



**IMAGE CLASSIFICATION**

**AND DETECTION**

**MINI PROJECT - II**

*Submitted by*

**LEKHA.S**

**Roll No: 727622MCA025**

*In partial fulfillment for the requirements*

*for the award of the degree of*

**MASTER OF COMPUTER APPLICATIONS**

**Dr. MAHALINGAM COLLEGE OF**

**ENGINEERING AND TECHNOLOGY**

**POLLACHI-642 003**

**(Approved by AICTE, Affiliated to Anna University**

**and Accredited by NBA & NAAC with ‘A++’ Grade)**

**JUNE 2023**

**Dr. MAHALINGAM COLLEGE OF ENGINEERING**

**AND TECHNOLOGY POLLACHI – 642 003**

Department of Computer Applications

**MINI PROJECT - II**

**JUNE 2023**

This is to certify that the project entitled

**IMAGE CLASSIFICATION AND DETECTION**

is the bonafide record of project work done by

**LEKHA.S**

**Roll No: 727622MCA025**

Of Master of Computer Applications during the year 2022-2024

**---------------------------------**   **---------------------------------------**

Project Guide Head of the Department

**Dr. S. RAJA RAJESWARI, MCA.,** **Dr. R .MUTHUSAMI , MCA.,**

**M.Phil., Ph.D M.Phil., Ph.D**

**Assistant professor/MCA**   **Assistant Professor(SG)/MCA**

Submitted for the Project Viva-Voce examination held on **\_\_\_\_\_\_\_\_\_\_\_\_**

**-----------------------**  **-------------------------**

Internal Examiner External Examiner

**i**

**DECLARATION**

**DECLARATION**

I affirm that the mini project work titled **“IMAGE CLASSIFICATION AND DETECTION”** being submitted in partial fulfillment for the award of **Master of Computer Applications** is the original work carried out by me. It has not formed the part of any other project work submitted for award of any degree or diploma, either in this or any other university.

**---------------------------------**

(Signature of the Candidate)

**LEKHA.S**

**Roll No: 727622MCA025**

I certify that the declaration made above by the candidate is true.

**-------------------------------**

(Signature of the Guide)

**Dr. S. RAJA RAJESWARI , MCA., M.Phil., Ph.D**

**AssistantProfessor/MCA**

**ii**

**ACKNOWLEDGEMENT**

**ACKNOWLEDGEMENT**

I express my gratitude to **Dr. C. RAMASWAMY, M.E., F.I.V., Ph.D., Secretary,** NIA Educational Institutions, Pollachi, for having provided me the facilities to do the project successfully.

I express my sincere thanks to **Dr. P. GOVINDASAMY, B.E., M.E., Ph.D ., Principal,** Dr. Mahalingam college of Engineering and Technology, Pollachi, for having provided me the facilities to do the project successfully.

It is a great privilege and pleasure for me to express my gratitude to **Dr. A. SENTHIL KUMAR, B.E., M.E., Ph.D., Dean (Academic and Autonomous),** **Dr. S. RAMAKRISHNAN, B.E., M.E., Ph.D., Dean (Research & Innovation), Dr. CALVIN SOPHISTUS KING, M.Tech., Ph.D, Dean (Industry Relations and Talent Development),** Dr. Mahalingam College of Engineering and Technology, Pollachi, for persistent encouragement.

I owe deep sense of gratitude to **Dr. R. MUTHUSAMI, MCA., M.Phil**.**, Ph.D.,** Head of Department of Computer Applications for appreciating my goal. I express my sincere thanks to his for his constant encouragement.

I express my thanks to **Dr. S. RAJA RAJESWARI, MCA., M.Phil., Ph.D, Assistant Professor, Department of Computer Applications** for her valuable guidance and support to meet the successful completion of my project.

I express my sincere thanks to **all staff members of Department of Computer Applications** for their encouragement and valuable guidance throughout this project.

Last but not the least, I would like to thank **my family** and **my friends** for putting up with me spending so much time providing encouragement and valuable suggestions, throughout the project tenure.

**iii**

**ABSTRACT**

**ABSTRACT**

Image classification and detection using machine learning is a powerful techniques for automatic image analysis and understanding. These techniques leverage algorithms and models to classify images into predefined categories or detect and locate specific objects or patterns within images. This system deals with machine learning approach to image classification and detection using Support Vector Machines (SVM). The purpose of this project is to classify and detect objects within images using machine learning with the Support Vector Machines (SVM) algorithm. The project uses a datasets of images labelled with different categories to train the SVM model to recognize and classify images.

