A Project Report

on

Image Classifier Using ML

*carried out as part of the **Minor Project IT3270** Submitted*

by

Manas Jain
199302120

in partial fulfilment for the award of the degree of

Bachelor of Technology

in

Information Technology

MANIPAL UNIVERSITY
JAIPUR

School of Computing and Information Technology
Department of Information Technology

MANIPAL UNIVERSITY JAIPUR
JAIPUR-303007
RAJASTHAN, INDIA

May 2022

CERTIFICATE

Date: 20/05/2022

This is to certify that the minor project titled **Image Classifier Using Machine Learning** is a record of the bonafide work done by **Manas Jain** (199302120) submitted in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology in Information Technology of Manipal University Jaipur, during academic year 2021-22.

Mr. Dhananjay Kumar Singh
Project Guide, Department of Information Technology
Assistant Professor
Manipal University Jaipur

Dr. Pankaj Vyas
HoD, Department of Information Technology
Manipal University Jaipur

ABSTRACT

People are able to recognize the environment they are in as well as the various objects in their everyday life no matter the influence on the item's features or if their view is obstructed, as this is one of the very first skills, we learn from the moment we are born. Computers, on the other hand, require effort and powerful computation and complex algorithms to attempt to recognize correctly patterns and regions where a possible object might be. Object detection and recognition are two main ways that have been implemented over multiple decades that are at the centre of Computer Vision systems at the moment.

Here we will introduce one of the core problems in computer vision, which is image classification. It is defined as the task of classifying an image from a fixed set of categories. Many other computer vision challenges such as object detection and segmentation can be reduced to image classification. Throughout this project, we will start by exploring our dataset, then show how to pre-process and prepare the images to be a valid input for our learning algorithms. Finally, we will explore about the SVM (Support Vector Machine); the machine learning algorithm that has been implemented in the project.

LIST OF FIGURES

TABLE OF CONTENTS

1. Introduction

1.1. Introduction

We live in the era of data. With the Internet of Things and Artificial Intelligence becoming ubiquitous technologies, we now have huge volumes of data being generated. Differing in form, data could be speech, text, image, or a mix of any of these. In the form of photos or videos, images make up for a significant share of global data creation.

Since the vast amount of image data we obtain from cameras and sensors is unstructured, we depend on advanced techniques such as machine learning algorithms to analyze the images efficiently. Image classification is probably the most important part of digital image analysis. It uses AI-based deep learning models to analyze images with results that for specific tasks already surpass human-level accuracy (for example, in face recognition).

There are various factors that make Image Classification as motivational field of study, such as:

- It makes computer vision a possibility, hence enhancing power of Artificial Intelligence.
- In today's industry there is significant interest in creating light weight mobile systems that can identify objects using this system of image classification and computer vision.
- Numerous practical applications such as medical imaging, object identification in satellite images, traffic control systems, brake light detection, machine vision, and more; making Image Classification a motivating field of study.

1.2. **Problem Statement**

To build an image classifier using any machine learning algorithm such that in a given set of images and prior knowledge about the content of the images we will able to find the correct semantic label for the pixels in the image/s

1.3. **Objectives**

- To design an image classifier program such that it will improve efficiency of the normal user to classify and identify images.
- No prior knowledge will be needed for the user to use this system. Thus, it will be a user-friendly program.

1.4. **Scope of Project**

Some areas of study that will be covered in this project work are:

i. Determining suitable Classification system
ii. Selecting and Training of image data
iii. Image Preprocessing
iv. Feature Extraction
v. Selection of appropriate classification algorithm; like SVM.
vi. Accuracy assessment

2. Background Detail

2.1. Conceptual Overview / Literature Review

Image classification is the process of categorizing and labeling groups of pixels or vectors within an image based on specific rules. The categorization law can be devised using one or more spectral or textural characteristics. Two general methods of classification are ‘supervised’ and ‘unsupervised’.

