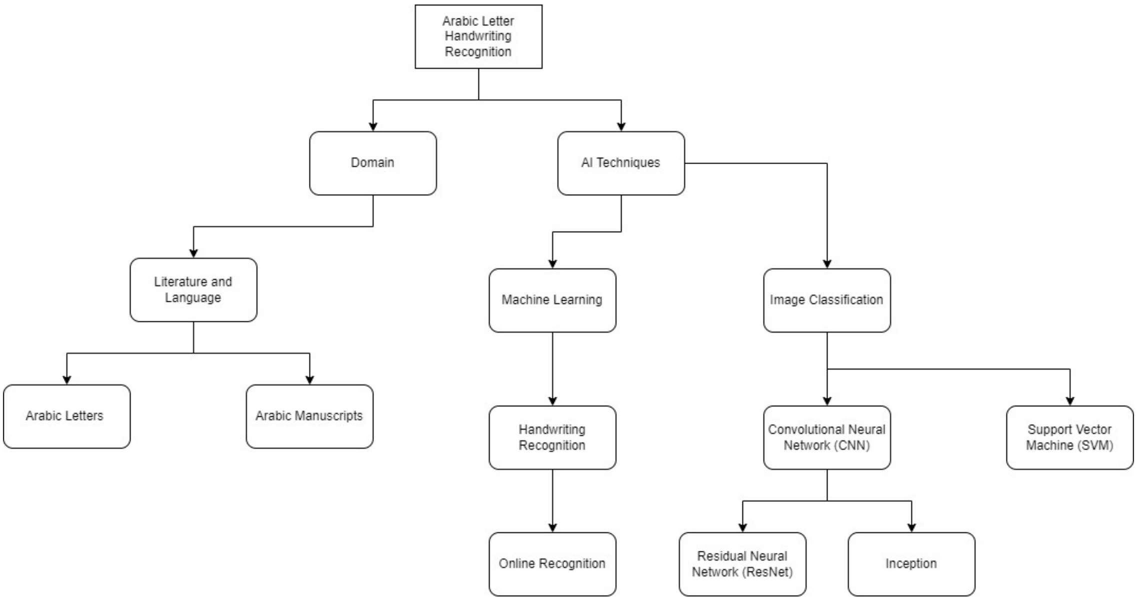
CHAPTER 3

RESEARCH METHODOLOGY

The study technique and the procedures used to carry out the project's goals are covered in this chapter. The study methodology discussed and described how and how the project was completed in accordance with the project's objectives. The stages of the research structure are preliminary studies, knowledge gathering and acquisition, data collection, data preprocessing, model development, model design, model evaluation, system design, system development, system testing and evaluation, and the last stage is the documentation of the final report.



3.1 OUTLINE OF RESEARCH

This research involves four phases with their activities. Phase 1 is the process for the first objective where it includes the preliminary studies, knowledge gathering and acquisition, data collection, data preprocessing. Next, phase 2 is the process for the second objective where phase 2 describes the system design, knowledge representation, system development. The last phase, which is phase 3 is where the system is tested, and evaluation followed by the documentation of the final year report.

3.2 RESEARCH DESIGN

The research design is the framework for all of the research approaches and techniques a researcher chooses to use to conduct a study. The design makes it possible for researchers to concentrate on creating research methods relevant to the subject and sets up our studies for success. For each of the many elements, it has a specific objective, phase, activities, and deliverables that should be worked on.

Research Objectives	**Phases**	**Activities**	**Deliverables**
To identify the features and handwriting styles of every Arabic letter	Preliminary study	Read and understand articles regarding the methods that are used to implement for Arabic letter recognition	• Background study • Problem Statement • Research questions • Project Scope • Significance
	Knowledge gathering and acquisition	Gaining knowledge and accumulate the results gain from articles on the preferred methods	• Literature Review
	Data Collection	Search for Arabic letters dataset and collect data of images of different handwriting styles from different people through Kaggle	• Data of Arabic letters and images of different handwriting styles
	Data Preprocessing	Transforming images of different handwritten Arabic letters by classification, splitting, converting class vector, reshaping, resizing, and data augmentation.	• Processed handwritten Arabic letters

