Project Development Phase

Model Performance Test

Date	19 November 2022
Team ID	PNT2022TMID16818
Project Name	AI-powered Nutrition Analyzer for Fitness Enthusiasts
Maximum Marks	10 Marks

ModelPerformance Testing:

Project team shall fill the following information in model performance testing template.

S.No.	Parameter	Values	Screenshot
1.	Model Summary	Total params: 21,885,485 Trainable params: 1,024,005 Non-trainable params: 20,861,480	Attached below
2.	Accuracy	Training Accuracy - 72% Validation Accuracy - 59%	Attached below
3.	Confidence Score (Only Yolo Projects)	Class Detected - NILL Confidence Score - NILL	NILL

User Acceptance Testing

PRIEE PROFESSIONAL READINESS INNOVATION EMPLOYABILITY ENTREPRENURSHIP › TEAM IBM › FINAL DELIVERABLES › Final Code › Sample_Images
Test_Image1.jpg
Test_Image2.jpg
Test_Image3.jpg
Test_Image4.jpg
Test_Image5.jpg
Test_Image11.jpg

We are performing **White Box Testing** for select the package module.

Pseudocode for select the package module is-

```
select_the_package() procedure begins
READ the package name, features, offers and duration from the package database
DISPLAY the package name, features, offers and duration
DO
GET the package name, features, offers and duration
STORE the package selected to the customer's database
PROCEED to payment screen //another module
WHILE select package is NULL
//End DO...WHILE
procedure ends
```

