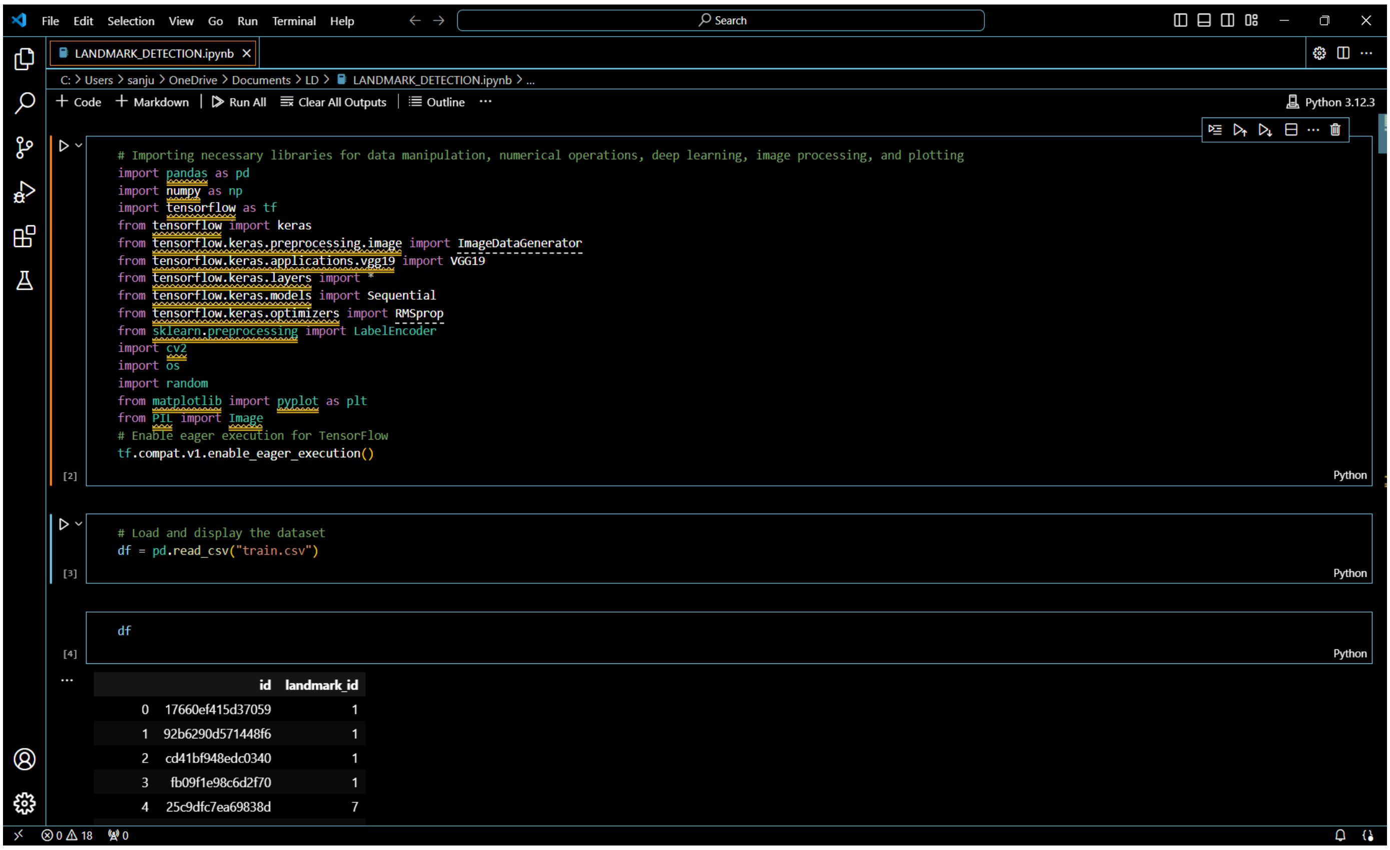
3. LANDMARK DETECTION AI-MODEL USING JUPYTER NOTEBOOK


File Edit Selection View Go Run Terminal Help
Search
LANDMARK_DETECTION.ipynb
C: > Users > sanju > OneDrive > Documents > LD > LANDMARK_DETECTION.ipynb > ...
+ Code + Markdown Run All Clear All Outputs Outline
Python 3.12.3
Python
0 18 0

[2]

```python
# Importing necessary libraries for data manipulation, numerical operations, deep learning, image processing, and plotting
import pandas as pd
import numpy as np
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications.vgg19 import VGG19
from tensorflow.keras.layers import *
from tensorflow.keras.models import Sequential
from tensorflow.keras.optimizers import RMSprop
from sklearn.preprocessing import LabelEncoder
import cv2
import os
import random
from matplotlib import pyplot as plt
from PIL import Image
# Enable eager execution for TensorFlow
tf.compat.v1.enable_eager_execution()
```

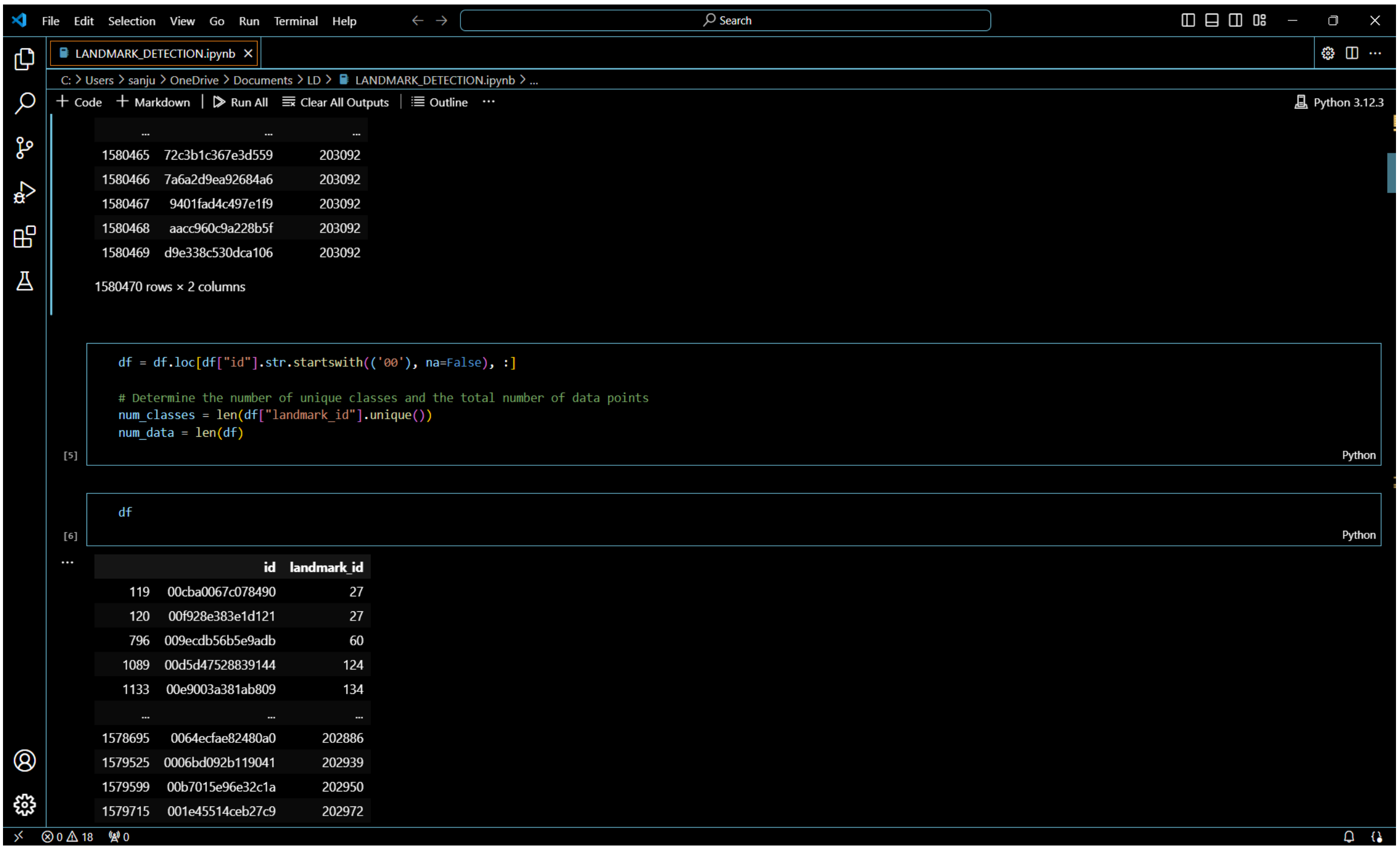
[3]

```python
# Load and display the dataset
df = pd.read_csv("train.csv")
```

[4]

```python
df
```

	id	landmark_id
0	17660ef415d37059	1
1	92b6290d571448f6	1
2	cd41bf948edc0340	1
3	fb09f1e98c6d2f70	1
4	25c9dfc7ea69838d	7


File Edit Selection View Go Run Terminal Help
Search
LANDMARK_DETECTION.ipynb
C: > Users > sanju > OneDrive > Documents > LD > LANDMARK_DETECTION.ipynb > ...
+ Code + Markdown Run All Clear All Outputs Outline
Python 3.12.3
Python
0 18 0

...	...	...
1580465	72c3b1c367e3d559	203092
1580466	7a6a2d9ea92684a6	203092
1580467	9401fad4c497e1f9	203092
1580468	aacc960c9a228b5f	203092
1580469	d9e338c530dca106	203092

1580470 rows × 2 columns

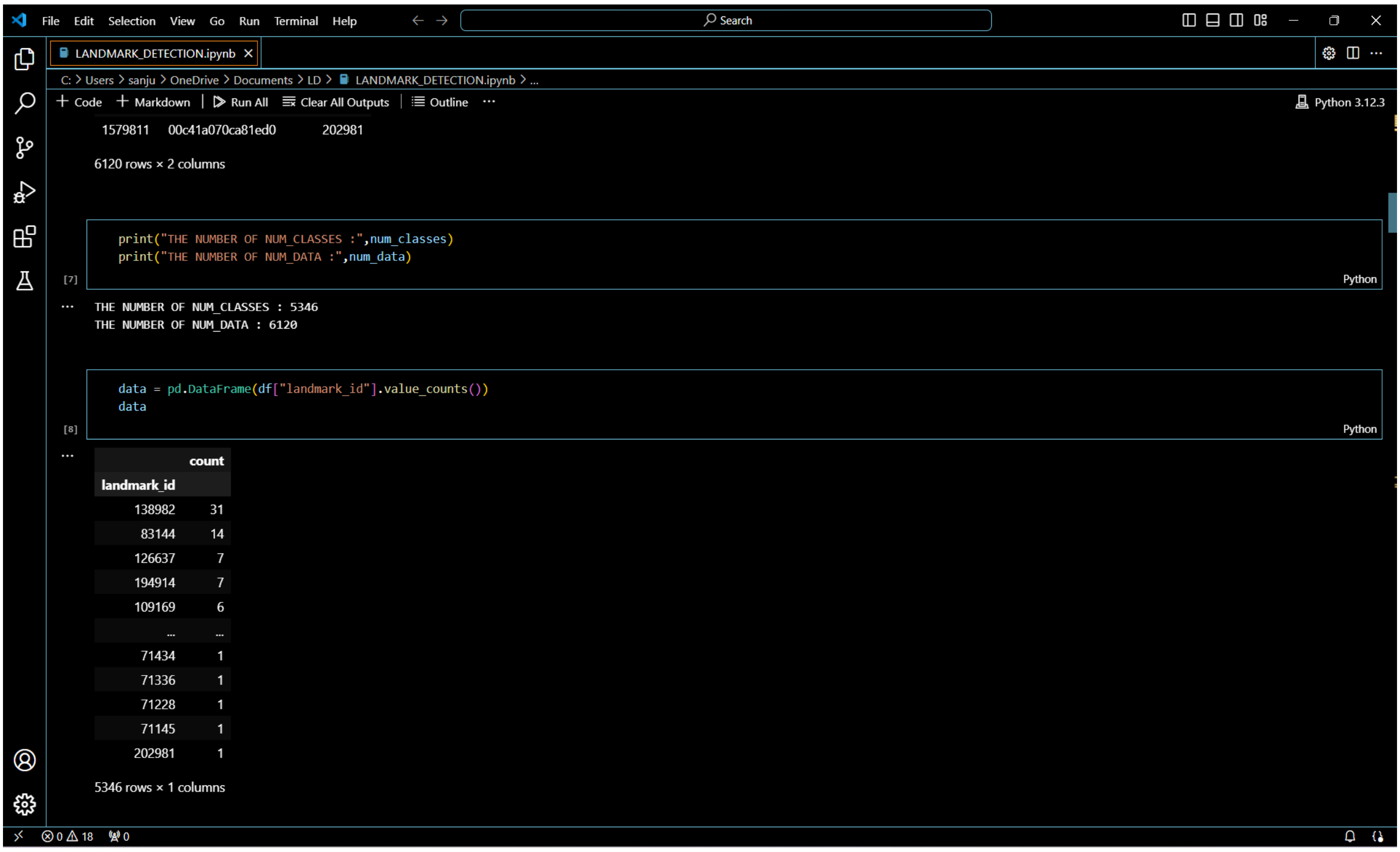
[5]

```python
df = df.loc[df["id"].str.startswith(('00'), na=False), :]

# Determine the number of unique classes and the total number of data points
num_classes = len(df["landmark_id"].unique())
num_data = len(df)
```

