyter Notebook
localhost:8888/notebooks/cnn_model.ipynb
Apps
Suggested Sites
Python String casef...
eYRC-2017: Track 2...
Remote API Functio...
Direct Benefit Trans...
Getting Started wit...
CAT Question Paper...
Reading list
jupyter cnn_model (autosaved)
Logout
File Edit View Insert Cell Kernel Help
Trusted
Python 3 (ipykernel)
Run
Code
# Initializing the CNN
classifier = Sequential()
# First convolution layer and pooling
classifier.add(Convolution2D(32, (3, 3), input_shape=(sz, sz, 1), activation='relu'))
classifier.add(MaxPooling2D(pool_size=(2, 2)))
# Second convolution layer and pooling
classifier.add(Convolution2D(32, (3, 3), activation='relu'))
# input_shape is going to be the pooled feature maps from the previous convolution layer
classifier.add(MaxPooling2D(pool_size=(2, 2)))
# Flattening the layers
classifier.add(Flatten())
# Adding a fully connected layer
classifier.add(Dense(units=128, activation='relu'))
classifier.add(Dropout(0.40))
classifier.add(Dense(units=96, activation='relu'))
classifier.add(Dropout(0.40))
classifier.add(Dense(units=64, activation='relu'))
classifier.add(Dense(units=27, activation='softmax')) # softmax for more than 2
classifier.summary()


Home Page - Select or create a n
cnn_model - Jupyter Notebook
localhost:8888/notebooks/cnn_model.ipynb
Apps
Suggested Sites
Python String casef...
eYRC-2017: Track 2...
Remote API Functio...
Direct Benefit Trans...
Getting Started wit...
CAT Question Paper...
Reading list
jupyter cnn_model (autosaved)
Logout
File Edit View Insert Cell Kernel Help
Trusted
Python 3 (ipykernel)
Run
Code
Model: "sequential_2"
Layer (type) Output Shape Param #
conv2d_4 (Conv2D) (None, 126, 126, 32) 320
max_pooling2d_4 (MaxPooling2 (None, 63, 63, 32) 0
conv2d_5 (Conv2D) (None, 61, 61, 32) 9248
max_pooling2d_5 (MaxPooling2 (None, 30, 30, 32) 0
flatten_2 (Flatten) (None, 28800) 0
dense_8 (Dense) (None, 128) 3686528
dropout_4 (Dropout) (None, 128) 0
dense_9 (Dense) (None, 96) 12384
dropout_5 (Dropout) (None, 96) 0
dense_10 (Dense) (None, 64) 6208
dense_11 (Dense) (None, 27) 1755
Total params: 3,716,443


Home Page - Select or create a n
cnn_model - Jupyter Notebook
localhost:8888/notebooks/cnn_model.ipynb
jupyter cnn_model (autosaved)
Logout
File Edit View Insert Cell Kernel Help
Trusted
Python 3 (ipykernel)
Run
Code
In [12]:
#fitting the model
h = classifier.fit(training_set,validation_data=test_set,epochs = 10, batch_size=128,shuffle = True)
Epoch 1/10
1285/1285 [==============================] - 519s 400ms/step - loss: 1.9630 - accuracy: 0.3905 - val_loss: 0.4726 - val_accuracy: 0.8599
Epoch 2/10
1285/1285 [==============================] - 211s 164ms/step - loss: 0.9230 - accuracy: 0.6830 - val_loss: 0.1786 - val_accuracy: 0.9541
Epoch 3/10
1285/1285 [==============================] - 175s 136ms/step - loss: 0.6312 - accuracy: 0.7825 - val_loss: 0.0977 - val_accuracy: 0.9815
Epoch 4/10
1285/1285 [==============================] - 179s 139ms/step - loss: 0.4804 - accuracy: 0.8407 - val_loss: 0.0474 - val_accuracy: 0.9920
Epoch 5/10
1285/1285 [==============================] - 172s 134ms/step - loss: 0.3966 - accuracy: 0.8694 - val_loss: 0.0306 - val_accuracy: 0.9916
Epoch 6/10
1285/1285 [==============================] - 173s 135ms/step - loss: 0.3600 - accuracy: 0.8808 - val_loss: 0.0215 - val_accuracy: 0.9951
Epoch 7/10
1285/1285 [==============================] - 173s 135ms/step - loss: 0.2964 - accuracy: 0.9048 - val_loss: 0.0171 - val_accuracy: 0.9958
Epoch 8/10
1285/1285 [==============================] - 173s 135ms/step - loss: 0.2870 - accuracy: 0.9071 - val_loss: 0.0109 - val_accuracy