**iv**

**TABLE OF CONTENT**

**TABLE OF CONTENT**

**CHAPTER NO TITLE PAGENO**

**ABSTRACT iv**

**LIST OF FIGURES vii**

**LIST OF ABBREVATIONS viii**

**1 INTRODUCTION 1**

1.1 Objectives **1**

**2 SYSTEM ANALYSIS 2**

2.1 Existing System **2**

2.2 Proposed System **2**

2.2.1 Advantages Of Proposed System **2**

2.3 Requirements Gathering **3**

2.3.1 Functional Requirement **3**

2.3.2 Non-Functional Requirement **4**

**3 SYSTEM SPECIFICATION 5**

3.1 Hardware Specification **5**

3.2 Software Specification **5**

**4 SOFTWARE DESCRIPTION 6**

4.1 Development Tools And Technologies **6**

4.1.1 Python **6**

4.1.2 Machine Learning **7**

4.1.3 Dataset **8**

**5 PROJECT DESCRIPTION 9**

5.1 Problem Definition **9**

5.2 Overview Of The Project **9**

5.3 Input Design **9**

5.4 Output Design **10**

5.5 System Design **11**

5.5.1 System Flow Diagram **11**

**v**

**6 SYSTEM IMPLEMENTATION 12**

6.1 System Maintenance **12**

6.2 System Architecture **16**

**7 CONCLUSION & FUTURE ENHANCEMENTS 17**

**8 APPENDICES 18**

8.1 Source Code **18**

8.2 Screen Shots **23**

**REFERENCES 26**

**vi**

**LIST OF FIGURES**

**FIG NO DESCRIPTION PAGENO**

**5.5.1 SFD 11**

**6.1.1 Types of Decision Boundary 13**

**6.1.2 Computer Readable Image 13**

**6.2.1 Training and Testing Phase 14**

**6.2.2 CNN Architecture 16**

**vii**

**LIST OF ABBREVIATIONS**

**DESCRIPTION ABBREVATION**

**SFD System Flow Diagram**

**SVM Support Vector Machine**

**CNN Convolution Neural Networks**

**viii**

**INTRODUCTION**

**CHAPTER 1**

**INTRODUCTION**

**1.1 Objective**

The main objective of image classification and detection using machine learning is to automatically analyze and understand the contents of an image. It involves the use of algorithms and models to classify images into predefined categories or detect and locate specific objects or patterns within an image. Overall, image classification and detection using machine learning is to automate the process of analyzing and understanding visual data, enabling a wide range of applications across various industries.

**1**

**SYSTEM ANALYSIS**

**CHAPTER 2**

**SYSTEM ANALYSIS**

**2.1 Existing System**

The existing systems for image classification and detection using machine learning include various traditional algorithms and frameworks such as Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Random Forests. These systems use image datasets to train the models and then classify and detect objects in new images. The frameworks for image classification and detection using machine learning include TensorFlow, Keras, PyTorch, and Caffe.

**2.2 Proposed System**

The proposed system for image classification and detection using machine learning with Support Vector Machine (SVM) uses labeled image datasets to train the models and then classify and detect objects in new images. In this project we have implemented “Matlab” and “Scikit-Learn” with SVM which helps in faster processing and segmentation of datasets and higher precision in detection of image objects.

**2.2.1 Advantages of the Proposed System**

* Involves fast processing of data
* Effective segmentation of datasets
* Higher precision in detection of image objects
* Cost effective

**2**

**2.3 Requirement Gathering**

**2.3.1 Functional Requirement**

Functional Requirements analysis is a software engineering technique that is composed of the various tasks that determine the needs or conditions that are to be met for a new or altered product taking into consideration the possible conflicting requirements of the various users. Functional requirements are those requirements that are used to illustrate the internal working nature of the system, the description of the system, and explanation of each subsystem It consists of what task the system should perform the processes involved, which data should the system holds and the interfaces with the user. The functional requirements identified are

* Importing the dependencies
* Data pre-processing
* Splitting the dataset to training and test the data
* Training the model.
* Evaluation or accuracy score
* Making predictive system

**2.3.2 Non-Functional Requirement**

**Performance**: Specifies the system's expected response time, throughput, and accuracy. It includes metrics such as prediction accuracy, training time, inference time, and model size.

**Scalability:** Defines how the system can handle increasing amounts of data, users, or computational resources. It ensures that the system can scale up or down to accommodate growing or fluctuating workloads.

**Reliability:** Addresses the system's ability to function correctly and consistently over time. It includes factors such as fault tolerance, error handling, and system recovery.

**3**

**Security:** Ensures that the machine learning system is protected against unauthorized access, data breaches, or malicious attacks. It covers areas such as data privacy, model security, secure communication, and access controls.

**Maintainability:** Refers to the ease with which the system can be modified, updated, or repaired. It includes factors such as code readability, documentation, modular design, and version control.

**Usability:** Focuses on the user experience and how easily users can interact with the system. It includes factors such as intuitive interfaces, clear documentation, user training, and error handling.

**4**

**SYSTEM SPECIFICATION**

**CHAPTER 3**

**SYSTEM SPECIFICATION**

**3.1 Hardware Specification**

Processor : Intel core i5

RAM : 8GB

Hard Disk : 512 GB

**3.2 Software Specification**

Operating system: Windows 10

Front-end : Python with Machine Learning

Backend : Google Drive

Editor : Google Colaboratory

**5**

**SOFTWARE DESCRIPTION**

**CHAPTER 4**

**SOFTWARE DESCRIPTION**

**4.1 Development Tools and Technologies**

**4.1.1 Python**

Python is a versatile and dynamic programming language that has gained immense popularity among developers and enthusiasts. Known for its simplicity and readability, Python allows users to write concise and elegant code, making it an ideal choice for beginners and experienced programmers alike. Its extensive library ecosystem provides a wide range of functionalities, enabling developers to efficiently tackle various tasks, from web development and data analysis to artificial intelligence and scientific computing.

