**E-WASTE MANAGEMENT SYSTEM**

**INTRODUCTION:**

One of the major problems in all the countries across the globe is the e-waste which is otherwise called electronic waste.The Electrical and Electronic Equipment (EEE) is strongly linked to global economic development. Higher levels of disposable incomes, growing urbanization and mobility, and further industrialization in some parts of the world are leading to growing amounts of EEE. After its use, EEE is disposed of, generating a waste stream that contains hazardous and valuable materials. This waste stream is referred to as E-waste, or Waste Electrical and Electronic Equipment (WEEE).

Hear the idea is to collect and separate all the components from the motherboard which is working and not working one. Working components are used in another devices and not working components are used for recycling to get precious materials. Here we are taking motherboard components.

**Purpose:**

1. Collect the devices which are not used by the user
2. Separate all the components and categorized all(working and not working)
3. Working components are reused in other devices
4. Not working components & materials are send it to recycle
5. Extracting the precious materials from the components.

**LITERATURE SURVEY:**

**TECHNOLOGY IMPLEMENTATION:**

**Technical Information**

* Algorithm used – Convolutional Neural Network.
* Technology used – Deep Learning i.e., Neural Network.
* Key Frameworks used – Tensorflow, Keras.
* Libraries and Functions used – Image Data Generator, Sequential, Dense, Convolution2D, MaxPooling2D, Flatten.
* Activation Functions used – ReLU, Softmax.
* Optimizer used – Adam.
* Loss Function used – Categorical Cross entropy

**Editors Requirement**

* Visual Studio Code
* Spyder

**Languages and Software Required:**

* Jupyter Notebook
* Deep Learning Algorithms
* HTML
* Flask
* Python(Keras, Tensorflow)

We developed this E-Waste management prediction by using the Python language which is a interpreted and high level programming language and using the Deep Learning algorithms. for coding we used the Jupyter Notebook environment of the Visual Studio Code / Anaconda distributions and the Spyder, it is an integrated scientific programming in the Loading the pre-processed data adding CNN Layers Configure the learning Process Adding Dense Layers Optimize the model Save the Model Train and Test the Model application building with HTML and Flask.

For creating a user interface for the prediction we used the Flask. It is a micro web framework written in Python and uses WSGI for web development. It is classified as a micro framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions, and a scripting language to create a webpage is HTML by creating the templates to use in the functions of the Flask and HTML.

**Proposed Solution:**

In vision of the problem statement described in the introduction section, a CNN model is proposed with boosted accuracy to AI Enabled E-Waste management Recogination System.The framework is composed of the following important phases:

* Dataset Collection (creating training and testing folders)
* Data Pre-processing Model Building
* Achieving trained model with highest accuracy
* Using trained model for prediction
* Application Building

Classification is principally done by making predictions based on known sample data that has been learned from training data. Designed algorithm is first trained on the known data labels and further uses this learning to predict the class labels for the new unknown set of data sample. The classification objective set for this study is to achieve enhanced accuracy by using ImageDataGenerator classifiers. We train the classifier with known sample data in a training dataset and check its performance by examining the test.

The Proposed work is to focus on the detection of working and not working components in the motherboard. Identification of components in images and Location of the component on the motherboard based on the position. Based on erosion followed by dilation segmentation algorithm. This algorithm can detect components and also classify it. Currently the algorithm is testing on four types of components.

**Theoretical Analysis:**

Data Collection

Data Pre-Processing

Configure images Data Generator class

Import the Image Data Generator library

Adding Dense Layers

Configure the learning Process

Application building with HTML and Flask

Save the Model

Optimize the model

Train and Test the Model

Adding CNN Layers

Initializing the model

Importing the model Building libraries

Model Building

Apply image Data Generator class to train and test set

Loading the pre-processed data

**Training Process:**

Initially for Training the data four motherboard components are taken and each component is again classified into damaged and undamaged which has some set of images given for training. The current components given for training are ATX Connector, CMOS Battery, IO Ports, and Processor. Secondly the ImageDataGenerator is used to augment the images and we can apply any random transformations on each training image.