FLOWGRAPH

Flowgraph

CYCLOMATIC COMPLEXITY OF RESULTANT GRAPH

V (G) = Number of regions

= 2

V (G) = Edges-Nodes+2

= 8-8+2

=2

V (G) = Predicate nodes+1

= 1+1

= 2

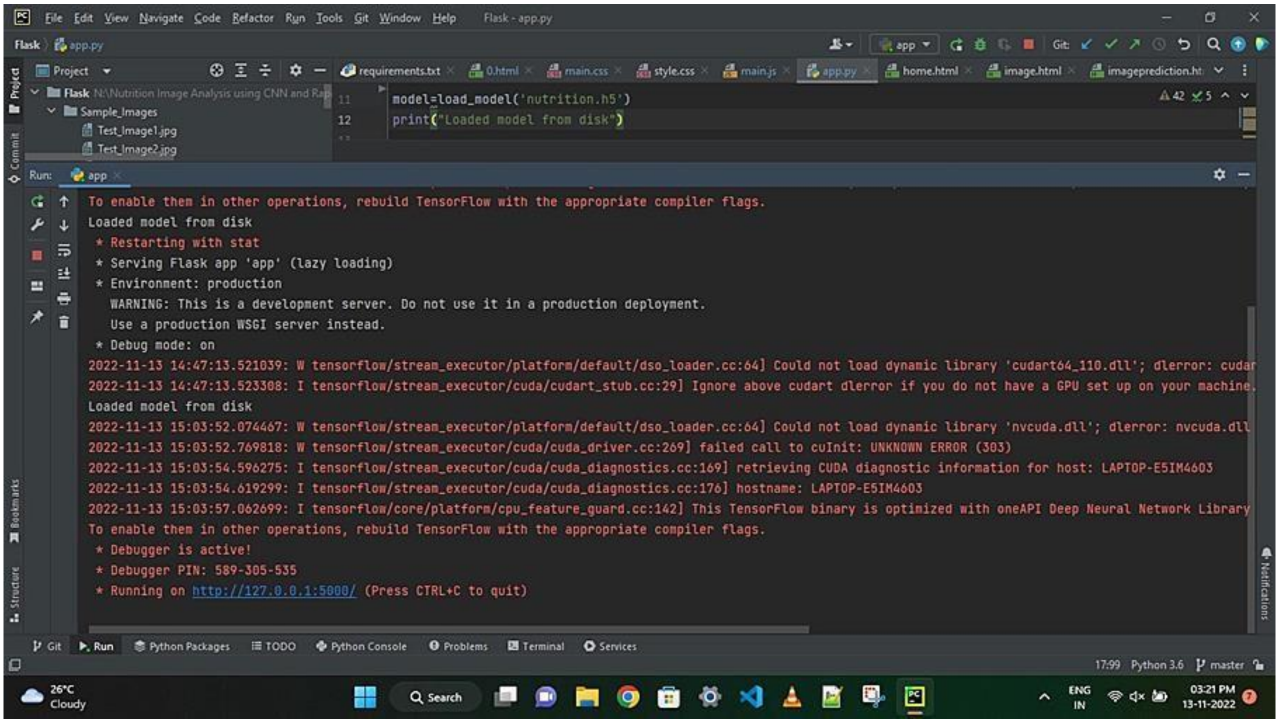
LINEARLY INDEPENDENT PATHS FOR FLOW GRAPHS

Path 1: 1-2-3-4-5-6-7-8-9-10

Path 2: 1-2-3-4-5-6-7-8-4-5-6-7-8-9-10

TEST ID	INPUT VALUES	ACTUAL OUTPUT	EXPECTED OUTPUT
1	Package is selected	To be observed after execution	Display the selected package
2	Package is not selected	To be observed after execution	Show the packages to select until one is selected