[6]

```python
df
```

	id	landmark_id
119	00cba0067c078490	27
120	00f928e383e1d121	27
796	009ecdb56b5e9adb	60
1089	00d5d47528839144	124
1133	00e9003a381ab809	134
...	...	...
1578695	0064ecfae82480a0	202886
1579525	0006bd092b119041	202939
1579599	00b7015e96e32c1a	202950
1579715	001e45514ceb27c9	202972


File Edit Selection View Go Run Terminal Help
Search
LANDMARK_DETECTION.ipynb
C: > Users > sanju > OneDrive > Documents > LD > LANDMARK_DETECTION.ipynb > ...
+ Code + Markdown | Run All Clear All Outputs | Outline ...
Python 3.12.3
Python
Python
0 18 0

1579811	00c41a070ca81ed0	202981

6120 rows × 2 columns

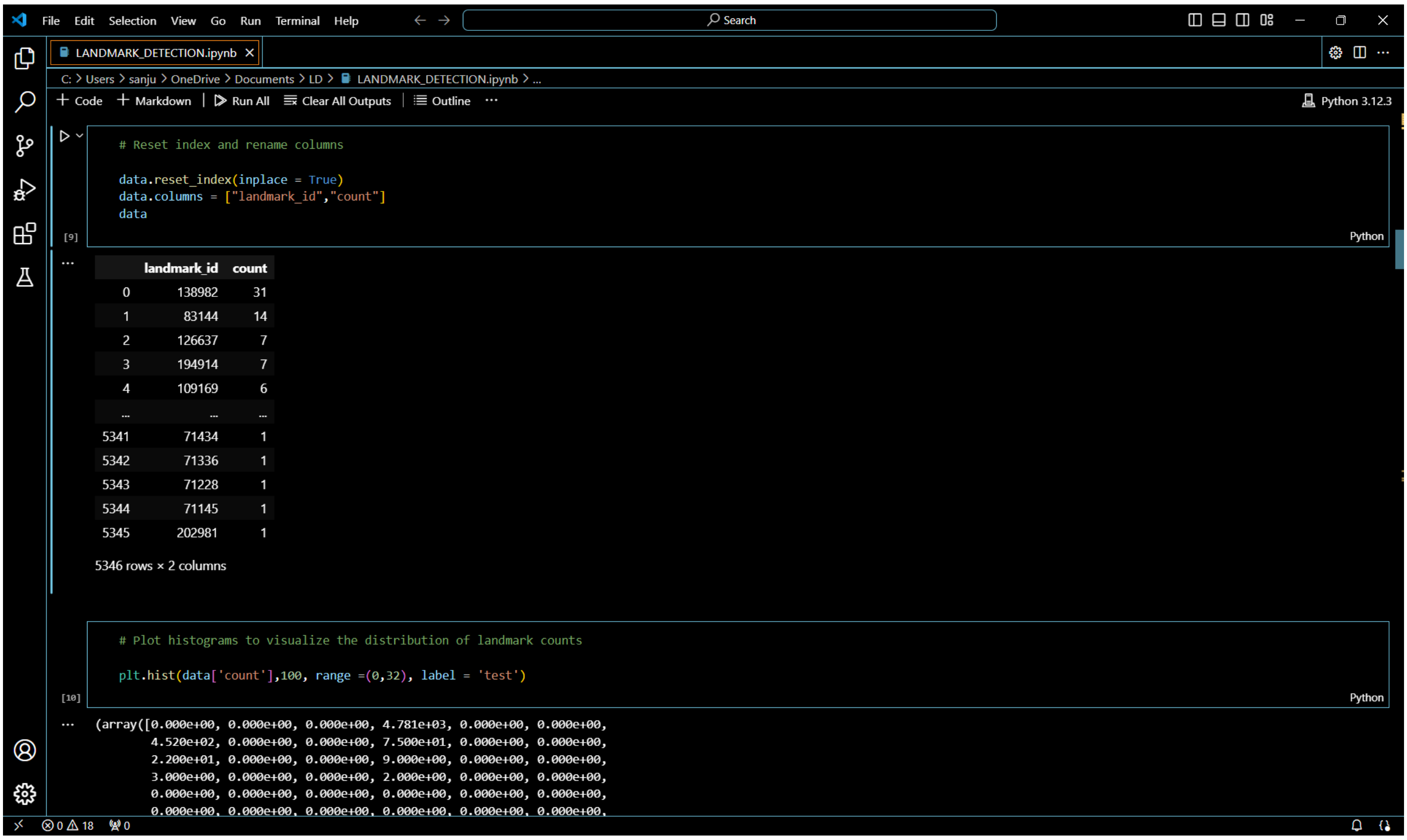
```
[7]
print("THE NUMBER OF NUM_CLASSES :",num_classes)
print("THE NUMBER OF NUM_DATA :",num_data)
```

```
THE NUMBER OF NUM_CLASSES : 5346
THE NUMBER OF NUM_DATA : 6120
```

```
[8]
data = pd.DataFrame(df["landmark_id"].value_counts())
data
```

	count
landmark_id	
138982	31
83144	14
126637	7
194914	7
109169	6
...	...
71434	1
71336	1
71228	1
71145	1
202981	1

5346 rows × 1 columns


File Edit Selection View Go Run Terminal Help
Search
LANDMARK_DETECTION.ipynb
C: > Users > sanju > OneDrive > Documents > LD > LANDMARK_DETECTION.ipynb > ...
+ Code + Markdown | Run All Clear All Outputs | Outline ...
Python 3.12.3
Python
Python
0 18 0

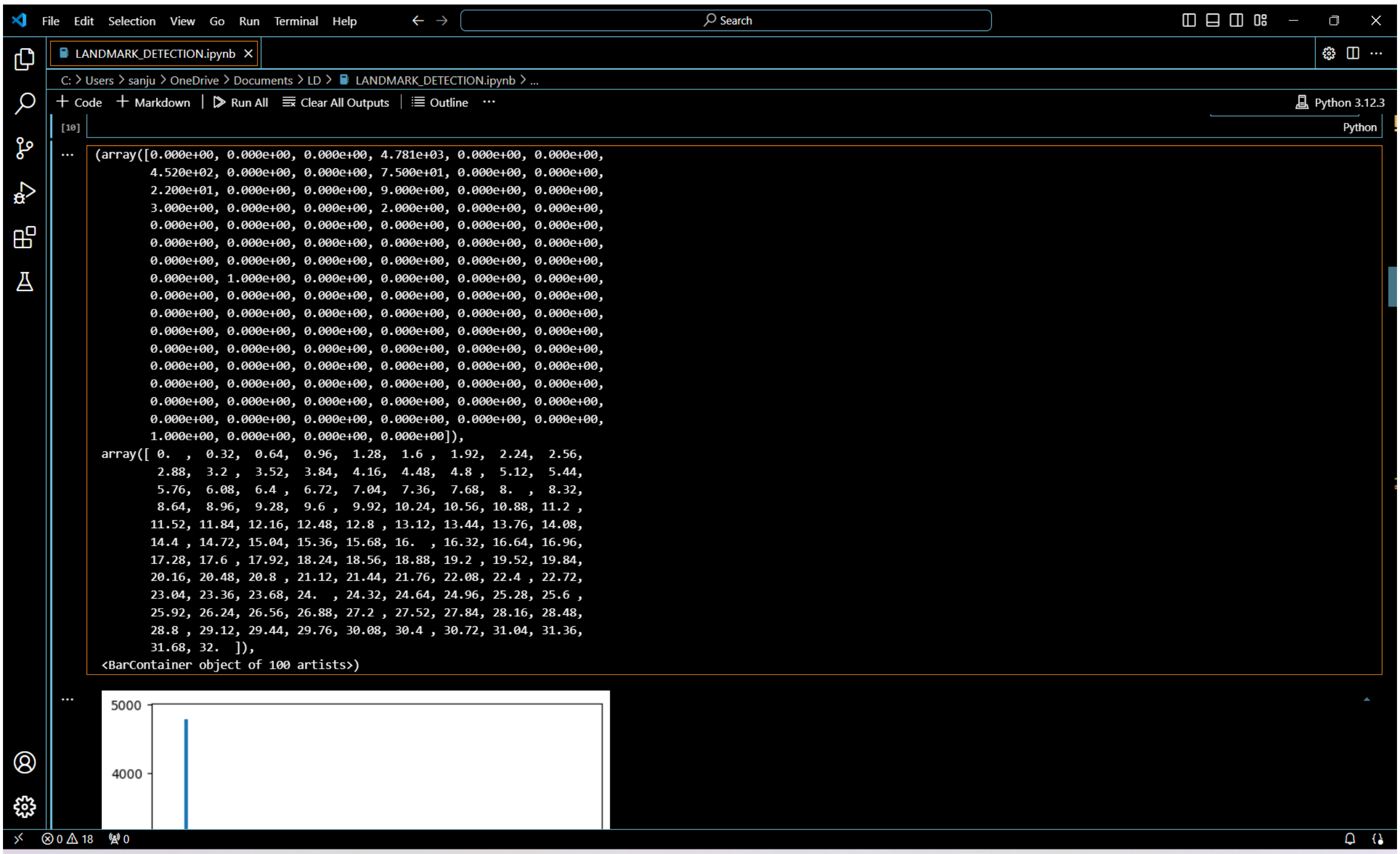
```
[9]
# Reset index and rename columns

data.reset_index(inplace = True)
data.columns = ["landmark_id","count"]
data
```

	landmark_id	count
0	138982	31
1	83144	14
2	126637	7
3	194914	7
4	109169	6
...	...	...
5341	71434	1
5342	71336	1
5343	71228	1
5344	71145	1
5345	202981	1

