

**Kingdom of Saudi Arabia**

**Ministry of Education**

**King Faisal University**

**College of Computer Sciences & Information Technology**

**Multi agents Project**

**"Traffic Sign Recognition"**



**Section 60**

**By Group 2**

|  |  |
| --- | --- |
| **Name** | **ID** |
| **Norah Almarri** | **217041639** |
| **Rabab albeladi** | **217027224** |
| **Asalah Albreih** | **217027174** |

**Instructor:** Dr. Hafiz Farooq Ahmed

**April, 2021**

**Contents**

[1. Project Description 3](#_Toc68639625)

[2. Software Requirements Specification (SRS) 3](#_Toc68639626)

[3. Project Methodology 4](#_Toc68639627)

[4. API Used for implementation 4](#_Toc68639628)

# **Project Description**

An integral part of our road infrastructure is traffic signs which are represented by pictures, where each image expresses the rules and information that the driver and road users must adhere to, and traffic lights use pictures instead of words to make them easier to understand by all people. They contribute to preserving the safety of life, regulating traffic and pedestrians, avoiding the expenses of repairing damage, avoiding court procedures and legal issues. We would most likely be faced with further incidents without such helpful signs, as drivers would not be given crucial guidance about how easily they could go safely or told about road works, sharp turns, or school crossings ahead. Traffic signs recognition has a very important role to play in self-driving cars, which is the automotive industry's future [1]. In our project, we want to build a program using the AI methodology and Python language to recognize the traffic signs on the roads and classified them correctly.

# **Software Requirements Specification (SRS)**

* The system shall be able to give an accurate result.
* The system shall be able to upload images from the desktop directory.
* The system shall be able to recognize the images of traffic signs using CNN method.

# **Project Methodology**

3.1 Class diagram

Timeline

Description automatically generated

Figure : Class diagram

# **API Used for Implementation**

Our project which is titled Traffic Sign Recognition will consist of two classes the first one is the main class that includes created the module and applied the CNN algorithm, and the second class is GUI which we can run the code from this interface. Now, we will show each class separately to explain their packages and why we were needed to implement this project.

* **Main class**

In the main class, we typed packages that we need to explore the dataset, build a CNN model, and the following will explain their use in our project:

* Explore the dataset

We need *pandas* library to import and read the dataset, *image* package we used to open the traffic sign image into an array, *numpy* which is a python library used to convert the list into arrays for feeding to the model, *train\_test\_split* library used to split the dataset into random train and test and because in ‘train’ folder that contains 43 folders each representing a different class. So, we need to iterate over all those classes and append images and their respective labels in the data and labels list, with the help of the *OS* library. After we extracted the image path and labels using pandas and from the *sklearn.metrics*, we imported the *accuracy\_score* and observed how our model predicted the actual labels. Then, we used *matplotlib.pyplot* package for plotting graphs of accuracy and loss as shown in **Figure** **3**.

* Build a CNN model we need

Convolutional Neural Network is best for image classification purposes and we used this model to classify the images into their respective categories by using *Sequential, load\_model*, and from the *keras.utils* package, we use *to\_categorical* method to convert the labels present in y\_train and t\_test into one-hot encoding. As well we have multiple classes to categorize so we use *Conv2D, MaxPool2D, Dense, Flatten, Dropout* package.

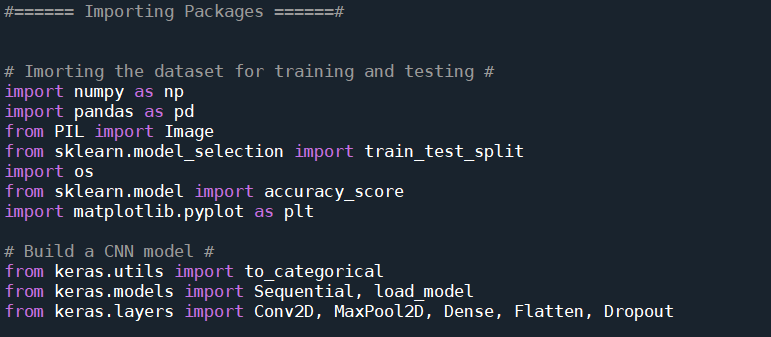


Figure : Main Class packages

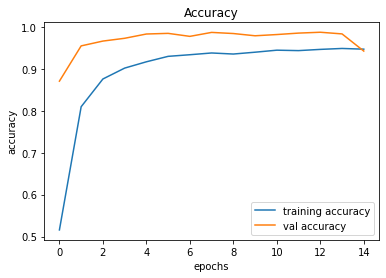
****

Figure : The accuracy graph

* **GUI**

We built a graphical user interface for our traffic signs classifier with Tkinter which is a GUI toolkit in the standard python library, and we use this tool in *tkinter* package. And then, we used *from tkinter import \** for build the GUI. *from tkinter import filedialog* librarywhich gave us a set of unique dialogs to be used when dealing with files. After that, we used *from PIL import ImageTk, Image* packageto upload the image and a button that used in GUI. Finally, *load\_model* package helped us to load the trained model in order to classify traffic signs.

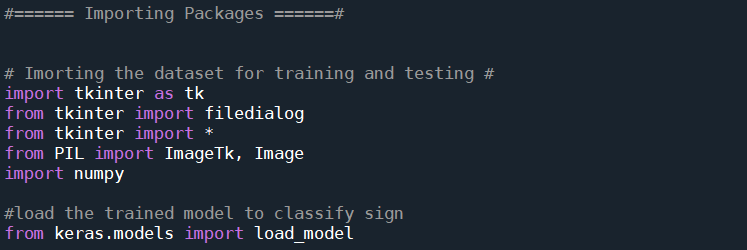


Figure : GUI packages

References:

[1] Vagadia, H. (2020, July 20). German Traffic Sign Recognition Benchmark. Retrieved January 31, 2021, from https://medium.com/analytics-vidhya/german-traffic-sign-recognition-benchmark-5477ca13daa0