To design a machine learning model through CNN for the Arabic Letter Handwritten Recognition system	Model Development	• Choosing CNN as the machine learning model for Arabic letter recognition. • Splitting the dataset into training and validation sets. Training selected model using the training data. Use the validation data to tune hyperparameters and prevent overfitting. • Fine-tune the model by adjusting hyperparameters such as learning rate, batch size, and network architecture to achieve better performance.	• Documentation of the hyperparameter tuning process, listing the selected hyperparameters and the performance impact on the model.
	Model Design	• Design a Hybrid Convolutional Neural Network and SVM model for image processing. • Design the model with different types of transfer learning which are Resnet50 and InceptionV3	• Model of the Arabic letter Recognition system
	Model Evaluation	• Evaluate the trained model on a separate test dataset to	• A detailed evaluation report showcasing the model's

		assess its performance. • The models are compared to see the best accuracy among them.	performance metrics on the test dataset, including accuracy, precision, training and validation graphs, recall, F1-score, and confusion matrix.
To develop Arabic letter handwriting recognition prototype that can accurately recognize and classify Arabic letters.	System Development	• Develop a system for the prototype through python	• Prototype of the Arabic Handwritten letter recognition system
	System Design	• Designing an interface to obtain written letters that are drawn live and transformed into text form	• UI Design and system interface design through Python • System Architecture
	System testing and evaluation	• Testing the prototype • Test the models and compare the accuracy of the methods. • Test the functionality of the project by prototyping the actual dashboard of interface.	• Testing and result evaluation
	Documentation	Writing the report	• Final year report

Table 3.1: Project Phase Timeline

3.3 PHASE 1

Phase 1 focused to identify the features and handwriting styles of every Arabic letter. Phase 1's goal is to recognise Arabic letters characteristics. Phase 1 activities include preliminary study, knowledge acquisition, data collecting, and pre-processing of the data.

3.3.1 PREMILINARY STUDY

Preliminary study has been done to help gain a better knowledge of the materials that are actually present and the opinions that are being expressed that is also known as preliminary research. This kind of research has enlarged or helped me narrow down topics regarding Arabic Handwritten Letter Recognition and strengthening it. Using the data acquired, it has provided me with better keywords. Preliminary research, sometimes referred to as the first step in the research process, evaluates the entire project by taking into account its context, defining the issue, and observing and comprehending how Arabic Letter Handwritten Recognition is related to other topics such as Optical Character Recognition, digit recognition, and handwriting recognition which gave access to the data that is needed to effectively plan and carry out the research. Additionally, it provides background information about the key subjects that is essential for assessing the research's progress. It also helps in decision-making and background research for the subtopics in letter recognition. The preliminary study has been helpful in determining the project's background study, problem statement, scope, research questions, and significance. In order to stay updated of the most recent developments in the field of Arabic letter recognition, additional online study was done.

