CAMERA CLASSIFIER

MINOR PROJECT-II REPORT

Submitted by

AREEBA SABRI 2020-310-032

in partial fulfillment for the award of the degree of

B.TECH COMPUTER SCIENCE AND ENGINEERING

Under the supervision of

DR. IHTIRAM RAZA KHAN

Department of Computer Science& Engineering
School of Engineering Sciences & Technology

JAMIA HAMDARD

New Delhi-110062

(2023)

DECLARATION

I, **Ms. Areeba Sabri** a student of **Bachelors Of Computer Science And Technology (B.Tech CSE) , Enrolment No:2020-310-032,** hereby declare that the Project/Dissertation entitled **" Camera Classifier "** which is being submitted by me to the Department of Computer Science, Jamia Hamdard, New Delhi in partial fulfillment of the requirement for the award of the degree of **Bachelors Of Computer Science And Technology (B.Tech CSE)**,is my original work and has not been submitted anywhere else for the award of any Degree, Diploma, Associateship, Fellowship or other similar title or recognition.

Areeba Sabri

(Signature and Name of the Applicant)

Date: 15 November 2023

Place: New Delhi

ACKNOWLEDGEMENT

I express my sincere thanks to **Dr Ihtiram Raza Khan** (Dept. of Computer Science and Engineering), my project in charge, who guided me through the project for his valuable suggestions and guidance for completing the project. This project has been a success only because of his guidance.

I deeply express my sincere thanks to our Head of Department **Dr. Farheen Siddiqui** for encouraging me and allowing me to present the project on the topic "**Camera Classifier**" at the department premises for the partial fulfilment of the requirements leading to the award of B.Tech degree.

I am also thankful to the whole computer science and engineering department for providing the technical support to carry out the project work, letting us utilize all the necessary facilities of the institute and providing guidance at each & every step during the project work.

INDEX

OBJECTIVE

A python application that uses camera input to train a Support Vector Machine (SVM) to respond to specific action that classifies what is being shown to the camera into separate classes.

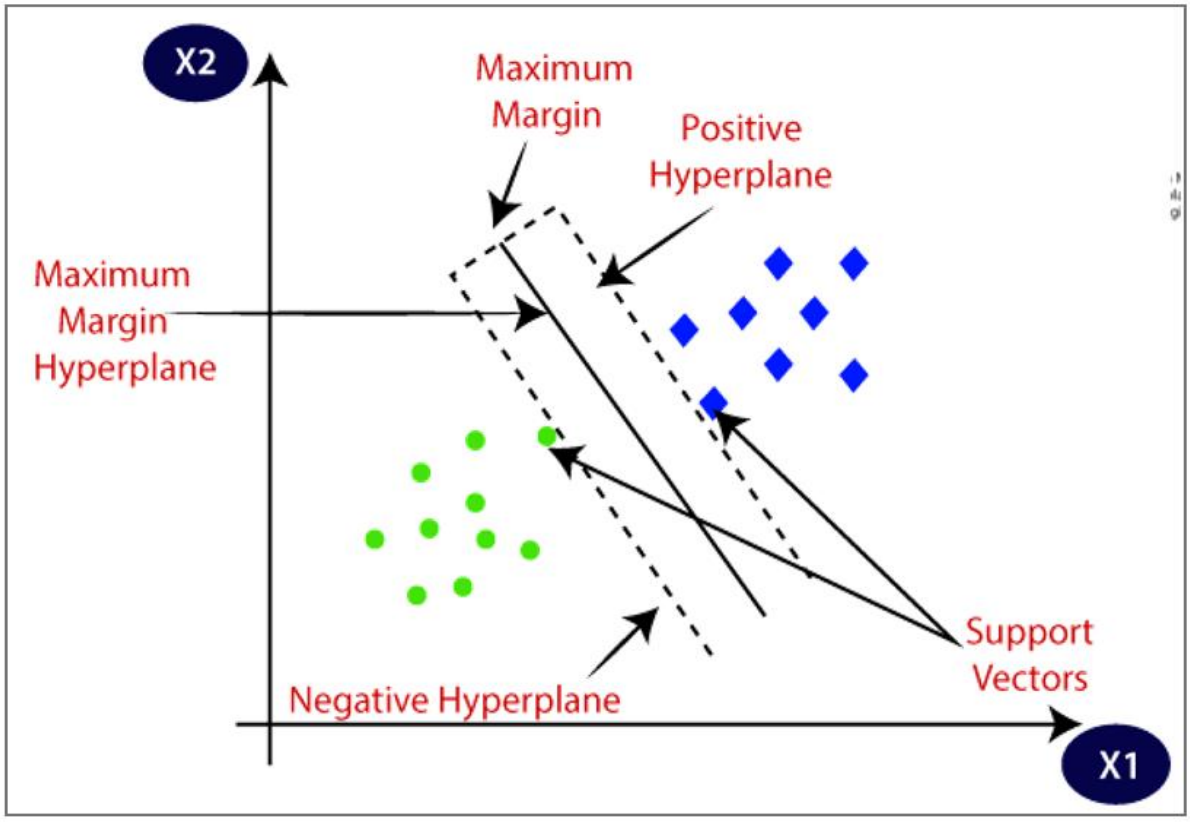
INTRODUCTION

Support Vector Machine Algorithm

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

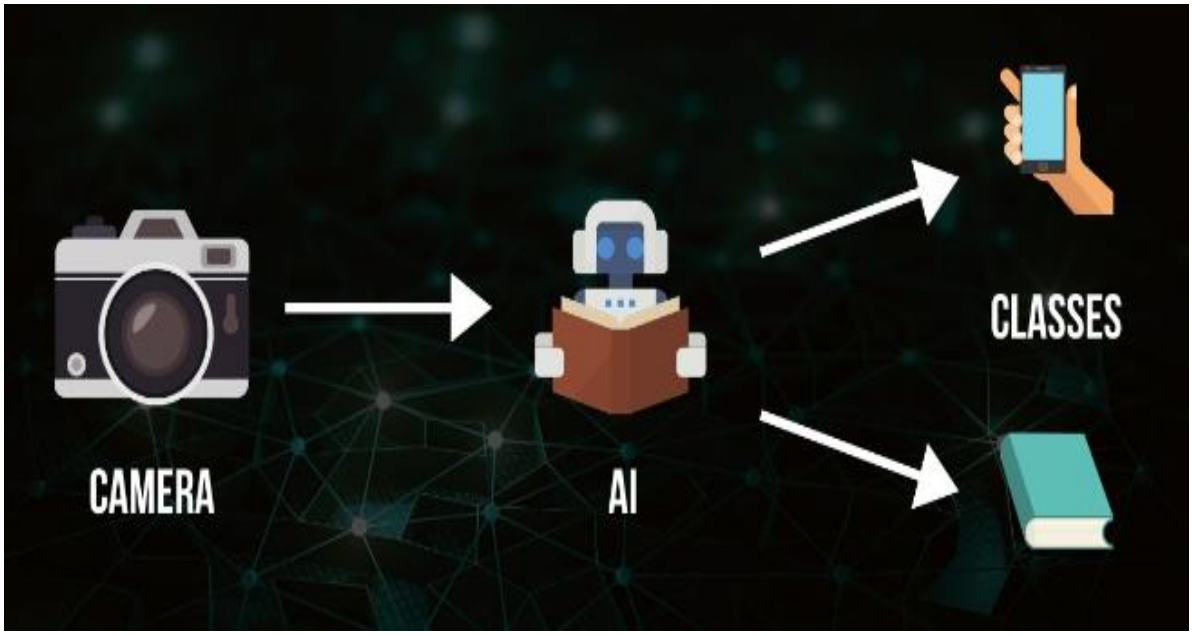
The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:



The SVM algorithm works by finding the hyperplane that separates the different classes in the feature space. The key idea behind SVMs is to find the hyperplane that maximizes the margin, which is the distance between the closest points of the different classes. The points that are closest to the hyperplane are called support vectors.

One of the main advantages of using SVMs for image classification is that they can effectively handle high-dimensional data, such as images. Additionally, SVMs are less prone to overfitting than other algorithms such as neural networks.



To create such a model, it is necessary to go through the following phases:

- Import required libraries
- Load the image and convert it to a data frame.
- Separate input features and targets.
- Split train and test value.
- Build and train the model
- Model evaluation.
- Prediction

What is Classification in Machine Learning?

Classification is a supervised machine learning method where the model tries to predict the correct label of a given input data. In classification, the model is fully trained using the training data, and then it is evaluated on test data before being used to perform prediction on new unseen data.
There are two types of learners in machine learning classification: lazy and eager learners

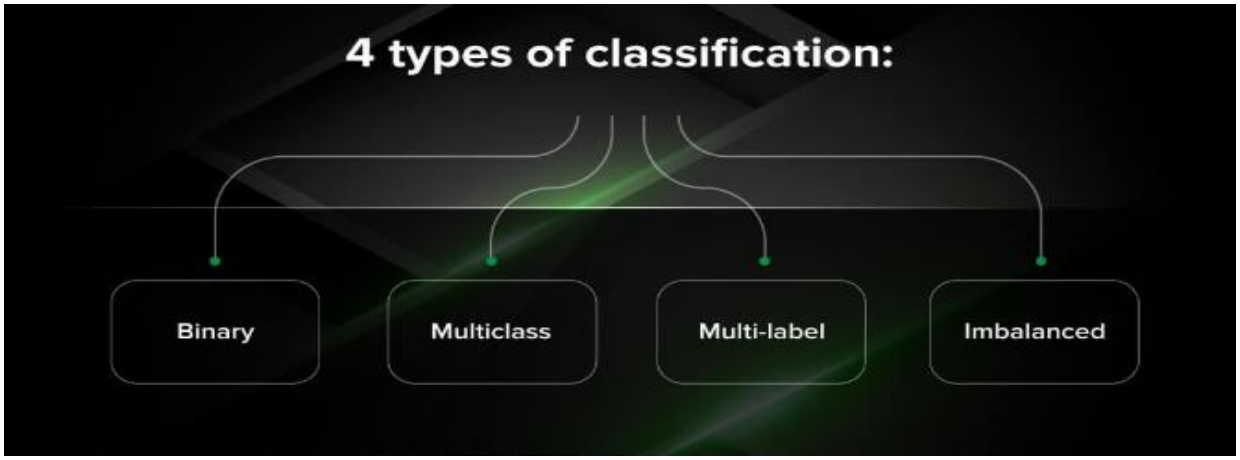
Eager learners are machine learning algorithms that first build a model from the training dataset before making any prediction on future datasets. They spend more time during the training process because of their eagerness to have a better generalization during the training from learning the weights, but they require less time to make predictions.
Most machine learning algorithms are eager learners, and below are some examples:

- Logistic Regression.
- Support Vector Machine.
- Decision Trees.
- Artificial Neural Networks.

Lazy learners or instance-based learners, on the other hand, do not create any model immediately from the training data, and this is where the lazy aspect comes from. They just memorize the training data, and each time there is a need to make a prediction, they search for the nearest neighbor from the whole training data, which makes them very slow during prediction. Some examples of this kind are:

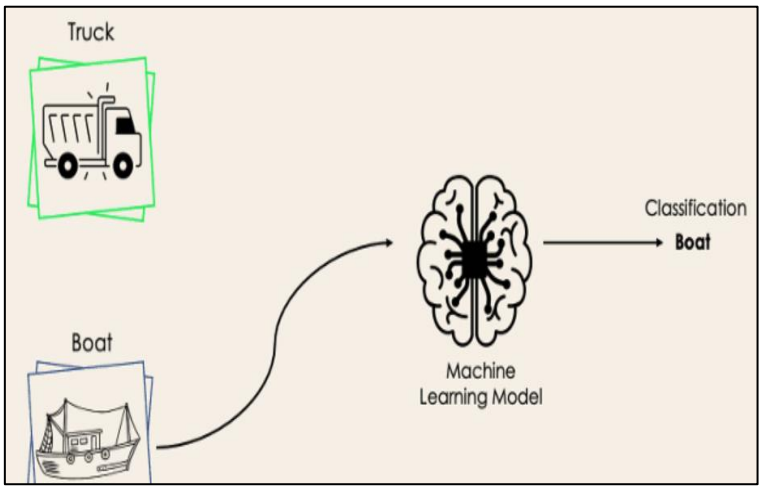
- K-Nearest Neighbor.
- Case-based reasoning.

Different Types of Classification Tasks in Machine Learning



Binary Classification

In a binary classification task, the goal is to classify the input data into two mutually exclusive categories. The training data in such a situation is labeled in a binary format: true and false; positive and negative; O and 1; spam and not spam, etc. depending on the problem being tackled. For instance, we might want to detect whether a given image is a truck or a boat.