Python's versatility extends beyond its syntax. Its cross-platform compatibility allows developers to write code once and run it on different operating systems without any major modifications. Furthermore, Python supports multiple programming paradigms, including procedural, object-oriented, and functional programming, providing flexibility in design and implementation.

Python's extensive standard library, coupled with its vast collection of third-party packages, makes it a powerful tool for diverse applications. Whether you need to interact with databases, create web applications using frameworks like Django or Flask, perform complex scientific computations with libraries like NumPy and SciPy, or dive into machine learning with TensorFlow or PyTorch, Python has you covered.

**6**

**4.1.2 Machine Learning**

Machine learning, a subfield of artificial intelligence, has revolutionized various industries by enabling computers to learn from data and make intelligent decisions. Python has emerged as a dominant language in the field of machine learning, providing researchers, data scientists, and developers with powerful tools and libraries to build intelligent systems.

Python's popularity in machine learning is largely attributed to its rich ecosystem of libraries. Libraries such as NumPy, Pandas, and Matplotlib offer efficient data manipulation, analysis, and visualization capabilities, allowing practitioners to preprocess and explore datasets easily. Scikit-learn provides a comprehensive set of algorithms for tasks such as classification, regression, clustering, and dimensionality reduction, making it a go-to choice for traditional machine learning tasks.

However, the true power of Python in machine learning lies in its deep learning frameworks. TensorFlow and PyTorch have gained significant traction and have become industry standards. TensorFlow, with its computational graph abstraction, offers scalability and deployment flexibility, making it suitable for large-scale machine learning projects. On the other hand, PyTorch's dynamic computational graph and intuitive interface make it a favorite among researchers and developers for prototyping and experimentation.

**7**

**4.1.3 Dataset**

For image classification tasks, the dataset typically contains images belonging to different classes or categories. Each image is associated with a specific label or class, indicating the object or concept it represents. The dataset aims to cover a diverse range of images, capturing various angles, backgrounds, lighting conditions, and variations within each class.

Labeling each image with the correct class or bounding box annotations demands manual annotation by human annotators, or in some cases, leveraging advanced techniques like active learning or crowdsourcing. The datasets need to be carefully curated to ensure a balanced representation of classes and variations, reducing biases and potential errors in the model training process.

Moreover, it is essential to split the dataset into training, validation, and test sets. The training set is used to train the machine learning models, while the validation set helps fine-tune the models and tune parameters. The test set serves as an independent evaluation benchmark to assess the model's performance on unseen data

In summary, datasets for image classification and detection in machine learning are carefully curated collections of labeled images. These datasets play a critical role in training and evaluating models, enabling them to learn patterns, generalize to unseen data, and perform accurate classification and object detection tasks. Proper dataset creation, annotation, splitting, and augmentation are essential for building robust and reliable machine learning models in the domain of image analysis.

**8**

**PROJECT DESCRIPTION**

**CHAPTER 5**

**PROJECT DESCRIPTION**

**5.1 Problem Definition**

The dataset of images and their corresponding labels or classes, the task is to develop a machine learning model that can accurately classify new images into the correct categories. The model should be able to generalize from the training data and correctly predict the class of previously unseen images.

The ultimate goal is to build models that can automatically analyse and understand visual content, enabling applications such as automated image categorization and more.

**5.2 Overview of the Project**

An image classification and detection project in machine learning involves building a model that can accurately classify images into different predefined categories or classes. Throughout the project, it is important to iterate and refine the steps based on the evaluation results and domain-specific requirements. Additionally, regular monitoring and retraining of the model may be necessary to maintain its accuracy as new data becomes available.

**5.3 Input Design**

Input design is the process of converting external user oriented description of the input systems to a machine-oriented format. Also there is major activities of the input design are,

* Data Collection
* Data Pre-processing
* Splitting the Dataset
* Data Loading

**9**

* Model Selection
* Model Training
* Hyperparameter Tuning
* Model Evaluation
* Deployment and Inference

**5.4 Output Design**

Output design in an image classification and detection project refers to how the results or outputs of the machine learning models are presented or utilized. An overview of the output design for image classification and detection projects,

1. Classification Output

* Predicted Class Labels
* Probability Scores

1. Detection Output

* Class Labels
* Confidence Scores

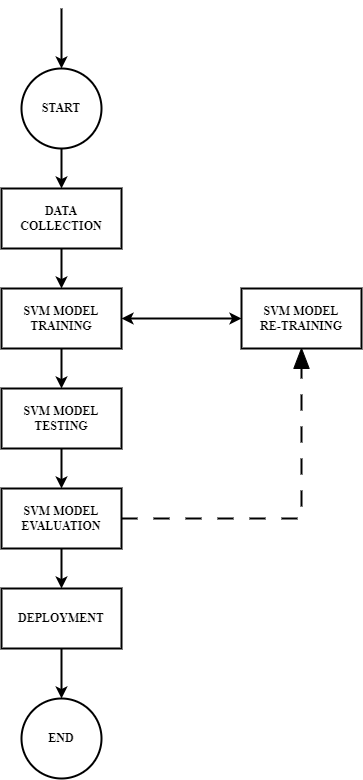
1. Visualization

By carefully designing the output presentation and considering the specific needs and requirements, the outputs of image classification and detection models can be effectively utilized, analyzed, and integrated into various applications and workflow

**10**

**5.5 System Design**

**5.5.1 System Flow Diagram**

****

**11**

**SYSTEM IMPLEMENTATION**

**CHAPTER 6**

**SYSTEM IMPLEMENTATION**

**6.1 Methodologies**

The Implementation stage in a system involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the change in our procedures, and evaluation of change over methods.