Unsupervised classification method is a fully automated process without the

use of training data. Using a suitable algorithm, the specified characteristics of an image is detected systematically during the image processing stage. The classification methods used in here are 'image clustering' or 'pattern recognition'.

Supervised classification method is the process of visually selecting samples (training data) within the image and assigning them to pre-selected categories (i.e., roads, buildings, water body, vegetation, etc.) in order to create statistical measures to be applied to the entire image. 'maximum likelihood' and 'minimum distance' are two common methods to categorize the entire image using the training data. For example, 'maximum likelihood' classification uses the statistical characteristics of the data where the mean and standard deviation values of each spectral and textural indices of the image are computed first. Then, considering a normal distribution for the pixels in each class and using some classical statistics and probabilistic relationships, the likelihood of each pixel to belong to individual classes is computed. Finally, the pixels are labeled to a class of features that show the highest likelihood.

Some examples of image classification include:

- Labeling an x-ray as cancer or not (binary classification).
- Classifying a handwritten digit (multiclass classification).
- Assigning a name to a photograph of a face (multiclass classification)

<u>Structure of an Image Classification Task</u>

i. Image Preprocessing - The aim of this process is to improve the image data(features) by suppressing unwanted distortions and enhancement of some important image features so that our Computer Vision models can benefit from this improved data to work on.
ii. Detection of an object - Detection refers to the localization of an object which means the segmentation of the image and identifying the position of the object of interest.
iii. Feature extraction and Training- This is a crucial step wherein statistical or deep learning methods are used to identify the most interesting patterns of

the image, features that might be unique to a particular class and that will, later on, help the model to differentiate between different classes. This process where the model learns the features from the dataset is called model training.

iv. Classification of the object - This step categorizes detected objects into predefined classes by using a suitable classification technique that compares the image patterns with the target patterns.

v. Accuracy assessment - An accuracy assessment is realized to identify possible sources of errors and as an indicator used in comparisons.

Classification Technique Used-SVM:

i. A support vector machine builds a hyper plane or set of hyper planes in a high or infinite dimensional space used for classification.

ii. Good separation is achieved by the hyper plane that has the largest distance to the nearest training data point of any class, generally larger the margin lower the generalization error of the classifier.

iii. SVM uses non-parametric with binary classifier approach and can handle more input data very efficiently.

iv. Performance and accuracy depend on the Hyper-plane selections and kernel parameter.

v. Advantages and Disadvantages of SVM: -

- Advantage
  a. SVM works relatively well when there is a clear margin of separation between classes.
  b. SVM is more effective in high dimensional spaces.
  c. SVM is effective in cases where the number of dimensions is greater than the number of samples.
  d. SVM is relatively memory efficient.

- Disadvantage: -
  a. SVM algorithm is not suitable for large data sets.
  b. SVM does not perform very well when the data set has more noise i.e.,

target classes are overlapping.

c. In cases where the number of features for each data point exceeds the number of training data samples, the SVM will underperform.