A library such as Dense is used to create a new layer. Functions such as Convolution2D used to create a Convolutional layer(to convert image into an array) , MaxPooling2D is used to create a pooling layer(to reduce image size by using filter), Flatten is used to convert the pooled feature map to a single column that is passed to the fully connected layer.

**Flow Chart:**

Start

Upload an image

Test

Not Working category

Working category

Recycle

Extract

**OUTPUT:**

The expected output is choose the image which should be tested by our algorithm and detect whether is it working or non-working component of the motherboard.

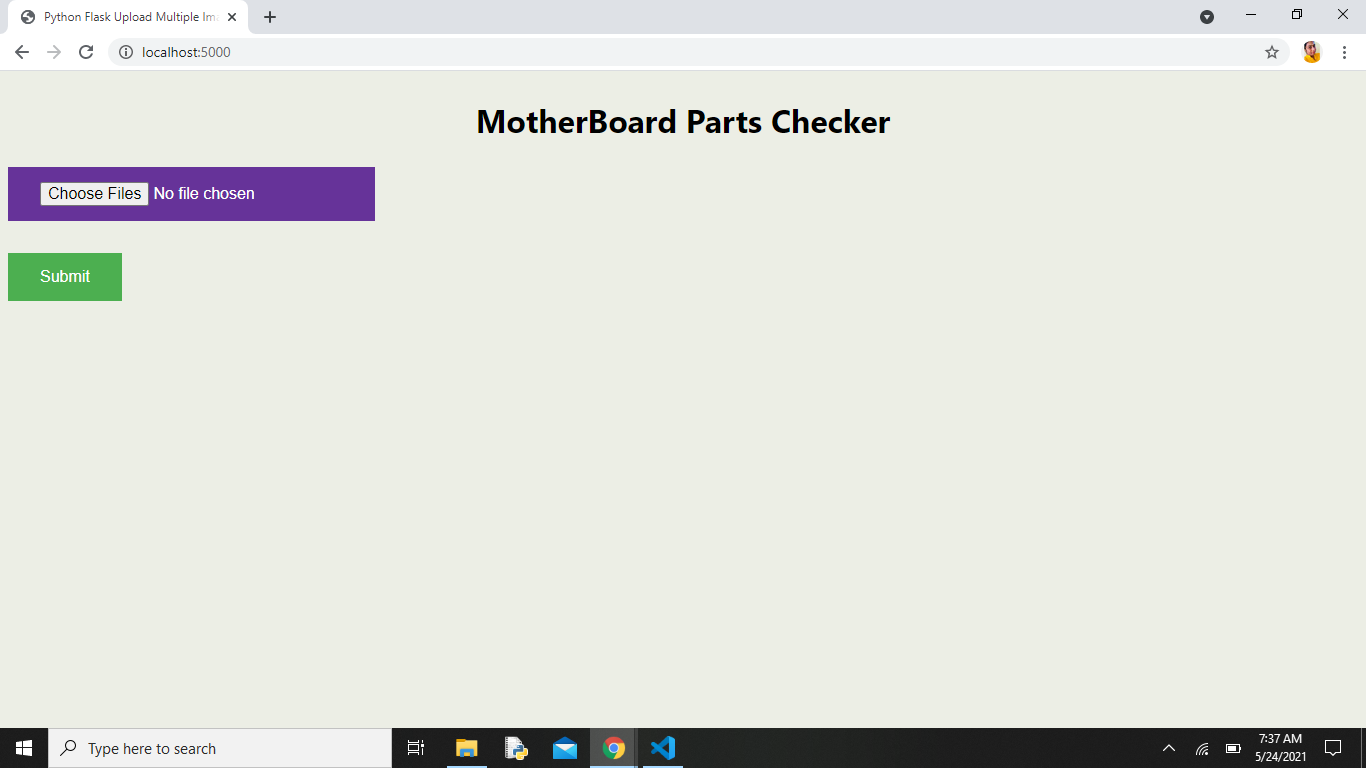


Figure 1

* The Figure 1 is the initial page of the output.