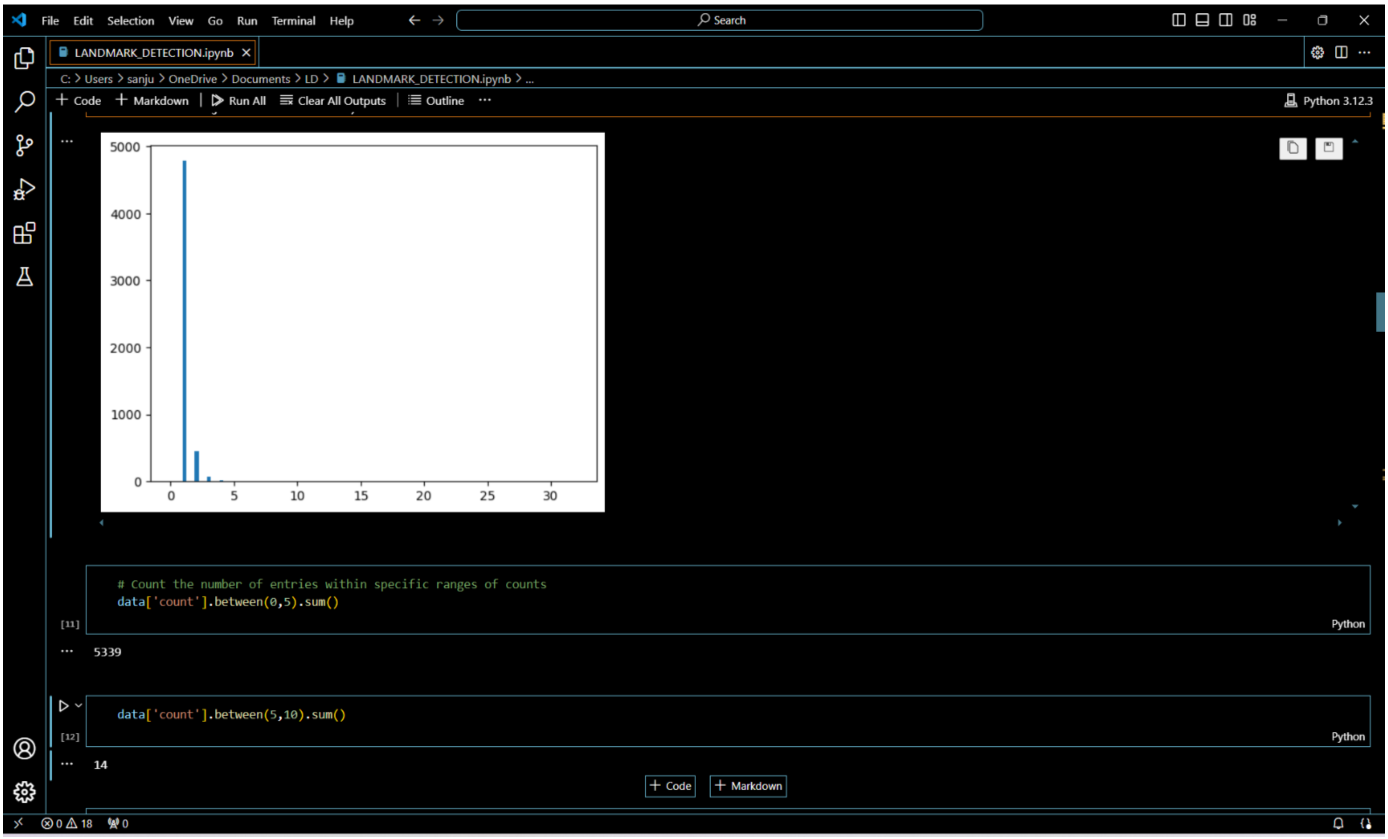
5346 rows × 2 columns

```
[10]
# Plot histograms to visualize the distribution of landmark counts

plt.hist(data['count'],100, range =(0,32), label = 'test')
```

```
(array([0.000e+00, 0.000e+00, 0.000e+00, 4.781e+03, 0.000e+00, 0.000e+00,
        4.520e+02, 0.000e+00, 0.000e+00, 7.500e+01, 0.000e+00, 0.000e+00,
        2.200e+01, 0.000e+00, 0.000e+00, 9.000e+00, 0.000e+00, 0.000e+00,
        3.000e+00, 0.000e+00, 0.000e+00, 2.000e+00, 0.000e+00, 0.000e+00,
        0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00,
        0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00,
```


File Edit Selection View Go Run Terminal Help
Search
LANDMARK_DETECTION.ipynb
C: > Users > sanju > OneDrive > Documents > LD > LANDMARK_DETECTION.ipynb > ...
+ Code + Markdown | Run All Clear All Outputs | Outline ...
Python 3.12.3
[10]
Python
5000
4000
⊗ 0 ⚠ 18 0

```
(array([0.000e+00, 0.000e+00, 0.000e+00, 4.781e+03, 0.000e+00, 0.000e+00,
        4.520e+02, 0.000e+00, 0.000e+00, 7.500e+01, 0.000e+00, 0.000e+00,
        2.200e+01, 0.000e+00, 0.000e+00, 9.000e+00, 0.000e+00, 0.000e+00,
        3.000e+00, 0.000e+00, 0.000e+00, 2.000e+00, 0.000e+00, 0.000e+00,
        0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00,
        0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00,
        0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00,
        0.000e+00, 1.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00,
        0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00,
        0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00,
        0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00,
        0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00,
        0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00,
        0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00,
        0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00,
        0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00,
        1.000e+00, 0.000e+00, 0.000e+00, 0.000e+00]),
 array([ 0.  ,  0.32,  0.64,  0.96,  1.28,  1.6 ,  1.92,  2.24,  2.56,
         2.88,  3.2 ,  3.52,  3.84,  4.16,  4.48,  4.8 ,  5.12,  5.44,
         5.76,  6.08,  6.4 ,  6.72,  7.04,  7.36,  7.68,  8.  ,  8.32,
         8.64,  8.96,  9.28,  9.6 ,  9.92, 10.24, 10.56, 10.88, 11.2 ,
        11.52, 11.84, 12.16, 12.48, 12.8 , 13.12, 13.44, 13.76, 14.08,
        14.4 , 14.72, 15.04, 15.36, 15.68, 16.  , 16.32, 16.64, 16.96,
        17.28, 17.6 , 17.92, 18.24, 18.56, 18.88, 19.2 , 19.52, 19.84,
        20.16, 20.48, 20.8 , 21.12, 21.44, 21.76, 22.08, 22.4 , 22.72,
        23.04, 23.36, 23.68, 24.  , 24.32, 24.64, 24.96, 25.28, 25.6 ,
        25.92, 26.24, 26.56, 26.88, 27.2 , 27.52, 27.84, 28.16, 28.48,
        28.8 , 29.12, 29.44, 29.76, 30.08, 30.4 , 30.72, 31.04, 31.36,
        31.68, 32.  ]),
 <BarContainer object of 100 artists>)
```

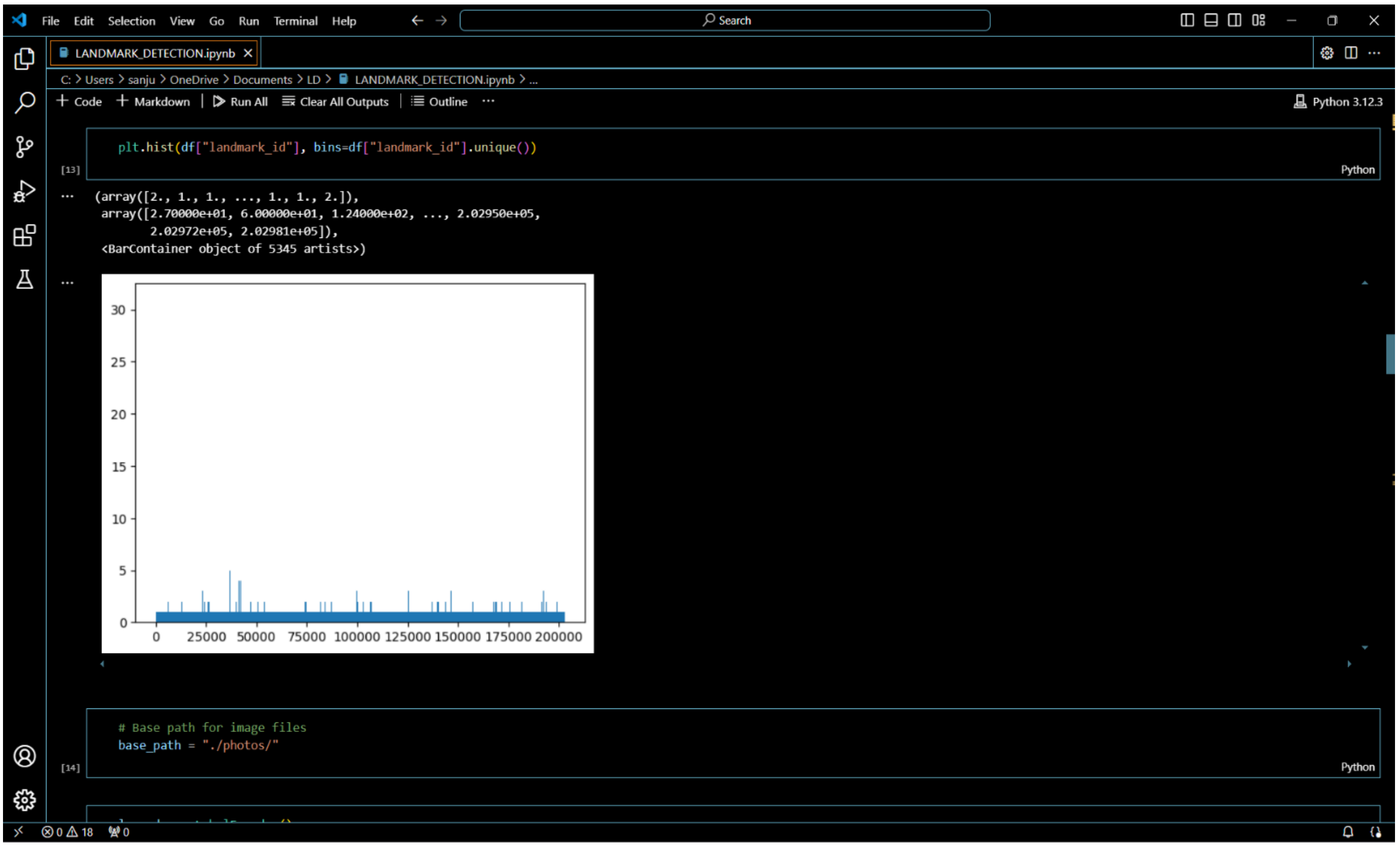

File Edit Selection View Go Run Terminal Help
Search
LANDMARK_DETECTION.ipynb
C: > Users > sanju > OneDrive > Documents > LD > LANDMARK_DETECTION.ipynb > ...
+ Code + Markdown | Run All Clear All Outputs | Outline ...
Python 3.12.3
5000
4000
3000
2000
1000
0
0
5
10
15
20
25
30
[11]
Python
[12]
Python
+ Code
+ Markdown
⊗ 0 ⚠ 18 0

```
# Count the number of entries within specific ranges of counts
data['count'].between(0,5).sum()
```

```
5339
```

```
data['count'].between(5,10).sum()
```

```
14
```