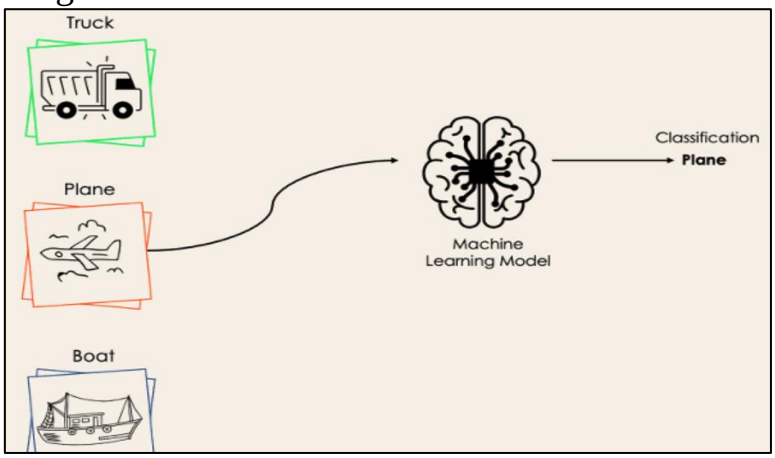


Multi-Class Classification

The multi-class classification, on the other hand, has at least two mutually exclusive class labels, where the goal is to predict to which class a given input example belongs to. In the following case, the model correctly classified the image to be a plane.

Most of the binary classification algorithms can be also used for multi-class classification. These algorithms include but are not limited to:

- Random Forest
- Naive Bayes
- K-Nearest Neighbors
- Gradient Boosting
- SVM
- Logistic Regression.



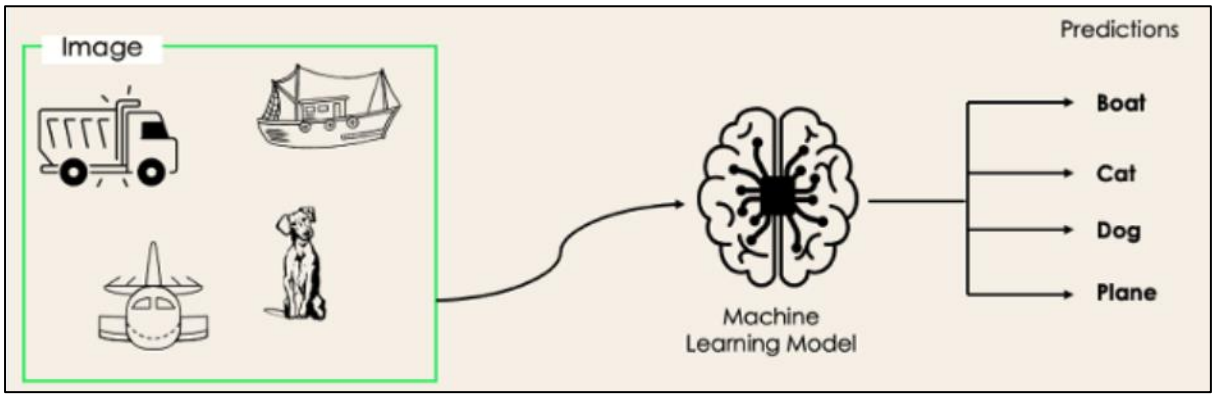
Multi-Label Classification

In multi-label classification tasks, we try to predict 0 or more classes for each input example. In this case, there is no mutual exclusion because the input example can have more than one label.

Such a scenario can be observed in different domains, such as auto-tagging in Natural Language Processing, where a given text can contain multiple topics. Similarly, to computer vision, an image can contain multiple objects, as illustrated below: the model predicted that the image contains: a plane, a boat, a truck, and a dog.

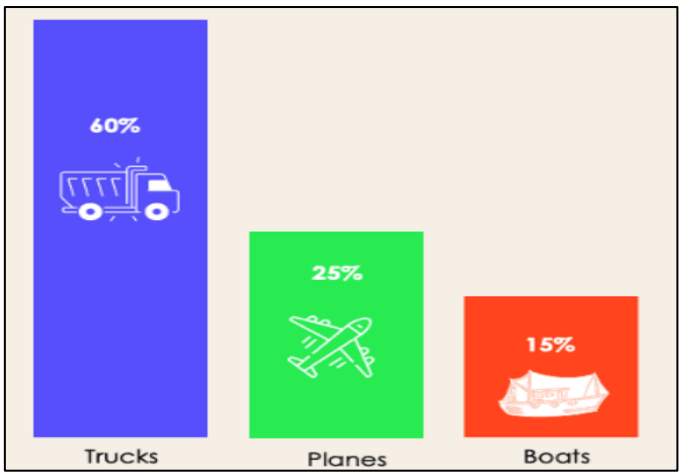
It is not possible to use multi-class or binary classification models to perform multi-label classification. However, most algorithms used for those standard classification tasks have their specialized versions for multi-label classification. We can cite:

- Multi-label Decision Trees
- Multi-label Gradient Boosting
- Multi-label Random Forests



Imbalanced Classification

For the imbalanced classification, the number of examples is unevenly distributed in each class, meaning that we can have more of one class than the others in the training data. Let's consider the following 3-class classification scenario where the training data contains: 60% of trucks, 25% of planes, and 15% of boats.



PROBLEM STATEMENT

To develop a machine learning model using supervised learning approach- SVM - that uses objects shown to the camera as labelled training dataset of images and testing it on similar/same objects that classify what is shown to the camera into different classes. The model must be able to Auto-Predict and reset itself whenever instructed to do so

SOFTWARE REQUIREMENTS AND SPECIFICATIONS

Visual Studio Code (VS Code):
Visual Studio Code is a free open source text editor by Microsoft. VS Code is available for Windows, Linux, and macOS. Although the editor is relatively lightweight, it includes some powerful features that have made VS Code one of the most popular development environment tools in recent times.

Python:
Python, one of the most popular programming languages in the world, has created everything from Netflix's recommendation algorithm to the software that controls self-driving cars. Python is a general-purpose language, used to create a range of applications, including data science, software and web development, automation, and improving the ease of everyday tasks.

Libraries used:

- OpenCV: OpenCV stands for Open Source Computer Vision. To put it simply, it is a library used for image processing. In fact, it is a huge open-source library used for computer vision applications, in areas powered by Artificial Intelligence or Machine Learning algorithms, and for completing tasks that need image processing.

- Tkinter: Tkinter is a Python library that can be used to construct basic graphical user interface (GUI) applications. In Python, it is the most widely used module for GUI applications.

