Project Report

Garbage Classification Using Deep Learning

BY

Mohammed Marzuk Ali .S

Akash Kumar .S

Sridhar .S

Nikesh .C

INDEX

INTRODUCTION

1.1 Overview

A computer vision approach to classify garbage into recycling categories could be an efficient way to process waste. The project aims to take garbage waste images and classify them into six classes: cardboard, glass, paper, metal, plastic and trash. We use garbage image database that contains around 2696 images for each i.e. 450 approx. In this project, we import the images in data generator library and configure the class and then apply the image data generator functionality then we import the model building libraries. Initialize the model, then we add the layers, first CNN and then dense.

Then we have built the HTML web page and for doing this we used:

- Git
- Git Hub
- Python 3.7
- Anaconda Spyder
- Html5
- CSS3
- Java Script
- Flask Framework

1.2 Purpose

This code set shows how to differentiate the garbage type by pulling it aside on the basis of 2696 images we have uploaded and described which will be recognized by CNN. the six type of garbage's are: cardboard, glass, paper, metal, plastic and trash.

On the basis of recognition, the model differentiates the garbage. which is possibly harmful or hazardous to humans labours.

The main purpose of garbage collection system is to provide another way for the labours for reducing the hand-picking method and giving the e-waste materials.

The goals for the system are:

- to make the waste separate by image identification
- to rescue electronic material by recycling or deploy
- to decrease the mixing of waste material from every household

LITERATURE SURVEY

2.1 Existing Problems

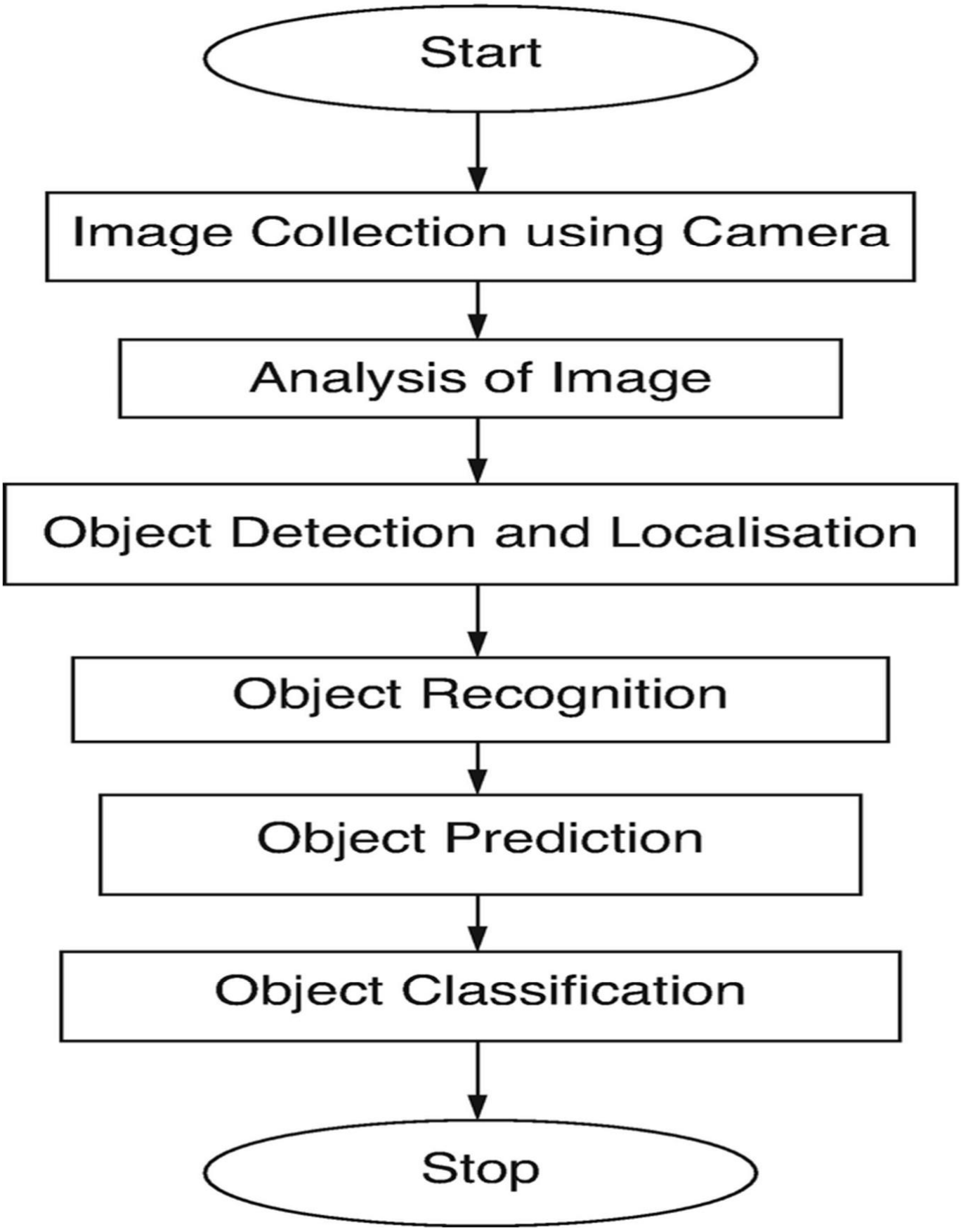
The accumulation of solid waste in the urban area is becoming a great concern, and it would result in environmental pollution and may be hazardous to human health if it is not properly managed. It is important to have an advanced/intelligent waste management system to manage a variety of waste materials. One of the most important steps of waste management is the separation of the waste into the different components and this process is normally done manually by hand-picking. Convolutional Neural Network model which is used to classify the waste into different groups/types such as glass, metal, paper, and plastic etc. The proposed system is tested on the trash image data set which was developed by Gary Thung and Mindy Yang, and is able to achieve an accuracy of 87% on the data set. The separation process of the waste will be faster and intelligent using the proposed waste material classification system without or reducing human involvement.

2.2 Proposed Solution

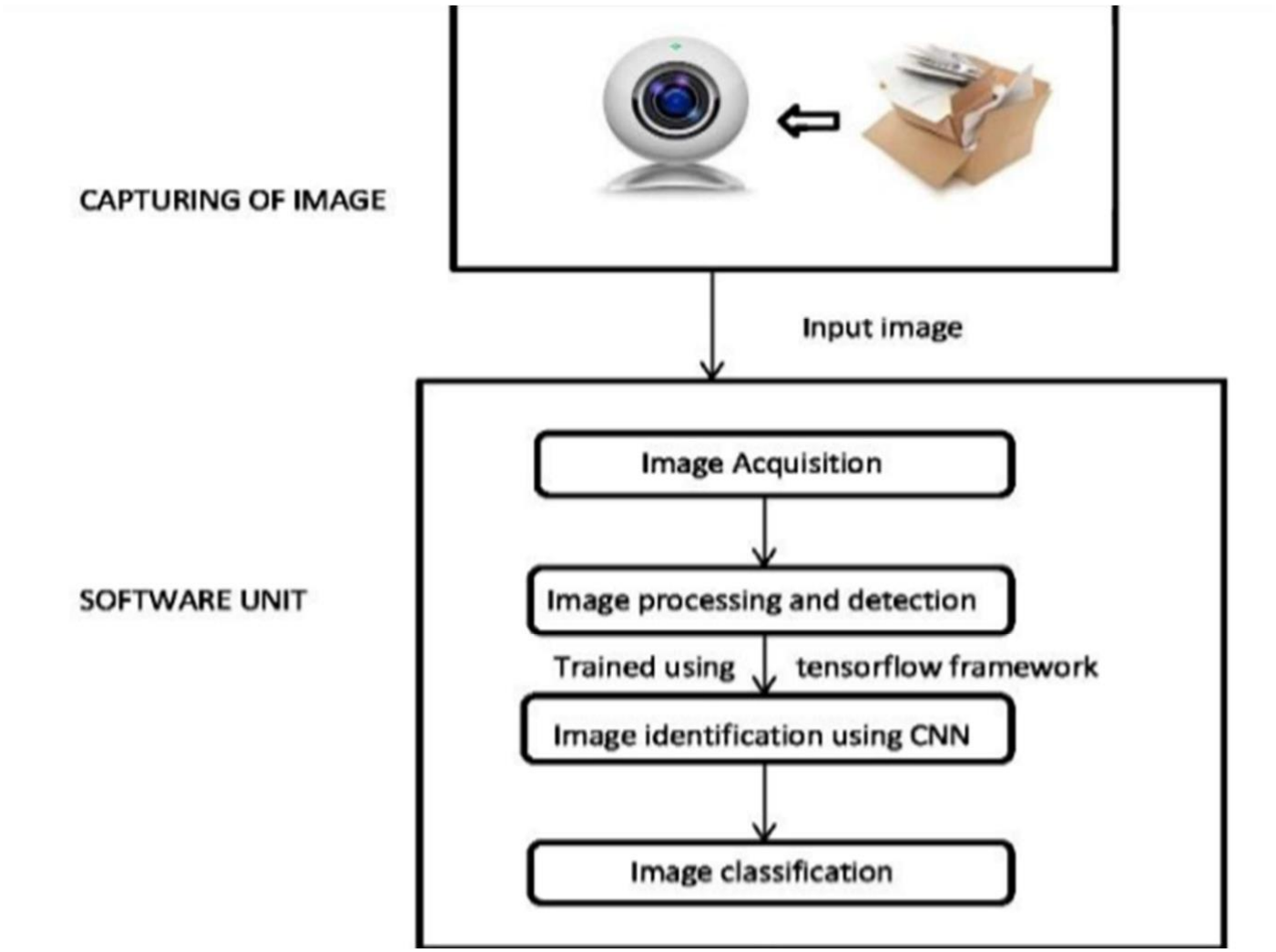
The present way of separating waste/garbage is the hand-picking method, whereby someone is employed to separate out the different objects/materials. The person, who separate waste, is prone to diseases due to the harmful substances in the garbage. With this in mind, it motivated us to develop an automated system which is able to sort the waste and this system can take short time to sort the waste, and it will be more accurate in sorting than the manual way. With the system in place, the beneficial separated waste can still be recycled and converted to energy and fuel for the growth of the economy. The system that is developed for the separation of the accumulated waste is based on the combination of Convolutional Neural Network (CNN) with recognition and classification.

THEORITICAL ANALYSIS

3.1 **Block Diagram**



3.2 Hardware/Software Designing

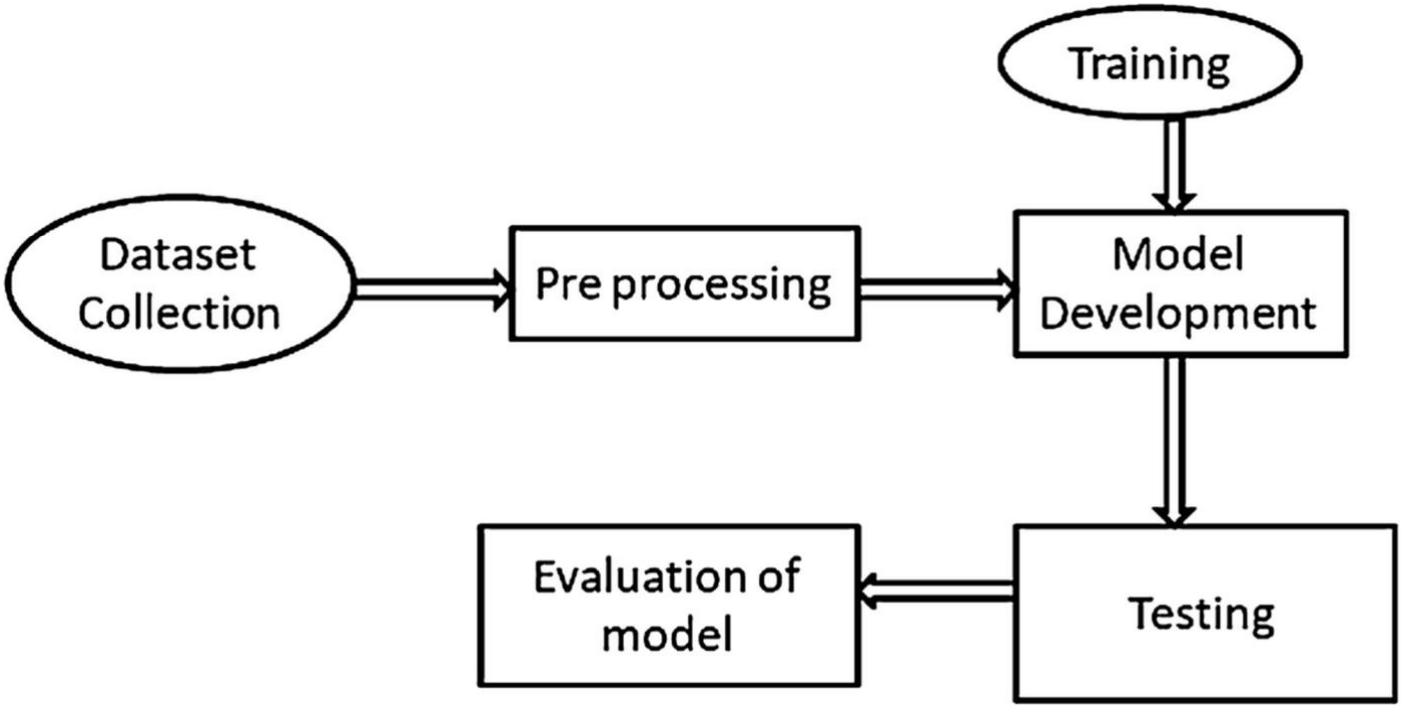


EXPERIMENTAL INVESTIGATION

In this process of developing the project we have undergone many investigation processes to learn and understand the concept of smart garbage classification on the basis of CNN and to make it successfully. For this we have to learn and investigate following:

- Convolution Neural Network
- ZOHO Writer
- Anaconda Spyder
- Slack
- Git Hub
- Git
- Alex net
- Sublime Text for CSS3

FLOWCHART



RESULTS

1) **Data Collection**:

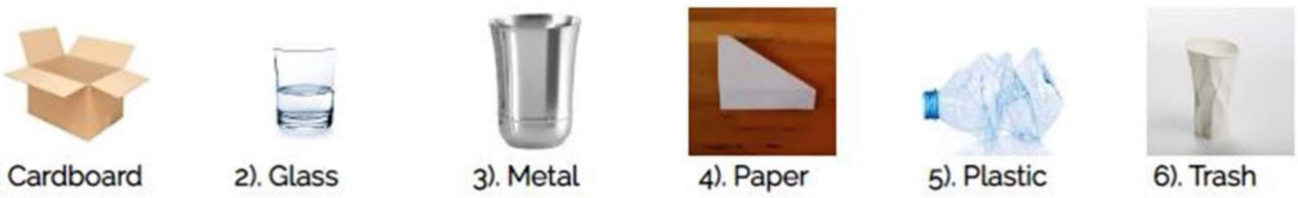
In our data sets, non-organic recyclable waste has been categorized in 6 groups. These are- cardboard, plastic, metal, glass, paper, and trash. For plastic, we have collected 480 images of each plastic bottles and boxes. In a similar fashion, for each group, we have chosen six objects and for each of them, we have collected 20 images. Hence, our total data sets have 2696 images of many different objects along with their recycle values.