The implementation plan consists of the following steps.

**Support Vector Machine**:

“Support Vector Machine” (SVM) is a supervised machine learning algorithm that can be used for both classification or regression challenges. However, it is mostly used in classification problems. In this SVM algorithm, we plot each data item as a point in n-dimensional space (where n is the number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well.

**Some of the key parameters in SVM are:**

**→ Gamma :** defines how far the influence of single training examples reaches values leads to biased results.

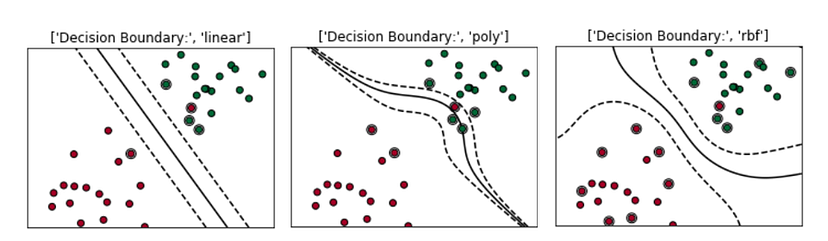
**→ C :** Controls the cost of miscalculations

1. **Small C** — makes the cost of misclassification LOW
2. **Large C** — makes the cost of misclassification HIGH

**→ Kernel :** SVM algorithms use a set of mathematical functions that are defined as the kernel.

**Types of Kernels**: Linear, RBF(Radial Basis Function), Polynomial Kernel

**12**

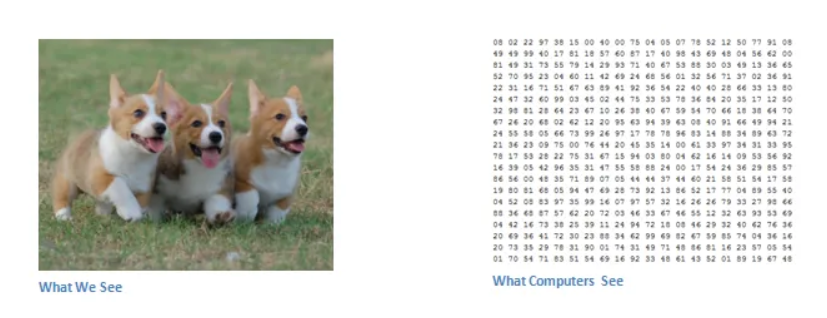
****

**Fig 6.1.1 Types of Decision Boundary**

**Computer Readable Image:**

The main task of image Classification is to read the input image, the computer sees the image quite differently:

The computer sees the image as an array of pixels, if the size of the image is 200 X 200, the size of the array will be 200 X 200 X 3 wherein the first 200 is the width and second 200 is height and the next 3 is RGB channel values. The values in the array would range from 0–255 which describes the intensity of the pixel at each point.

****

**Fig 6.1.2 Computer Readable Image**

**13**

**GridSearchCV**

It is a library function that is a member of sklearn’s model\_selection package. It helps to loop through predefined hyperparameters and fit your estimator (model) on your training set. So, in the end, you can select the best parameters from the listed hyperparameters.

**Process**

It is one of the ways of machine learning where the model is trained by input data and expected output data.

**To Create a Model**

1.Data Collection

2. Data Preprocessing

3. Splitting the Dataset

4. Data Loading

5.Model Selection

6. Model Training

7. Hyperparameter Tuning

8. Model Evaluation

9. Deployment and Inference

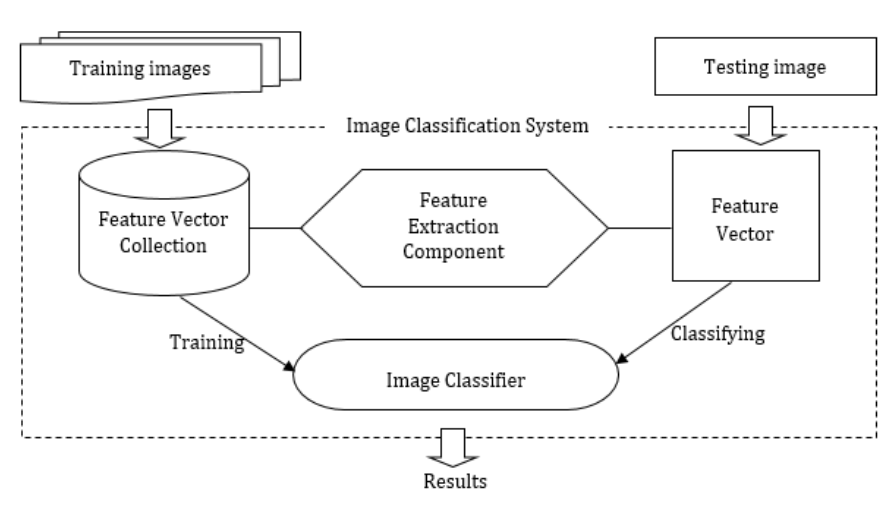
* **Data Collection:** Collect a diverse and representative dataset of images for training and evaluation. The dataset should include images from each class you want the model to classify.
* **Data Preprocessing:** Prepare the dataset by performing necessary preprocessing steps such as resizing images to a consistent size, normalizing pixel values, and potentially augmenting the dataset to increase its size and diversity.
* **Splitting the Dataset:** Divide the dataset into training, validation, and testing subsets. The training set is used to train the model, the validation set is used for hyperparameter tuning and model evaluation during training, and the testing set is used to assess the final performance of the trained model.