Performance Testing



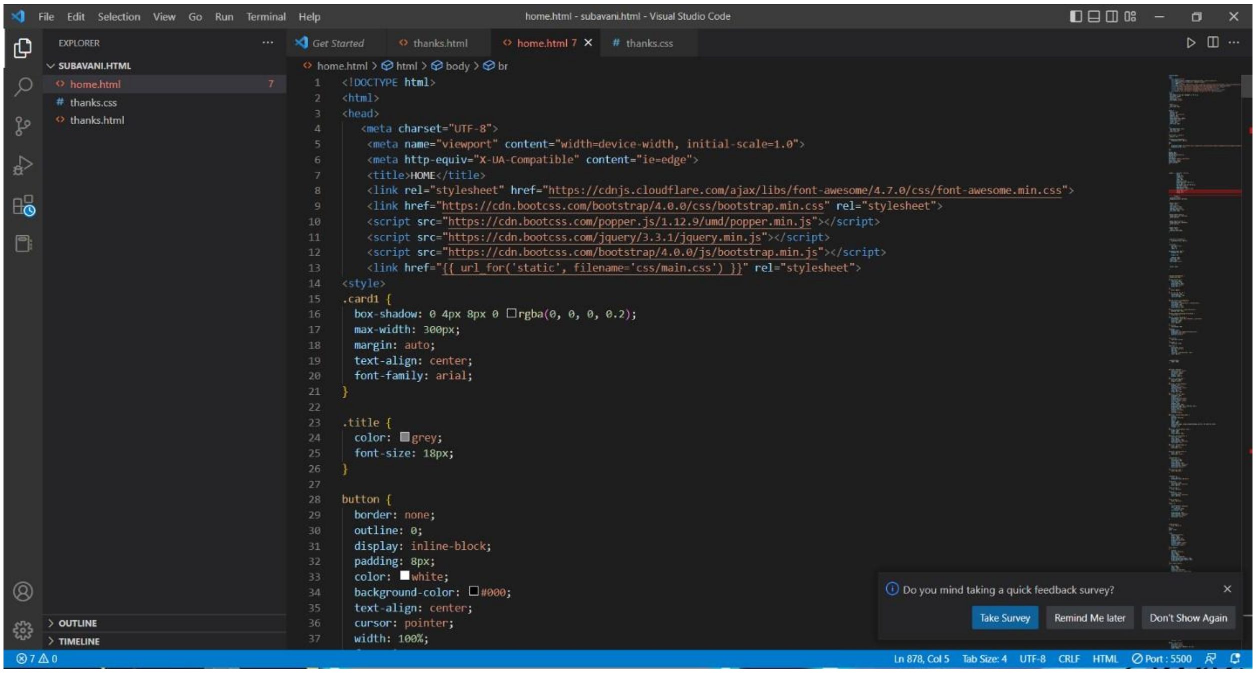
```
model=load_model('nutrition.h5')
print("Loaded model from disk")
```

```
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
Loaded model from disk
 * Restarting with stat
 * Serving Flask app 'app' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
2022-11-13 14:47:13.521039: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudart64_110.dll'; dlerror: cudar
2022-11-13 14:47:13.523308: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
Loaded model from disk
2022-11-13 15:03:52.074467: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'nvcuda.dll'; dlerror: nvcuda.dll
2022-11-13 15:03:52.769818: W tensorflow/stream_executor/cuda/cuda_driver.cc:269] failed call to cuInit: UNKNOWN ERROR (303)
2022-11-13 15:03:54.596275: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:169] retrieving CUDA diagnostic information for host: LAPTOP-E5IM4603
2022-11-13 15:03:54.619299: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:176] hostname: LAPTOP-E5IM4603
2022-11-13 15:03:57.062699: I tensorflow/core/platform/cpu_feature_guard.cc:142] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
 * Debugger is active!
 * Debugger PIN: 589-305-535
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

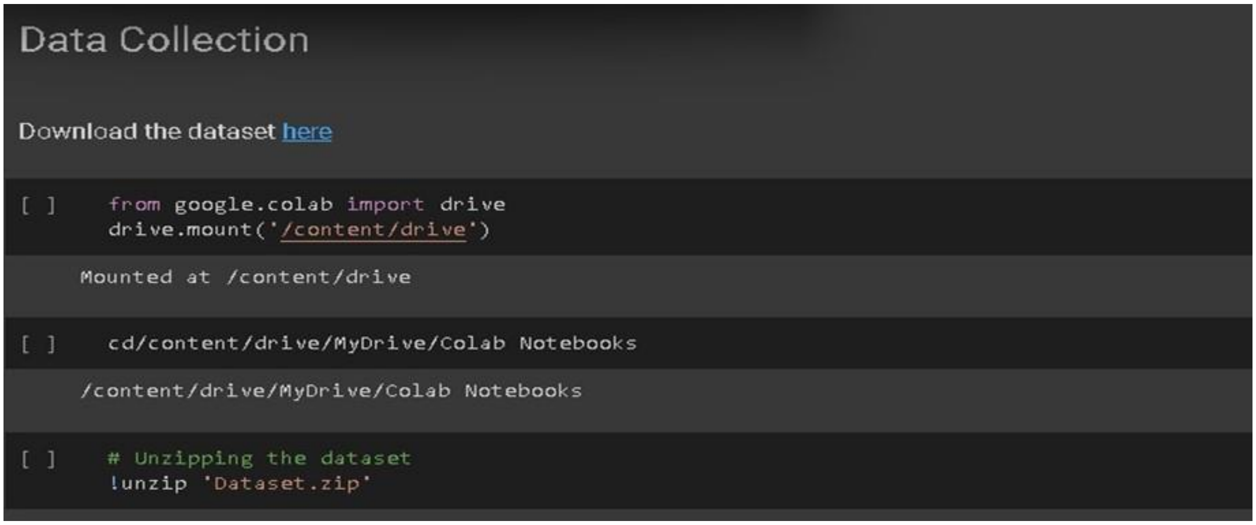
Future Scope

- Offers payment convenience in personal training subscriptions.
- Useful articles.
- Video instructions.
- Diet Plans.
- Individual progress tracking.

SCREENSHOTS :



```
<!DOCTYPE html>
<html>
<head>
   <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>HOME</title>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
    <link href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css" rel="stylesheet">
    <script src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"></script>
    <script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>
    <script src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
    <link href="{{ url_for('static', filename='css/main.css') }}" rel="stylesheet">
<style>
.card1 {
  box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2);
  max-width: 300px;
  margin: auto;
  text-align: center;
  font-family: arial;
}

.title {
  color: grey;
  font-size: 18px;
}

button {
  border: none;
  outline: 0;
  display: inline-block;
  padding: 8px;
  color: white;
  background-color: #000;
  text-align: center;
  cursor: pointer;
  width: 100%;
```



Data Collection

Download the dataset here

```
[ ]  from google.colab import drive
     drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
[ ]  cd/content/drive/MyDrive/Colab Notebooks
```