2) **Object Detection**:

The process initiates by introducing a small region in the whole image and gradually merging them with hierarchical grouping. By analyzing the similarities between all neighboring regions, two most similar regions are selected and then grouped together. After that, the resulting group region is compared with all other neighboring regions and a new group is formed. The whole process continues until it becomes a single region. A demo of object detection and segmentation.

3) **Object Recognition**:

After detection and segmentation, the images of the objects are fed into the trained CNN for recognition and labeling purposes. The output of the CNN-based model along with a few labeled objects.



ADVANTAGES

- Reduce the cost and complexity of the edge devices.
- Labours life will be safe from harmful and hazardous waste.
- In household waste, there are some substances that are not easy to be degraded and these substances will pollute the soil badly. By garbage classification, we can remove substances that are recyclable and substances that are hard to be degraded. Thus reducing more than 50% of the waste.
- Reduce environmental pollution. The waste battery contains metal mercury, cadmium and other toxic substances that will do harm to human, the waste plastics in the soil will reduce the production of crop and the waste plastics can also lead to the death of animals. Thus recycling these wastes can reduce the damage.

DISADVANTAGE

- High cost
- Need skilled personals
- Labours that earn small amount of money by collecting the hand picked garbage's for their living will be effected .

APPLICATION

This garbage classification application powered by CNN is used to classify the type of garbage mainly glass, paper, cardboard and metal. Which recognized by the CNN image recognition process that will help us to classify the garbage and the type of recycling process is to be processed for different type of garbage classified.

CONCLUSION

In conclusion, we proposed a garbage classification system that is able to separate different components of waste using the Machine learning tools. This system can be used to automatically classify waste and help in reducing human intervention and preventing infection and pollution. From the result, when tested against the trash data set, we got an accuracy of 87%. The separation process of the waste will be faster and intelligent using our system without or reducing human involvement. If more image is added to the data set, the system accuracy can be improved In the future, we will tend to improve our system to be able to categories more waste items, by turning some of the parameters used.

FUTURE SCOPE

The product developed can also find application in various food industries, fabrication industries and many more where all the process of segregation is involved.

According to the need, the images of the materials to be separated will be entered in the data set and the algorithm is trained.

BIBILOGRAPHY

Name: Mohammed Marzuk Ali .S

College Name: Coimbatore Institute Of Technology

Work Title: Garbage Classification Using Deep Learning

Name: Akash Kumar .S

College Name: Coimbatore Institute Of Technology

Work Title: Garbage Classification Using Deep Learning

Name: Sridhar .S

College Name: Coimbatore Institute Of Technology

Work Title: Garbage Classification Using Deep Learning

Name: Nikesh .C

College Name: Coimbatore Institute Of Technology

Work Title: Garbage Classification Using Deep Learning

Reference:

1. **Train and Test Image Data Set**
https://drive.google.com/drive/folders/1_RZkL0ZcFfSk6jnysyQ1UVDgMp_2ssm5?usp=sharing

2. **HTML, CSS, JavaScript and Python**
https://stackoverflow.com/

3. **HTML, CSS and JavaScript**
https://www.w3schools.com/

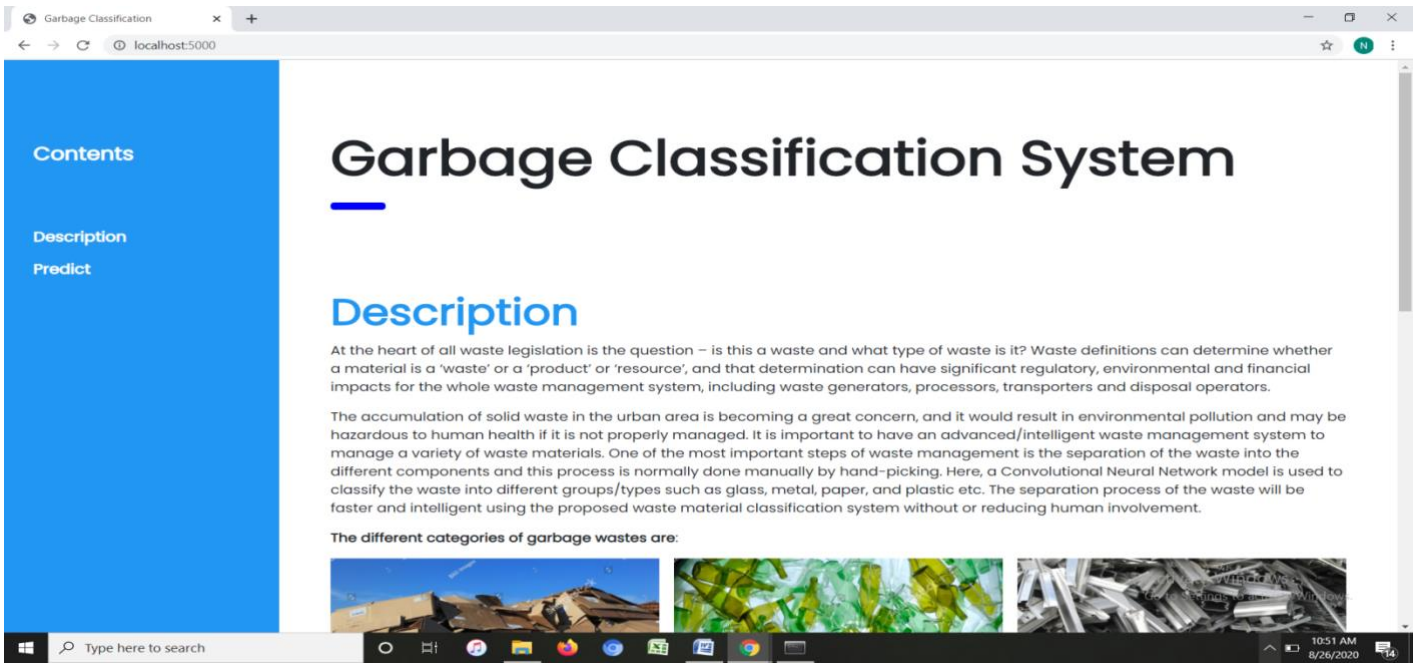
4. **Python CNN Issues**
https://github.com/

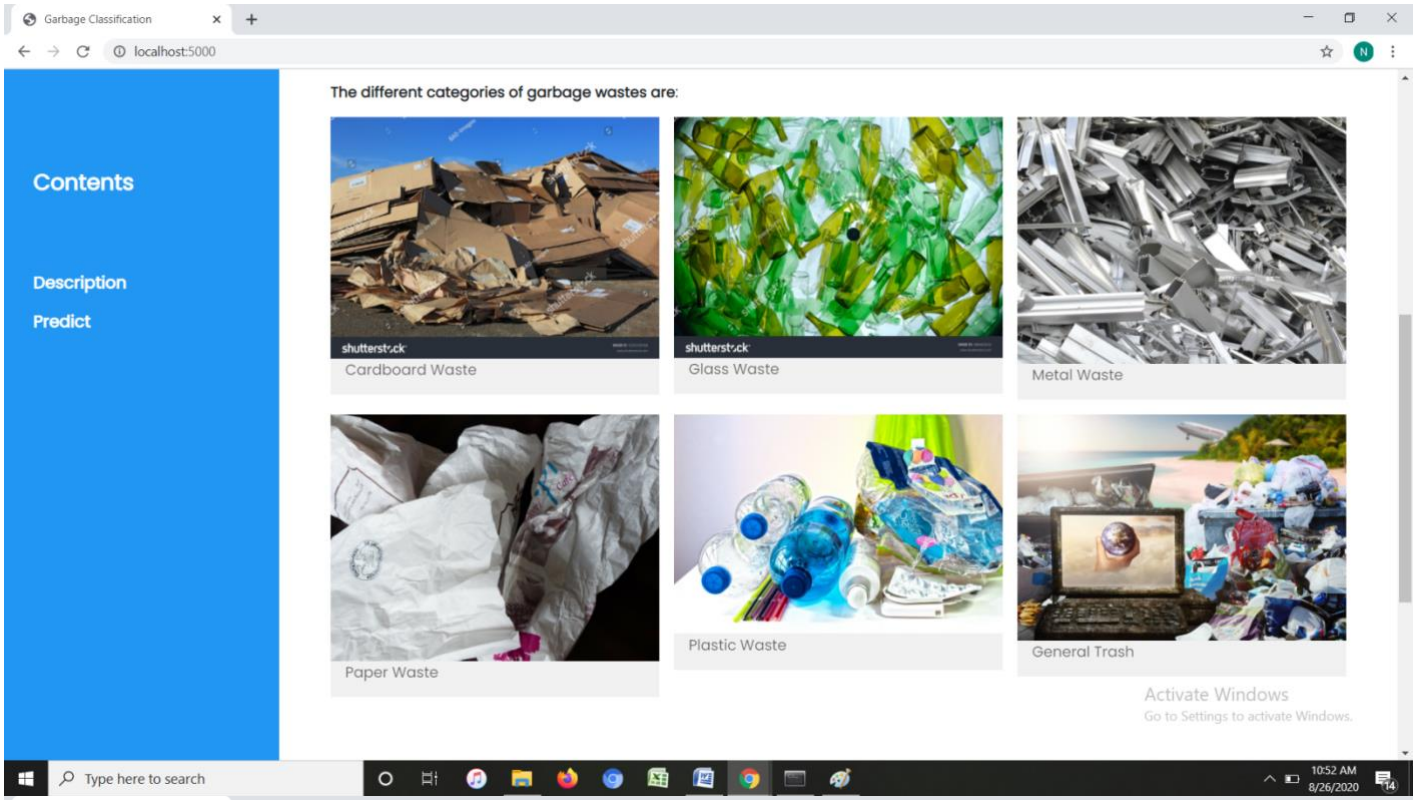
5. **Python Preprocessor and Model Building**
https://www.kaggle.com/rkuo2000/garbage-cnn

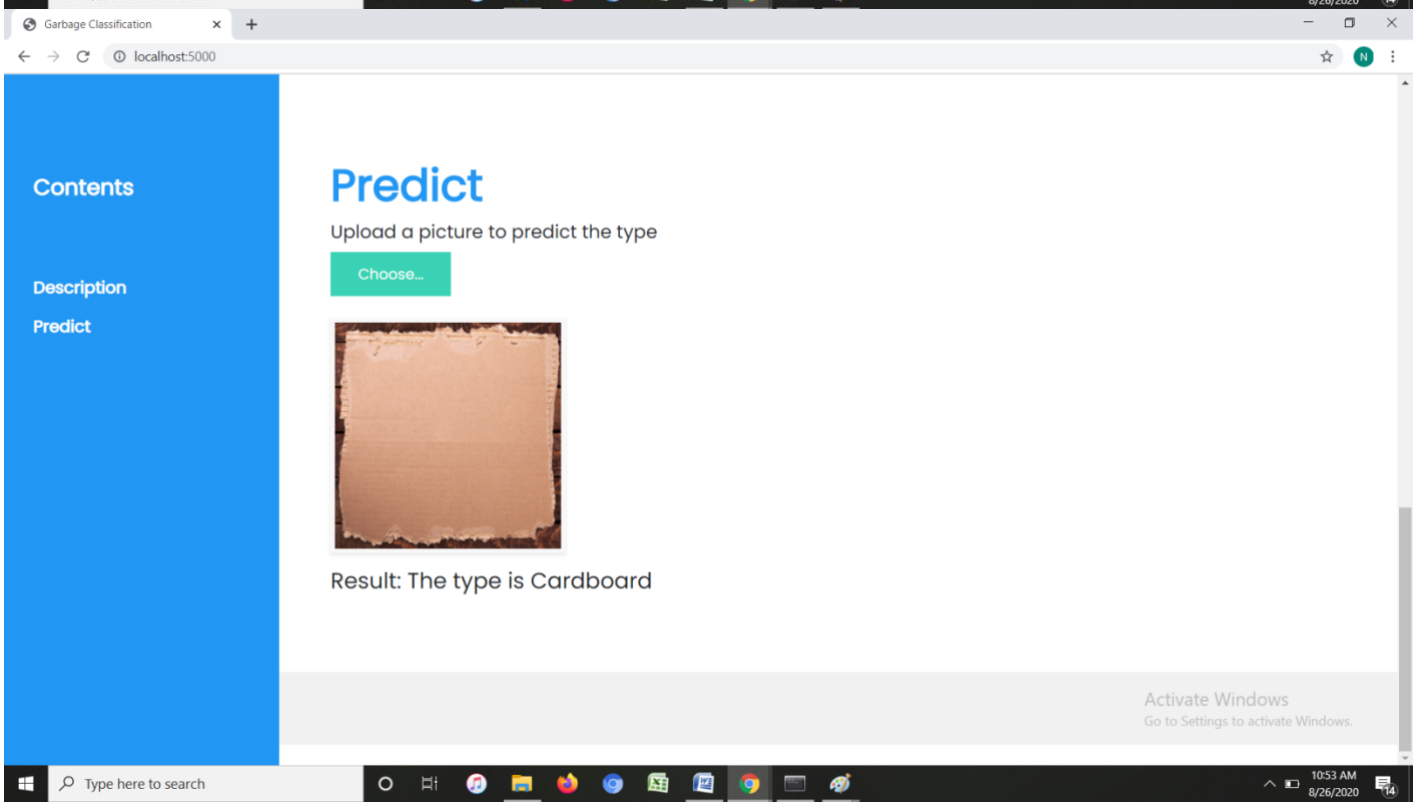
6. **Working with Flask Framework**
https://palletsprojects.com/p/flask/