**14**

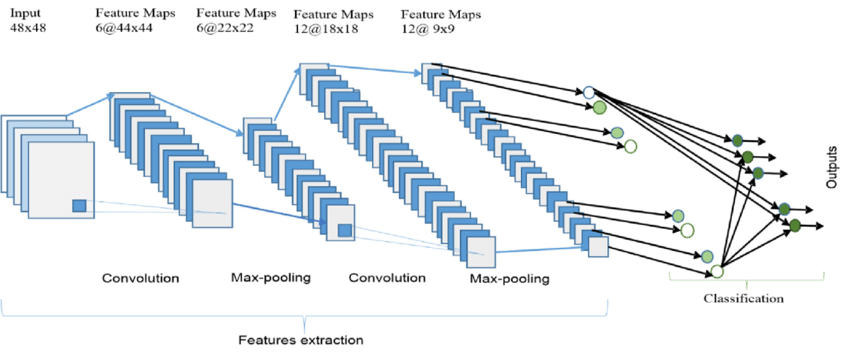
* **Data Loading**: Implement a data loading mechanism to efficiently load and process the input images during training and inference. Depending on the size of the dataset, you may need to consider strategies like batch loading, parallel loading, or utilizing data loading libraries and frameworks optimized for image data
* **Model Selection:** Choose an appropriate model architecture for image classification. Convolutional Neural Networks (CNNs) are commonly used for this task due to their ability to learn spatial hierarchies from images. You can select a pre-existing CNN architecture or design your own model architecture.
* **Model Training**: Train the selected model using the training dataset. During training, the model learns to recognize patterns and features in the images and adjusts its internal parameters to minimize the prediction error.
* **Hyperparameter Tuning:** Optimize the hyperparameters of the model, such as learning rate, batch size, and regularization parameters, using the validation set. This process involves adjusting the hyperparameters and evaluating the model's performance on the validation set until satisfactory results are achieved.
* **Model Evaluation:** Evaluate the trained model using the testing dataset to assess its performance on unseen data. Common evaluation metrics include accuracy, precision, recall, and F1 score. Analyze the model's performance and identify any potential issues or areas for improvement.
* **Deployment and Inference:** Once the model meets the desired performance criteria, it can be deployed for real-world applications. The trained model can take input images and provide predictions of their classes or labels. Ensure the model performs well in different scenarios and handle any specific requirements for the deployment environment.

**15**

**6.2 SYSTEM ARCHITECTURE**

****

**Fig 6.2.1 Training and Testing Phase**

****

**Fig 6.2.2 CNN Architecture**

**16**

**CONCLUSION AND FUTURE ENHANCEMENT**

**CHAPTER 7**

**CONCLUSION & FUTURE ENHANCEMENTS**

**7.1 Conclusion**

To conclude, image classification and detection using machine learning, particularly with Support Vector Machines (SVM), have proven to be effective in various applications.

In this project, I assembled and trained the SVM model to classify images of ice cream cone, cricket ball, and cars. I used GridSearchCV to find out the best parameters for SVM to classify the images and have measured the accuracy of the model.

**7.2 Scope for Future Development**

The Future enhancements in image classification and detection using SVM can involve exploring deep learning approaches, leveraging transfer learning, utilizing ensemble methods, optimizing for real-time and embedded systems, incorporating multi-modal approaches, and improving explainability and interpretability. These advancements can further enhance the accuracy, efficiency, and applicability of image analysis systems in various domains.

**17**

**APPENDICES**

**CHAPTER 8**

**APPENDICES**

**8.1 Source Code**

import pandas as pd

from sklearn import svm

from sklearn.model\_selection import GridSearchCV

import os

import matplotlib.pyplot as plt

from skimage.transform import resize

from skimage.io import imread

import numpy as np

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import classification\_report,accuracy\_score,confusion\_matrix

import pickle

Categories=['Cars','Ice cream cone','Cricket ball']

print("Type y to give categories or type n to go with classification of Cars,Ice Cream cone and Cricket ball");

while(True):

check=input()

if(check=='n' or check=='y'):

break

print("Please give a valid input (y/n)")

if(check=='y'):

print("Enter How Many types of Images do you want to classify")

n=int(input())

Categories=[]

print(f'please enter {n} names')

**18**

for i in range(n):

name=input()

Categories.append(name)

print(f"If not drive Please upload all the {n} category images in google collab with the same names as given in categories")

from google.colab import drive

drive.mount('/content/gdrive')