File Edit Selection View Go Run Terminal Help
new.py - work - Visual Studio Code
EXPLORER
WORK
.ipynb_checkpoints
.vscode
data
test
tp
train
app.ipynb
cnn_model.ipynb
main.py
model-bw_dru.h5
model-bw_dru.json
model-bw_smn.h5
model-bw_smn.json
model-bw_tkdi.h5
model-bw_tkdi.json
model-bw.h5
model-bw.json
model.h5
model.ipynb
new.py
update.py
OUTLINE
Get Started
new.py
new.py > ...
import cv2
from tensorflow.keras.models import model_from_json
import operator
from string import ascii_uppercase
from PIL import Image
json_file = open("model-bw.json", "r")
model_json = json_file.read()
json_file.close()
loaded_model = model_from_json(model_json)
loaded_model.load_weights("model-bw.h5")
json_file_dru = open("model-bw_dru.json" , "r")
model_json_dru = json_file_dru.read()
json_file_dru.close()
loaded_model_dru = model_from_json(model_json_dru)
loaded_model_dru.load_weights("model-bw_dru.h5")
json_file_tkdi = open("model-bw_tkdi.json" , "r")
model_json_tkdi = json_file_tkdi.read()
json_file_tkdi.close()
loaded_model_tkdi = model_from_json(model_json_tkdi)
loaded_model_tkdi.load_weights("model-bw_tkdi.h5")
json_file_smn = open("model-bw_smn.json" , "r")
model_json_smn = json_file_smn.read()
json_file_smn.close()
loaded_model_smn = model_from_json(model_json_smn)
Python 3.9.7 64-bit
Live Share
tabnine
Ln 11, Col 41
Spaces: 4
UTF-8
CRLF
Python