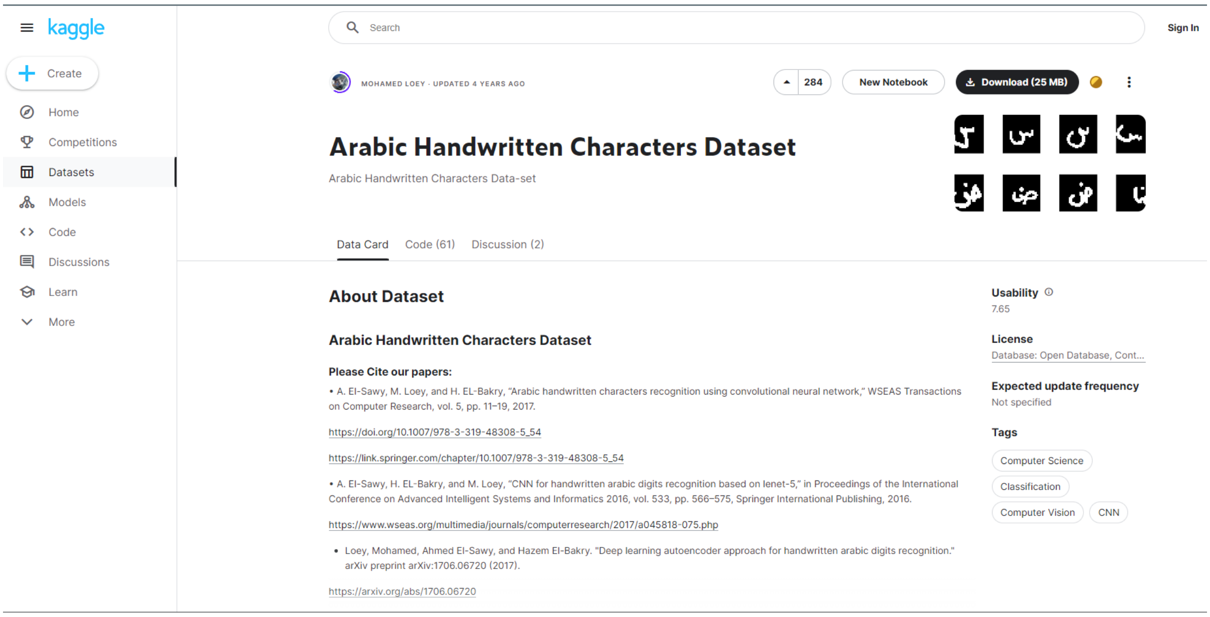
3.3.2 KNOWLEDGE GATHERING AND ACQUISITION

The process of extracting, categorising, and analysing data from a single source typically a human expert is known as knowledge acquisition. However, for this project, knowledge acquisition was done using Google Scholar, IEEE Xplore, Scopus, Web of Science, and other open-source articles. The issue was identified, together with its goal, range, and importance in connection to the associated topic, which is letter recognition, by applying the literature analysis technique. It refers to the process of gathering all the facts and information.

Earlier studies conducted by other analysts were examined in the evaluation of the literature, which has helped to further understand the requirements in related subtopics of Arabic Handwritten Letter Recognition. Since there are numerous ways to frame through findings and address each issue, this phase enabled the acquisition of new knowledge. The methods and approaches, or method algorithms, that has been chosen and used on this project will be supported by this phase based on their efficiency and accuracy in the development of the models. The analysis of the project, method, and approaches has also been determined based on how they are described in the previous research that are done related to letter recognition and on how to carry out each of the procedures in developing the model for Arabic Handwritten Letter Recognition. To better understand how to select the best method for letter recognition, research is also conducted using the same technique in various domains. This was accomplished by reviewing related deep learning works on letter recognition. In conclusion, CNN has been extensively utilised for image recognition in order to categorise each image into a unique class. Consequently, CNN is selected to be used in conjunction with SVM in order to increase accuracy for this project. The CNN model is used to categorise Arabic letter images into their respective classes once it has been trained.

3.3.3 DATA COLLECTION

In the data collection phase, the data on Arabic letters was searched and taken from a website named Kaggle as shown in Figure 3.1. Under Google LLC, Kaggle is a platform for data science competitions and an online community for data scientists and machine learning practitioners. Users can discover and publish datasets on Kaggle, explore and develop models in a web-based data science environment, collaborate with other data scientists and machine learning experts, and participate in competitions to address data science challenges. The dataset contains 16,800 letters written by 60 participants, the age range is between 19 to 40 years, and 90% of participants are right-hand. The database is partitioned into two sets: a training set (13,440 letters to 480 images per class) and a test set (3,360 letters to 120 images per class). Writers of training set and test set are exclusive. These images have 32 × 32 resolution pixels and are in the Portable Network Graphics (PNG) format.