```
/content/drive/MyDrive/Colab Notebooks
```

```
[ ]  # Unzipping the dataset
     !unzip 'Dataset.zip'
```

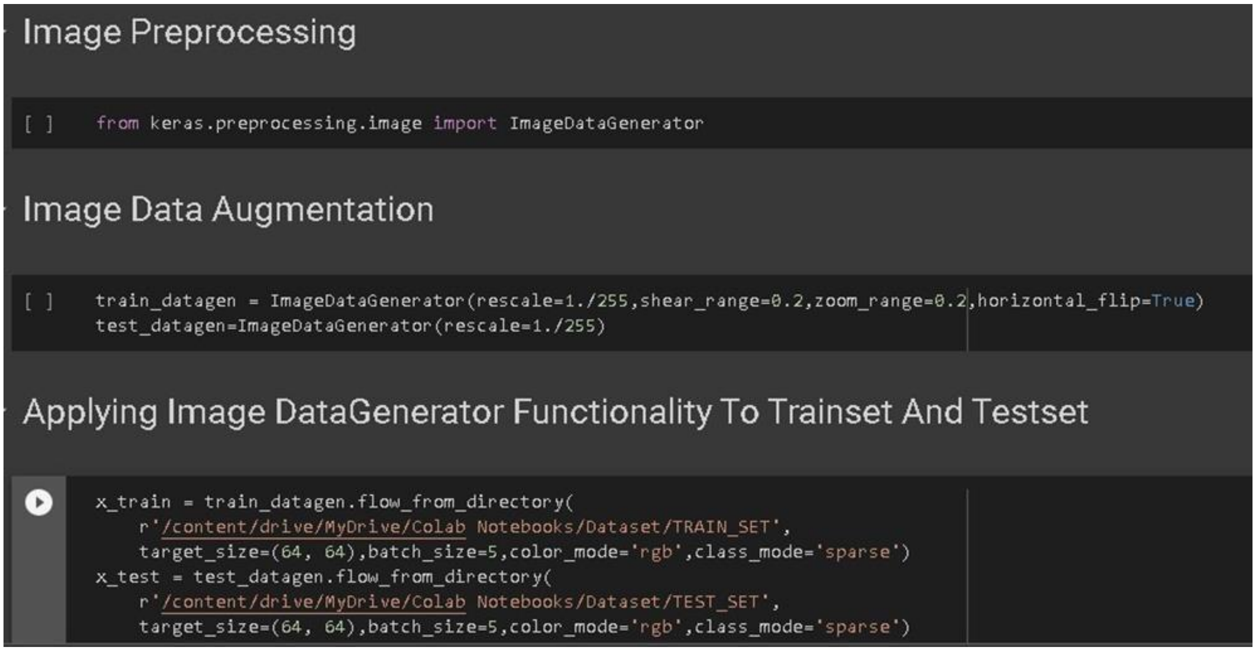


Image Preprocessing

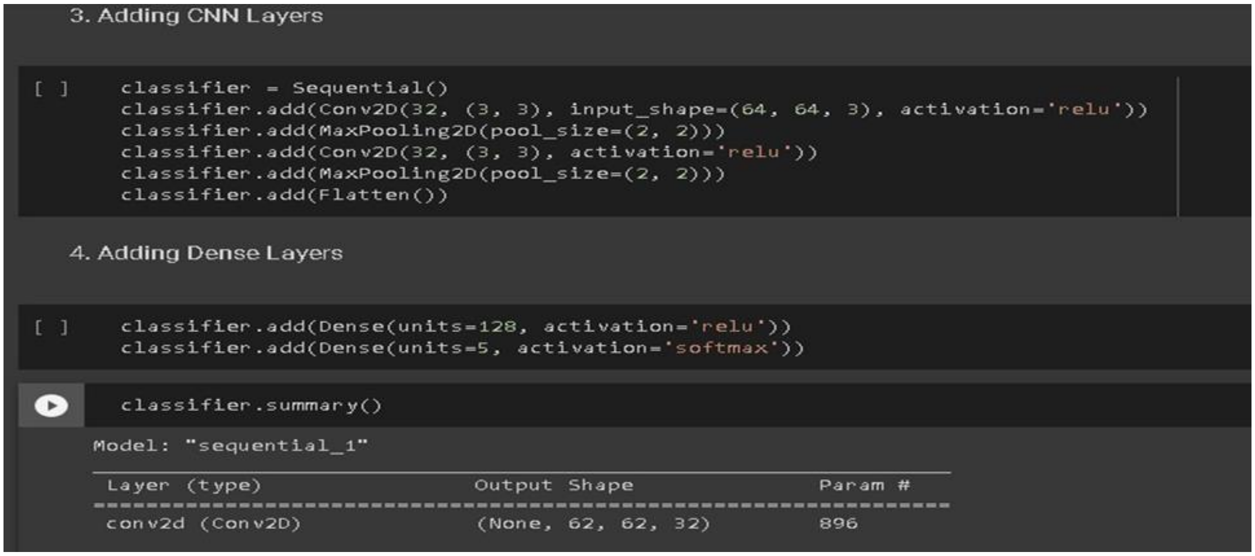
```
[ ]   from keras.preprocessing.image import ImageDataGenerator
```

Image Data Augmentation