APPENDIX

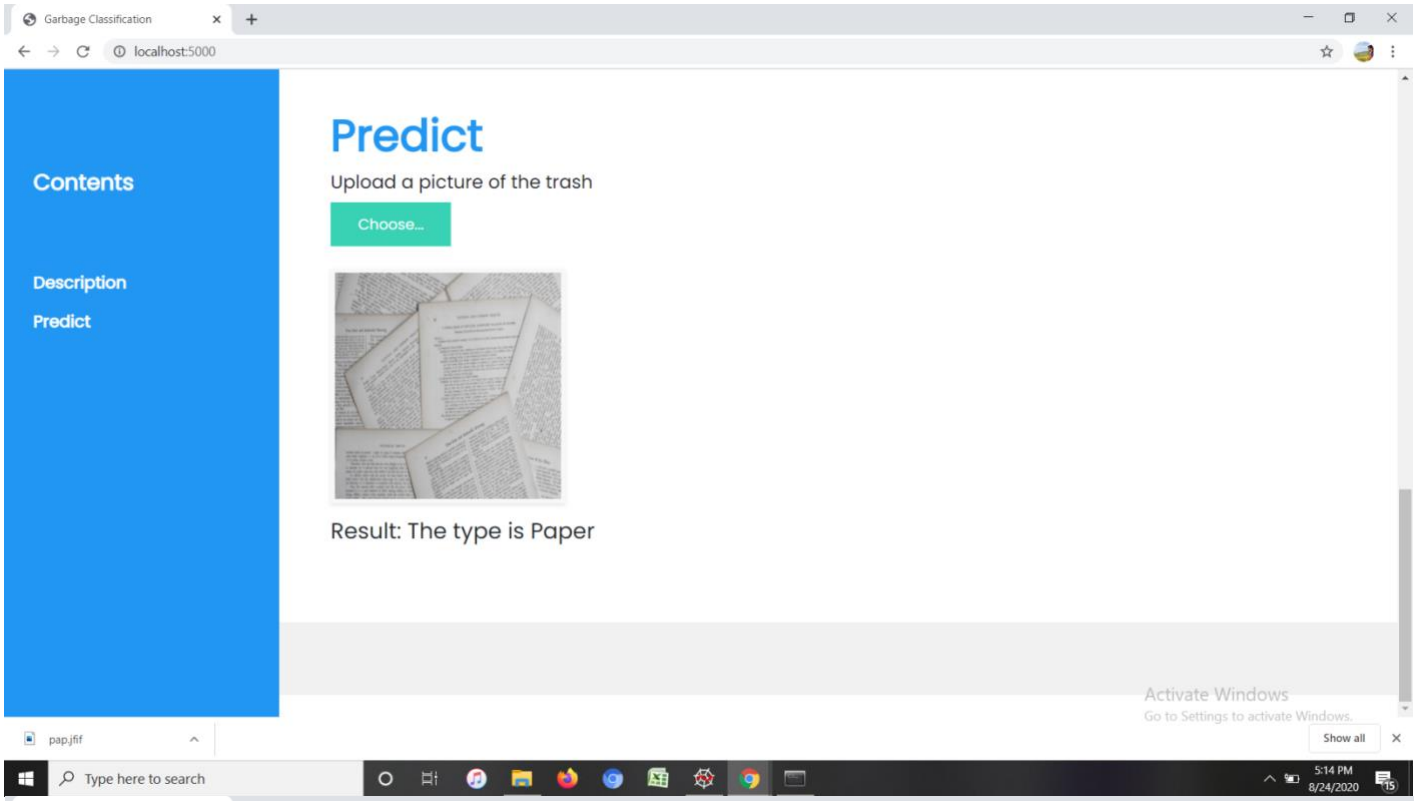
A. User Interface:

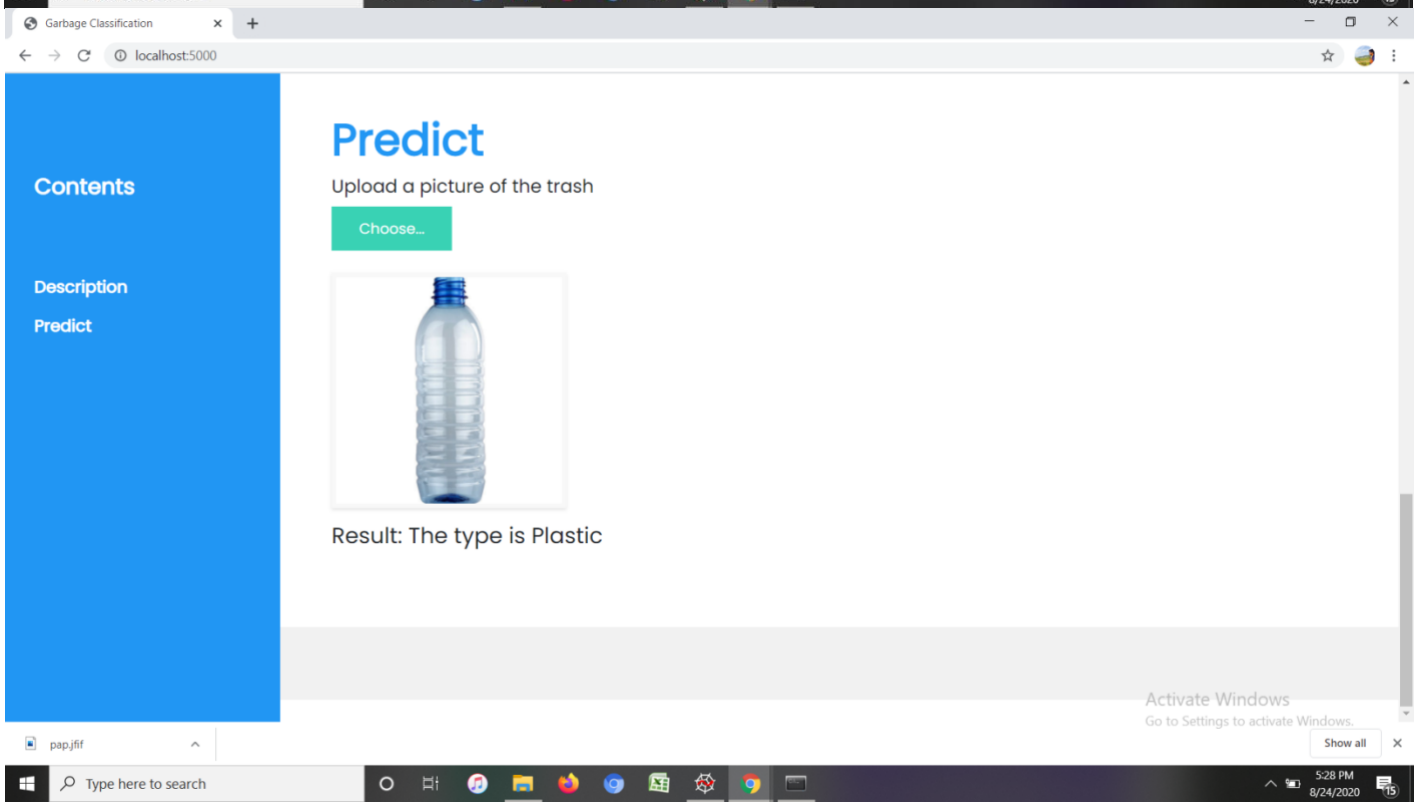



Garbage Classification
localhost:5000
Contents
Description
Predict
The different categories of garbage wastes are:
Cardboard Waste
Glass Waste
Metal Waste
Paper Waste
Plastic Waste
General Trash
Activate Windows
Go to Settings to activate Windows.
Type here to search
10:52 AM
8/26/2020


Garbage Classification
localhost:5000
Contents
Description
Predict
Predict
Upload a picture to predict the type
Choose...
Result: The type is Cardboard
Activate Windows
Go to Settings to activate Windows.
Type here to search
10:53 AM
8/26/2020

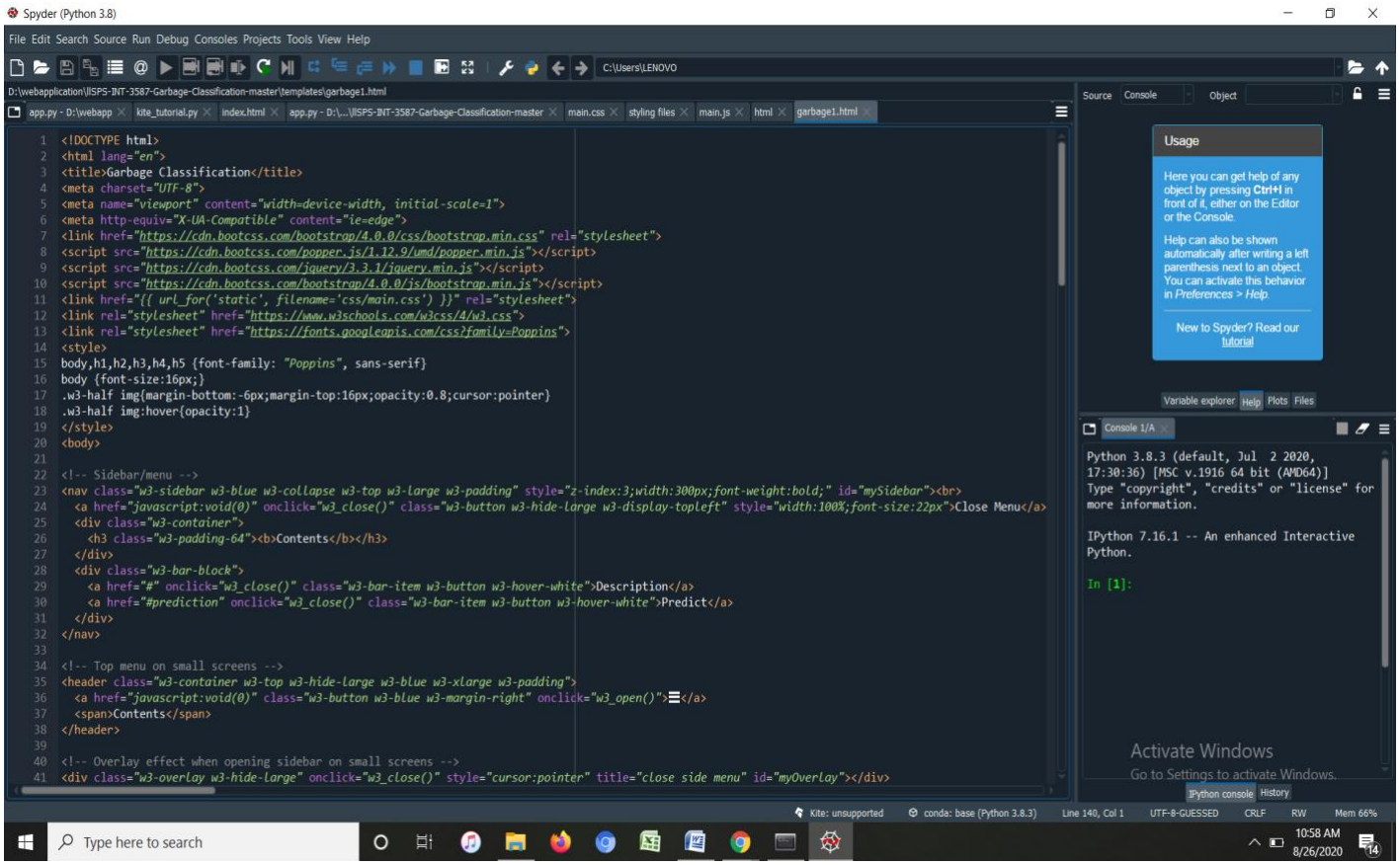
➔Predicting the type of garbage:

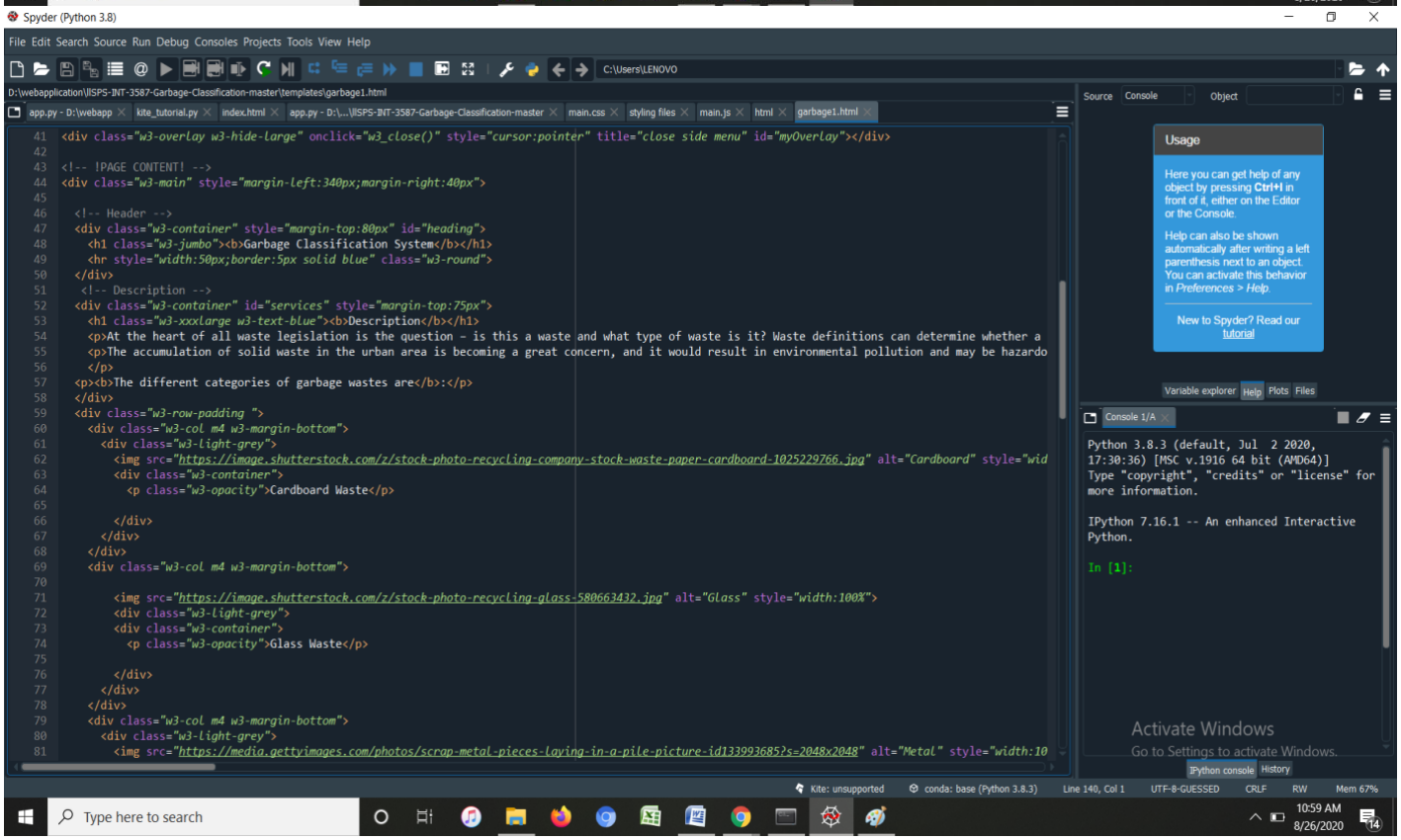


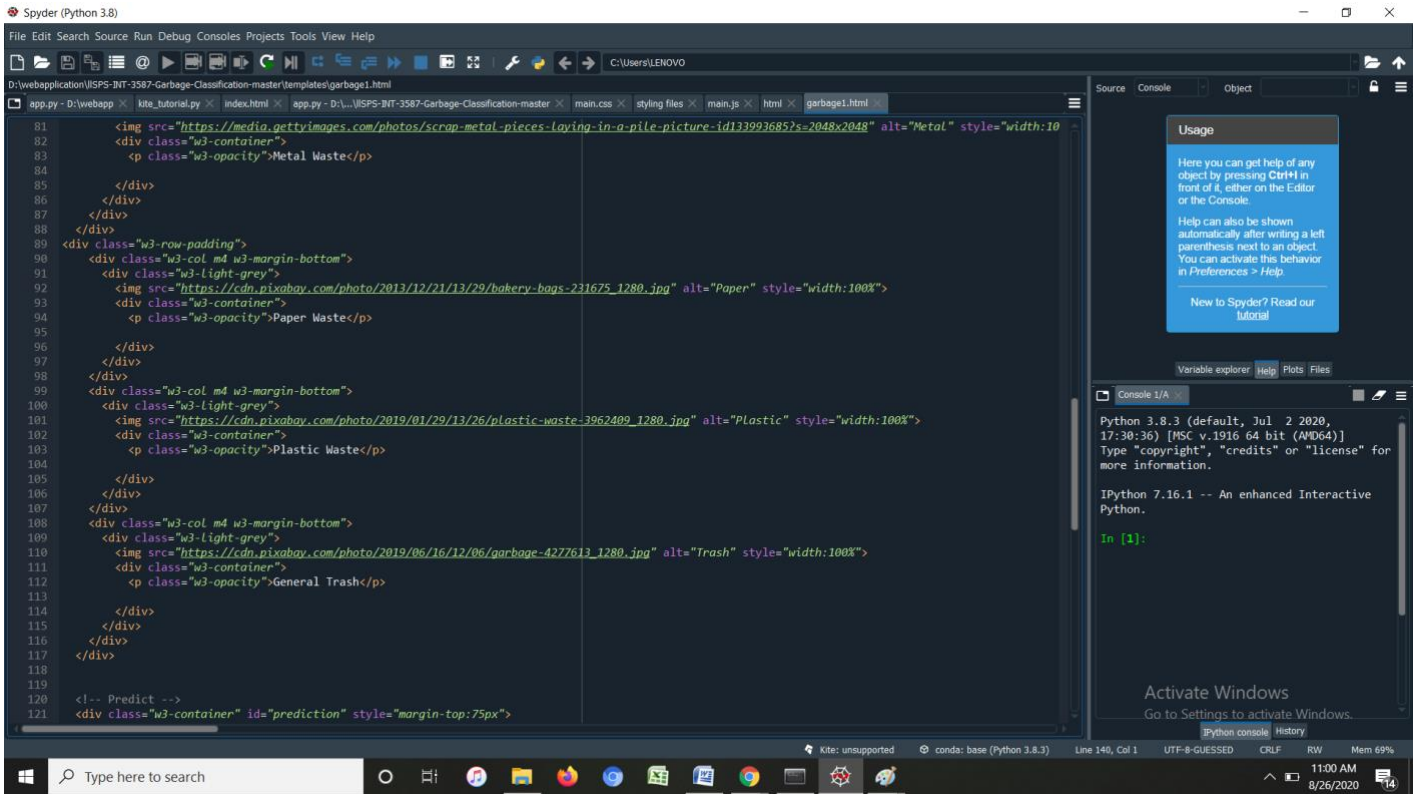


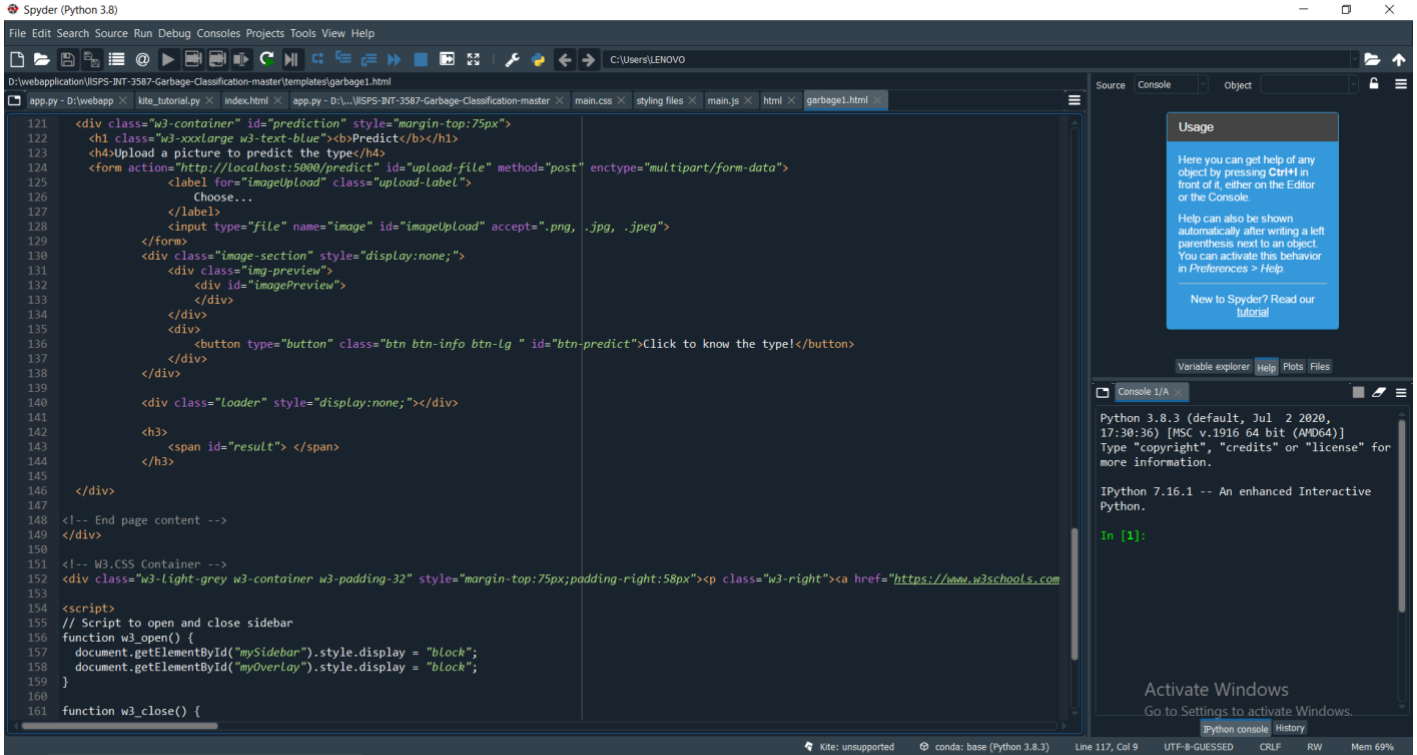
B. **Source Code**:

1. **garbage.html**



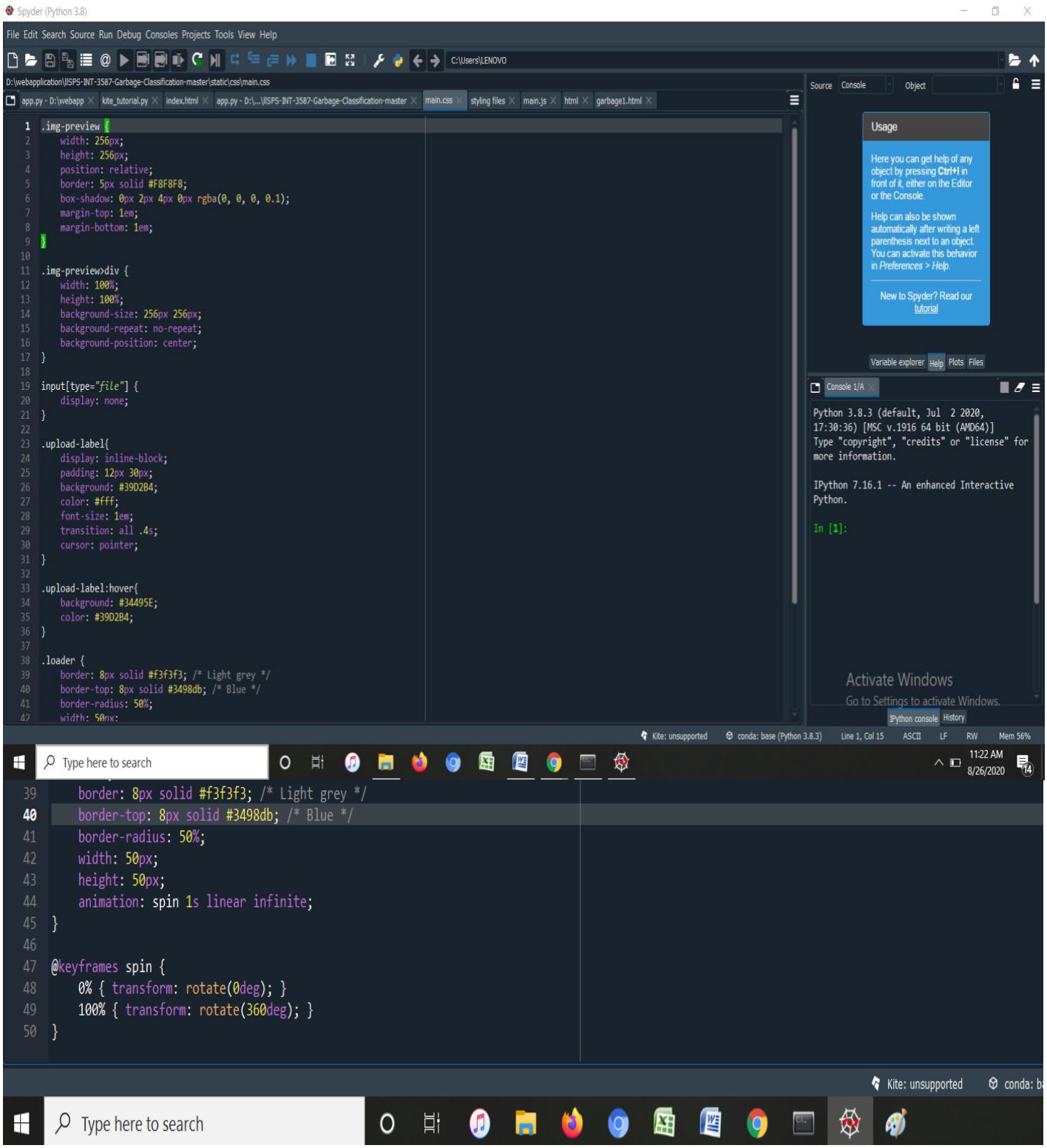






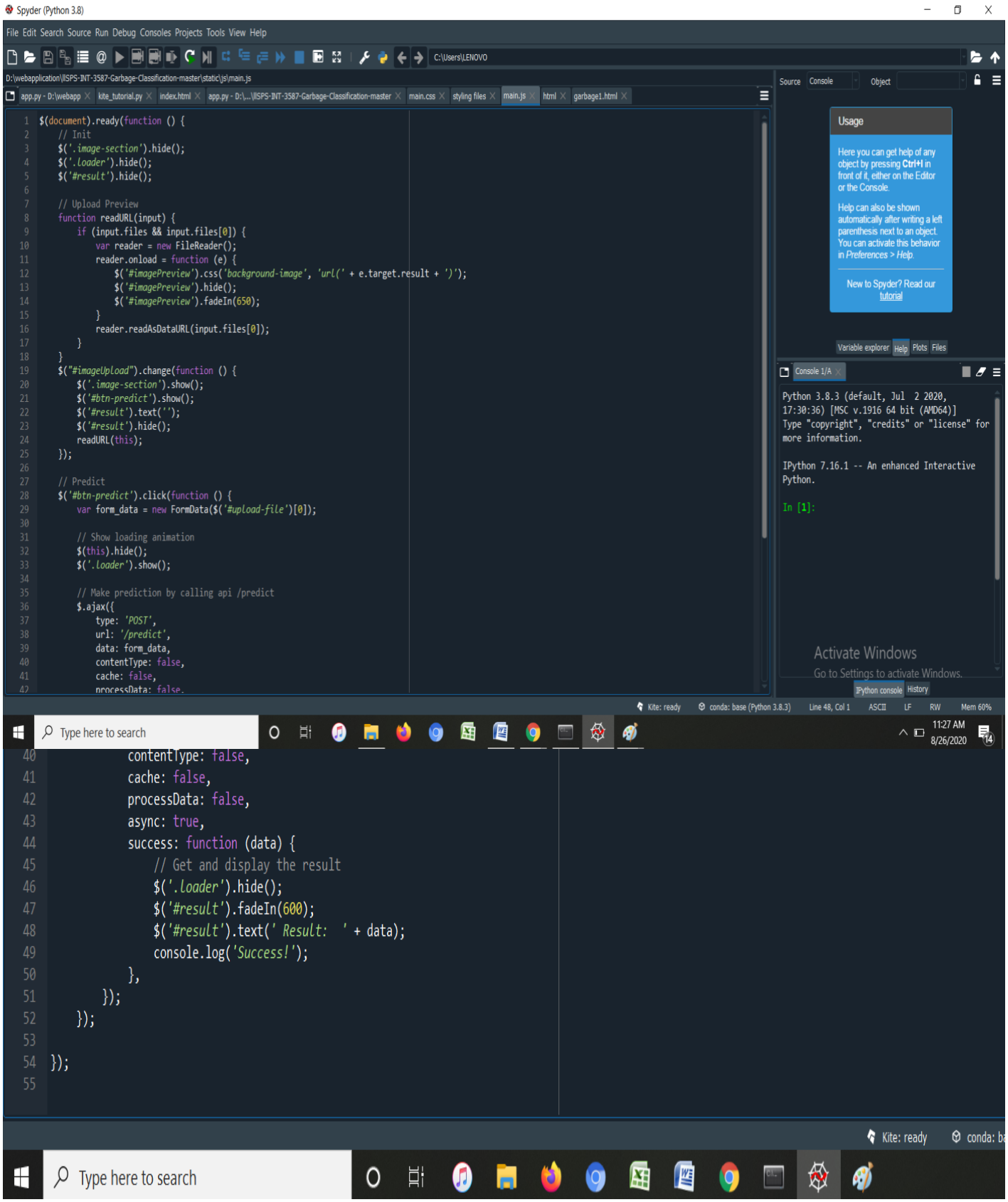

```
function w3_close() {
  document.getElementById("mySidebar").style.display = "none";
  document.getElementById("myOverlay").style.display = "none";
}


</script>

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

2. main.css



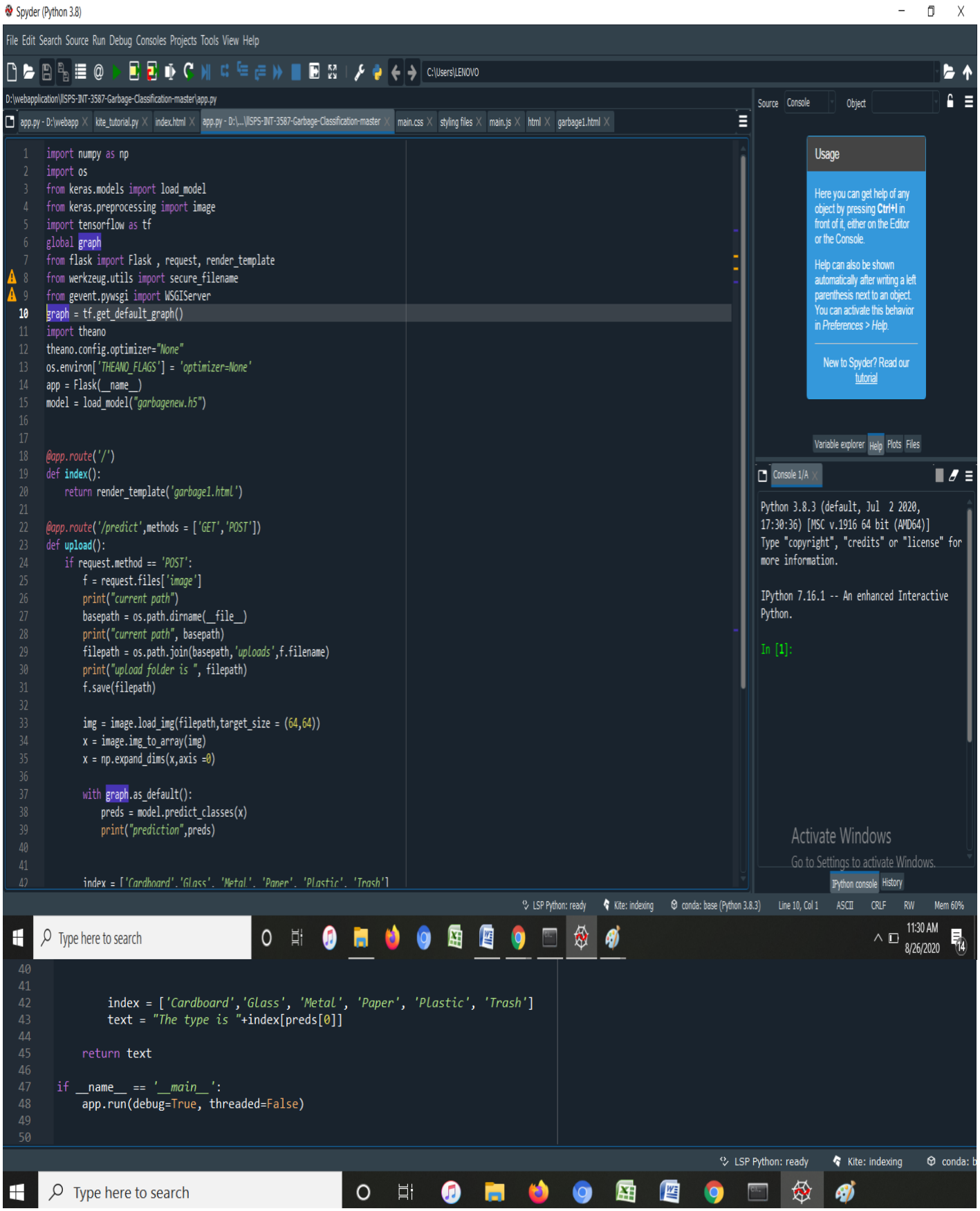
```
.img-preview {
    width: 256px;
    height: 256px;
    position: relative;
    border: 5px solid #F8F8F8;
    box-shadow: 0px 2px 4px 0px rgba(0, 0, 0, 0.1);
    margin-top: 1em;
    margin-bottom: 1em;
}

.img-preview>div {
    width: 100%;
    height: 100%;
    background-size: 256px 256px;
    background-repeat: no-repeat;
    background-position: center;
}

input[type="file"] {
    display: none;
}

.upload-label{
    display: inline-block;
    padding: 12px 30px;
    background: #39D2B4;
    color: #fff;
    font-size: 1em;
    transition: all .4s;
    cursor: pointer;
}

.upload-label:hover{
    background: #34495E;
    color: #39D2B4;
}

.loader {
    border: 8px solid #f3f3f3; /* Light grey */
    border-top: 8px solid #3498db; /* Blue */
    border-radius: 50%;
    width: 50px;
    height: 50px;
    animation: spin 1s linear infinite;
}

@keyframes spin {
    0% { transform: rotate(0deg); }
    100% { transform: rotate(360deg); }
}
```

3. main.js

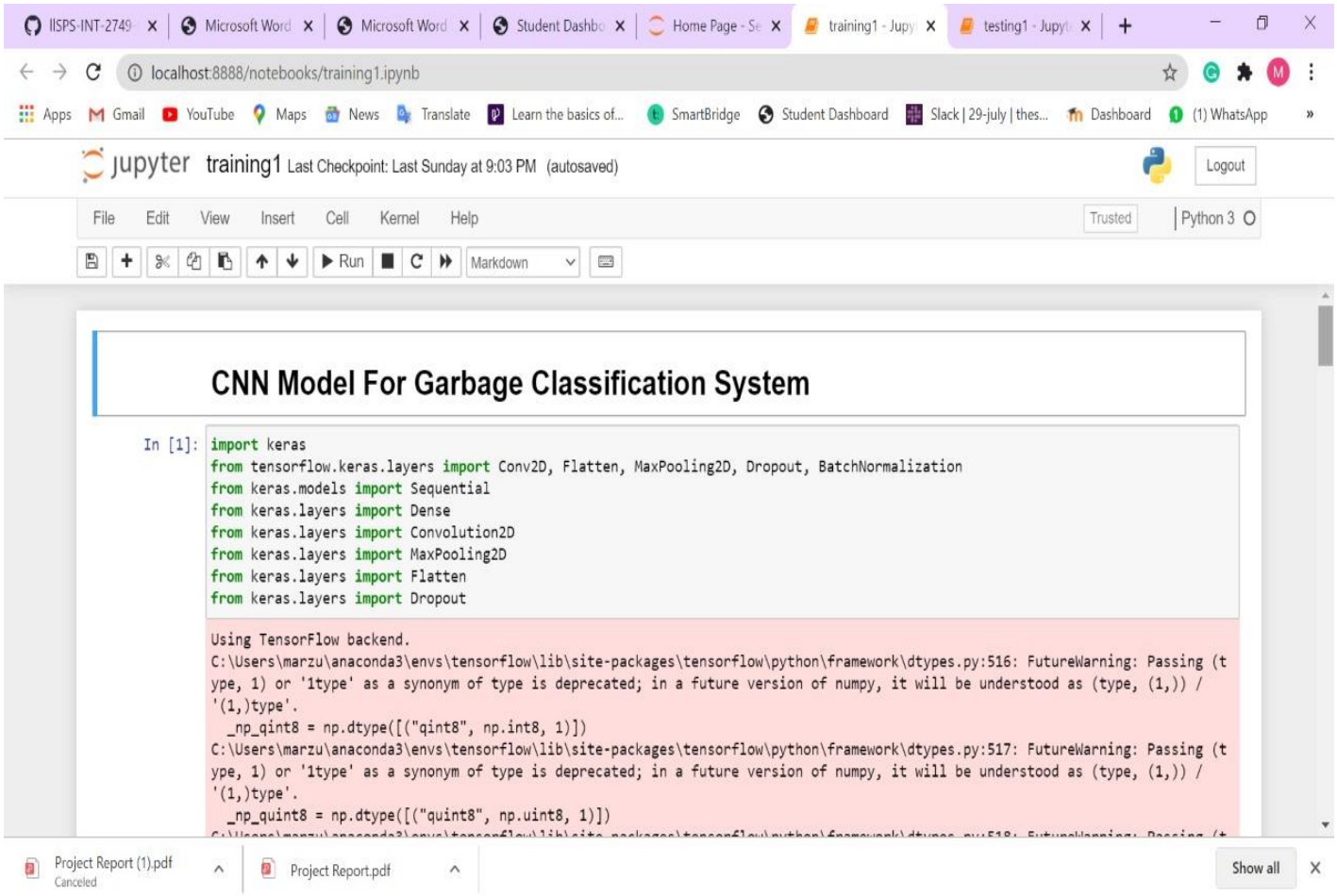


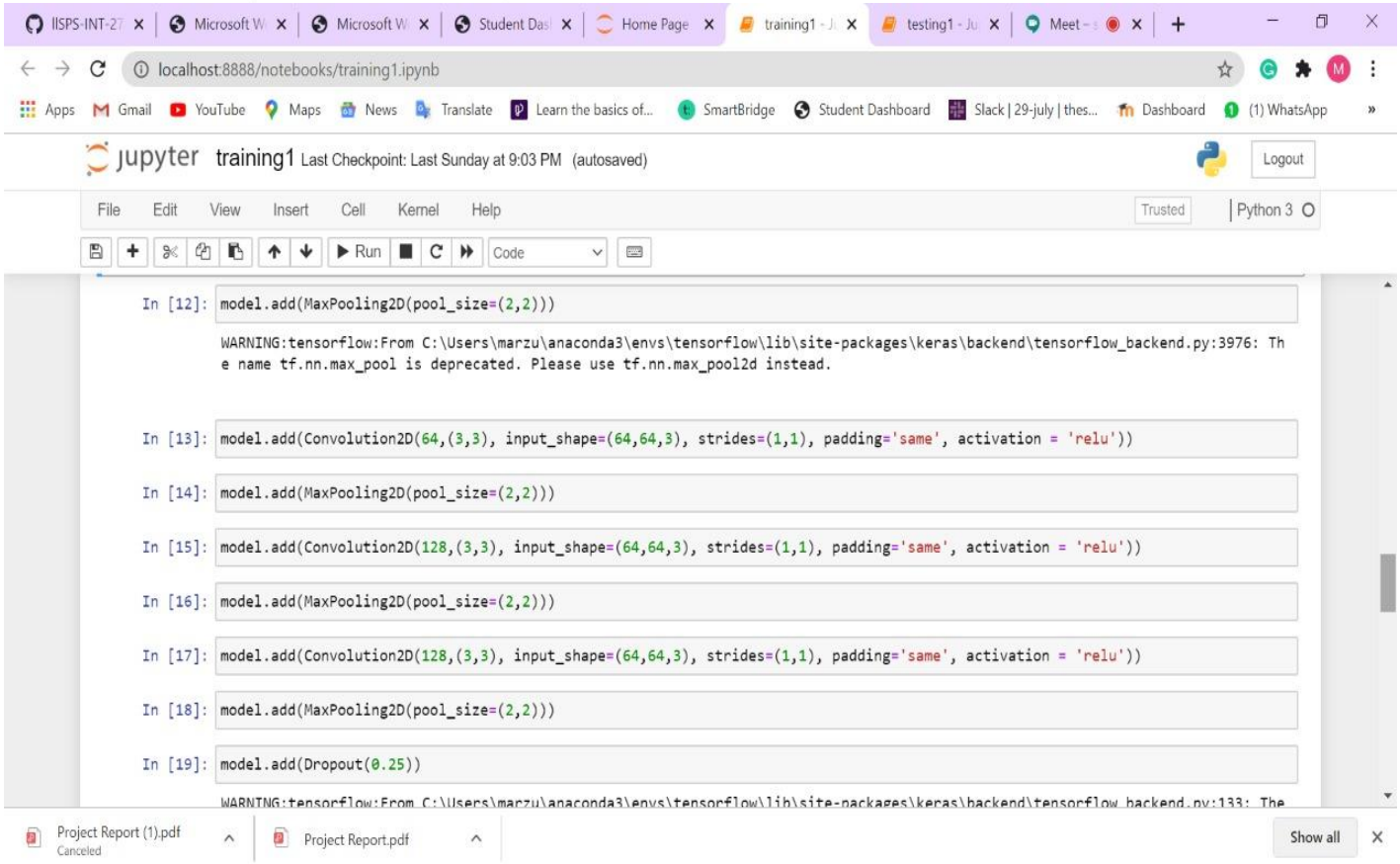