Arabic Handwritten Characters Dataset

Arabic Handwritten Characters Data-set

About Dataset

Arabic Handwritten Characters Dataset

Please Cite our papers:

• A. El-Sawy, M. Loey, and H. EL-Bakry, "Arabic handwritten characters recognition using convolutional neural network," WSEAS Transactions on Computer Research, vol. 5, pp. 11–19, 2017.

https://doi.org/10.1007/978-3-319-48308-5_54

https://link.springer.com/chapter/10.1007/978-3-319-48308-5_54

• A. El-Sawy, H. EL-Bakry, and M. Loey, "CNN for handwritten arabic digits recognition based on lenet-5," in Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2016, vol. 533, pp. 566–575, Springer International Publishing, 2016.

https://www.wseas.org/multimedia/journals/computerresearch/2017/a045818-075.php

- Loey, Mohamed, Ahmed El-Sawy, and Hazem El-Bakry. "Deep learning autoencoder approach for handwritten arabic digits recognition." arXiv preprint arXiv:1706.06720 (2017).

https://arxiv.org/abs/1706.06720

Usability
7.65

License
Database: Open Database, Cont...

Expected update frequency
Not specified

Tags
Computer Science
Classification
Computer Vision
CNN

Figure 3.1: Screenshot of Arabic letter dataset

3.3.4 DATA PREPROCESSING

Data pre-processing is the final phase 1 step that is covered. To train AI and machine learning models and run inferences against them, data preparation techniques have lately undergone modifications. Data preprocessing transforms raw data into information so that data mining and machine learning can process it more quickly and effectively. The methodologies are frequently used at the very beginning of the machine learning and AI development pipeline in order to get dependable results. In this step, the dataset is cleaned in order to create a clean and balanced dataset for later analysis stage. These processes reshaping the images, resizing the images, data augmentation, and renaming the labels of every image to fit the classes of every Arabic Letter. Here, the dataset of Arabic letters that was gathered during the data collection activity was processed and cleaned. This activity's deliverable was an Arabic letter dataset that has been processed.