!touch "/content/gdrive/My Drive/ML"

flat\_data\_arr=[]

target\_arr=[]

#please use datadir='/content' if the files are upload on to google collab

#else mount the drive and give path of the parent-folder containing all category images folders.

datadir='/content/gdrive/My Drive/ML'

for i in Categories:

print(f'loading... category : {i}')

path=os.path.join(datadir,i)

for img in os.listdir(path):

img\_array=imread(os.path.join(path,img))

img\_resized=resize(img\_array,(150,150,3))

flat\_data\_arr.append(img\_resized.flatten())

target\_arr.append(Categories.index(i))

print(f'loaded category:{i} successfully')

flat\_data=np.array(flat\_data\_arr)

target=np.array(target\_arr)

df=pd.DataFrame(flat\_data)

df['Target']=target

df

**19**

x=df.iloc[:,:-1]

y=df.iloc[:,-1]

x\_train,x\_test,y\_train,y\_test=train\_test\_split(x,y,test\_size=0.20,random\_state=77,stratify=y)

print('Splitted Successfully')

param\_grid={'C':[0.1,1,10,100],'gamma':[0.0001,0.001,0.1,1],'kernel':['rbf','poly']}

svc=svm.SVC(probability=True)

print("The training of the model is started, please wait for while as it may take few minutes to complete")

model=GridSearchCV(svc,param\_grid)

model.fit(x\_train,y\_train)

print('The Model is trained well with the given images')

model.best\_params\_

y\_pred=model.predict(x\_test)

print("The predicted Data is :")

y\_pred

print("The actual data is:")

np.array(y\_test)

#classification\_report(y\_pred,y\_test)

print(f"The model is {accuracy\_score(y\_pred,y\_test)\*100}% accurate")

#confusion\_matrix(y\_pred,y\_test)

pickle.dump(model,open('img\_model.p','wb'))

print("Pickle is dumped successfully")

#print(os.path.abspath(os.getcwd()))

model=pickle.load(open('img\_model.p','rb'))

**20**

url=input('Enter URL of Image')

img=imread(url)

plt.imshow(img)

plt.show()

img\_resize=resize(img,(150,150,3))

l=[img\_resize.flatten()]

probability=model.predict\_proba(l)

for ind,val in enumerate(Categories):

print(f'{val} = {probability[0][ind]\*100}%')

print("The predicted image is : "+Categories[model.predict(l)[0]])

print(f'Is the image a {Categories[model.predict(l)[0]]} ?(y/n)')

while(True):

b=input()

if(b=="y" or b=="n"):

break

print("please enter either y or n")

if(b=='n'):

print("What is the image?")

for i in range(len(Categories)):

print(f"Enter {i} for {Categories[i]}")

k=int(input())

while(k<0 or k>=len(Categories)):

print(f"Please enter a valid number between 0-{len(Categories)-1}")

k=int(input())

print("Please wait for a while for the model to learn from this image :)")

flat\_arr=flat\_data\_arr.copy()

tar\_arr=target\_arr.copy()

tar\_arr.append(k)

flat\_arr.extend(l)

**21**

tar\_arr=np.array(tar\_arr)

flat\_df=np.array(flat\_arr)

df1=pd.DataFrame(flat\_df)

df1['Target']=tar\_arr

model1=GridSearchCV(svc,param\_grid)

x1=df1.iloc[:,:-1]

y1=df1.iloc[:,-1]

x\_train1,x\_test1,y\_train1,y\_test1=train\_test\_split(x1,y1,test\_size=0.20,random\_state=77,stratify=y1)

d={}

for i in model.best\_params\_:

d[i]=[model.best\_params\_[i]]

model1=GridSearchCV(svc,d)

model1.fit(x\_train1,y\_train1)

y\_pred1=model.predict(x\_test1)

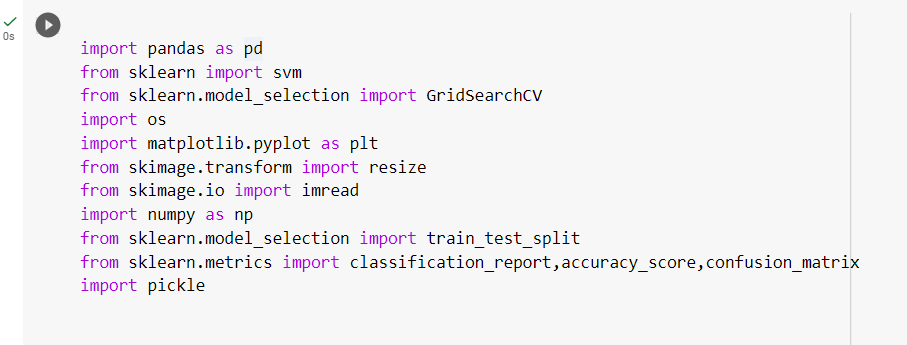
print(f"The model is now {accuracy\_score(y\_pred1,y\_test1)\*100}% accurate")

pickle.dump(model1,open('img\_model.p','wb'))

print("Thank you for your feedback")