```
            contentType: false,
            cache: false,
            processData: false,
            async: true,
            success: function (data) {
                // Get and display the result
                $('.loader').hide();
                $('#result').fadeIn(600);
                $('#result').text(' Result:  ' + data);
                console.log('Success!');
            },
        });
    });

});
```

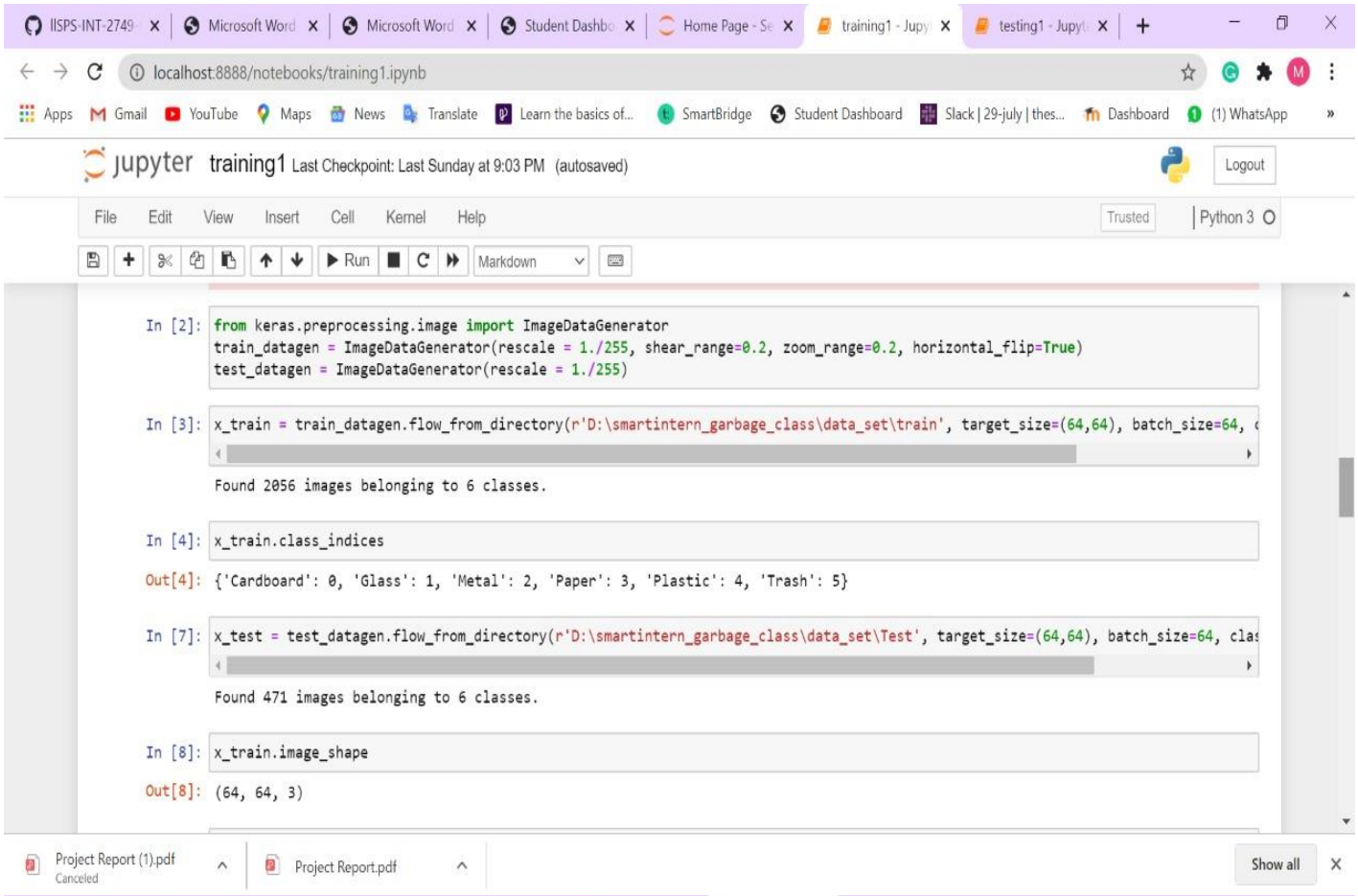
4. app.py

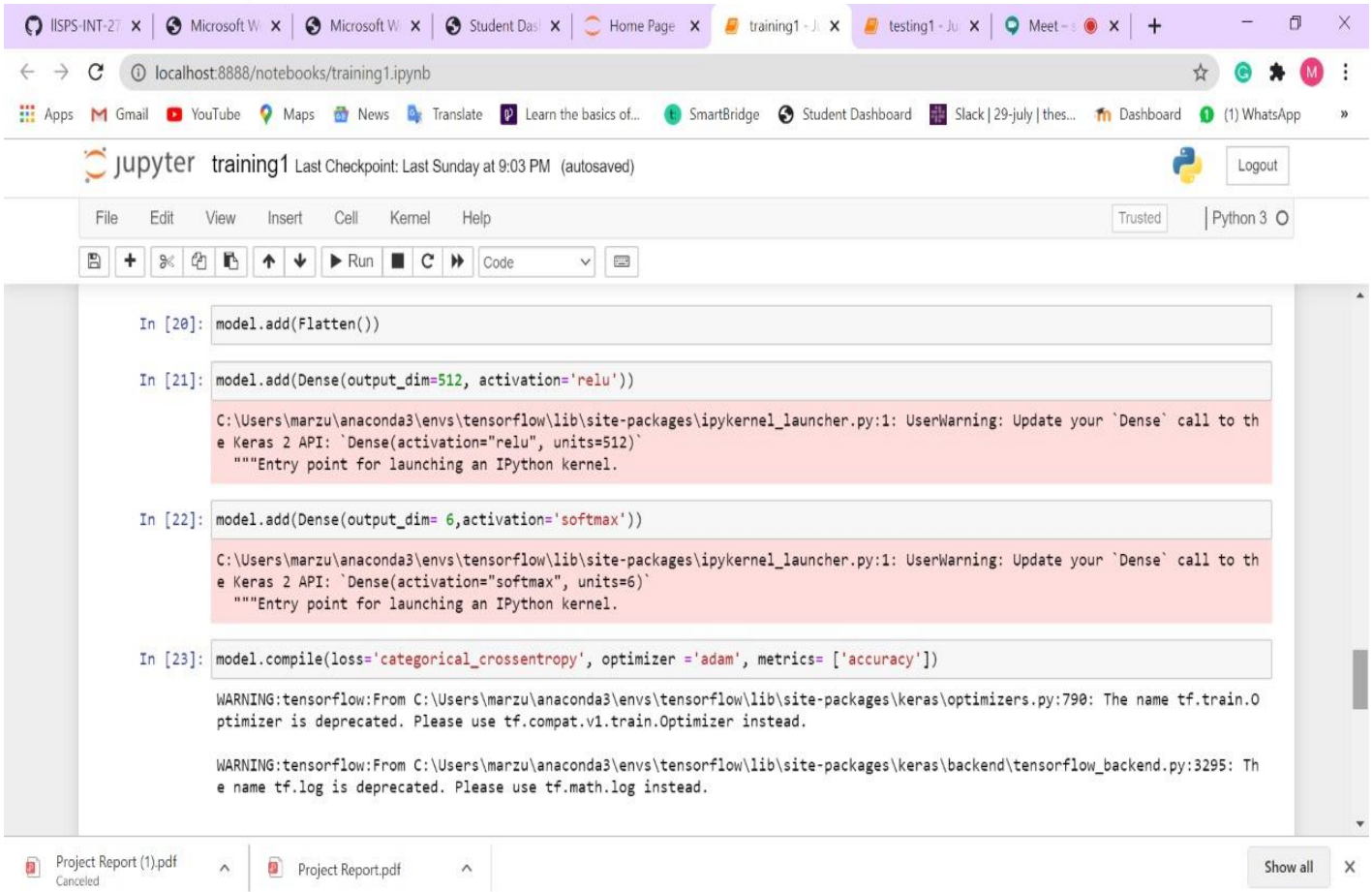

Spyder (Python 3.8)
File Edit Search Source Run Debug Consoles Projects Tools View Help
app.py - D:\webapp
kite_tutorial.py
index.html
app.py - D:\...\IISPS-INT-3587-Garbage-Classification-master
main.css
styling files
main.js
html
garbage1.html
import numpy as np
import os
from keras.models import load_model
from keras.preprocessing import image
import tensorflow as tf
global graph
from flask import Flask , request, render_template
from werkzeug.utils import secure_filename
from gevent.pywsgi import WSGIServer
graph = tf.get_default_graph()
import theano
theano.config.optimizer="None"
os.environ['THEANO_FLAGS'] = 'optimizer=None'
app = Flask(__name__)
model = load_model("garbagenew.h5")
@app.route('/')
def index():
return render_template('garbage1.html')
@app.route('/predict',methods = ['GET','POST'])
def upload():
if request.method == 'POST':
f = request.files['image']
print("current path")
basepath = os.path.dirname(__file__)
print("current path", basepath)
filepath = os.path.join(basepath,'uploads',f.filename)
print("upload folder is ", filepath)
f.save(filepath)
img = image.load_img(filepath,target_size = (64,64))
x = image.img_to_array(img)
x = np.expand_dims(x,axis =0)
with graph.as_default():
preds = model.predict_classes(x)
print("prediction",preds)
index = ['Cardboard','Glass', 'Metal', 'Paper', 'Plastic', 'Trash']
text = "The type is "+index[preds[0]]
return text
if __name__ == '__main__':
app.run(debug=True, threaded=False)
Usage
Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.
New to Spyder? Read our tutorial
Variable explorer
Help
Plots
Files
Console 1/A
Python 3.8.3 (default, Jul 2 2020, 17:30:36) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.
IPython 7.16.1 -- An enhanced Interactive Python.
In [1]:
Activate Windows
Go to Settings to activate Windows.
IPython console
History
LSP Python: ready
Kite: indexing
conda: base (Python 3.8.3)
Line 10, Col 1
ASCII
CRLF
RW
Mem 60%
Type here to search
11:30 AM
8/26/2020

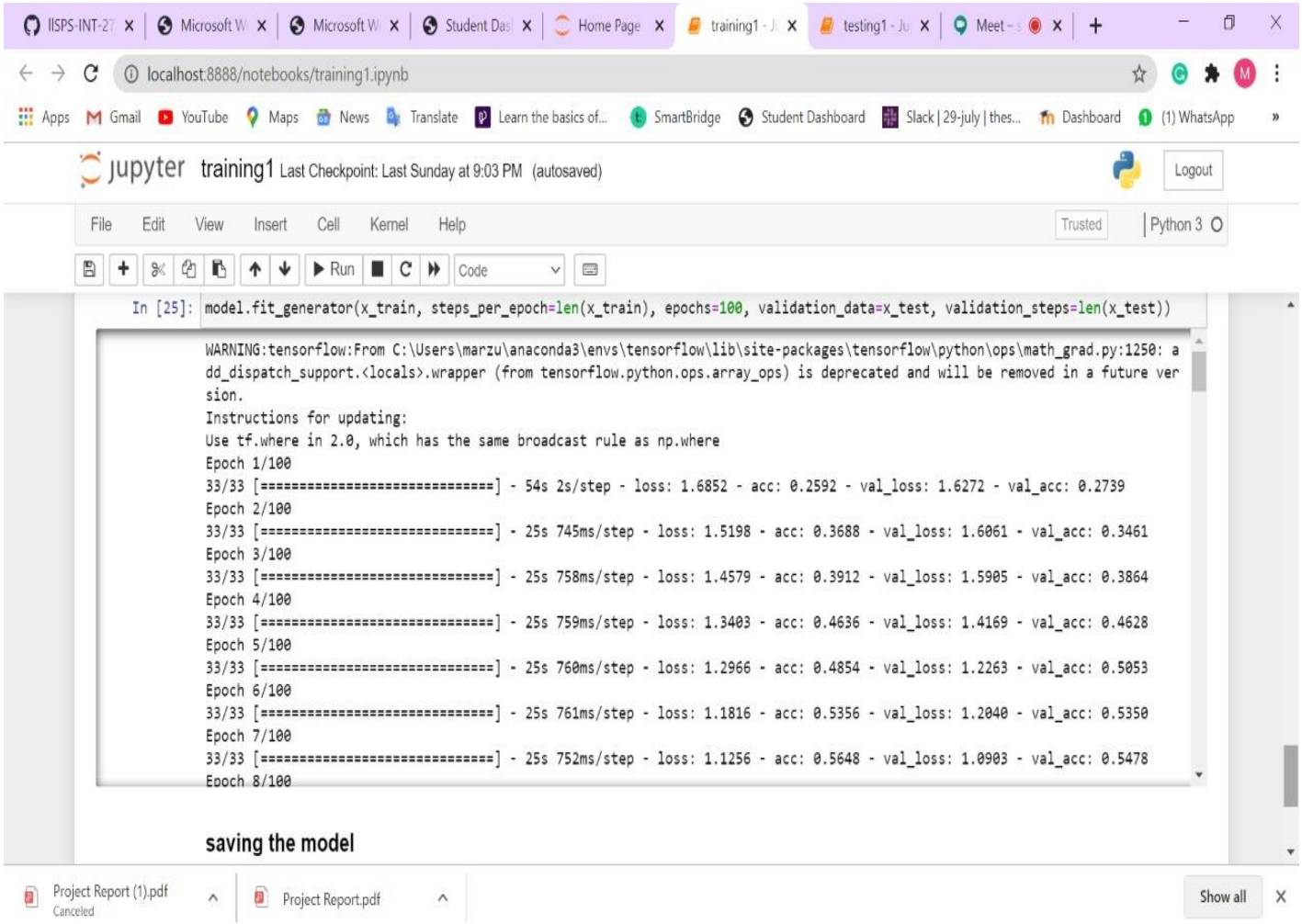
5. Training.ipynb

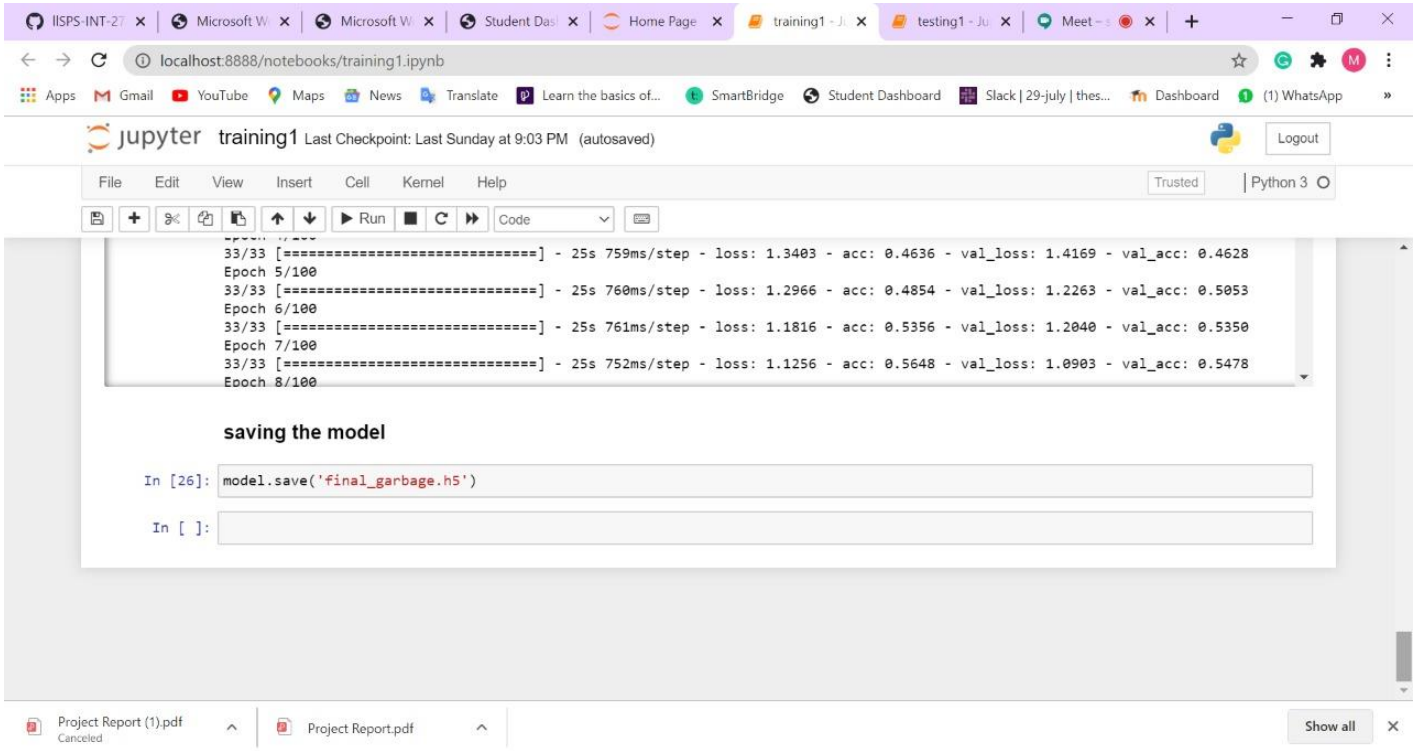




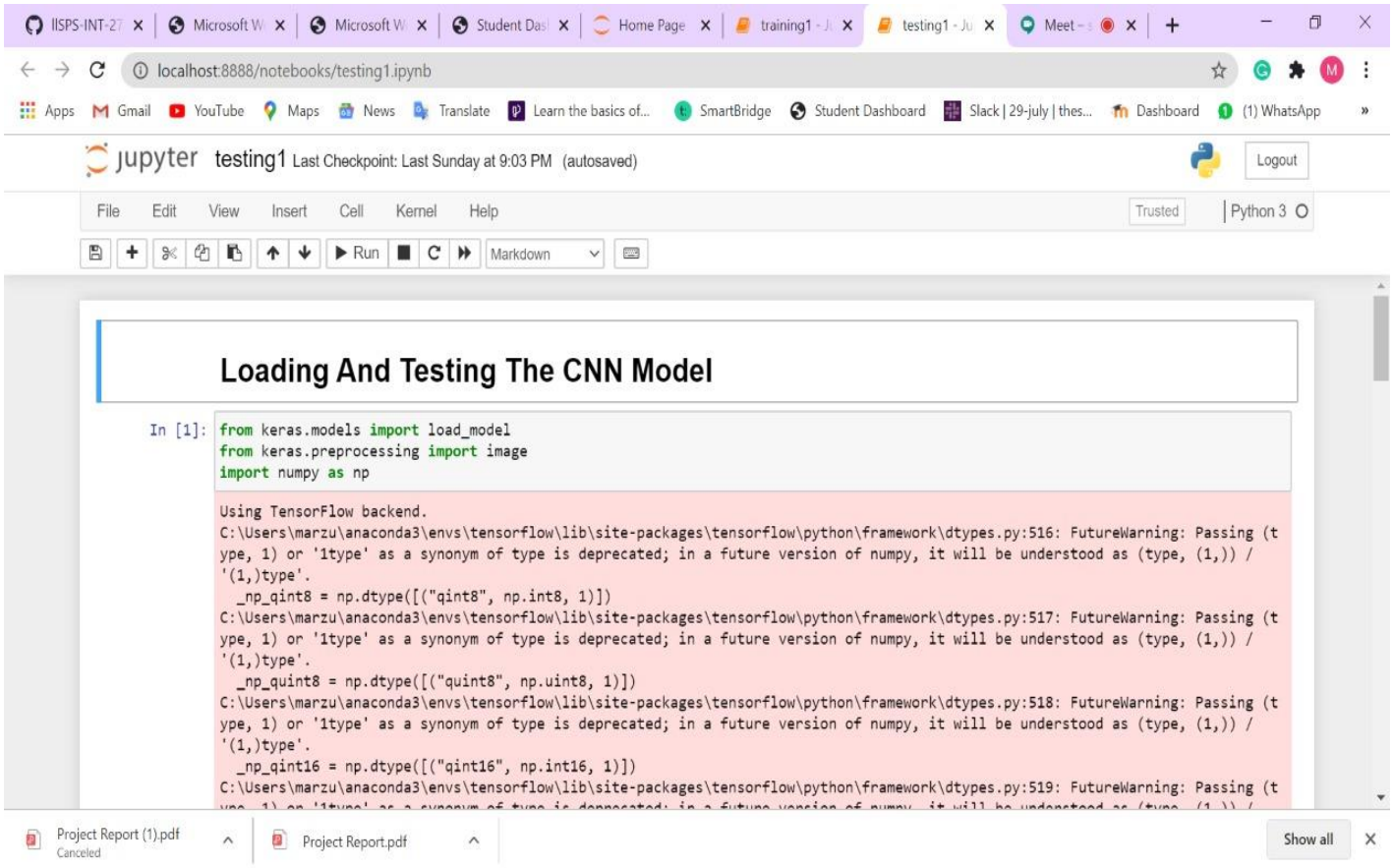

localhost:8888/notebooks/training1.ipynb
jupyter training1 Last Checkpoint: Last Sunday at 9:03 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Help
Trusted
Python 3
Run
Markdown
In [2]:
from keras.preprocessing.image import ImageDataGenerator
train_datagen = ImageDataGenerator(rescale = 1./255, shear_range=0.2, zoom_range=0.2, horizontal_flip=True)
test_datagen = ImageDataGenerator(rescale = 1./255)
In [3]:
x_train = train_datagen.flow_from_directory(r'D:\smartintern_garbage_class\data_set\train', target_size=(64,64), batch_size=64,
Found 2056 images belonging to 6 classes.
In [4]:
x_train.class_indices
Out[4]:
{'Cardboard': 0, 'Glass': 1, 'Metal': 2, 'Paper': 3, 'Plastic': 4, 'Trash': 5}
In [7]:
x_test = test_datagen.flow_from_directory(r'D:\smartintern_garbage_class\data_set\Test', target_size=(64,64), batch_size=64, clas
Found 471 images belonging to 6 classes.
In [8]:
x_train.image_shape
Out[8]:
(64, 64, 3)
Project Report (1).pdf
Canceled
Project Report.pdf
Show all

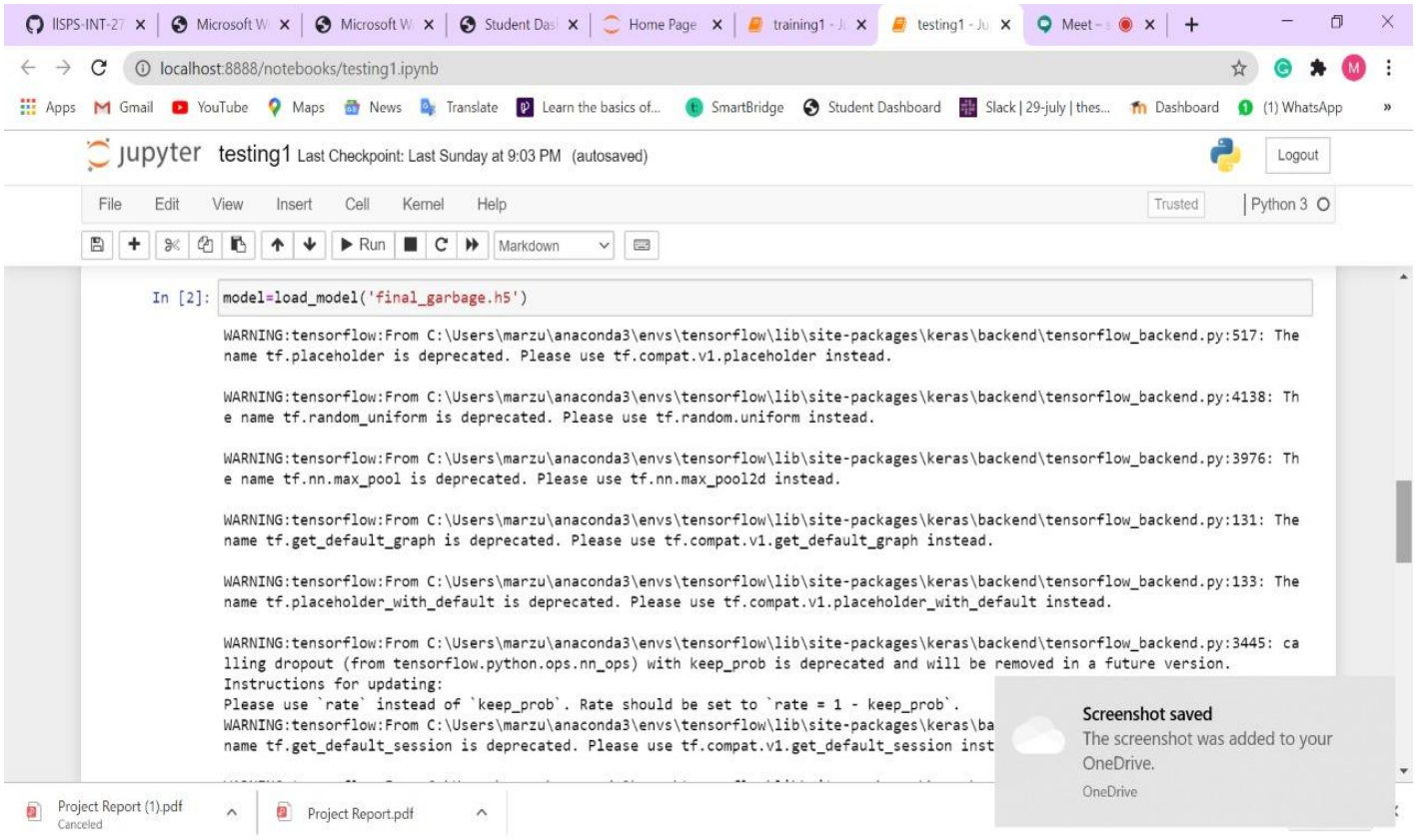

localhost:8888/notebooks/training1.ipynb
jupyter training1 Last Checkpoint: Last Sunday at 9:03 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Help
Trusted
Python 3
Run
Code
In [20]:
model.add(Flatten())
In [21]:
model.add(Dense(output_dim=512, activation='relu'))
C:\Users\marzu\anaconda3\envs\tensorflow\lib\site-packages\ipykernel_launcher.py:1: UserWarning: Update your `Dense` call to th
e Keras 2 API: `Dense(activation="relu", units=512)`
"""Entry point for launching an IPython kernel.
In [22]:
model.add(Dense(output_dim= 6,activation='softmax'))
C:\Users\marzu\anaconda3\envs\tensorflow\lib\site-packages\ipykernel_launcher.py:1: UserWarning: Update your `Dense` call to th
e Keras 2 API: `Dense(activation="softmax", units=6)`
"""Entry point for launching an IPython kernel.
In [23]:
model.compile(loss='categorical_crossentropy', optimizer ='adam', metrics= ['accuracy'])
WARNING:tensorflow:From C:\Users\marzu\anaconda3\envs\tensorflow\lib\site-packages\keras\optimizers.py:790: The name tf.train.O
ptimizer is deprecated. Please use tf.compat.v1.train.Optimizer instead.
WARNING:tensorflow:From C:\Users\marzu\anaconda3\envs\tensorflow\lib\site-packages\keras\backend\tensorflow_backend.py:3295: Th
e name tf.log is deprecated. Please use tf.math.log instead.
Project Report (1).pdf
Canceled
Project Report.pdf
Show all

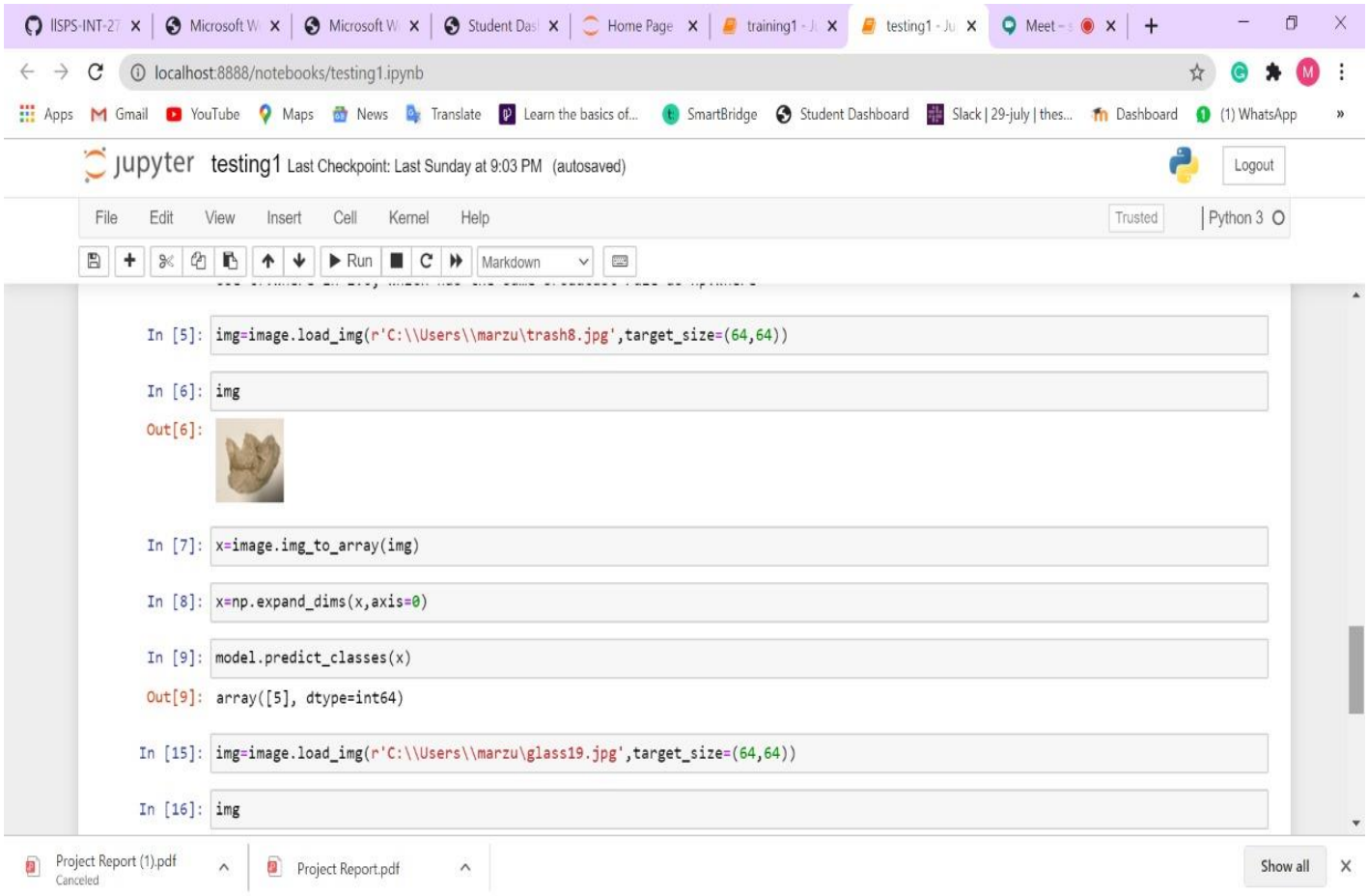

localhost:8888/notebooks/training1.ipynb
jupyter training1 Last Checkpoint: Last Sunday at 9:03 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Help
Trusted
Python 3
Run
Code
In [25]: model.fit_generator(x_train, steps_per_epoch=len(x_train), epochs=100, validation_data=x_test, validation_steps=len(x_test))
WARNING:tensorflow:From C:\Users\marzu\anaconda3\envs\tensorflow\lib\site-packages\tensorflow\python\ops\math_grad.py:1250: add_dispatch_support.<locals>.wrapper (from tensorflow.python.ops.array_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
Epoch 1/100
33/33 [==============================] - 54s 2s/step - loss: 1.6852 - acc: 0.2592 - val_loss: 1.6272 - val_acc: 0.2739