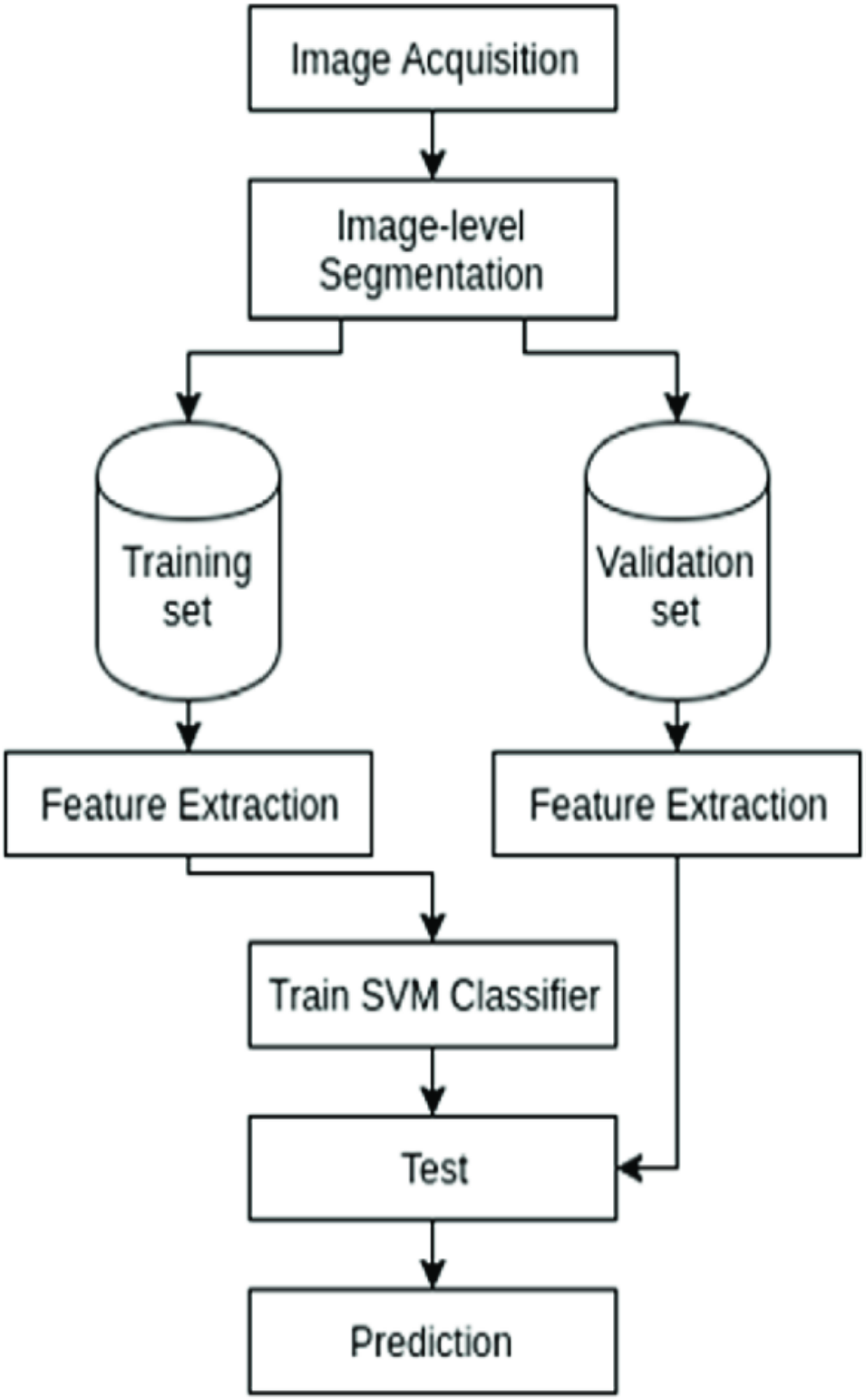
- PIL: PIL is an additional, free, open-source library for the Python programming language that provides support for opening, manipulating, and saving many different image file formats.

- Os: The OS module in Python provides functions for interacting with the operating system. This module provides a portable way of using operating system-dependent functionality. The *os* and *os. path* modules include many functions to interact with the file system.

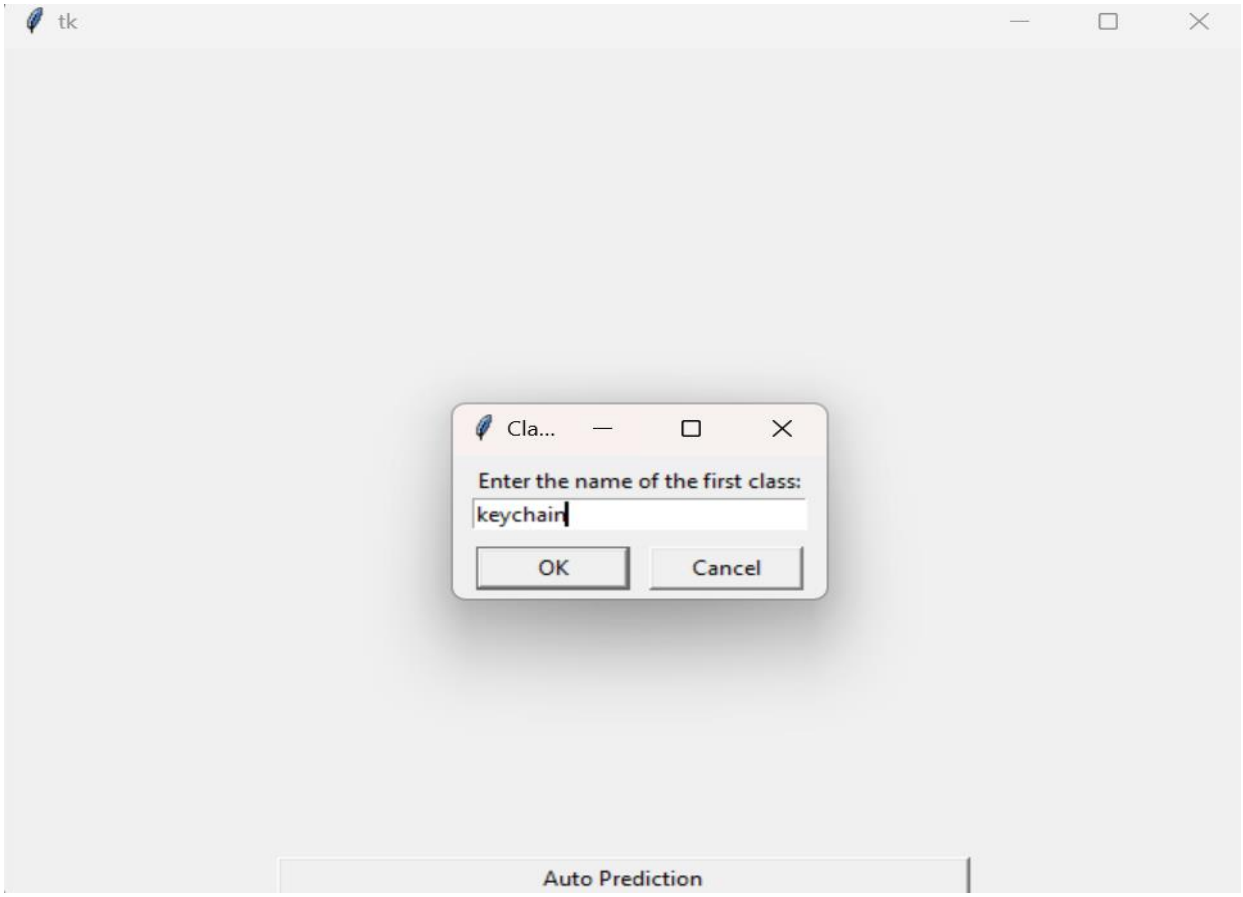
- Sklearn: The sklearn library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction.