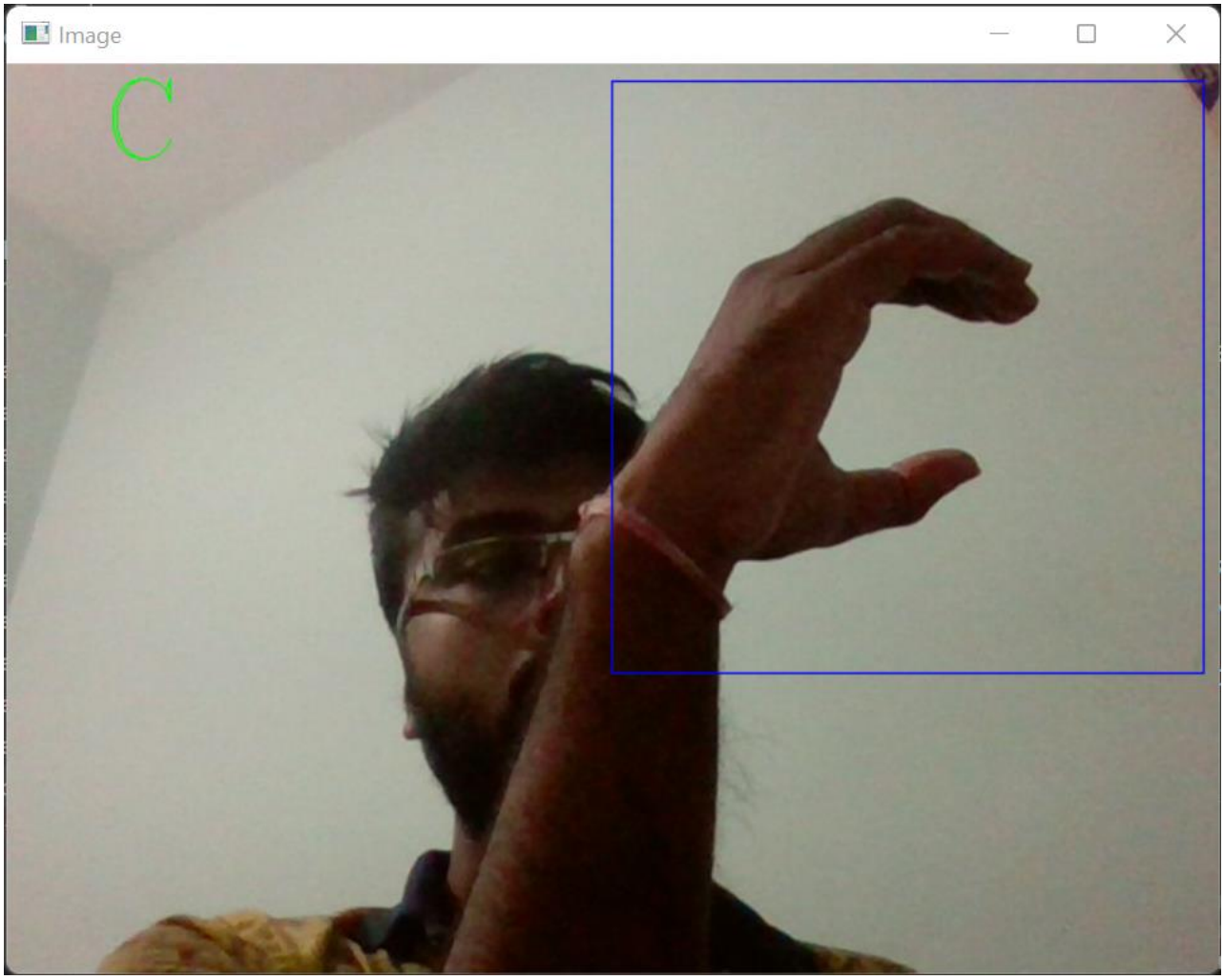


Image
C