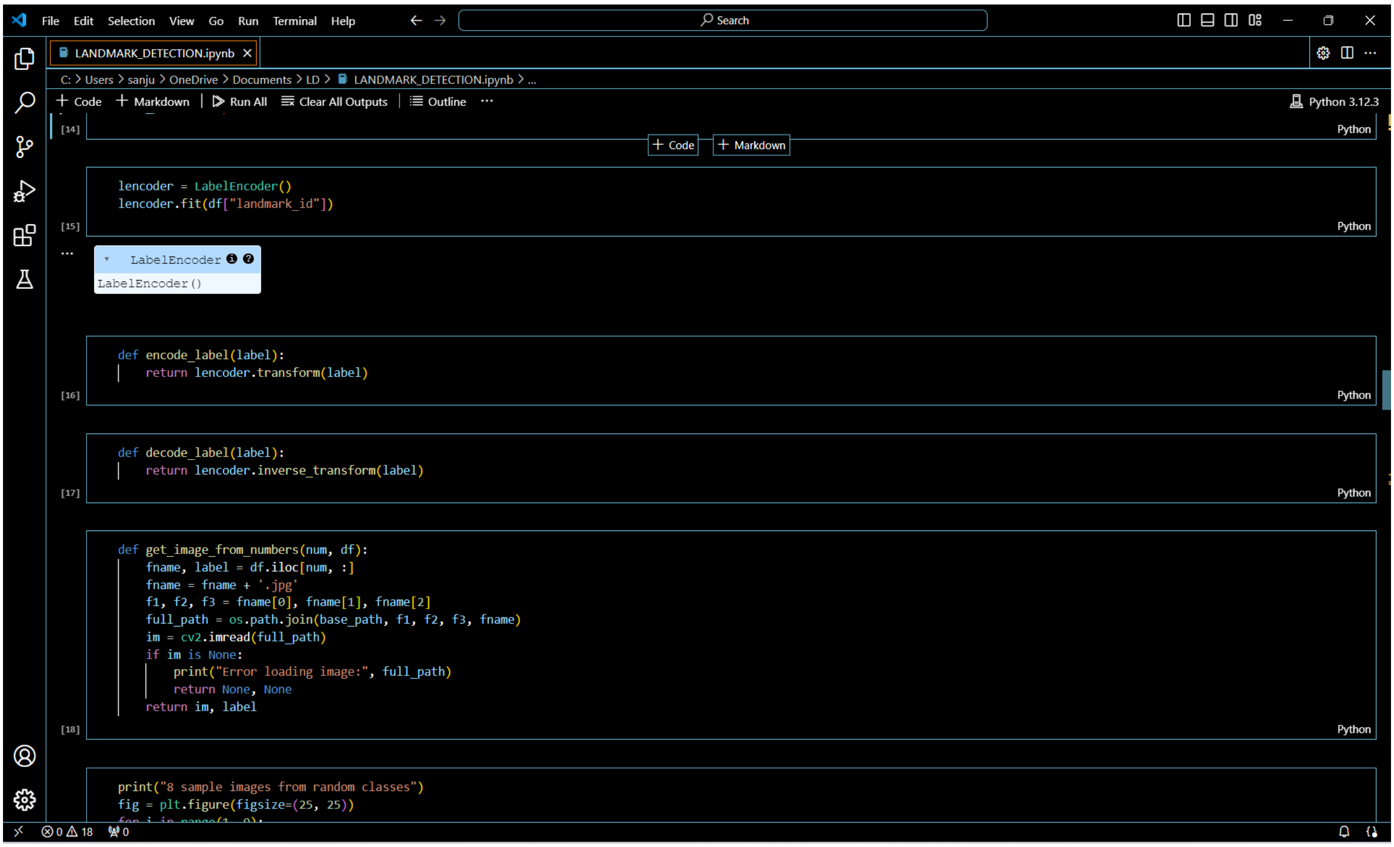


File Edit Selection View Go Run Terminal Help
Search
LANDMARK_DETECTION.ipynb
C: > Users > sanju > OneDrive > Documents > LD > LANDMARK_DETECTION.ipynb > ...
+ Code + Markdown | Run All Clear All Outputs | Outline ...
Python 3.12.3
[13]
Python
30
25
20
15
10
5
0
0 25000 50000 75000 100000 125000 150000 175000 200000
[14]
Python
0 18 0

```
plt.hist(df["landmark_id"], bins=df["landmark_id"].unique())
```

```
(array([2., 1., 1., ..., 1., 1., 2.]),
 array([2.70000e+01, 6.00000e+01, 1.24000e+02, ..., 2.02950e+05,
        2.02972e+05, 2.02981e+05]),
 <BarContainer object of 5345 artists>)
```

```
# Base path for image files
base_path = "./photos/"
```



File Edit Selection View Go Run Terminal Help
Search
LANDMARK_DETECTION.ipynb
C: > Users > sanju > OneDrive > Documents > LD > LANDMARK_DETECTION.ipynb > ...
+ Code + Markdown | Run All Clear All Outputs | Outline ...
Python 3.12.3
[14]
Python
+ Code + Markdown
[15]
Python
[16]
Python
[17]
Python
[18]
Python
0 18 0

```
lencoder = LabelEncoder()
lencoder.fit(df["landmark_id"])
```

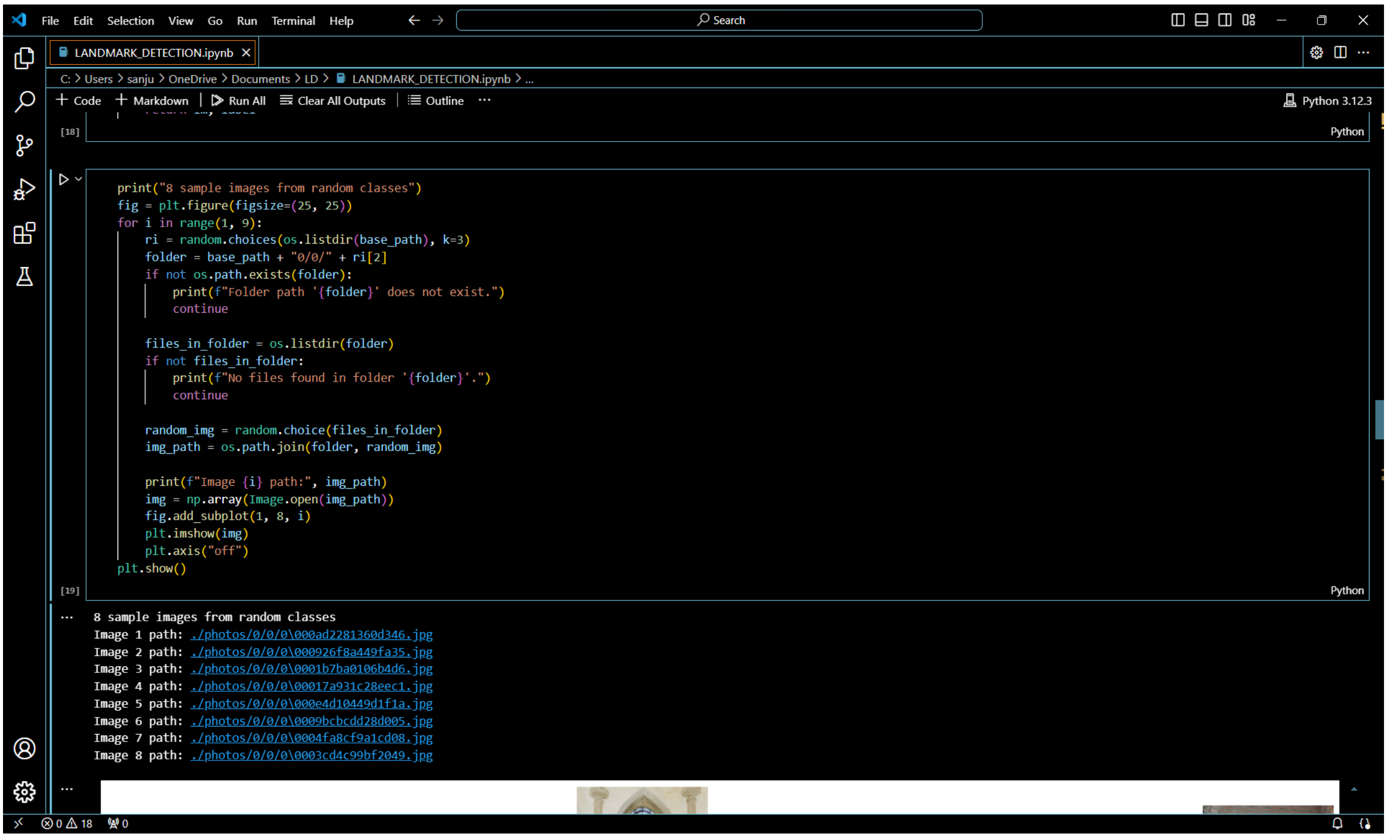
```
LabelEncoder
LabelEncoder()
```

```
def encode_label(label):
    return lencoder.transform(label)
```

```
def decode_label(label):
    return lencoder.inverse_transform(label)
```

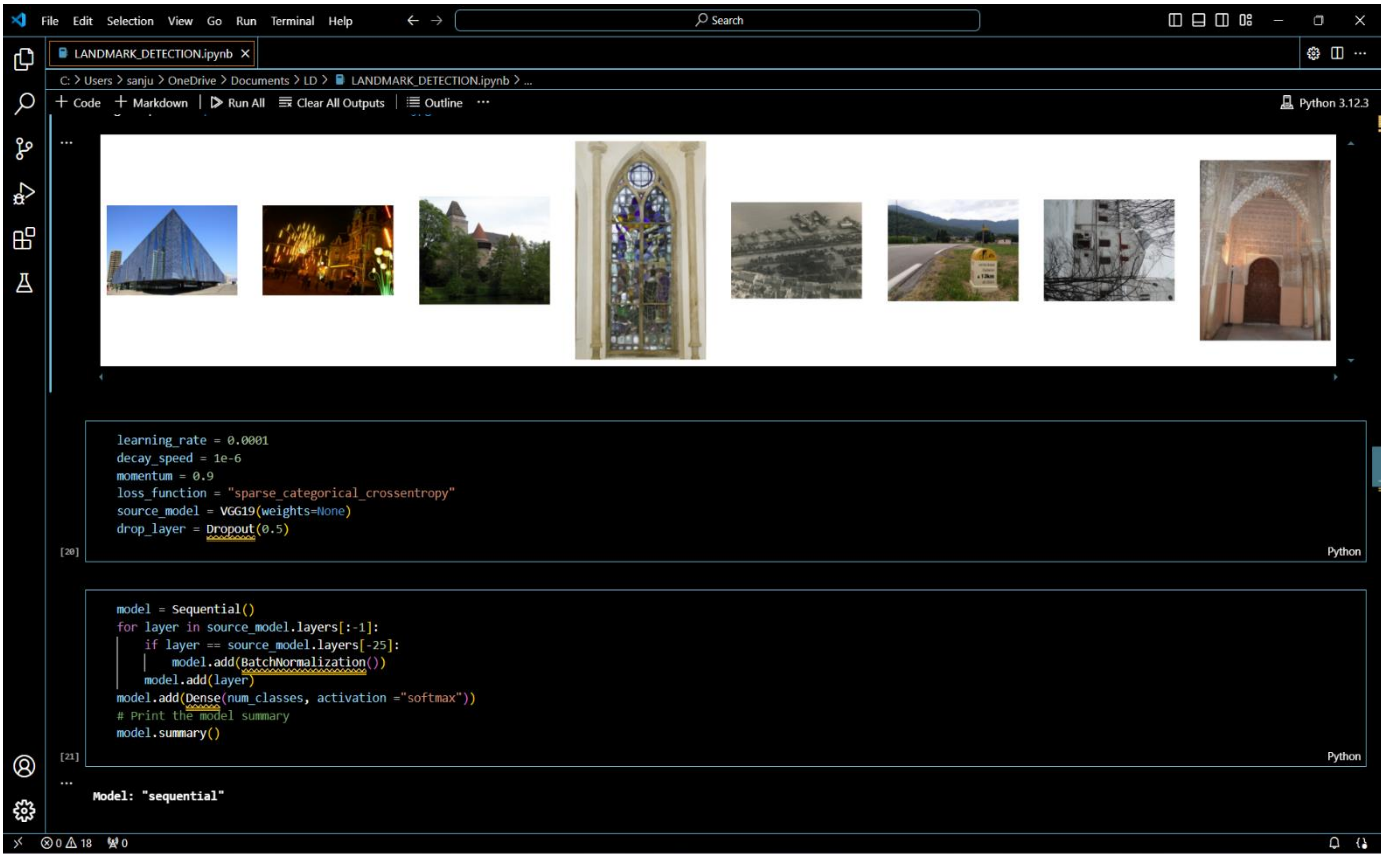
```
def get_image_from_numbers(num, df):
    fname, label = df.iloc[num, :]
    fname = fname + '.jpg'
    f1, f2, f3 = fname[0], fname[1], fname[2]
    full_path = os.path.join(base_path, f1, f2, f3, fname)
    im = cv2.imread(full_path)
    if im is None:
        print("Error loading image:", full_path)
        return None, None
    return im, label
```

```
print("8 sample images from random classes")
fig = plt.figure(figsize=(25, 25))
```