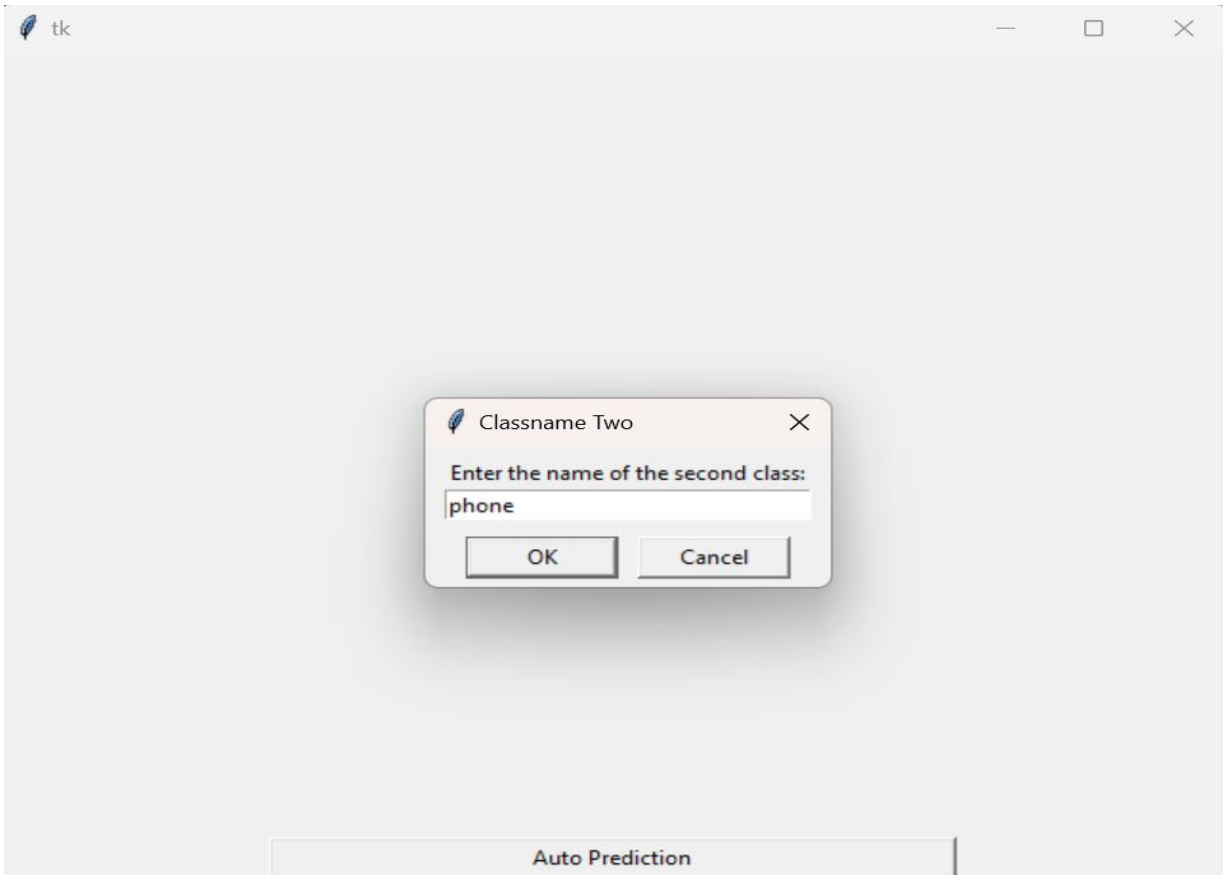
- NumPy: NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting and much more.