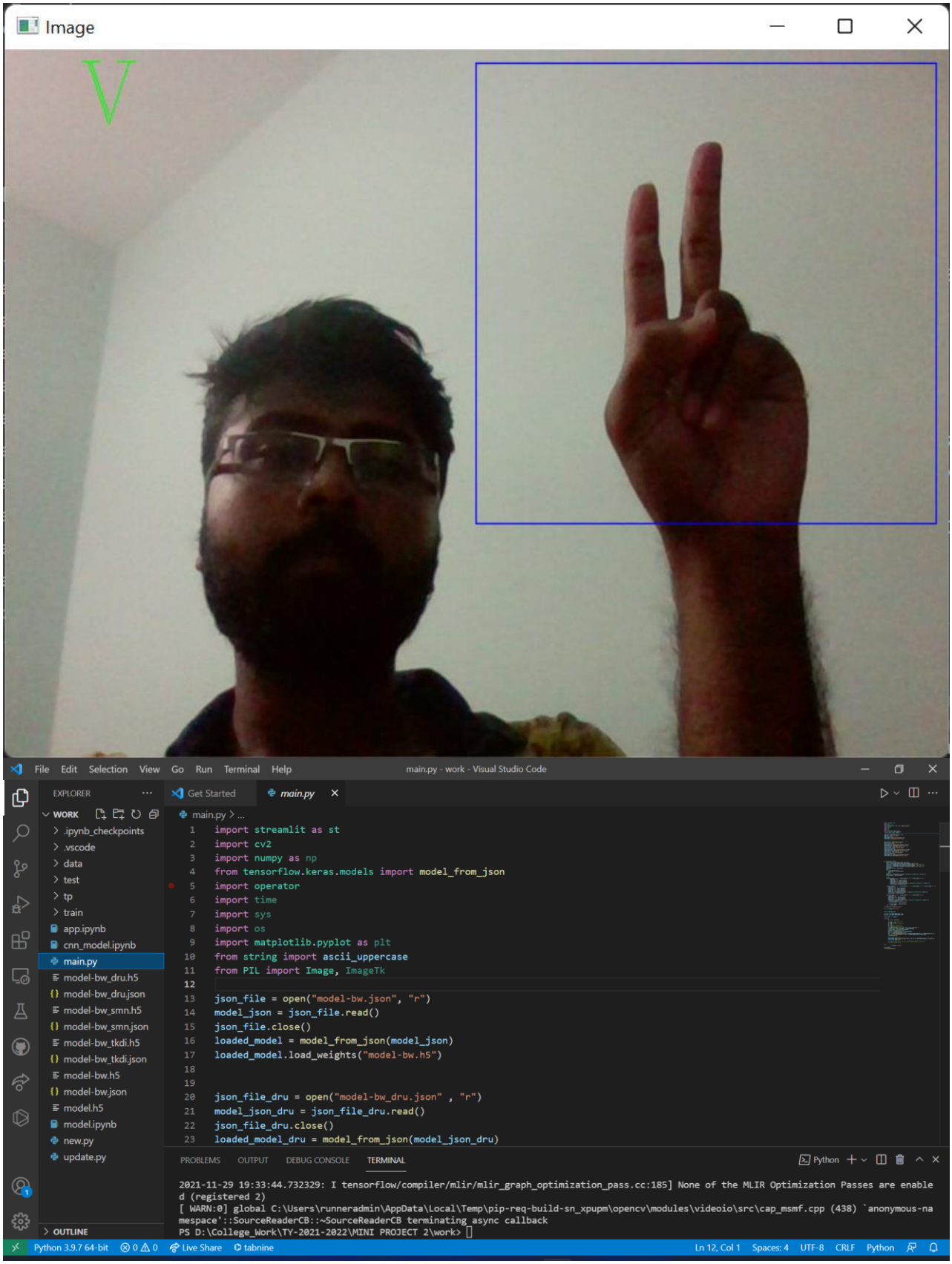
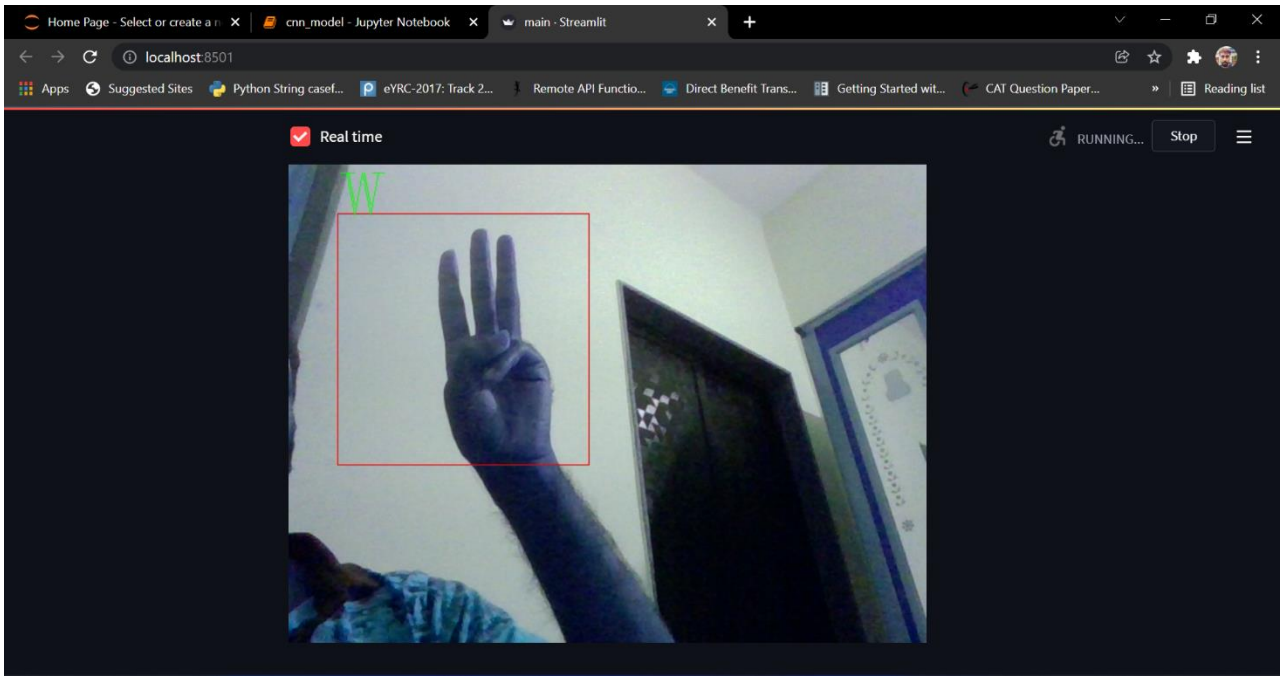


