

# A Gesture-based Tool for Sterile Browsing of Radiology. Image CNN and Open CV

## **INDEX:**

- 1.INTRODUCTION
- 2.LITERATURE SURVEY
- 3.THEORITICAL ANALYSIS
- 4.EXPERIMENTAL INVESTIGATIONS
- 5.FLOW CHARY
- 6.RESULT
- 7.ADVANTAGES AND DISADVANTAGES
- 8.APPLICATIONS
- 9.CONCLUSION
- 10.FUTURE SCOPE
- 11.BIBILOGRAPHY
- 12.APPENDIX
  - a.Source Code
  - b.UI output Screenshot

## **1.INTRODUCTION:**

Humans are able to recognize body and sign language easily. This is possible due to the combination of vision and synaptic interactions that were formed along brain development. In order to replicate this skill in computers, some problems need to be solved: how to separate objects of interest in images and which image capture technology and classification technique are more appropriate, among others.

In this project Gesture based Desktop automation, First model is trained pre trained on the images of different hand gestures, such as showing numbers with fingers as 1,2,3,4. This model uses the integrated webcam to capture video frame. The image of the gesture captured in the video frame is compared with the Pre-trained model and the gesture is identified. If the gesture predicted is 1 then image is blurred; 2, image is resized; 3, image is rotated etc..

### Overview:

1. Defining our classification categories
2. Collect training images
3. Train the model
4. Test our mode

## **2.LITERATURE SURVEY :**

To complete this project, you must require following software's, concept and packages

- **Anconda navigator**

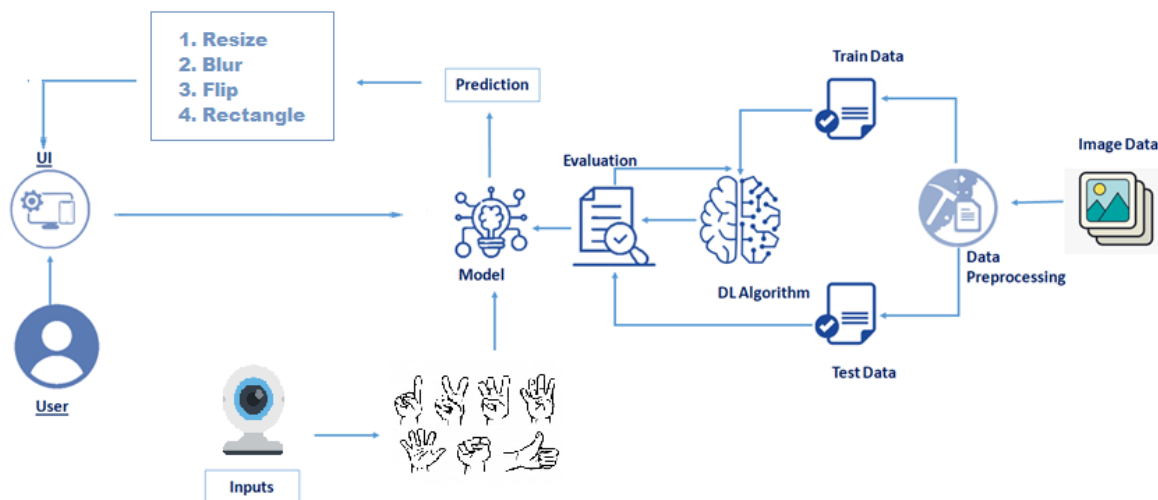
📁 Refer to the link below to download anaconda navigator

- **Python packages:**

- 📁 Open anaconda prompt as administrator
- 📁 Type "pip install tensorflow" (make sure you are working on python 64 bit)
- 📁 Type "pip install opencv-python"
- 📁 Type "pip install flask".

### 3.THEORITICAL ANALYSIS:

#### BLOCK DIAGRAM:



### 4.EXPERIMENTAL INVESTIGATIONS:

**CNN** : A convolutional neural network is a class of deep neural networks, most commonly applied to analyzing visual imagery.

**Opencv** - It is an Open Source Computer Vision Library which are mainly used for image

processing, video capture and analysis including features like face detection and object detection.

**Flask:** Flask is a popular web python web framework, meaning it is a third-party Python library used for developing web applications.

## 5.FLOWCHART:

Project Flow:

- User interacts with the UI(User Interface) to upload image input
- Depending on the different gesture inputs different operations are applied to the input image.
- Once model analyses the gesture, the prediction with operation applied on image is showcased on the UI.