ENTITY-RELATONSHIP DIAGRAM

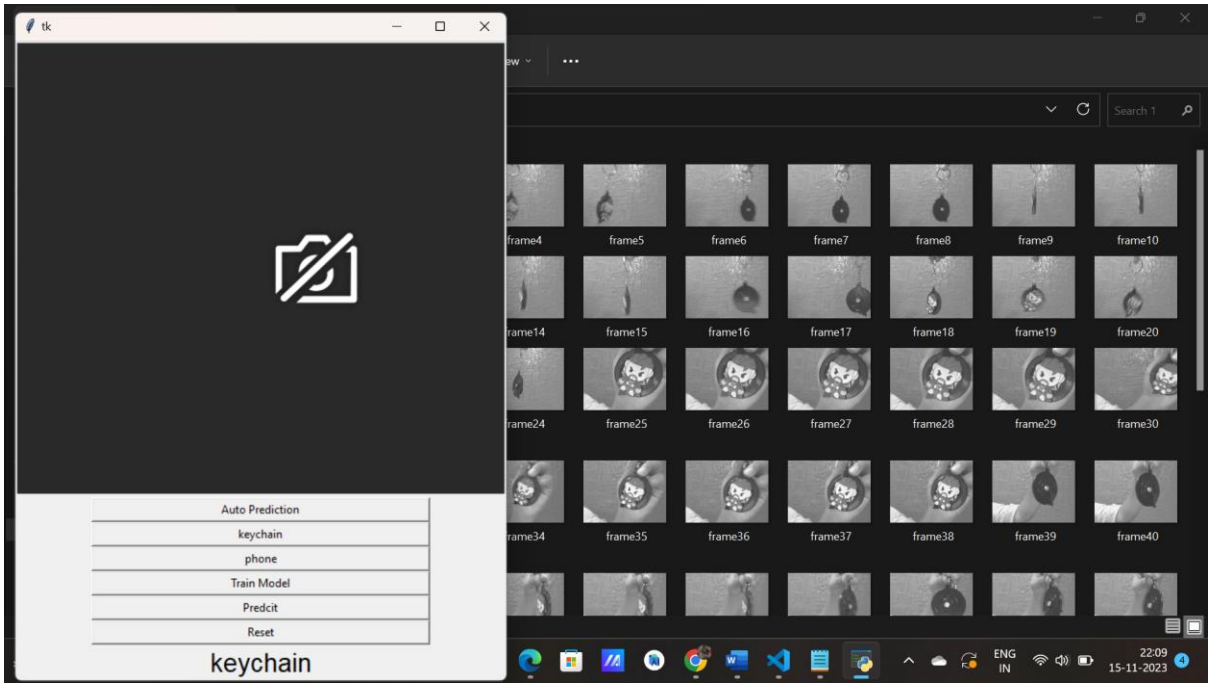


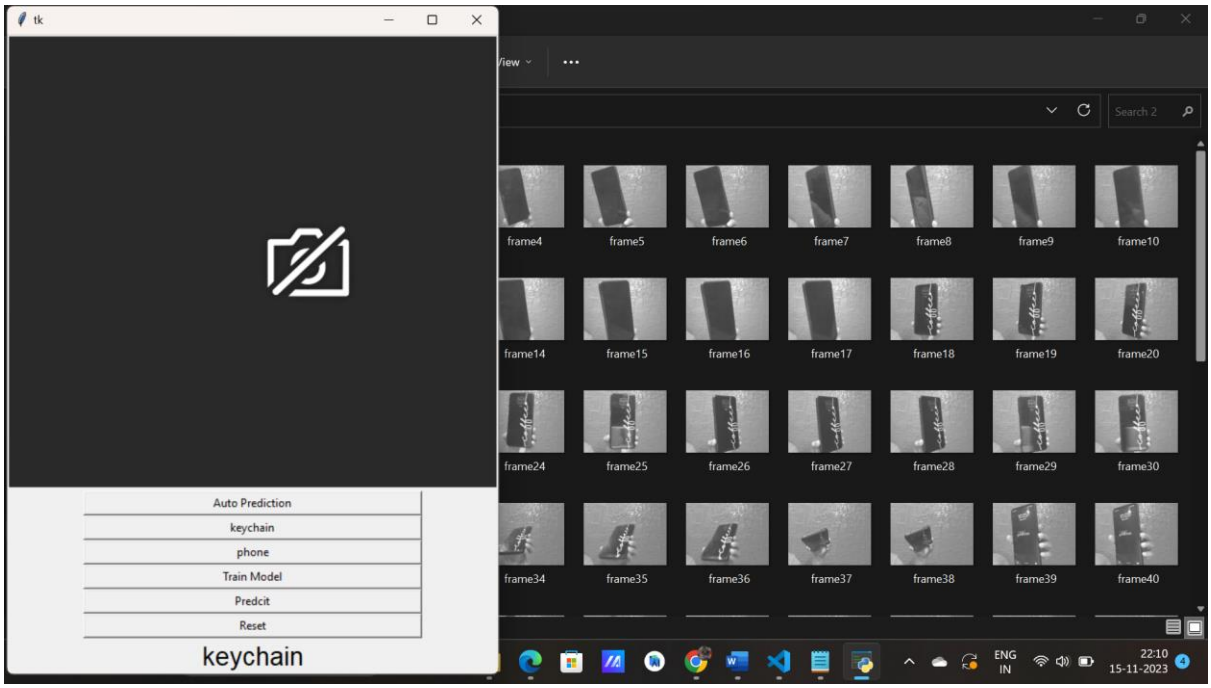
SNAPSHOTS OF INPUT AND OUTPUT SCREENS


tk
Cla...
Enter the name of the first class:
keychain
OK
Cancel
Auto Prediction

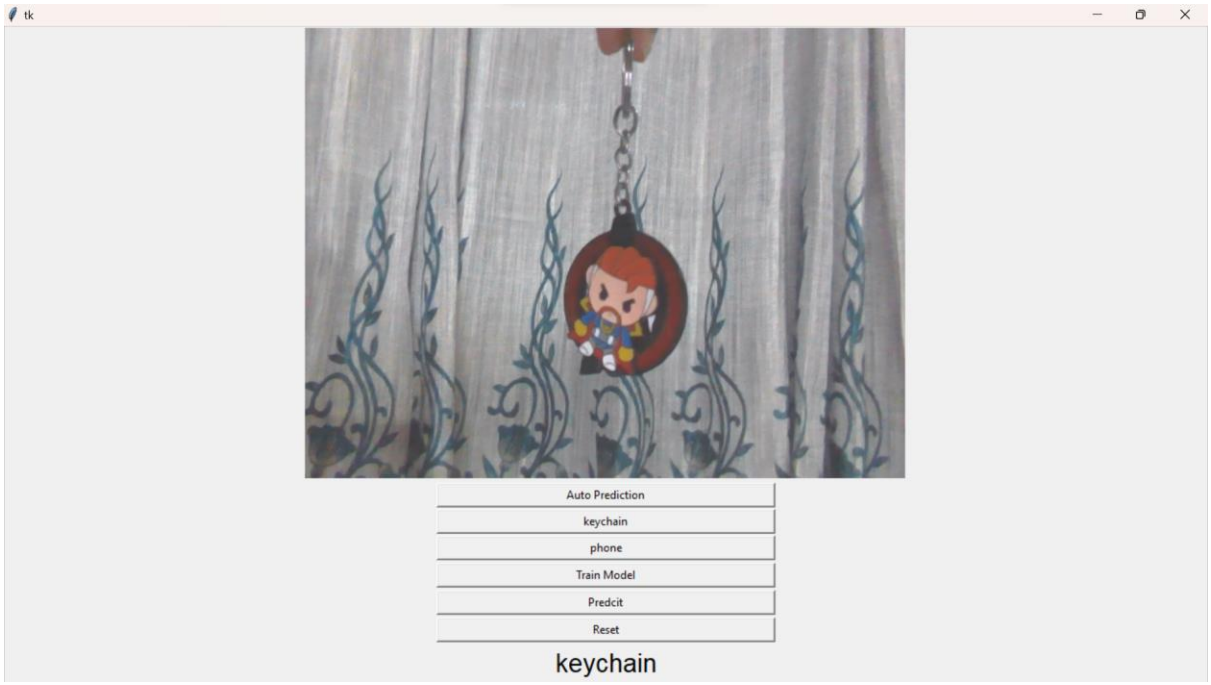

tk
Classname Two
Enter the name of the second class:
phone
OK
Cancel
Auto Prediction