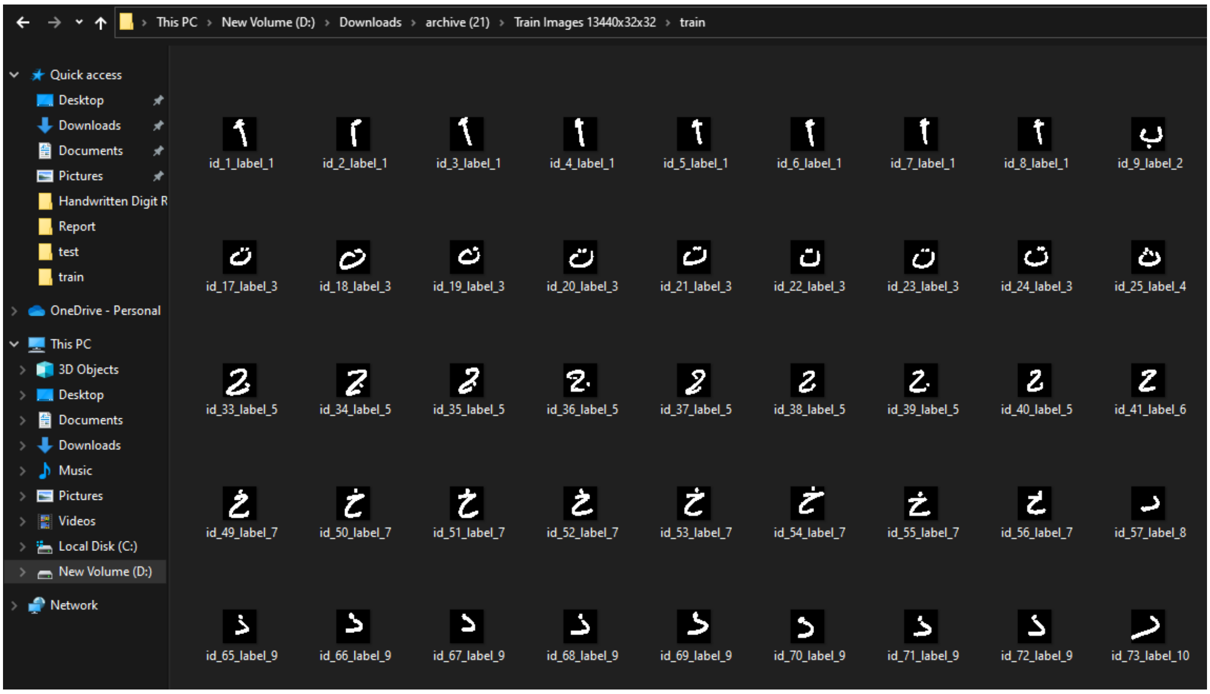


Figure 3.2: Screenshot of Arabic letter before classification

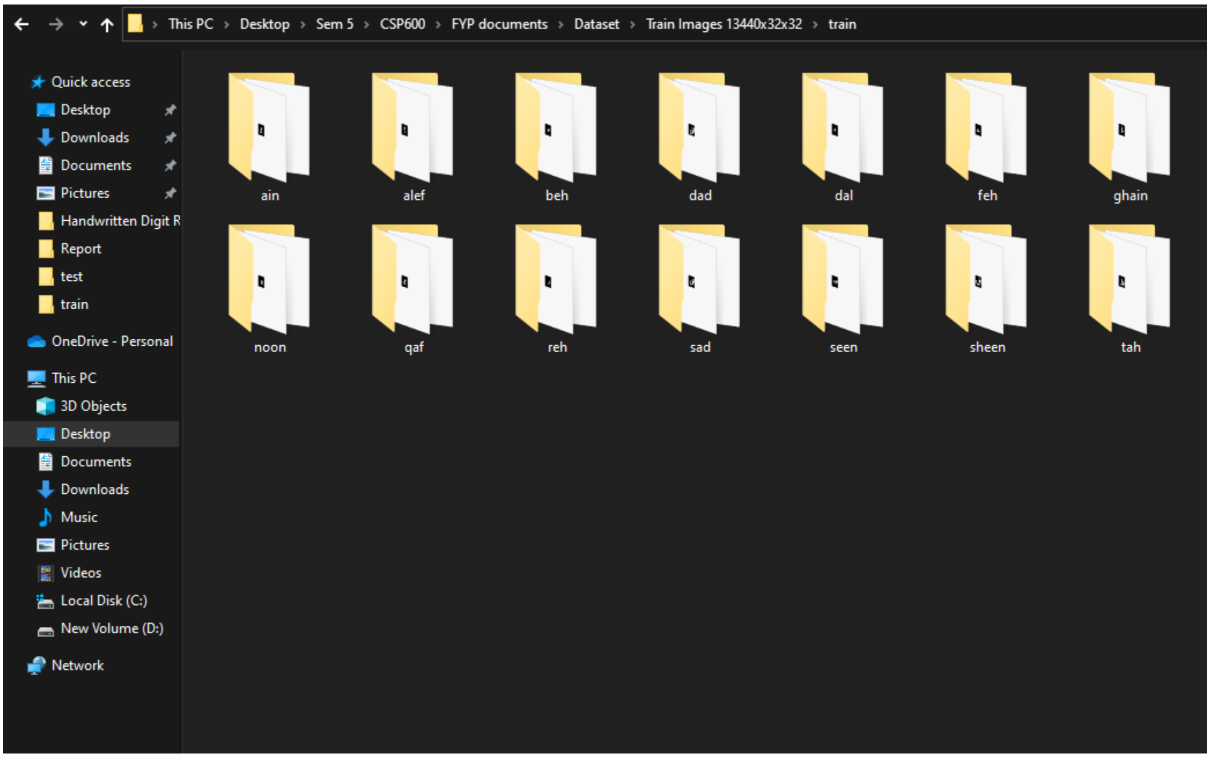
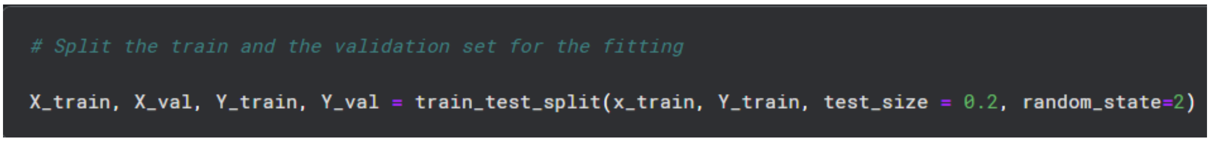


Figure 3.3: Screenshot of Arabic letter after classification

First of all, the images of the Arabic letters are classed according to their own letters into files. Then the name of the folders was changed for an easier understanding and can be read as letters with label. Next, the name of every image was too long which is then renamed to make it easier to be coded in python.