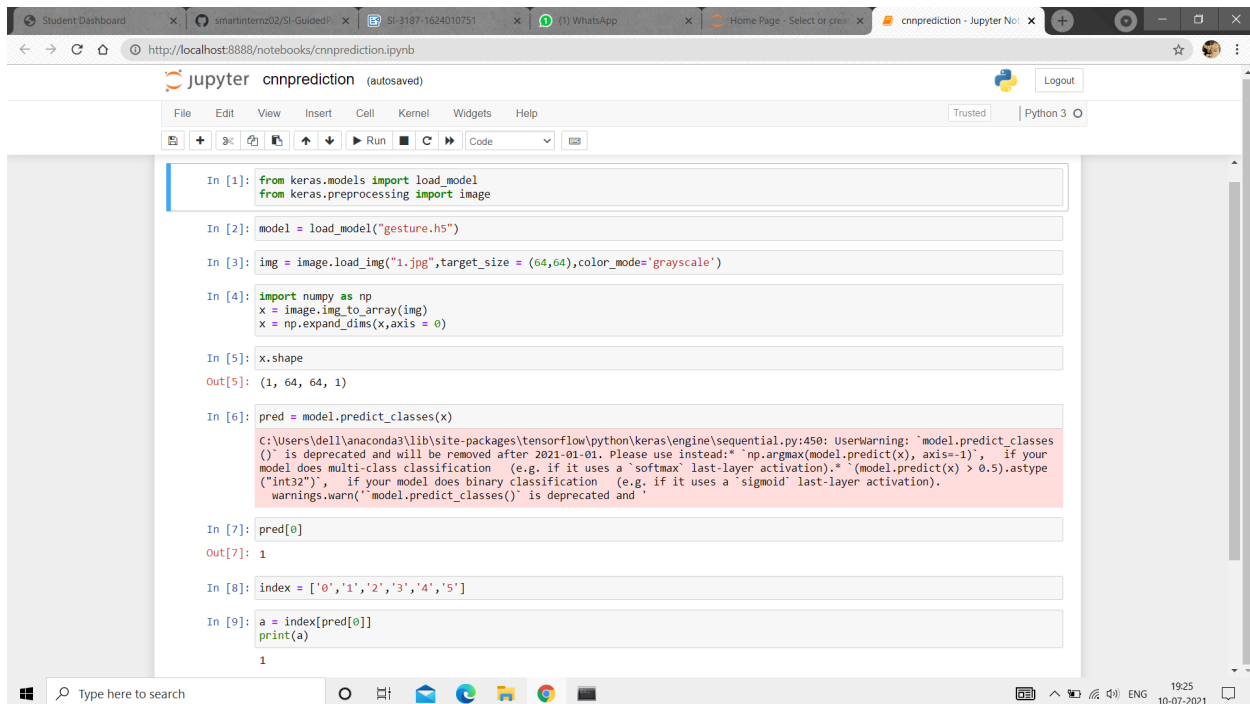
To accomplish this, we have to complete all the activities and tasks listed below

- Data Collection
- Data Preprocessing
- Import the ImageDataGenerator library
- Configure ImageDataGenerator class
- Model Building
  - Import the model building libraries
  - Initializing the model
  - Adding Input Layer
  - Adding Hidden Layer
  - Configure the learning process
  - Training and Testing the model
  - Save the Model
- Application Building
  - Create an HTML file
  - Build Python Code

## 6.RESULT:

This project aims at developing a system which is a sterile gesture interface for users,such as doctors/surgeons, to browse medical images in a dynamic medical environment.

A vision-based gesture capture system interprets user's gesture in realtime to navigatr through and manipulate an image and data visualization evnironment.Developed a deep learning model using CNN implemented in Keras backend Tensorflow and OpenCV which can recognize various gestures.



The screenshot displays a Jupyter Notebook titled 'cnnprediction' running on a local host. The notebook contains the following code and output:

```
In [1]: from keras.models import load_model
        from keras.preprocessing import image

In [2]: model = load_model("gesture.h5")

In [3]: img = image.load_img("1.jpg",target_size = (64,64),color_mode='grayscale')

In [4]: import numpy as np
        x = image.img_to_array(img)
        x = np.expand_dims(x,axis = 0)

In [5]: x.shape
Out[5]: (1, 64, 64, 1)

In [6]: pred = model.predict_classes(x)
C:\Users\dell\anaconda3\lib\site-packages\tensorflow\python\keras\engine\sequential.py:450: UserWarning: 'model.predict_classes()' is deprecated and will be removed after 2021-01-01. Please use instead: 'np.argmax(model.predict(x), axis=-1)', if your model does multi-class classification (e.g. if it uses a 'softmax' last-layer activation).*(model.predict(x) > 0.5).astype('int32')', if your model does binary classification (e.g. if it uses a 'sigmoid' last-layer activation).
  warnings.warn("'model.predict_classes()' is deprecated and

In [7]: pred[0]
Out[7]: 1

In [8]: index = ['0','1','2','3','4','5']

In [9]: a = index[pred[0]]
        print(a)
1
```

The output shows the model's prediction for the input image, which is '1'.