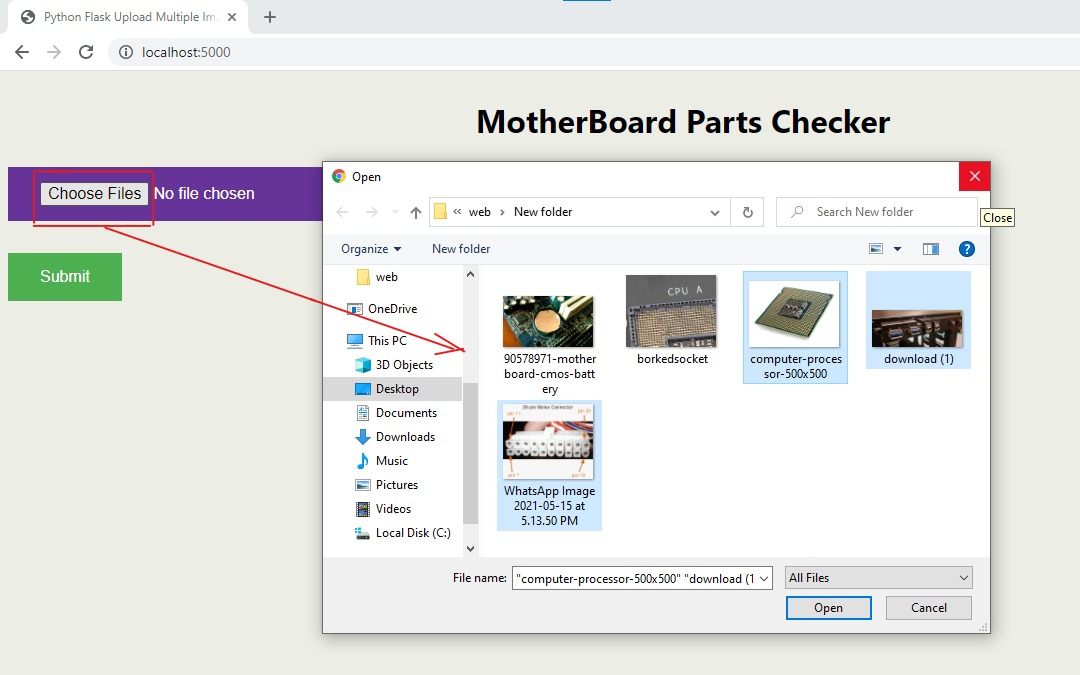


Figure 2

* Choose the file (Component of motherboard to be tested) as shown in figure 2.