Image
V
File Edit Selection View Go Run Terminal Help
main.py - work - Visual Studio Code
EXPLORER
WORK
.ipynb_checkpoints
.vscode
data
test
tp
train
app.ipynb
cnn_model.ipynb
main.py
model-bw_dru.h5
model-bw_dru.json
model-bw_smn.h5
model-bw_smn.json
model-bw_tkdi.h5
model-bw_tkdi.json
model-bw.h5
model-bw.json
model.h5
model.ipynb
new.py
update.py
OUTLINE
Get Started
main.py
main.py > ...
1 import streamlit as st
2 import cv2
3 import numpy as np
4 from tensorflow.keras.models import model_from_json
5 import operator
6 import time
7 import sys
8 import os
9 import matplotlib.pyplot as plt
10 from string import ascii_uppercase
11 from PIL import Image, ImageTk
12
13 json_file = open("model-bw.json", "r")
14 model_json = json_file.read()
15 json_file.close()
16 loaded_model = model_from_json(model_json)
17 loaded_model.load_weights("model-bw.h5")
18
19
20 json_file_dru = open("model-bw_dru.json" , "r")
21 model_json_dru = json_file_dru.read()
22 json_file_dru.close()
23 loaded_model_dru = model_from_json(model_json_dru)
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Python
2021-11-29 19:33:44.732329: I tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:185] None of the MLIR Optimization Passes are enabled (registered 2)
[WARN:0] global C:\Users\runneradmin\AppData\Local\Temp\pip-req-build-sn_xpupm\opencv\modules\videoio\src\cap_msmf.cpp (438) `anonymous-namespace'::SourceReaderCB::~SourceReaderCB terminating async callback
PS D:\College_Work\TY-2021-2022\MINI PROJECT 2\work>
Python 3.9.7 64-bit
0 0
Live Share
tabnine
Ln 12, Col 1
Spaces: 4
UTF-8
CRLF
Python


Home Page - Select or create a n
cnn_model - Jupyter Notebook
main - Streamlit
localhost:8501
Apps
Suggested Sites
Python String casef...
eYRC-2017: Track 2...
Remote API Functio...
Direct Benefit Trans...
Getting Started wit...
CAT Question Paper...
Reading list
Real time
RUNNING...
Stop
W

9. Results and Conclusion

- **CNN can be used to provide improved sign language recognition using hand gestures results.**
- **This model will be beneficiary in many fields and will provide natural, innovative and user-friendly way of communication with the computer which is more familiar to human beings.**
- **We see this application having real potential in improving the lives of hearing-impaired and it would be worthy to continue development.**
- **This model could progress if we apply the concept in real-time.**

10. References

[1] 9 A. A. T. Yang, Y. Xu, and “A., Hidden Markov Model for Gesture Recognition”, CMU-RI-TR-94 10, Robotics Institute, Carnegie Mellon Univ., Pittsburgh, PA, May 1994.

[2] Zaki, M.M., Shaheen, S.I.: Sign language recognition using a combination of new vision-based features. Pattern Recognition Letters 32(4), 572–577 (2011) F. Dornaika, I. Arganda-Carreras, and C. Belver, “Age estimation in facial images through transfer learning,” Mach. Vis. Appl., vol. 30, no. 1, pp. 177–187, 2019.