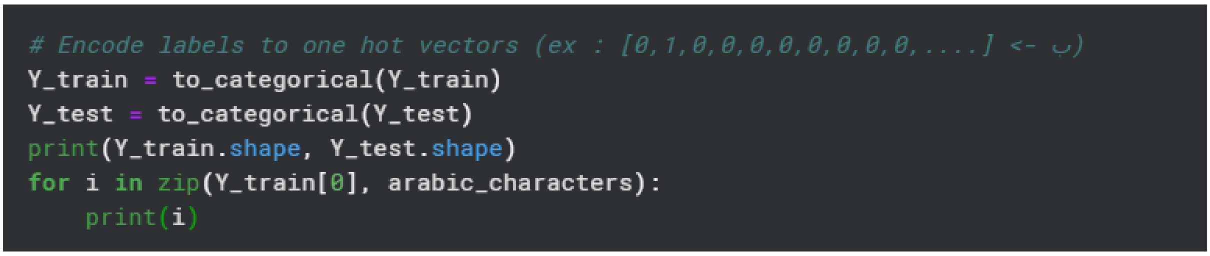


```
# Split the train and the validation set for the fitting

X_train, X_val, Y_train, Y_val = train_test_split(x_train, Y_train, test_size = 0.2, random_state=2)
```

Figure 3.4: Code for splitting dataset into training and testing set.

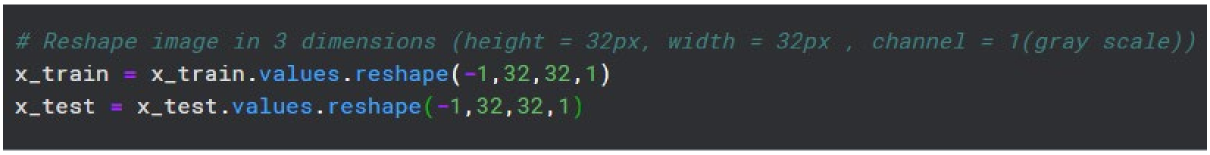
The dataset was already split between test and train, so further data splitting is done through the process of coding in python. Twenty percent of the data were used for testing, while eighty percent were used for training. The Sklearn model selection tool's "train_test_split" method was used to do this. Data arrays are frequently divided into two subsets: training data and testing data. The proportion that was specified in the coding will be used by the system to divide it. The coding of the dataset was divided into two sets of data, as shown in Figure 3.4. For the testing data, the "test_size" was set to 0.20, or 20%. The "random_state" function is utilised to distribute the dataset in an unbiased manner, as a fixed value ensures that the same set of random numbers is generated each time the code is executed. Two datasets, test and train, each containing 28 Arabic Letters, are the process's output.



```
# Encode labels to one hot vectors (ex : [0,1,0,0,0,0,0,0,0,0,....] <- ب)
Y_train = to_categorical(Y_train)
Y_test = to_categorical(Y_test)
print(Y_train.shape, Y_test.shape)
for i in zip(Y_train[0], arabic_characters):
    print(i)
```