## 7.ADVANTAGES AND DISADVANTAGES :

### Advantages:

- Very high accuracy in image recognition problems.

- Automatically detects the important features without any human supervisions.
- Weight sharing

#### **Disadvantages:**

- CNN do not encode the position and orientation of object.
- Lack of ability to be spatially invariant to the input data
- Lots of training data is required

### **8.APPLICATIONS:**

- ✓ [Decoding Facia Recognition](#)
- ✓ [Analyzing Documents](#)
- ✓ [Historic and Environmental Collections](#)
- ✓ [Understanding Climate](#)
- ✓ [Grey Areas](#)
- ✓ [Advertising](#)
- ✓ [Other Interesting Fields](#)

### **9.CONCLUSION:**

- Know fundamental concepts and techniques of Convolution Neural Networks.
- Gain a broad understanding of image data.
- Know how to pre-process/clean the data using different data preprocessing techniques
- Know how to build a web application using Flask framework

### **10.FUTURE SCOPE:**

neural nets also have the power of flexibility. Once established, they can be applied to almost anything, whether it's helping people spot the issues interfering with their productivity or improving air traffic patterns for smoother flights. The core functionality of a neural net is to learn something efficiently, so if you have a system that can learn to recognize patterns, it could feasibly recognize patterns in almost any domain.

## 11.BIBILOGRAPHY:

We use References of previous works in websites.

[A Gesture-Based Tool For Sterile Browsing Of Radiology Im - Google Search](#)

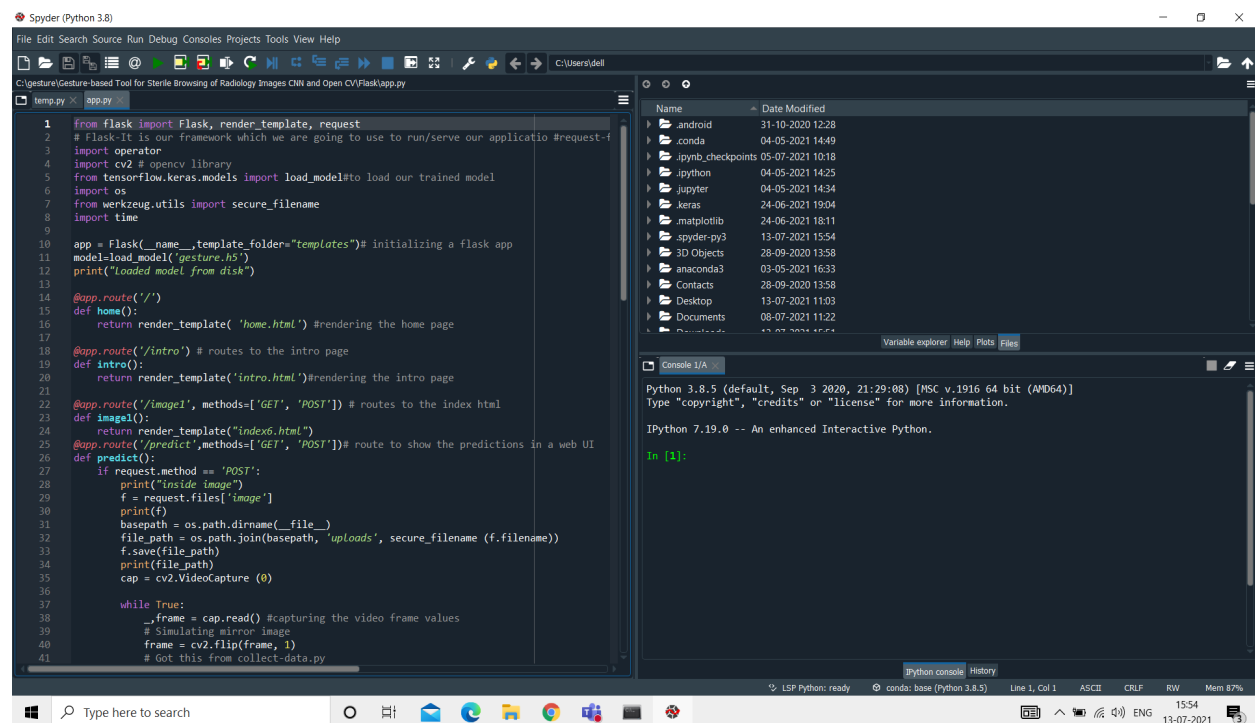
We used saw some Reference videos in You Tube.