[3] Felix Zhan, "Hand Gesture Recognition with Convolution Neural Networks",2019 IEEE 20th International Conference on Information Reuse and Integration for Data Science (IRI),2019.

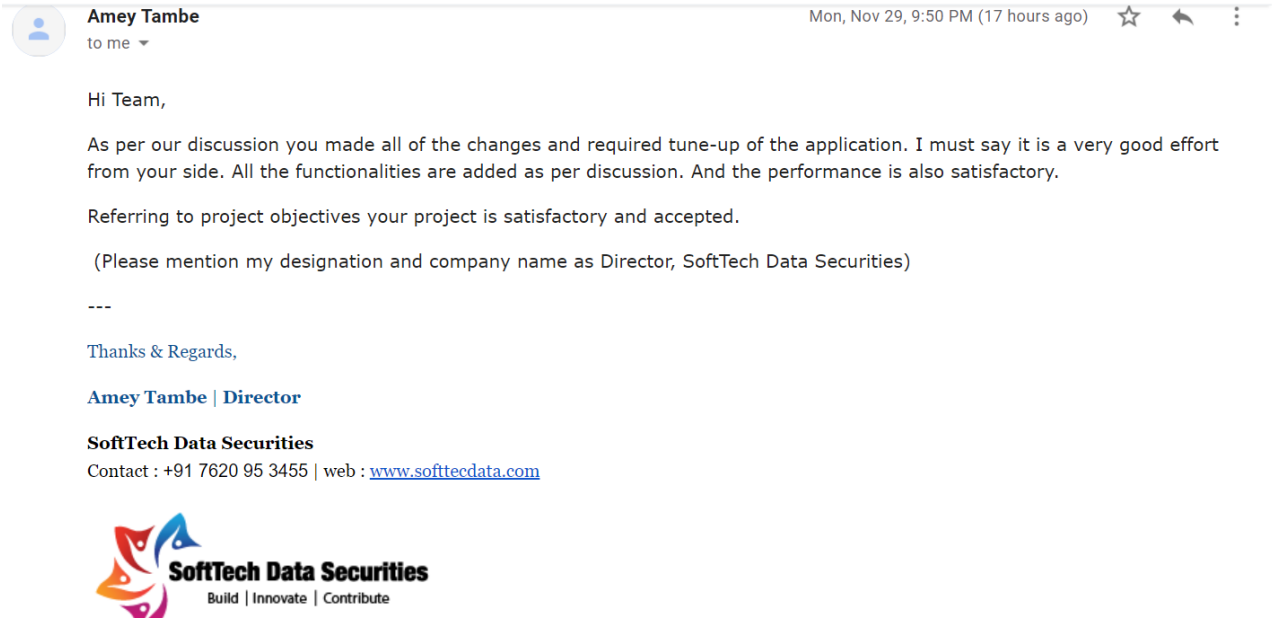
[4] Byeongkeun Kang, Subarna Tripathi, Truong Q. Nguyen” Real-time sign language fingerspelling recognition using convolutional neural networks from depth map” 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)

Annexure A

Meeting with Mentor

Sr.No	Day, Date and time of meeting	Discussion regarding (SRS, implementation, testing, problems during implementation, progress review etc.)	Meeting Summary
1	8 September	Topic Finalization	Topic was finalized
2	23 September	Project Introduction and Overview	Complete project was discussed
3	20 October	Progress meet	Problems faced were cleared
4	26 November	Project Finalization	Project implementation was done
5	29 November	Final Acceptance	Approval was received

Final acceptance received from mentor on implemented product (Email screenshot)



Amey Tambe Mon, Nov 29, 9:50 PM (17 hours ago)

to me

Hi Team,

As per our discussion you made all of the changes and required tune-up of the application. I must say it is a very good effort from your side. All the functionalities are added as per discussion. And the performance is also satisfactory.

Referring to project objectives your project is satisfactory and accepted.

(Please mention my designation and company name as Director, SoftTech Data Securities)

Thanks & Regards,

Amey Tambe | Director

SoftTech Data Securities
Contact : +91 7620 95 3455 | web : www.softtecdata.com

SoftTech Data Securities
Build | Innovate | Contribute