```
[ ]   train_datagen = ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
      test_datagen=ImageDataGenerator(rescale=1./255)
```

Applying Image DataGenerator Functionality To Trainset And Testset

```
x_train = train_datagen.flow_from_directory(
    r'/content/drive/MyDrive/Colab Notebooks/Dataset/TRAIN_SET',
    target_size=(64, 64),batch_size=5,color_mode='rgb',class_mode='sparse')
x_test = test_datagen.flow_from_directory(
    r'/content/drive/MyDrive/Colab Notebooks/Dataset/TEST_SET',
    target_size=(64, 64),batch_size=5,color_mode='rgb',class_mode='sparse')
```



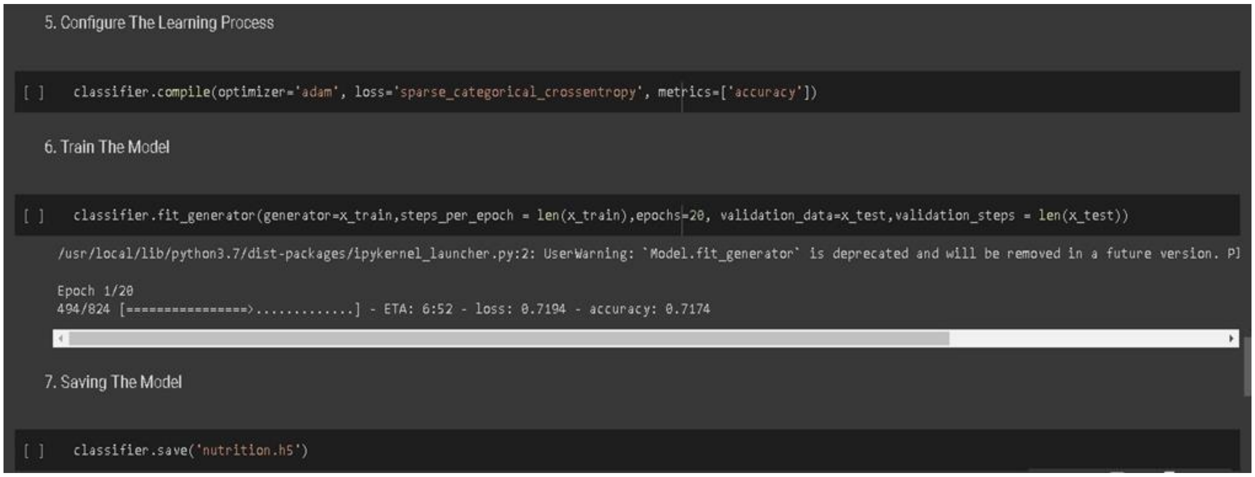
3. Adding CNN Layers

```
[ ]   classifier = Sequential()
      classifier.add(Conv2D(32, (3, 3), input_shape=(64, 64, 3), activation='relu'))
      classifier.add(MaxPooling2D(pool_size=(2, 2)))
      classifier.add(Conv2D(32, (3, 3), activation='relu'))
      classifier.add(MaxPooling2D(pool_size=(2, 2)))
      classifier.add(Flatten())
```

4. Adding Dense Layers