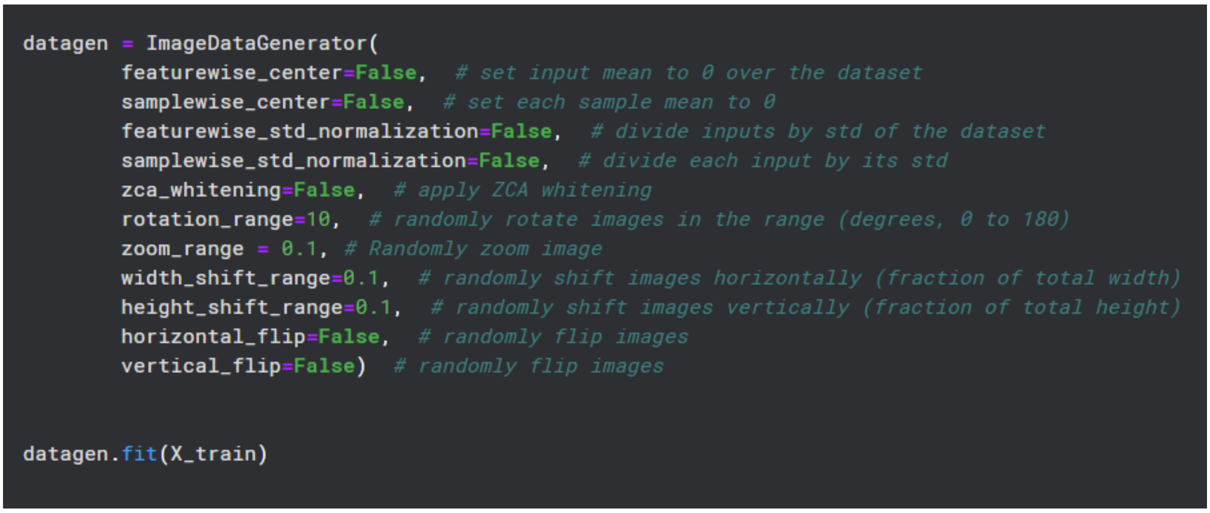
Figure 3.5: Code for converting class vector.

The images are categories according to its classes by converting the class vector in integers to binary class matrix for the model to class the images easier.



```
# Reshape image in 3 dimensions (height = 32px, width = 32px , channel = 1(gray scale))
x_train = x_train.values.reshape(-1,32,32,1)
x_test = x_test.values.reshape(-1,32,32,1)
```

Figure 3.6: Code for reshaping and resize image.



```
datagen = ImageDataGenerator(
        featurewise_center=False,  # set input mean to 0 over the dataset
        samplewise_center=False,  # set each sample mean to 0
        featurewise_std_normalization=False,  # divide inputs by std of the dataset
        samplewise_std_normalization=False,  # divide each input by its std
        zca_whitening=False,  # apply ZCA whitening
        rotation_range=10,  # randomly rotate images in the range (degrees, 0 to 180)
        zoom_range = 0.1, # Randomly zoom image
        width_shift_range=0.1,  # randomly shift images horizontally (fraction of total width)
        height_shift_range=0.1,  # randomly shift images vertically (fraction of total height)
        horizontal_flip=False,  # randomly flip images
        vertical_flip=False)  # randomly flip images


datagen.fit(X_train)
```

Figure 3.7: Code for data augmentation.

The resizing and reshaping of the images are done and then proceed to data augmentation where the images are then randomly flipped to create new samples within the same class of images. Finally, data augmentation is used to expand the dataset's image count. In addition to increasing the dataset's size, data augmentation produces a variety of newly altered image types. This enables the model to identify hidden features in the images and make better sense of the data. As a result, the model's accuracy may be raised. The dataset was supplemented using the ImageDataGenerator function found in the Python package Keras, as seen in Figure 3.7. In order to improve the dataset's resilience and store it in overhead memory, ImageDataGenerator augments it in real-time. The images in the dataset were augmented by flipping, rotating, and enlarging them while adjusting their width and height within a 0.1 range. However, some of the image's pixels were moved outside of the image, leaving an empty space, when the image was rotated and shifted. The nearest pixel values were used to fill the empty area of the image using the fill mode. Every transformation was applied to the dataset at random.