File Edit Selection View Go Run Terminal Help
Search
LANDMARK_DETECTION.ipynb
C: > Users > sanju > OneDrive > Documents > LD > LANDMARK_DETECTION.ipynb > ...
+ Code + Markdown | Run All Clear All Outputs | Outline ...
Python 3.12.3
[18]
Python
[19]
Python
0 18 0

```python
print("8 sample images from random classes")
fig = plt.figure(figsize=(25, 25))
for i in range(1, 9):
    ri = random.choices(os.listdir(base_path), k=3)
    folder = base_path + "0/0/" + ri[2]
    if not os.path.exists(folder):
        print(f"Folder path '{folder}' does not exist.")
        continue

    files_in_folder = os.listdir(folder)
    if not files_in_folder:
        print(f"No files found in folder '{folder}'.")
        continue

    random_img = random.choice(files_in_folder)
    img_path = os.path.join(folder, random_img)

    print(f"Image {i} path:", img_path)
    img = np.array(Image.open(img_path))
    fig.add_subplot(1, 8, i)
    plt.imshow(img)
    plt.axis("off")
plt.show()
```

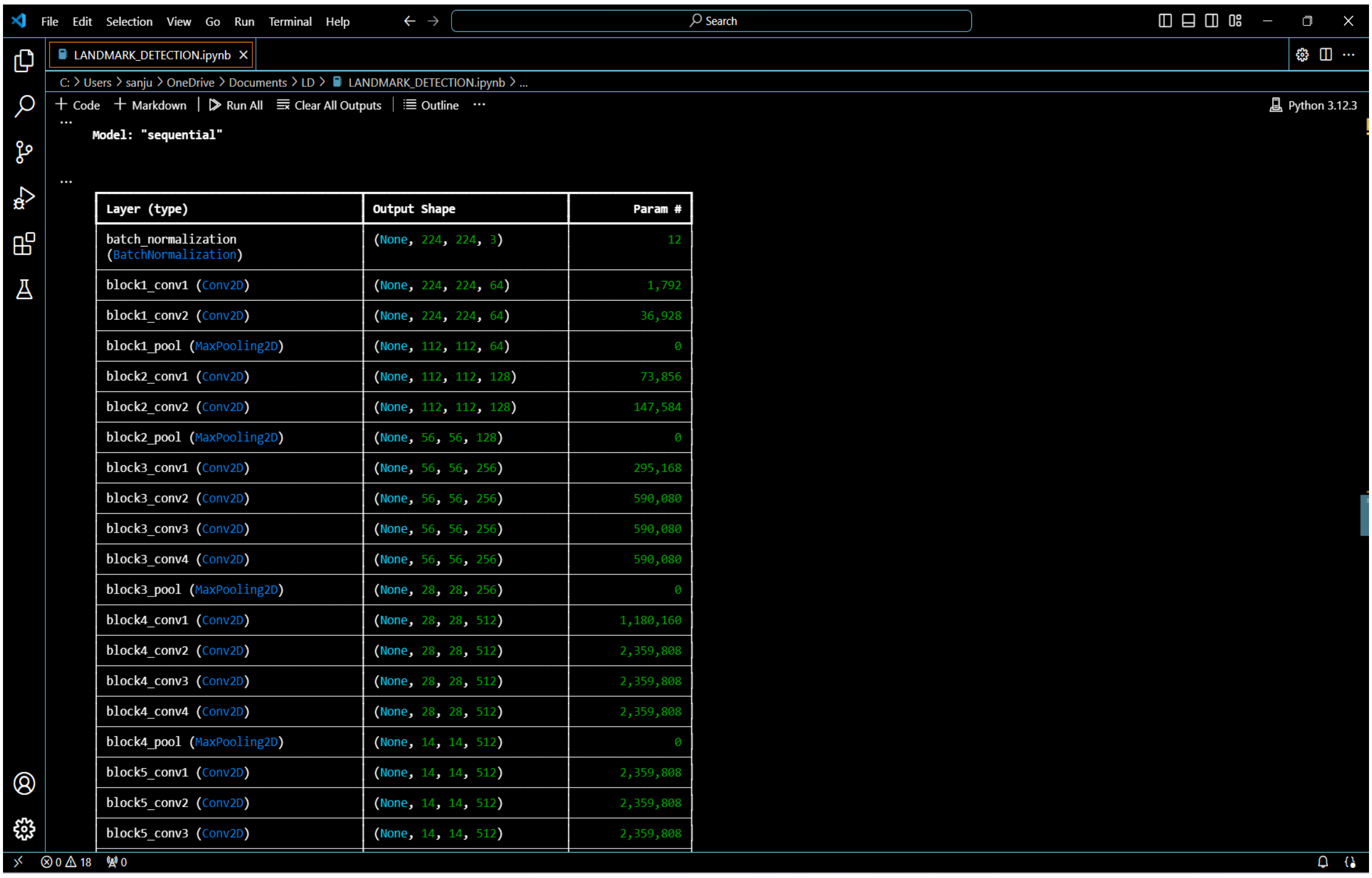
```
8 sample images from random classes
Image 1 path: ./photos/0/0/0\000ad2281360d346.jpg
Image 2 path: ./photos/0/0/0\000926f8a449fa35.jpg
Image 3 path: ./photos/0/0/0\0001b7ba0106b4d6.jpg
Image 4 path: ./photos/0/0/0\00017a931c28eec1.jpg
Image 5 path: ./photos/0/0/0\000e4d10449d1f1a.jpg
Image 6 path: ./photos/0/0/0\0009bcbcdd28d005.jpg
Image 7 path: ./photos/0/0/0\0004fa8cf9a1cd08.jpg
Image 8 path: ./photos/0/0/0\0003cd4c99bf2049.jpg
```


File Edit Selection View Go Run Terminal Help
Search
LANDMARK_DETECTION.ipynb
C: > Users > sanju > OneDrive > Documents > LD > LANDMARK_DETECTION.ipynb > ...
+ Code + Markdown | Run All Clear All Outputs | Outline ...
Python 3.12.3
[20]
Python
[21]
Python
0 18 0

```python
learning_rate = 0.0001
decay_speed = 1e-6
momentum = 0.9
loss_function = "sparse_categorical_crossentropy"
source_model = VGG19(weights=None)
drop_layer = Dropout(0.5)
```

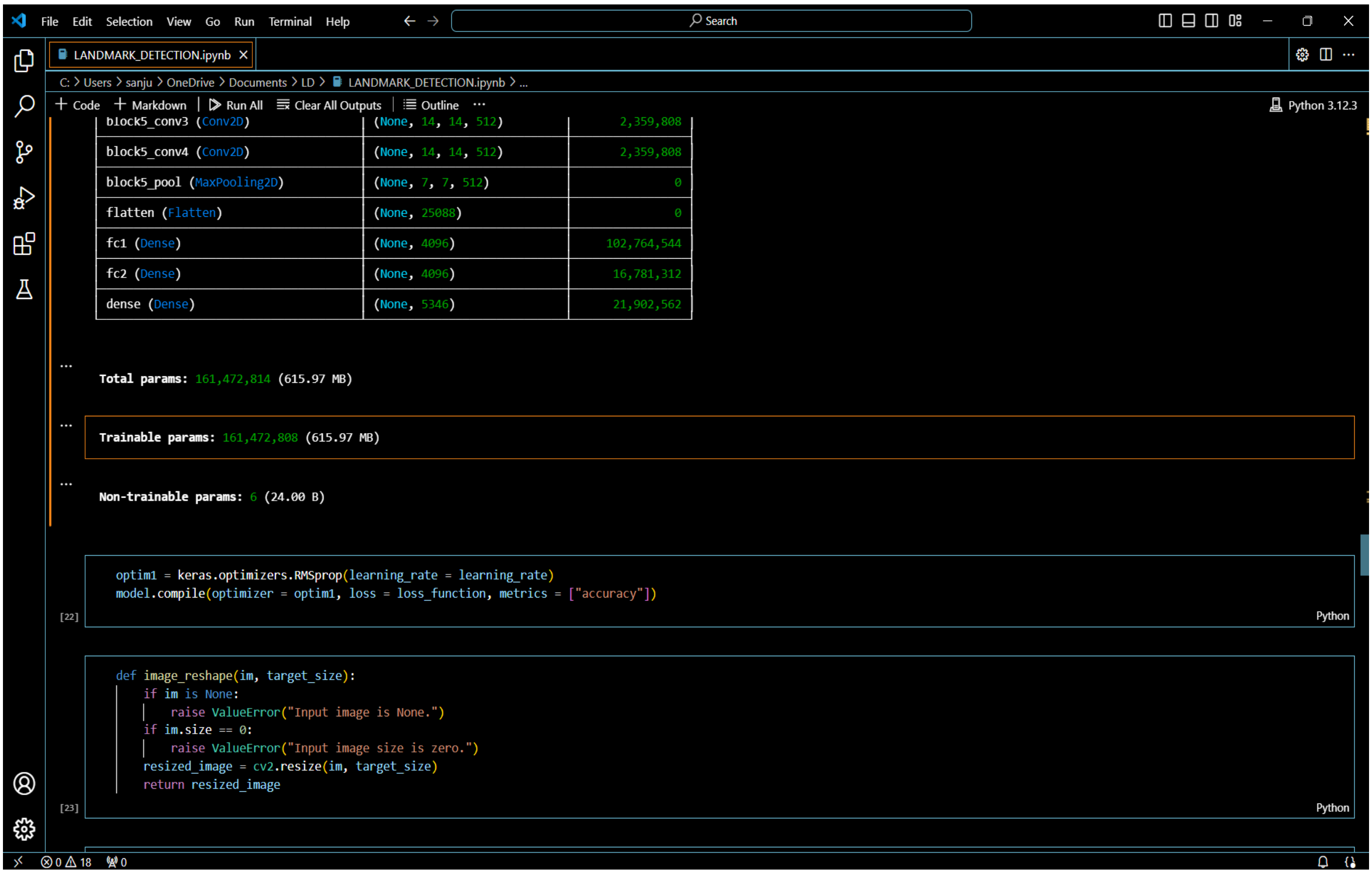
```python
model = Sequential()
for layer in source_model.layers[:-1]:
    if layer == source_model.layers[-25]:
        model.add(BatchNormalization())
    model.add(layer)
model.add(Dense(num_classes, activation ="softmax"))
# Print the model summary
model.summary()
```

```
Model: "sequential"
```