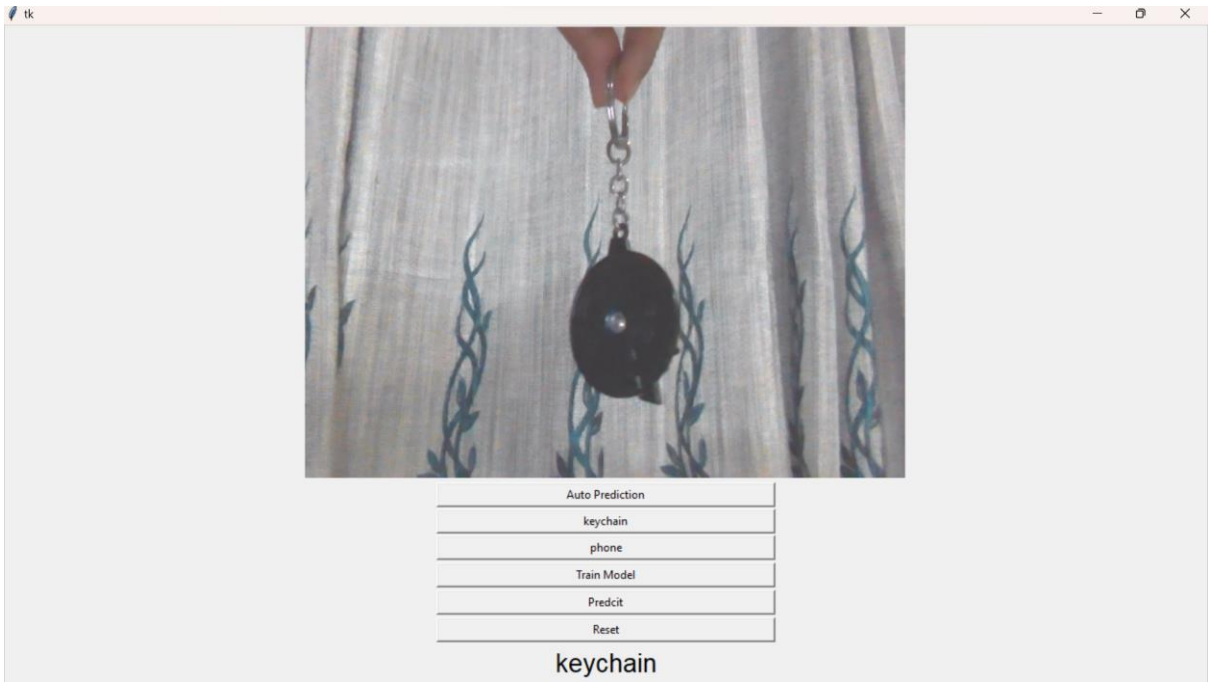
TRAINING MODEL ON THE BASIS OF TWO CLASSES