3.5.2 SYSTEM DESIGN

3.5.2.1 SYSTEM ARCHITECTURE

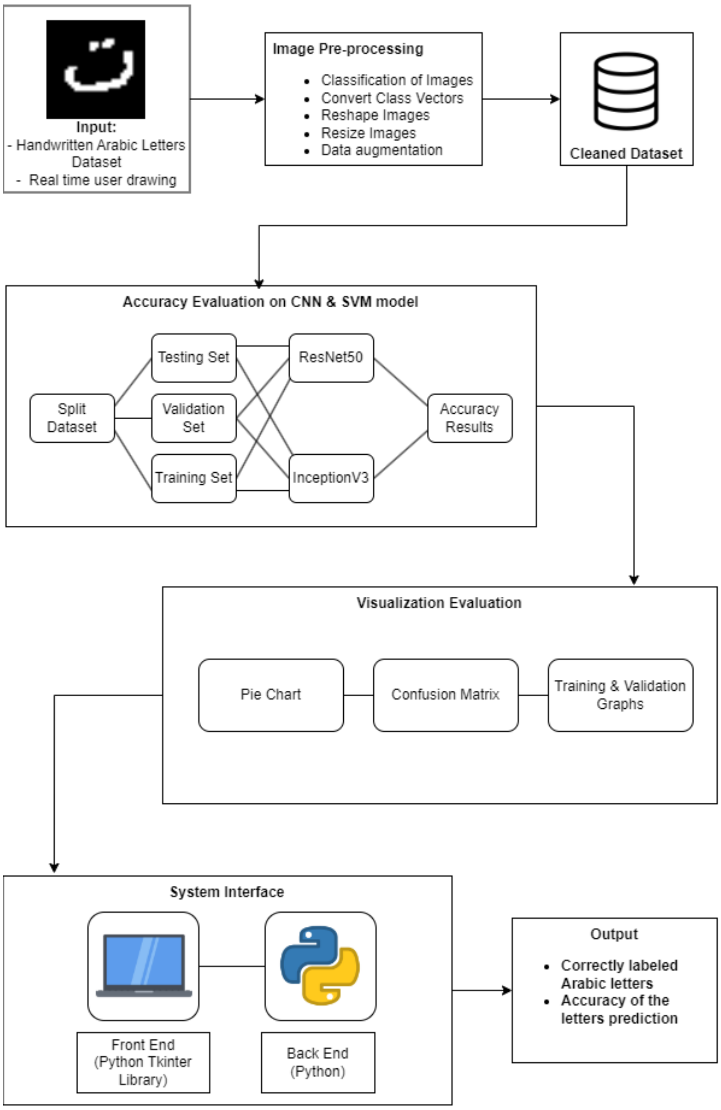


Figure 3.8: Image of System Architecture

The system architecture is a diagram that shows the structure, behaviour, and viewpoints of the system. The project's system architecture design is depicted in the picture below. There are five primary components to the architecture that must be completed in order to finish the system. The Arabic letter dataset's input images were first and foremost pre-processed. The pre-processed data includes classification, converting class vector, split dataset, reshaped, and augmented images. A testing set and a testing set were created from the final, cleaned dataset. The models ResNet50 and InceptionV3 were then used to obtain the accuracy result. The evaluation is then visualized through piecharts, confusion matrix, and training and validation graphs. Finally, Python was used to code both the system's front end and back end. A library in Python is used to improve the interface's appearance, Python code was combined with Tkinter. The system will also be able to receive real time drawing of handwritten Arabic letters. The output is then produced are correctly labelled Arabic letters and accuracy of the letter's prediction.

3.5.2.2 SYSTEM INTERFACE

The user interface (UI) of the system serves as a conduit for user interaction and communication with the system. Therefore, by having a user interface, the system can also be used by end users who lack the technical skills necessary to input the images. Python programming language and Hypertext Markup Language (HTML), which were both used to develop and show the system on the web browser, were combined to create the system interface. The web pages' structure, which includes text, tables, headings, and images, was created using HTML. The interface will also be linked to a touch pad where users can draw letters using a touch pen or a mouse. The handwritten letter will be detected through the recognition model. The interface will be adapted through a Python library which is Tkinter that allows to customize features and the design of the GUI in Python.