```
[ ]   classifier.add(Dense(units=128, activation='relu'))
      classifier.add(Dense(units=5, activation='softmax'))
```

```
      classifier.summary()

    Model: "sequential_1"
    _________________________________________________________________
     Layer (type)                Output Shape              Param #
    =================================================================
     conv2d (Conv2D)             (None, 62, 62, 32)        896
```



5. Configure The Learning Process

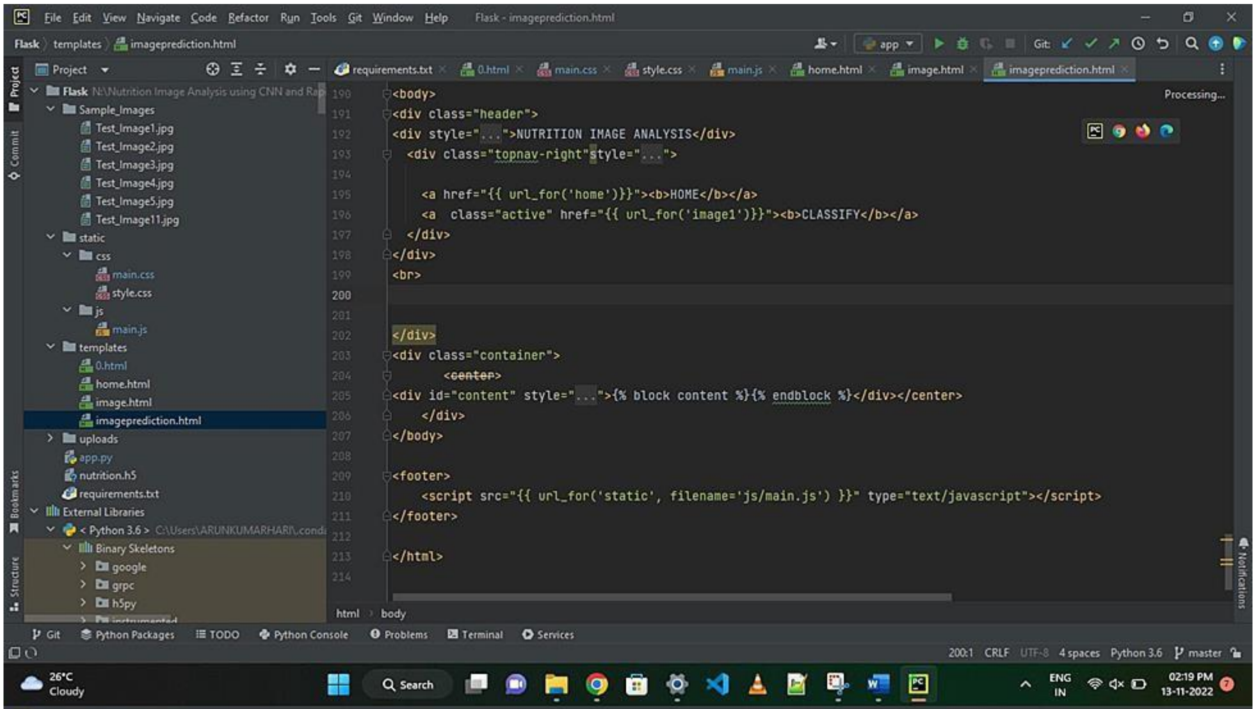
```
[ ]   classifier.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])
```

6. Train The Model

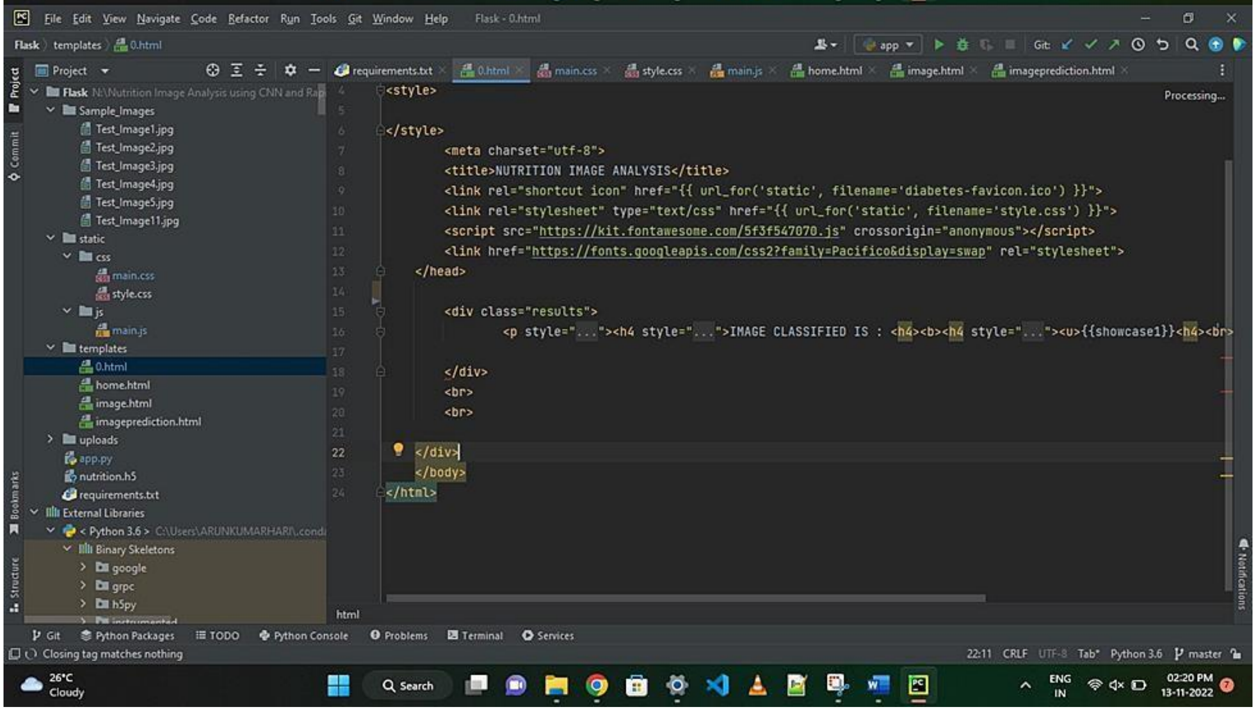
```
[ ]   classifier.fit_generator(generator=x_train,steps_per_epoch = len(x_train),epochs=20, validation_data=x_test,validation_steps = len(x_test))