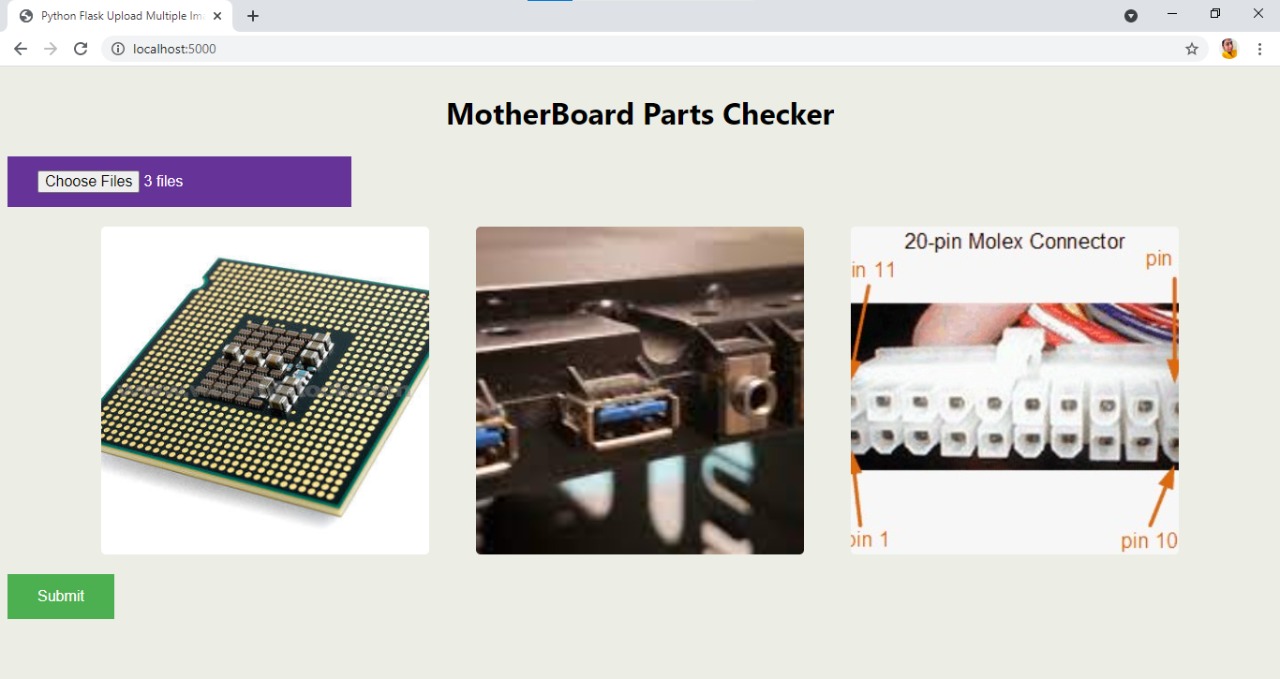


Figure 3

* The respective chosen images/ files are displayed on the web page as shown in figure 3, after submitting these images/files that have been chosen will be tested to make analysis that whether the image/file given is working or non-working component of the motherboard.

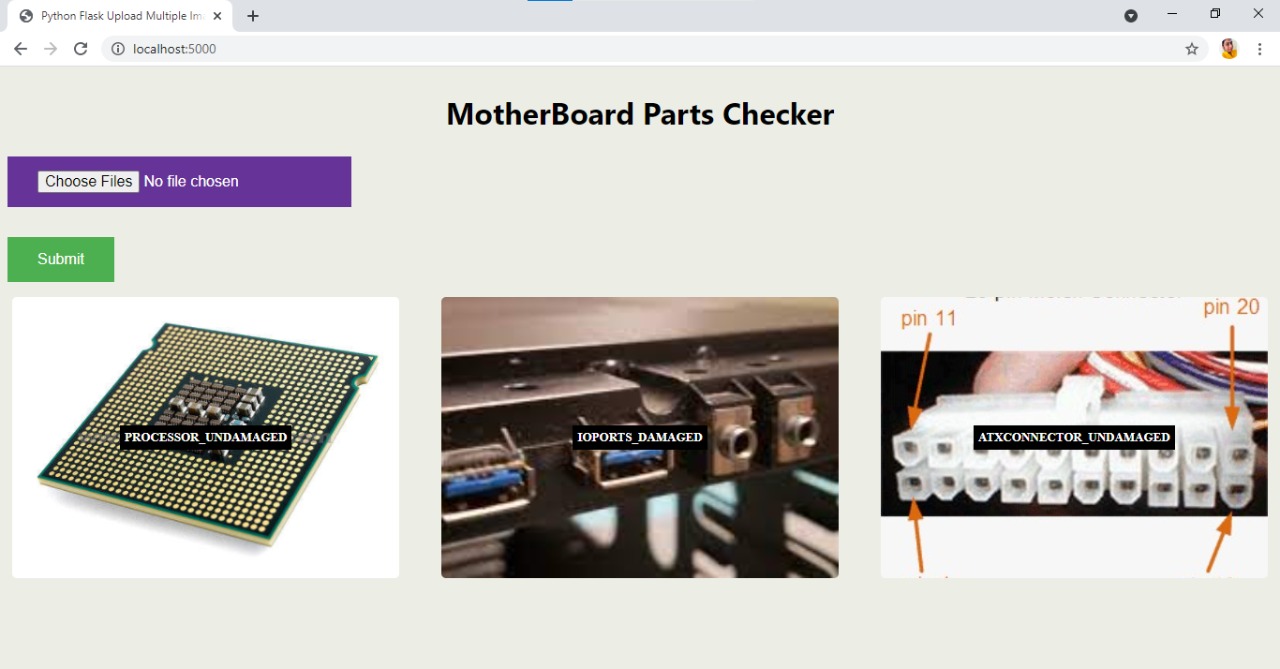


Figure 4

* After submitting the files i.e. after testing happens, we can see the output that the component is working or non-working(damaged/un-damaged) of the motherboard as shown in figure 4.

**FUTURE ENHANCEMENT:**