File Edit Selection View Go Run Terminal Help
Search
LANDMARK_DETECTION.ipynb
C: > Users > sanju > OneDrive > Documents > LD > LANDMARK_DETECTION.ipynb > ...
+ Code + Markdown | Run All Clear All Outputs | Outline ...
Python 3.12.3
0 18 0

```
Model: "sequential"

┌─────────────────────────────────┬────────────────────────┬───────────────┐
│ Layer (type)                    │ Output Shape           │       Param # │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ batch_normalization             │ (None, 224, 224, 3)    │            12 │
│ (BatchNormalization)            │                        │               │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ block1_conv1 (Conv2D)           │ (None, 224, 224, 64)   │         1,792 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ block1_conv2 (Conv2D)           │ (None, 224, 224, 64)   │        36,928 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ block1_pool (MaxPooling2D)      │ (None, 112, 112, 64)   │             0 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ block2_conv1 (Conv2D)           │ (None, 112, 112, 128)  │        73,856 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ block2_conv2 (Conv2D)           │ (None, 112, 112, 128)  │       147,584 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ block2_pool (MaxPooling2D)      │ (None, 56, 56, 128)    │             0 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ block3_conv1 (Conv2D)           │ (None, 56, 56, 256)    │       295,168 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ block3_conv2 (Conv2D)           │ (None, 56, 56, 256)    │       590,080 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ block3_conv3 (Conv2D)           │ (None, 56, 56, 256)    │       590,080 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ block3_conv4 (Conv2D)           │ (None, 56, 56, 256)    │       590,080 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ block3_pool (MaxPooling2D)      │ (None, 28, 28, 256)    │             0 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ block4_conv1 (Conv2D)           │ (None, 28, 28, 512)    │     1,180,160 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ block4_conv2 (Conv2D)           │ (None, 28, 28, 512)    │     2,359,808 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ block4_conv3 (Conv2D)           │ (None, 28, 28, 512)    │     2,359,808 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ block4_conv4 (Conv2D)           │ (None, 28, 28, 512)    │     2,359,808 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ block4_pool (MaxPooling2D)      │ (None, 14, 14, 512)    │             0 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ block5_conv1 (Conv2D)           │ (None, 14, 14, 512)    │     2,359,808 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ block5_conv2 (Conv2D)           │ (None, 14, 14, 512)    │     2,359,808 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ block5_conv3 (Conv2D)           │ (None, 14, 14, 512)    │     2,359,808 │
```



File Edit Selection View Go Run Terminal Help
Search
LANDMARK_DETECTION.ipynb
C: > Users > sanju > OneDrive > Documents > LD > LANDMARK_DETECTION.ipynb > ...
+ Code + Markdown | Run All Clear All Outputs | Outline ...
Python 3.12.3
0 18 0

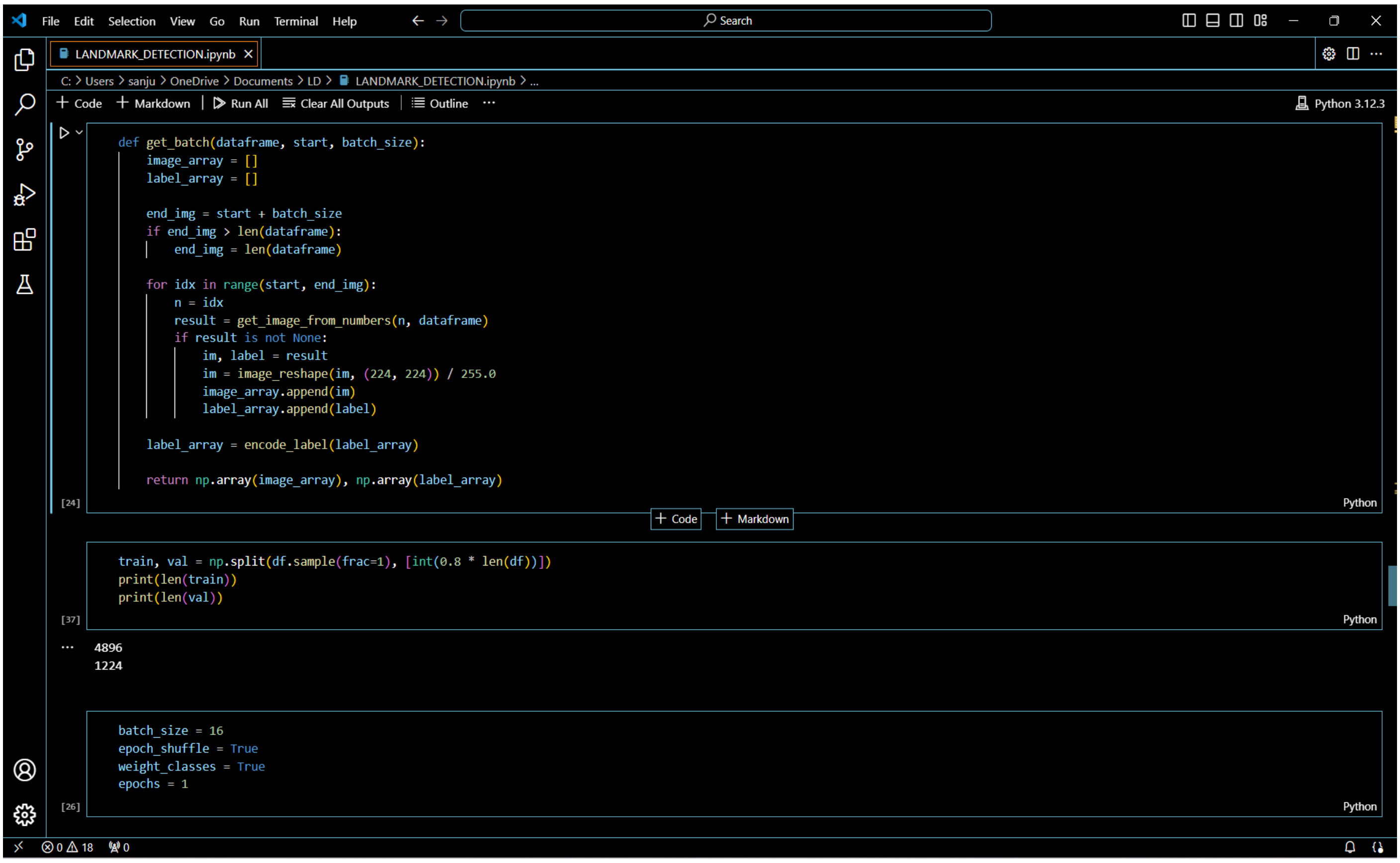
```
│ block5_conv3 (Conv2D)           │ (None, 14, 14, 512)    │     2,359,808 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ block5_conv4 (Conv2D)           │ (None, 14, 14, 512)    │     2,359,808 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ block5_pool (MaxPooling2D)      │ (None, 7, 7, 512)      │             0 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ flatten (Flatten)               │ (None, 25088)          │             0 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ fc1 (Dense)                     │ (None, 4096)           │   102,764,544 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ fc2 (Dense)                     │ (None, 4096)           │    16,781,312 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ dense (Dense)                   │ (None, 5346)           │    21,902,562 │
└─────────────────────────────────┴────────────────────────┴───────────────┘

 Total params: 161,472,814 (615.97 MB)

 Trainable params: 161,472,808 (615.97 MB)

 Non-trainable params: 6 (24.00 B)
```

```python
optim1 = keras.optimizers.RMSprop(learning_rate = learning_rate)
model.compile(optimizer = optim1, loss = loss_function, metrics = ["accuracy"])
```

[22] Python

```python
def image_reshape(im, target_size):
    if im is None:
        raise ValueError("Input image is None.")
    if im.size == 0:
        raise ValueError("Input image size is zero.")
    resized_image = cv2.resize(im, target_size)
    return resized_image
```

[23] Python


File Edit Selection View Go Run Terminal Help
Search
LANDMARK_DETECTION.ipynb
C: > Users > sanju > OneDrive > Documents > LD > LANDMARK_DETECTION.ipynb > ...
+ Code + Markdown | Run All Clear All Outputs | Outline ...
Python 3.12.3

```python
def get_batch(dataframe, start, batch_size):
    image_array = []
    label_array = []

    end_img = start + batch_size
    if end_img > len(dataframe):
        end_img = len(dataframe)

    for idx in range(start, end_img):
        n = idx
        result = get_image_from_numbers(n, dataframe)
        if result is not None:
            im, label = result
            im = image_reshape(im, (224, 224)) / 255.0
            image_array.append(im)
            label_array.append(label)

    label_array = encode_label(label_array)

    return np.array(image_array), np.array(label_array)
```

[24]

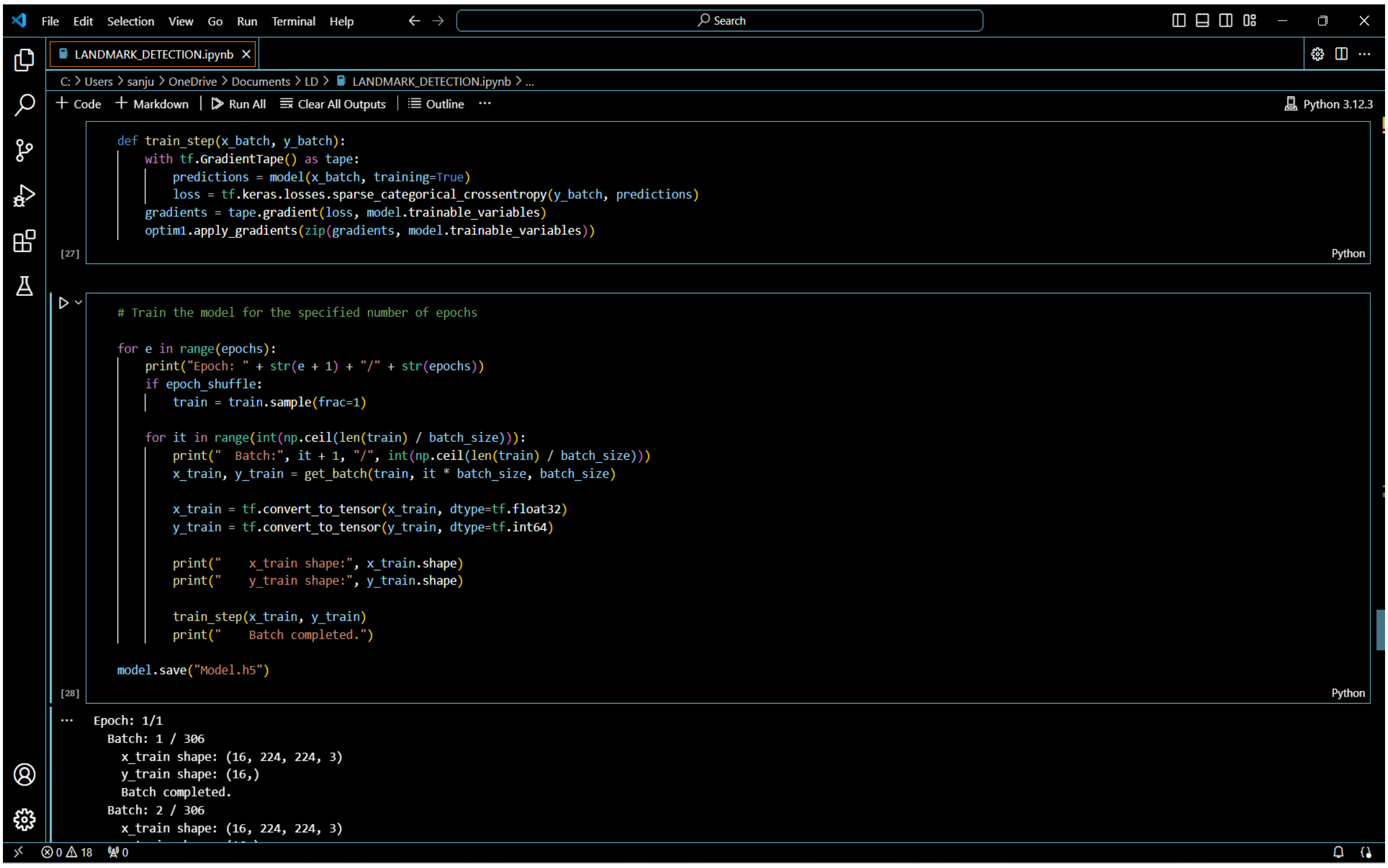
```python
train, val = np.split(df.sample(frac=1), [int(0.8 * len(df))])
print(len(train))
print(len(val))
```