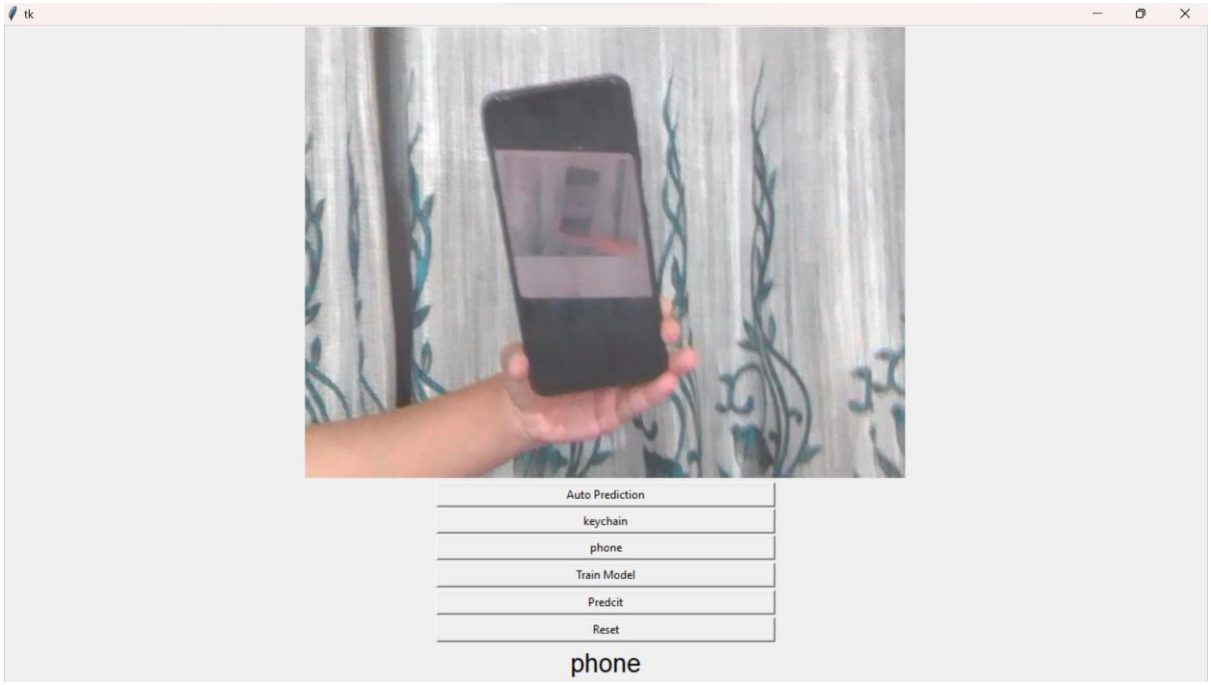


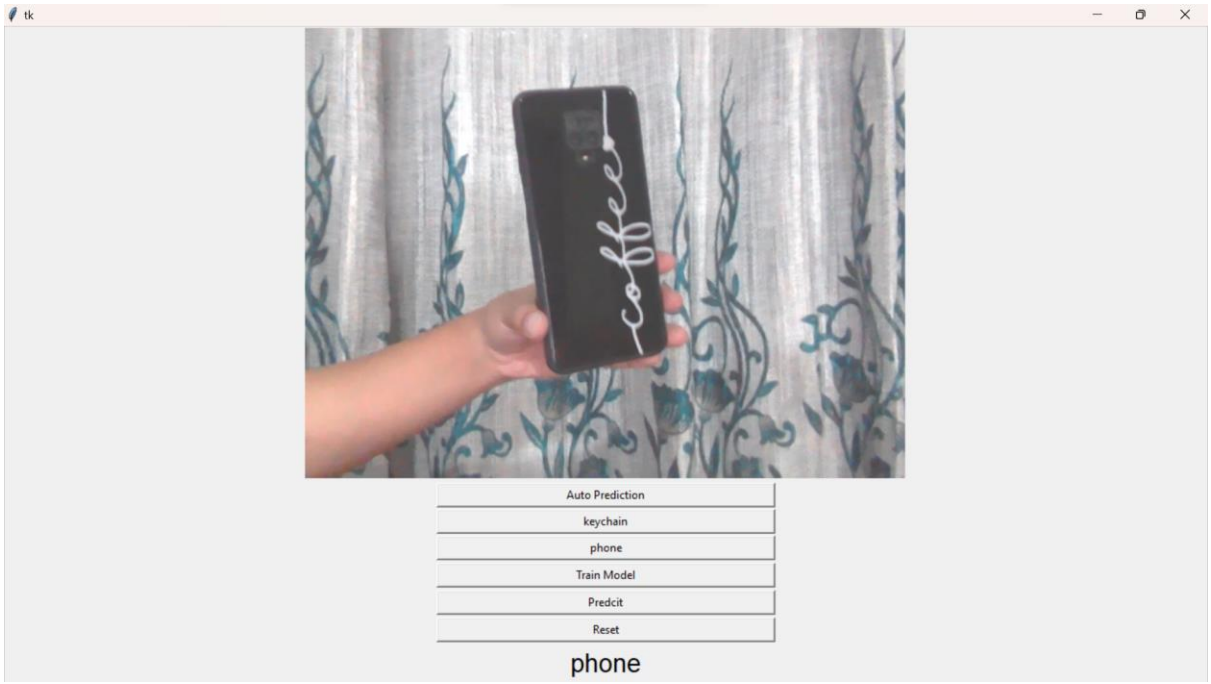
TESTING CLASS 1: KEYCHAIN



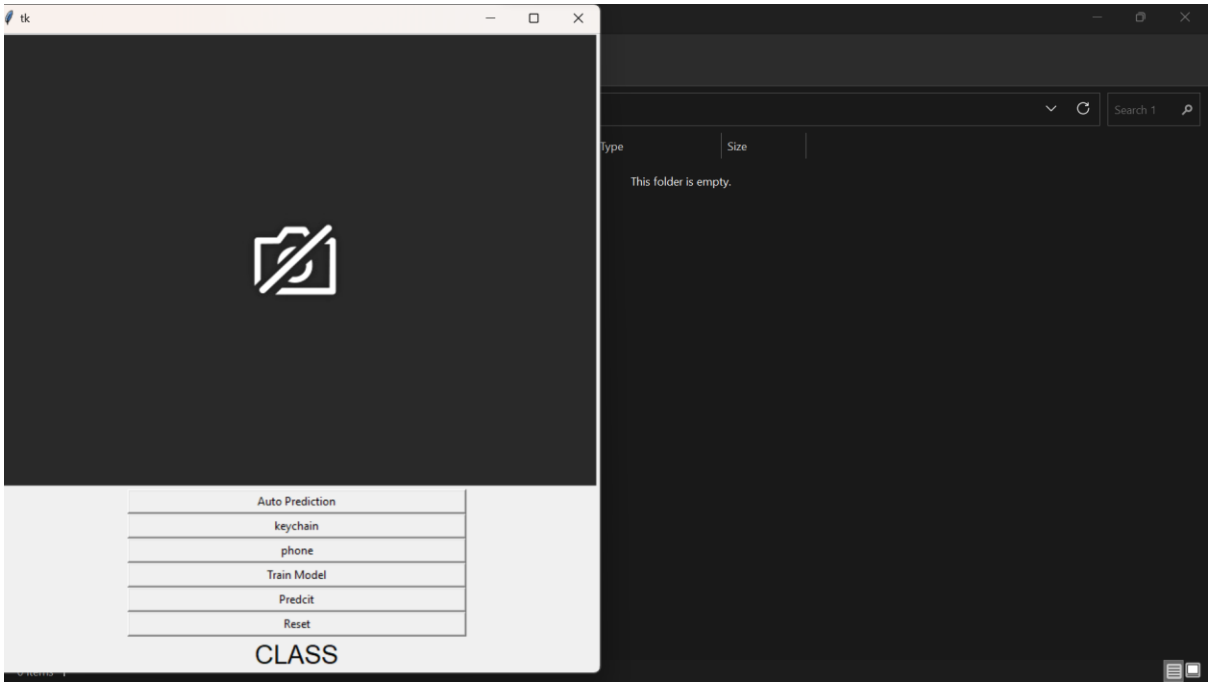


TESTING CLASS 2: PHONE





RESETTING THE MODEL



CONCLUSION

After successful completion of the training and testing of the above model, we can finally conclude that SVM is proved to be effective for real time classification of images. The main intent of the classification model was to categorize the images into one of the classes we created.

The following are some of the highlights that can be drawn from the model used:

- SVMs are effective in handling high-dimensional data, which is common in many applications such as image and text classification.
- SVMs are robust to noise in the data, as the decision boundary is determined by the support vectors, which are the closest data points to the boundary.
- SVMs have good generalization performance, which means that they are able to classify new, unseen data well.
- The input data is plotted in a high-dimensional space (with as many dimensions as the number of features), and the SVM algorithm finds the best boundary that separates the classes.
- SVM is more effective in high dimensional spaces and is relatively memory efficient

Overall the project has been able to demonstrate some key features of machine learning that is feature selection and in-depth study of how supervised learning algorithm works,

LIMITATIONS

- SVM does not perform very well when the data set has more noise i.e. target classes are overlapping. In cases where the number of features for each data point exceeds the number of training data samples, the SVM will underperform.
- As the support vector classifier works by putting data points, above and below the classifying hyperplane there is no probabilistic explanation for the classification.

BIBLOGRAPHY

- https://analyticsindiamag.com/understanding-the-basics-of-svm-with-example-and-python-implementation/
- https://stackoverflow.com/
- https://towardsdatascience.com/everything-about-svm-classification-above-and-beyond-cc665bfd993e
- https://www.geeksforgeeks.org/image-classification-using-support-vector-machine-svm-in-python/