[https://youtu.be/4y\\_zD-0Q3F8](https://youtu.be/4y_zD-0Q3F8)

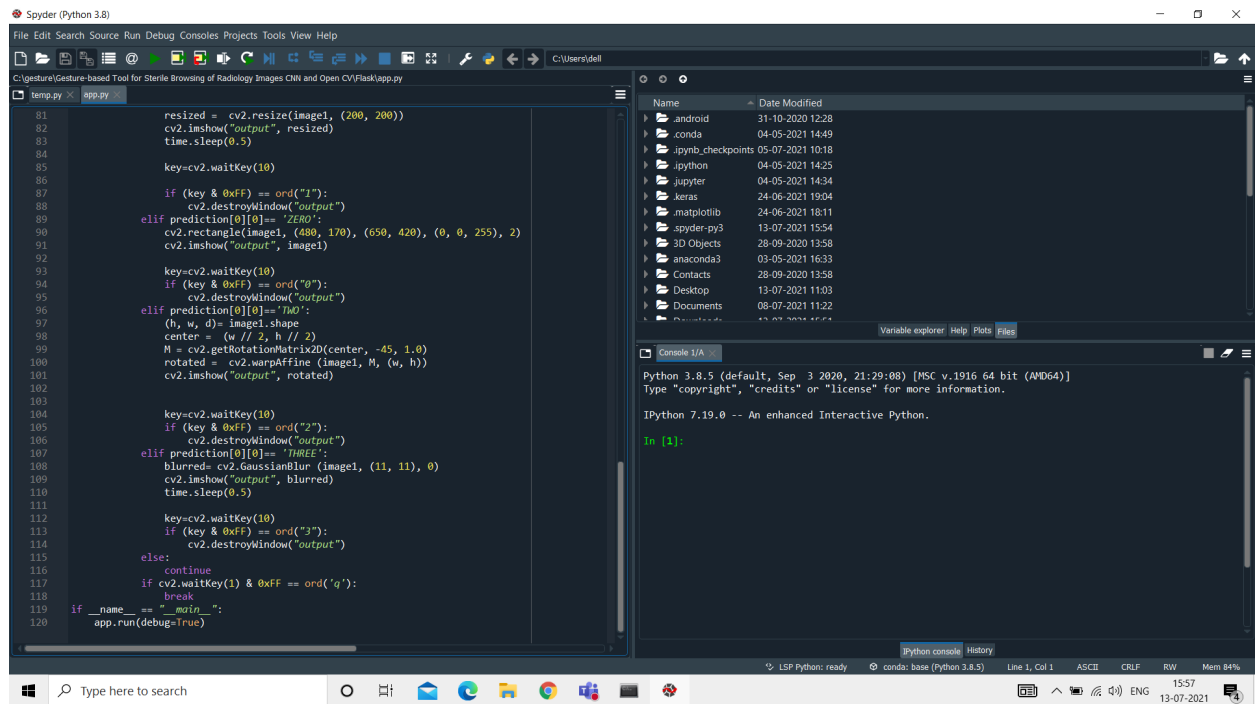
<https://youtu.be/BzougMGJ41k>

## 12.APPENDIX:

### a.Source Code



```
1 from flask import Flask, render_template, request
2 # flask-it is our framework which we are going to use to run/serve our applicatio #request-
3 import operator
4 import cv2 # opencv library
5 from tensorflow.keras.models import load_model#to load our trained model
6 import os
7 from werkzeug.utils import secure_filename
8 import time
9
10 app = Flask(__name__, template_folder="templates")# initializing a flask app
11 model=load_model('gesture.h5')
12 print("Loaded model from disk")
13
14 @app.route('/')
15 def home():
16     return render_template('home.html') #rendering the home page
17
18 @app.route('/intro') # routes to the intro page
19 def intro():
20     return render_template('intro.html')#rendering the intro page
21
22 @app.route('/image1', methods=['GET', 'POST']) # routes to the index html
23 def image1():
24     return render_template("index6.html")
25
26 @app.route('/predict',methods=['GET', 'POST'])# route to show the predictions in a web UI
27 def predict():
28     if request.method == 'POST':
29         print("inside image")
30         f = request.files['image']
31         print(f)
32         basepath = os.path.dirname(__file__)
33         file_path = os.path.join(basepath, 'uploads', secure_filename(f.filename))
34         f.save(file_path)
35         print(file_path)
36         cap = cv2.VideoCapture(0)
37
38         while True:
39             frame = cap.read() #capturing the video frame values
40             # Simulating mirror image
41             frame = cv2.flip(frame, 1)
42             # Got this from collect-data.py
```



## b.UI output Screenshot :