[37]

```
4896
1224
```

```python
batch_size = 16
epoch_shuffle = True
weight_classes = True
epochs = 1
```

[26]


File Edit Selection View Go Run Terminal Help
Search
LANDMARK_DETECTION.ipynb
C: > Users > sanju > OneDrive > Documents > LD > LANDMARK_DETECTION.ipynb > ...
+ Code + Markdown | Run All Clear All Outputs | Outline ...
Python 3.12.3

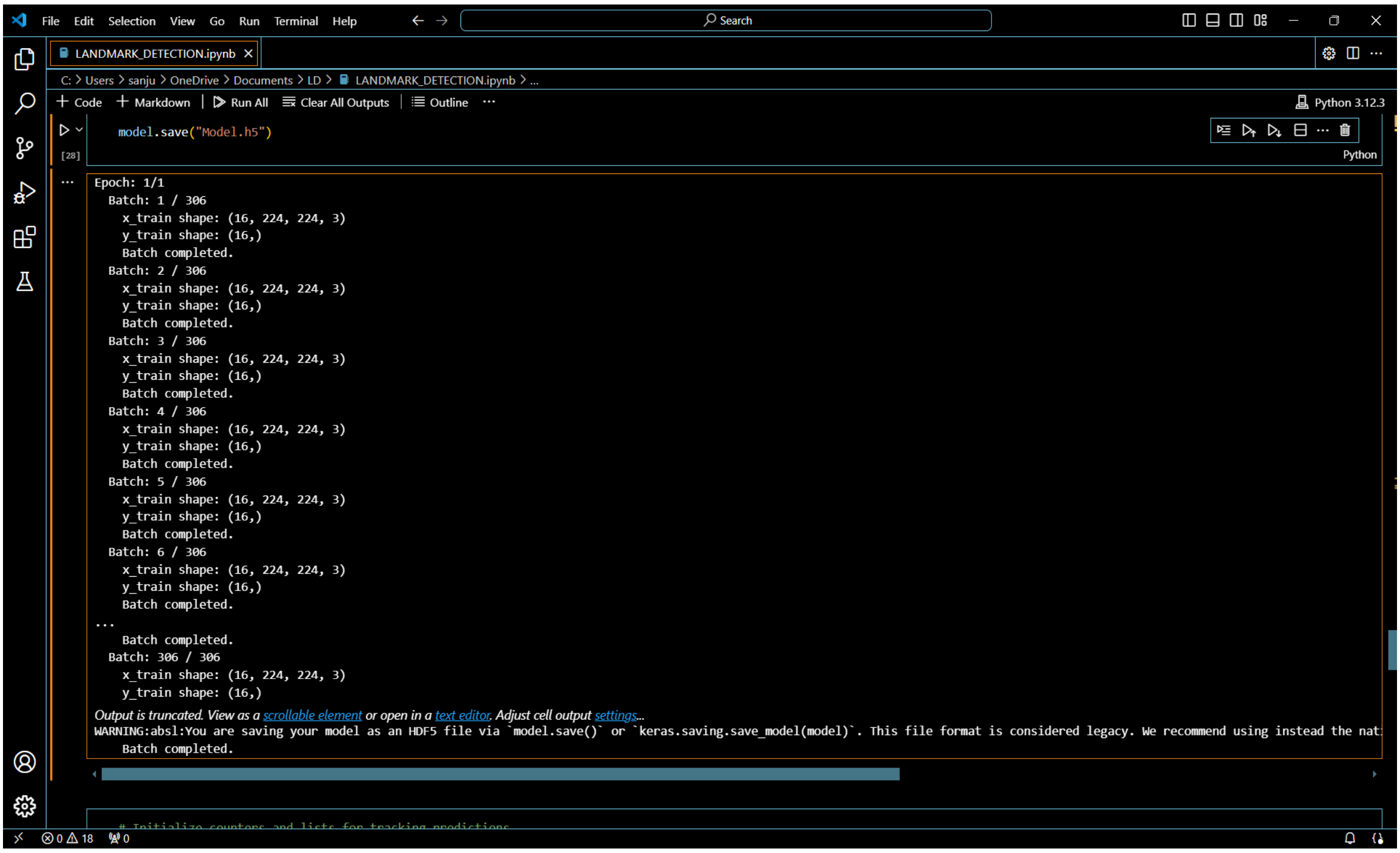
```python
def train_step(x_batch, y_batch):
    with tf.GradientTape() as tape:
        predictions = model(x_batch, training=True)
        loss = tf.keras.losses.sparse_categorical_crossentropy(y_batch, predictions)
    gradients = tape.gradient(loss, model.trainable_variables)
    optim1.apply_gradients(zip(gradients, model.trainable_variables))
```

[27]

```python
# Train the model for the specified number of epochs

for e in range(epochs):
    print("Epoch: " + str(e + 1) + "/" + str(epochs))
    if epoch_shuffle:
        train = train.sample(frac=1)

    for it in range(int(np.ceil(len(train) / batch_size))):
        print("  Batch:", it + 1, "/", int(np.ceil(len(train) / batch_size)))
        x_train, y_train = get_batch(train, it * batch_size, batch_size)

        x_train = tf.convert_to_tensor(x_train, dtype=tf.float32)
        y_train = tf.convert_to_tensor(y_train, dtype=tf.int64)

        print("    x_train shape:", x_train.shape)
        print("    y_train shape:", y_train.shape)

        train_step(x_train, y_train)
        print("    Batch completed.")

model.save("Model.h5")
```

[28]

```
Epoch: 1/1
  Batch: 1 / 306
    x_train shape: (16, 224, 224, 3)
    y_train shape: (16,)
    Batch completed.
  Batch: 2 / 306
    x_train shape: (16, 224, 224, 3)
```


File Edit Selection View Go Run Terminal Help
Search
LANDMARK_DETECTION.ipynb
C: > Users > sanju > OneDrive > Documents > LD > LANDMARK_DETECTION.ipynb > ...
Code
Markdown
Run All
Clear All Outputs
Outline
Python 3.12.3
[28]
Python
0 18
0

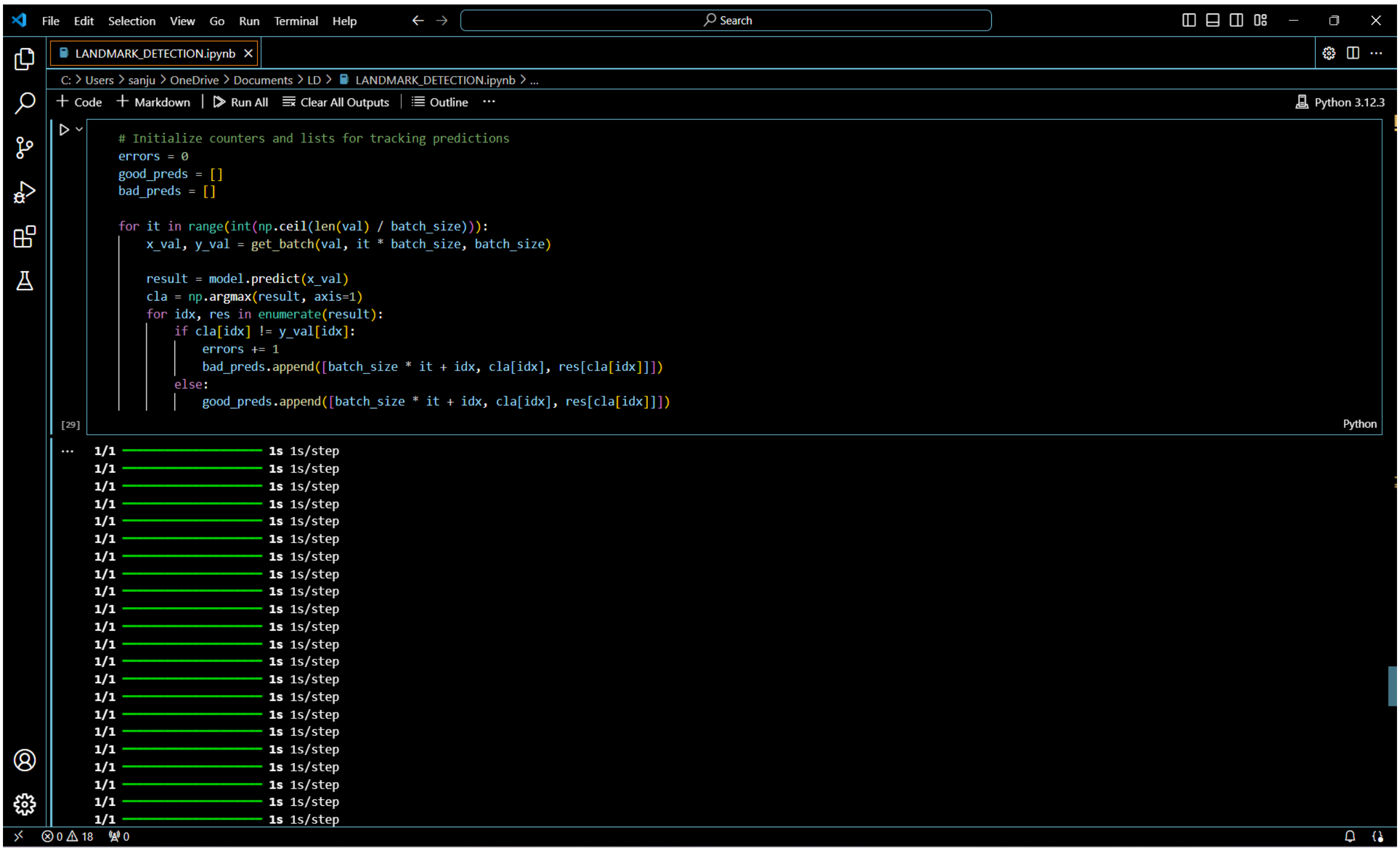
```python
model.save("Model.h5")
```

```
Epoch: 1/1
  Batch: 1 / 306
    x_train shape: (16, 224, 224, 3)
    y_train shape: (16,)
    Batch completed.
  Batch: 2 / 306
    x_train shape: (16, 224, 224, 3)
    y_train shape: (16,)
    Batch completed.
  Batch: 3 / 306
    x_train shape: (16, 224, 224, 3)
    y_train shape: (16,)
    Batch completed.
  Batch: 4 / 306
    x_train shape: (16, 224, 224, 3)
    y_train shape: (16,)
    Batch completed.
  Batch: 5 / 306
    x_train shape: (16, 224, 224, 3)
    y_train shape: (16,)
    Batch completed.
  Batch: 6 / 306
    x_train shape: (16, 224, 224, 3)
    y_train shape: (16,)
    Batch completed.
...
    Batch completed.
  Batch: 306 / 306
    x_train shape: (16, 224, 224, 3)
    y_train shape: (16,)
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

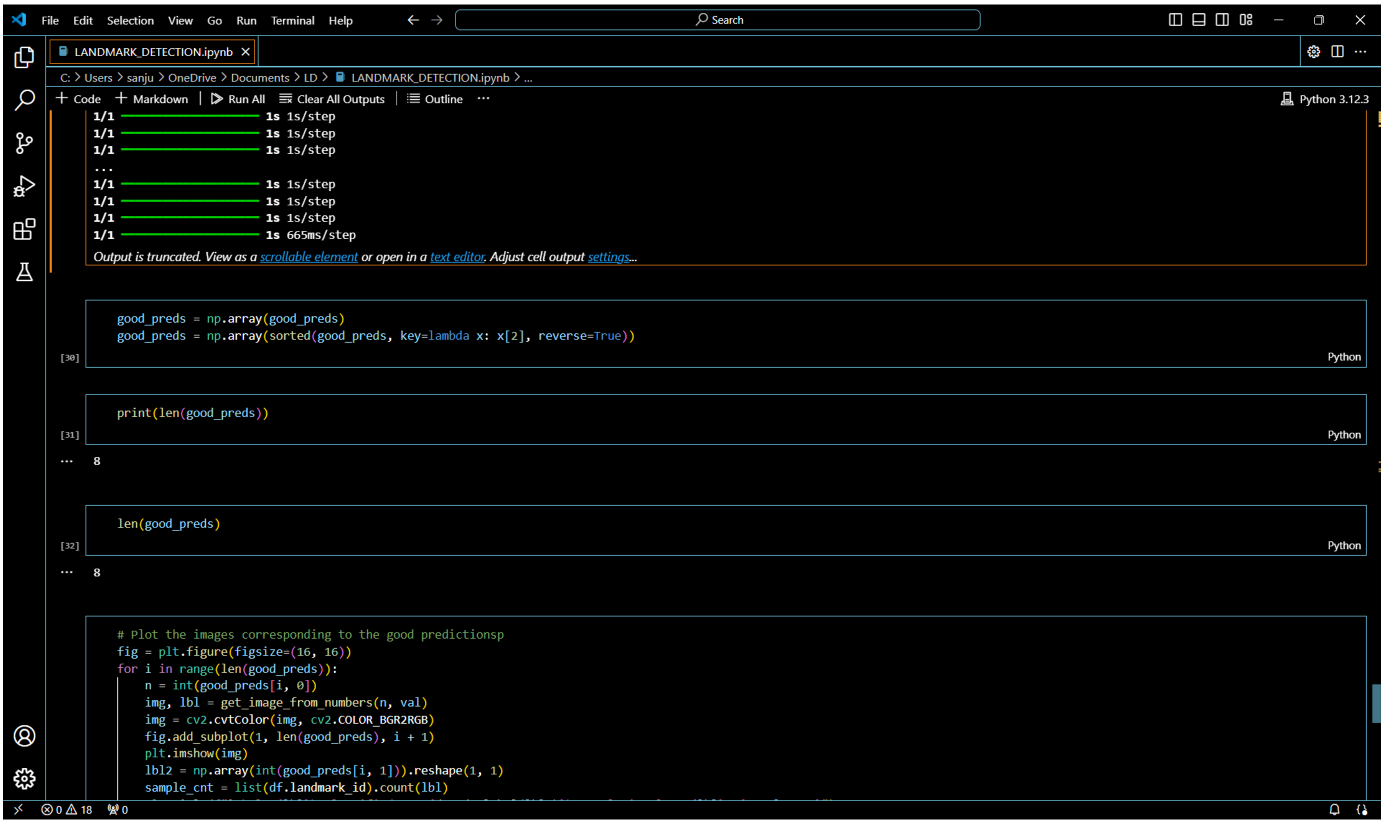
```
WARNING:absl:You are saving your model as an HDF5 file via `model.save()`. This file format is considered legacy. We recommend using instead the nat
    Batch completed.
```