    /usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Pl

    Epoch 1/20
    494/824 [================>............] - ETA: 6:52 - loss: 0.7194 - accuracy: 0.7174
```

7. Saving The Model

```
[ ]   classifier.save('nutrition.h5')
```

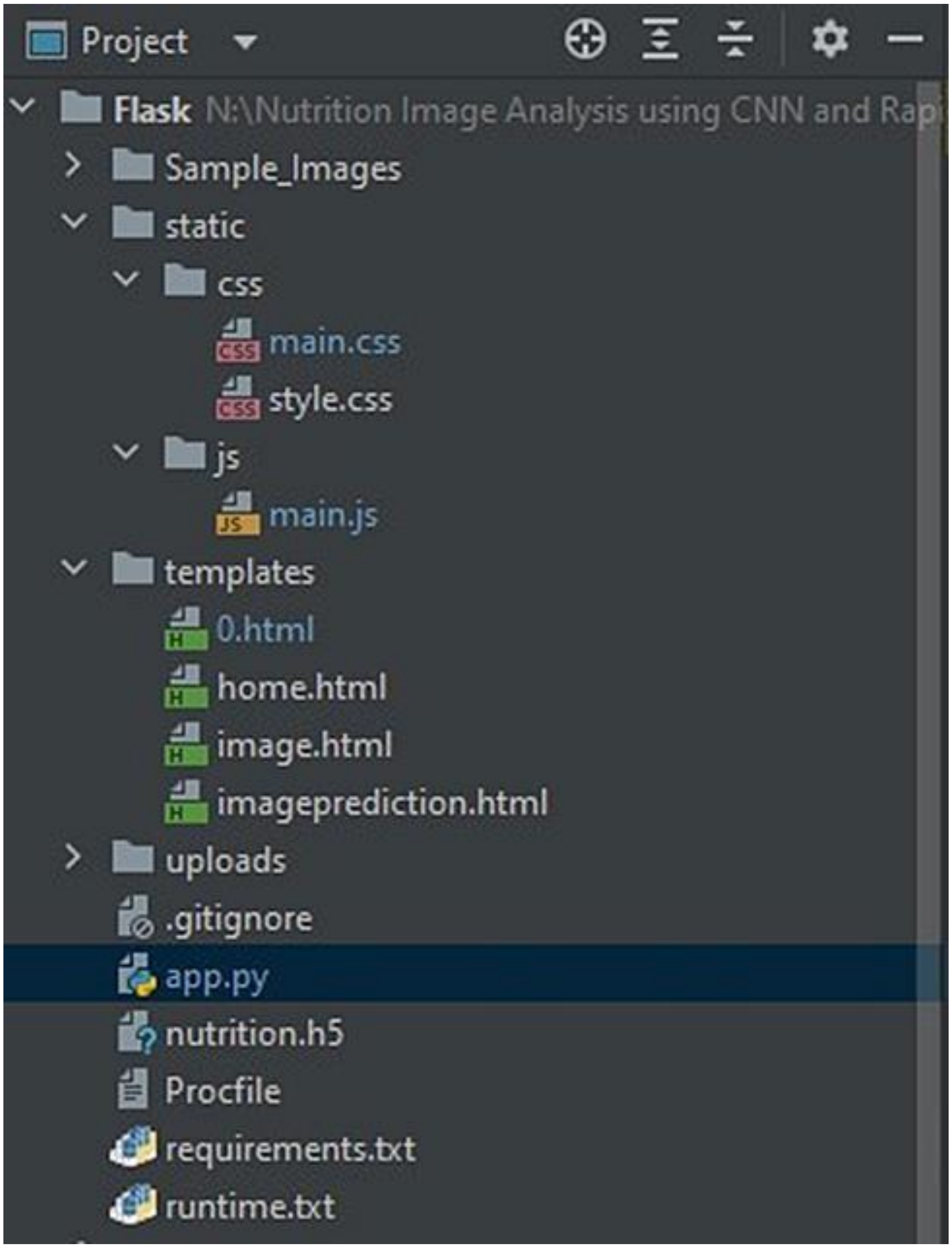


```
<body>
<div class="header">
<div style="...">NUTRITION IMAGE ANALYSIS</div>
  <div class="topnav-right"style="...">

    <a href="{{ url_for('home')}}"><b>HOME</b></a>
    <a  class="active" href="{{ url_for('image1')}}"><b>CLASSIFY</b></a>
  </div>
</div>
<br>


</div>
<div class="container">
        <center>
<div id="content" style="...">{% block content %}{% endblock %}</div></center>
    </div>
</body>

<footer>
    <script src="{{ url_for('static', filename='js/main.js') }}" type="text/javascript"></script>
</footer>

</html>
```



```
<style>

</style>
        <meta charset="utf-8">
        <title>NUTRITION IMAGE ANALYSIS</title>
        <link rel="shortcut icon" href="{{ url_for('static', filename='diabetes-favicon.ico') }}">
        <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='style.css') }}">
        <script src="https://kit.fontawesome.com/5f3f547070.js" crossorigin="anonymous"></script>
        <link href="https://fonts.googleapis.com/css2?family=Pacifico&display=swap" rel="stylesheet">
    </head>

        <div class="results">
                <p style="..."><h4 style="...">IMAGE CLASSIFIED IS : <h4><b><h4 style="..."><u>{{showcase1}}<h4><br>

        </div>
        <br>
        <br>

    </div>
    </body>
</html>
```

Closing tag matches nothing



```
Flask N:\Nutrition Image Analysis using CNN and Rap
  Sample_Images
  static
    css
      main.css
      style.css
    js
      main.js
  templates
    0.html
    home.html
    image.html
    imageprediction.html
  uploads
  .gitignore
  app.py
  nutrition.h5
  Procfile
  requirements.txt
  runtime.txt
```