Epoch 2/100
33/33 [==============================] - 25s 745ms/step - loss: 1.5198 - acc: 0.3688 - val_loss: 1.6061 - val_acc: 0.3461
Epoch 3/100
33/33 [==============================] - 25s 758ms/step - loss: 1.4579 - acc: 0.3912 - val_loss: 1.5905 - val_acc: 0.3864
Epoch 4/100
33/33 [==============================] - 25s 759ms/step - loss: 1.3403 - acc: 0.4636 - val_loss: 1.4169 - val_acc: 0.4628
Epoch 5/100
33/33 [==============================] - 25s 760ms/step - loss: 1.2966 - acc: 0.4854 - val_loss: 1.2263 - val_acc: 0.5053
Epoch 6/100
33/33 [==============================] - 25s 761ms/step - loss: 1.1816 - acc: 0.5356 - val_loss: 1.2040 - val_acc: 0.5350
Epoch 7/100
33/33 [==============================] - 25s 752ms/step - loss: 1.1256 - acc: 0.5648 - val_loss: 1.0903 - val_acc: 0.5478
Epoch 8/100
saving the model
Project Report (1).pdf
Canceled
Project Report.pdf
Show all

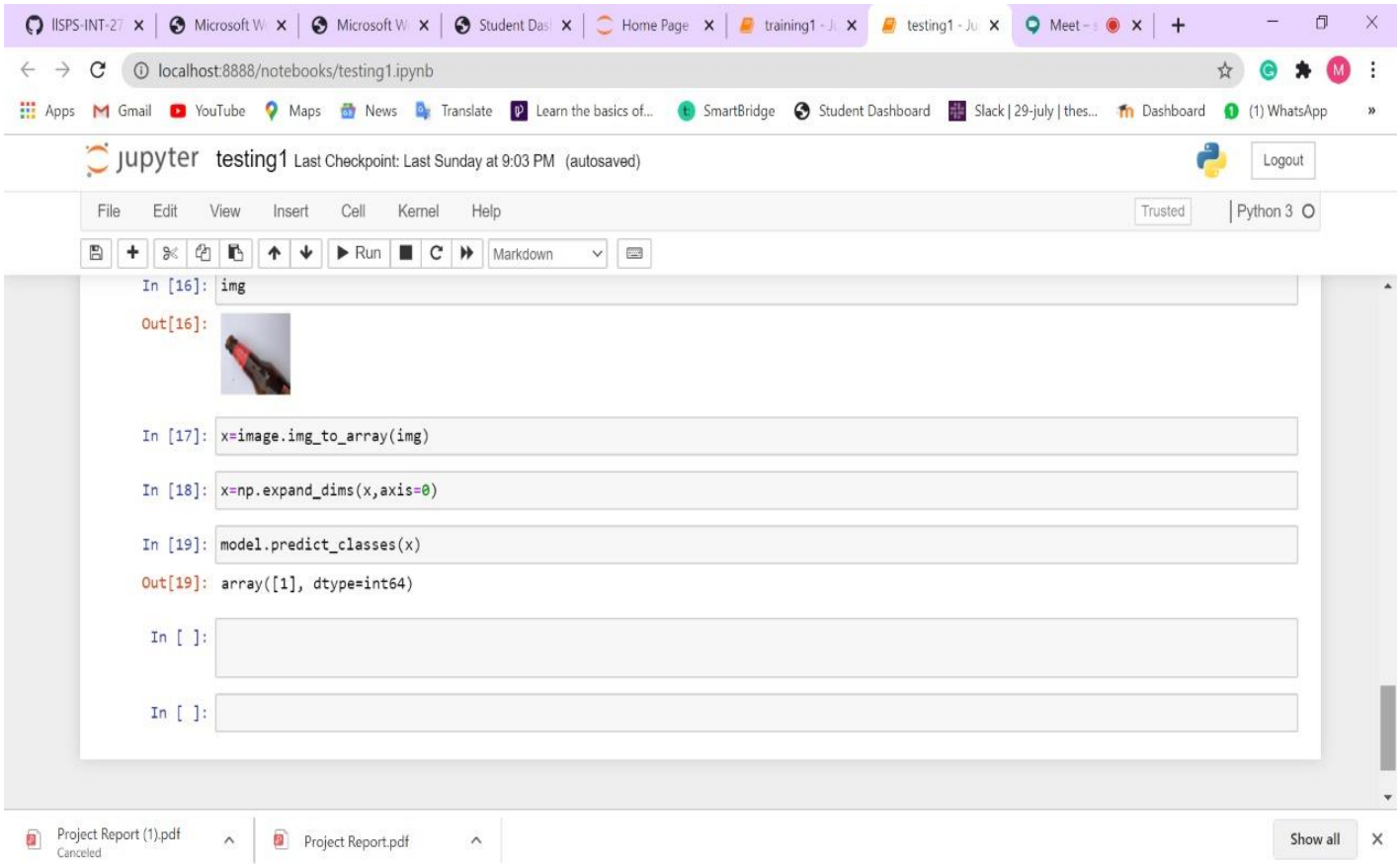

localhost:8888/notebooks/training1.ipynb
jupyter training1 Last Checkpoint: Last Sunday at 9:03 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Help
Trusted
Python 3
Run
Code
33/33 [==============================] - 25s 759ms/step - loss: 1.3403 - acc: 0.4636 - val_loss: 1.4169 - val_acc: 0.4628
Epoch 5/100
33/33 [==============================] - 25s 760ms/step - loss: 1.2966 - acc: 0.4854 - val_loss: 1.2263 - val_acc: 0.5053
Epoch 6/100
33/33 [==============================] - 25s 761ms/step - loss: 1.1816 - acc: 0.5356 - val_loss: 1.2040 - val_acc: 0.5350
Epoch 7/100
33/33 [==============================] - 25s 752ms/step - loss: 1.1256 - acc: 0.5648 - val_loss: 1.0903 - val_acc: 0.5478
Epoch 8/100
saving the model
In [26]: model.save('final_garbage.h5')
In [ ]:
Project Report (1).pdf
Canceled
Project Report.pdf
Show all

6. Testing .ipynb






localhost:8888/notebooks/testing1.ipynb
jupyter testing1 Last Checkpoint: Last Sunday at 9:03 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Help
Trusted
Python 3
Run
Markdown
In [5]: img=image.load_img(r'C:\\Users\\marzu\trash8.jpg',target_size=(64,64))
In [6]: img
Out[6]:
In [7]: x=image.img_to_array(img)
In [8]: x=np.expand_dims(x,axis=0)
In [9]: model.predict_classes(x)
Out[9]: array([5], dtype=int64)
In [15]: img=image.load_img(r'C:\\Users\\marzu\glass19.jpg',target_size=(64,64))
In [16]: img
Project Report (1).pdf
Canceled
Project Report.pdf
Show all


localhost:8888/notebooks/testing1.ipynb
jupyter testing1 Last Checkpoint: Last Sunday at 9:03 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Help
Trusted
Python 3
Run
Markdown
In [16]: img
Out[16]:
In [17]: x=image.img_to_array(img)
In [18]: x=np.expand_dims(x,axis=0)
In [19]: model.predict_classes(x)
Out[19]: array([1], dtype=int64)
In [ ]:
In [ ]:
Project Report (1).pdf
Canceled
Project Report.pdf
Show all