```python
# Initialize counters and lists for tracking predictions
```


File Edit Selection View Go Run Terminal Help
Search
LANDMARK_DETECTION.ipynb
C: > Users > sanju > OneDrive > Documents > LD > LANDMARK_DETECTION.ipynb > ...
Code
Markdown
Run All
Clear All Outputs
Outline
Python 3.12.3
[29]
Python
0 18
0

```python
# Initialize counters and lists for tracking predictions
errors = 0
good_preds = []
bad_preds = []

for it in range(int(np.ceil(len(val) / batch_size))):
    x_val, y_val = get_batch(val, it * batch_size, batch_size)

    result = model.predict(x_val)
    cla = np.argmax(result, axis=1)
    for idx, res in enumerate(result):
        if cla[idx] != y_val[idx]:
            errors += 1
            bad_preds.append([batch_size * it + idx, cla[idx], res[cla[idx]]])
        else:
            good_preds.append([batch_size * it + idx, cla[idx], res[cla[idx]]])
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 1s/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 1s/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 1s/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 1s/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 1s/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 1s/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 1s/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 1s/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 1s/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 1s/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 1s/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 1s/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 1s/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 1s/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 1s/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 1s/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 1s/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 1s/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 1s/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 1s/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 1s/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 1s/step
```


File Edit Selection View Go Run Terminal Help
Search
LANDMARK_DETECTION.ipynb
C: > Users > sanju > OneDrive > Documents > LD > LANDMARK_DETECTION.ipynb > ...
+ Code + Markdown | Run All Clear All Outputs | Outline ...
Python 3.12.3

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 1s/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 1s/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 1s/step
...
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 1s/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 1s/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 1s/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 665ms/step
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

```
good_preds = np.array(good_preds)
good_preds = np.array(sorted(good_preds, key=lambda x: x[2], reverse=True))
```

[30] Python

```
print(len(good_preds))
```

[31] Python

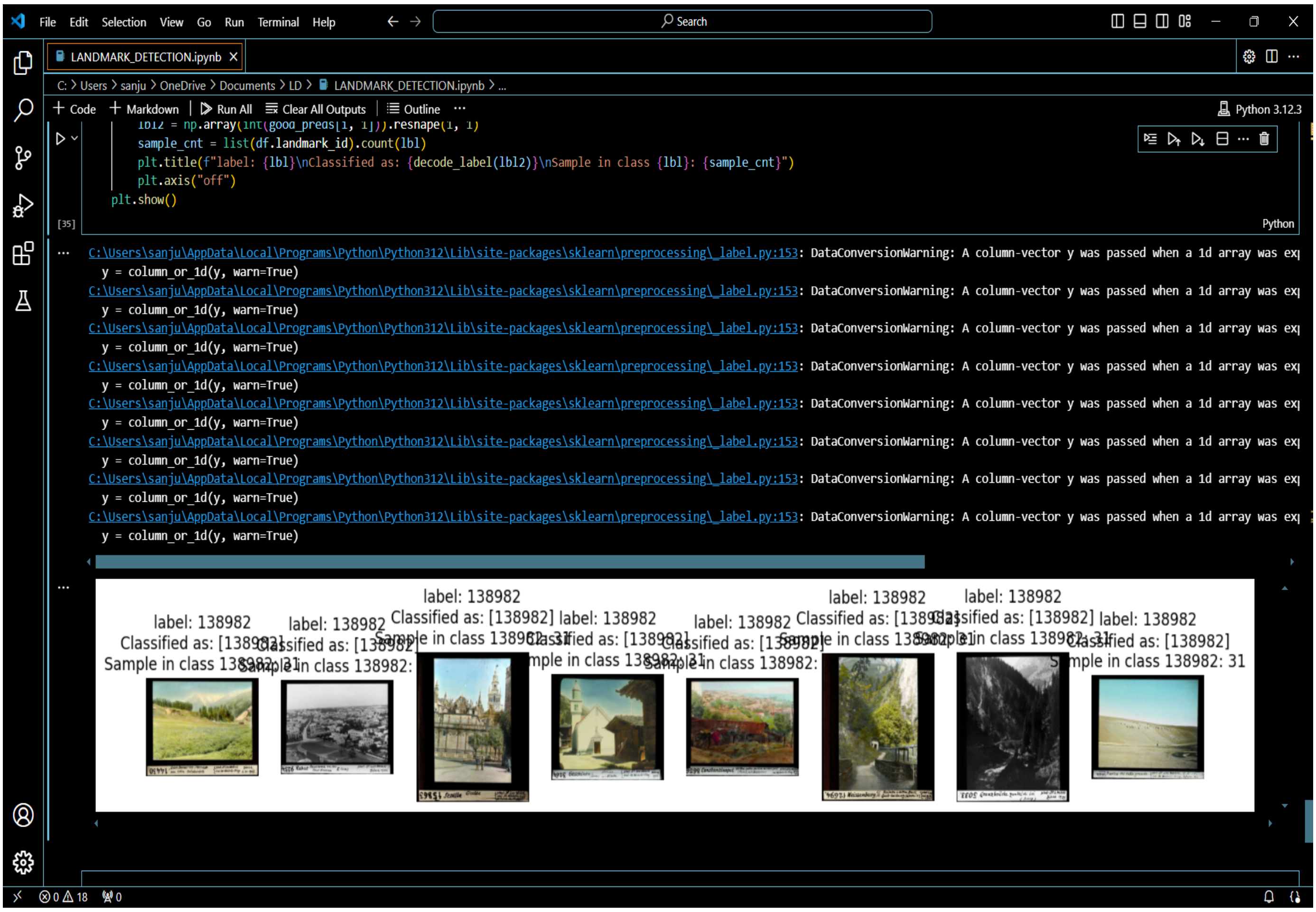
```
8
```

```
len(good_preds)
```

[32] Python

```
8
```

```
# Plot the images corresponding to the good predictionsp
fig = plt.figure(figsize=(16, 16))
for i in range(len(good_preds)):
    n = int(good_preds[i, 0])
    img, lbl = get_image_from_numbers(n, val)
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    fig.add_subplot(1, len(good_preds), i + 1)
    plt.imshow(img)
    lbl2 = np.array(int(good_preds[i, 1])).reshape(1, 1)
    sample_cnt = list(df.landmark_id).count(lbl)
```


File Edit Selection View Go Run Terminal Help
Search
LANDMARK_DETECTION.ipynb
C: > Users > sanju > OneDrive > Documents > LD > LANDMARK_DETECTION.ipynb > ...
+ Code + Markdown | Run All Clear All Outputs | Outline ...
Python 3.12.3

```
    lbl2 = np.array(int(good_preds[i, 1])).reshape(1, 1)
    sample_cnt = list(df.landmark_id).count(lbl)
    plt.title(f"label: {lbl}\nClassified as: {decode_label(lbl2)}\nSample in class {lbl}: {sample_cnt}")
    plt.axis("off")
plt.show()
```

[35] Python

```
C:\Users\sanju\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\preprocessing\_label.py:153: DataConversionWarning: A column-vector y was passed when a 1d array was ex
  y = column_or_1d(y, warn=True)
C:\Users\sanju\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\preprocessing\_label.py:153: DataConversionWarning: A column-vector y was passed when a 1d array was ex
  y = column_or_1d(y, warn=True)
C:\Users\sanju\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\preprocessing\_label.py:153: DataConversionWarning: A column-vector y was passed when a 1d array was ex
  y = column_or_1d(y, warn=True)
C:\Users\sanju\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\preprocessing\_label.py:153: DataConversionWarning: A column-vector y was passed when a 1d array was ex
  y = column_or_1d(y, warn=True)
C:\Users\sanju\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\preprocessing\_label.py:153: DataConversionWarning: A column-vector y was passed when a 1d array was ex
  y = column_or_1d(y, warn=True)
C:\Users\sanju\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\preprocessing\_label.py:153: DataConversionWarning: A column-vector y was passed when a 1d array was ex
  y = column_or_1d(y, warn=True)
C:\Users\sanju\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\preprocessing\_label.py:153: DataConversionWarning: A column-vector y was passed when a 1d array was ex
  y = column_or_1d(y, warn=True)
C:\Users\sanju\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\preprocessing\_label.py:153: DataConversionWarning: A column-vector y was passed when a 1d array was ex
  y = column_or_1d(y, warn=True)
```