**22**

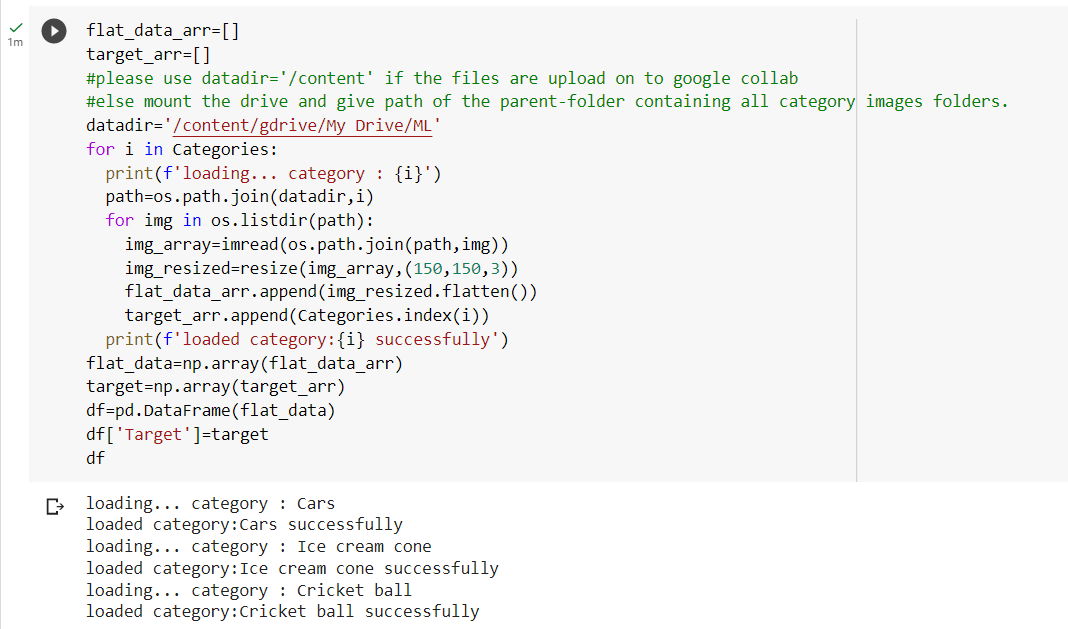
**8.2 Screen Shots**

****

**Fig 8.2.1 Importing Libraries**

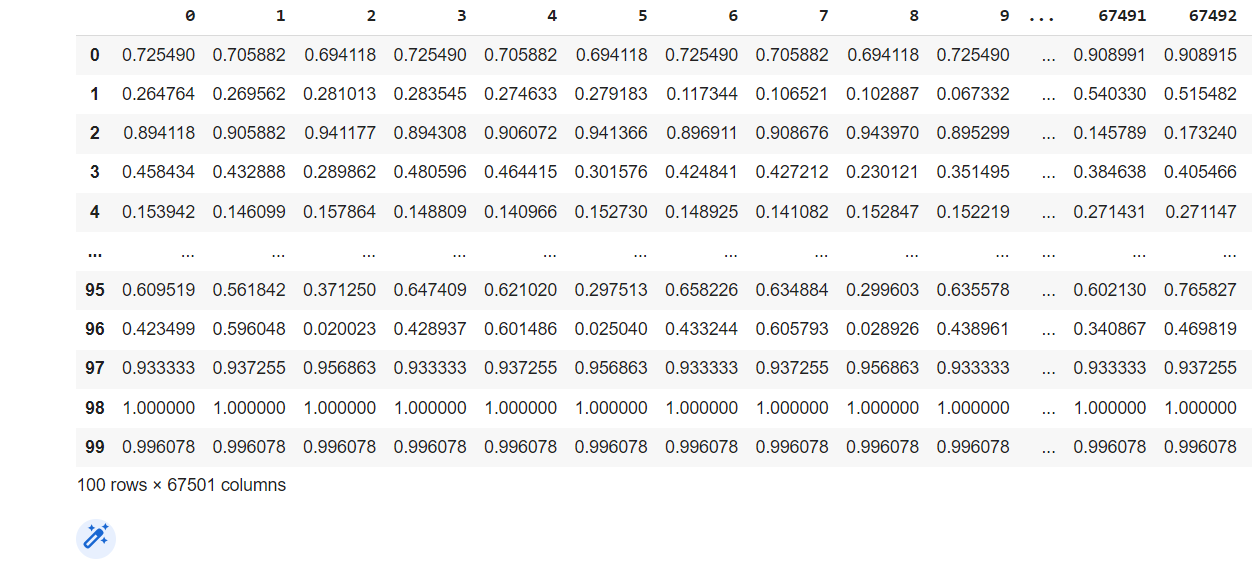
****

**Fig 8.2.3 Backend Connection**

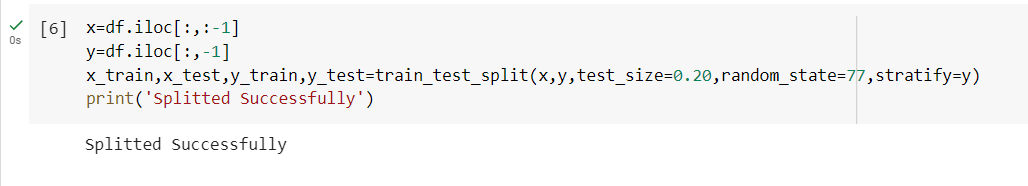
****

**Fig 8.2.3 Loading Data**

**23**

****

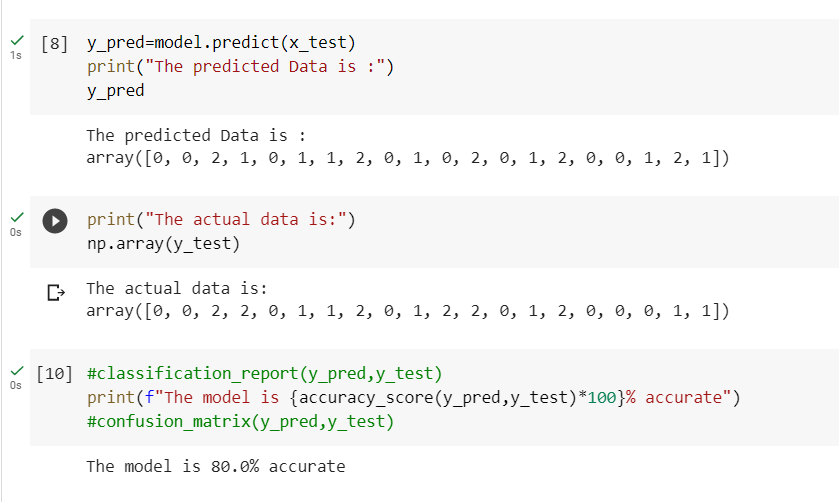
**Fig 8.2.4 Pre-processed Dataset**

****

**Fig 8.2.5 Splitting the Dataset**

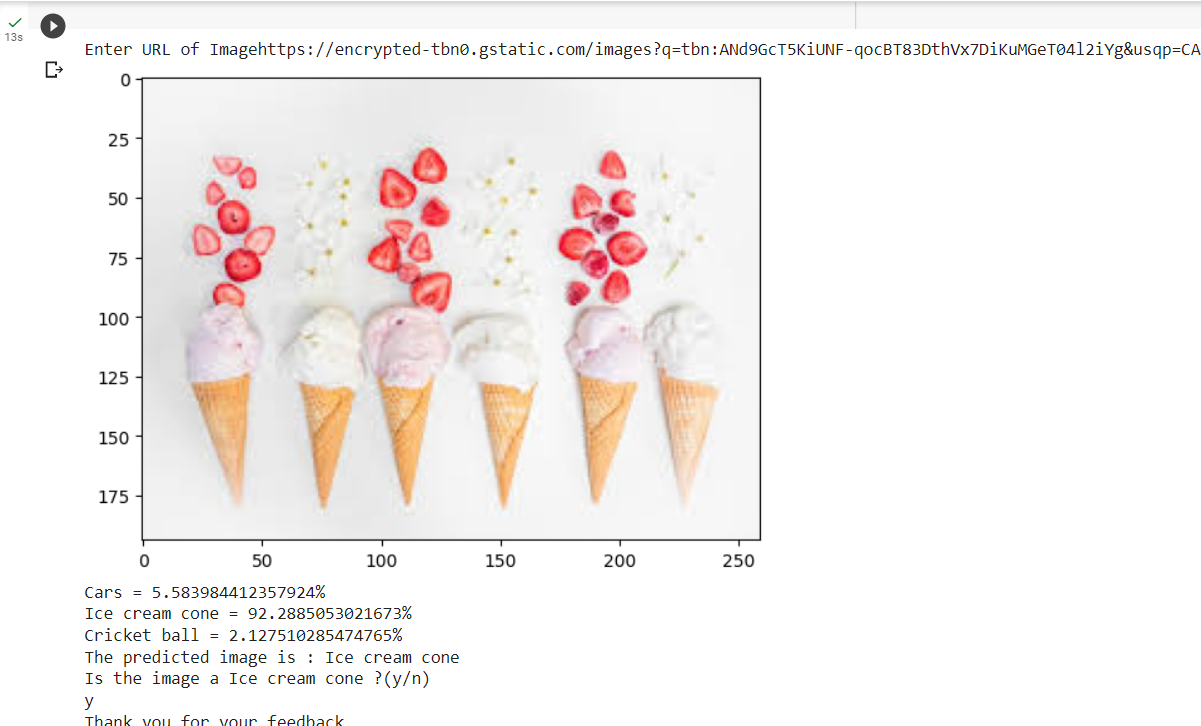
****

**Fig 8.2.6 Training the Model**

****

**Fig 8.2.7 Accuracy of the Dataset**

**24**

****

**Fig 8.2.8 Prediction of the Trained Model**

**25**

**REFERENCES**

**REFERENCES**

**Book Reference**

1. Shai Shalev- Shwartz and Shai Ben-David. "Understanding the machine learning:From theory to Algorithms" First published 2014. Library of Congress Cataloging in Publication data.

2. Allen B.Downey, "Think python How to think like a computer scientist", O'Reilly, Media,Inc.

**Web Reference**

1. <https://medium.com/analytics-vidhya/image-classification-using-machine-learning-support-vector-machine-svm-dc7a0ec92e01>
2. <https://developers.google.com/machine-learning/practica/image-classification>

**26**