3. System Design & Methodology

3.1. System Architecture

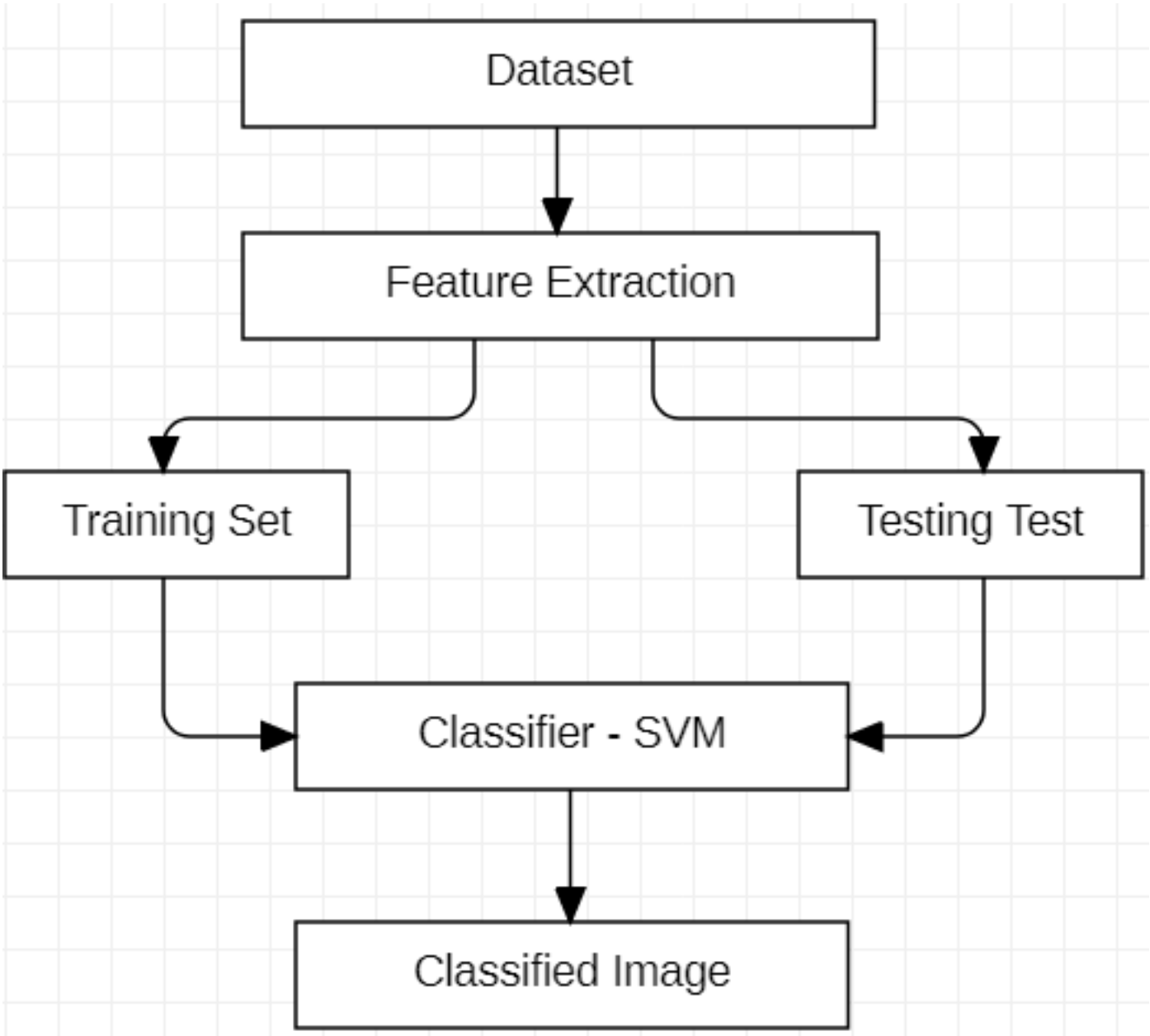


Figure 1

3.2. Development Environment. (H/w & S/W)

- Software:
    a) Google Colab
    b) Python 3.9.7
    c) Windows 7 or above

- Hardware:
    a) PC – 8gb RAM
    b) Processor – Intel Core i7 8th Gen

3.3. Methodology: Algorithm/Procedures

ML Algorithm that has been used here is SVM (Support Vector Machine). It is a supervised machine learning algorithm which can be used for classification or regression problems. It uses a technique called the kernel trick to transform your data and then based on these transformations it finds an optimal boundary between the possible outputs. The most commonly used kernels are:

- Linear Kernel
- Gaussian Kernel
- Polynomial Kernel

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.

Good separation is achieved by the hyper plane that has the largest distance to the nearest training data point of any class, generally larger the margin lower the generalization error of the classifier.

Performance and accuracy depend on the Hyper-plane selections and kernel parameter.

Simply put, it does some extremely complex data transformations, then figures out how to separate your data based on the labels or outputs you've defined.

Work Flow Chart

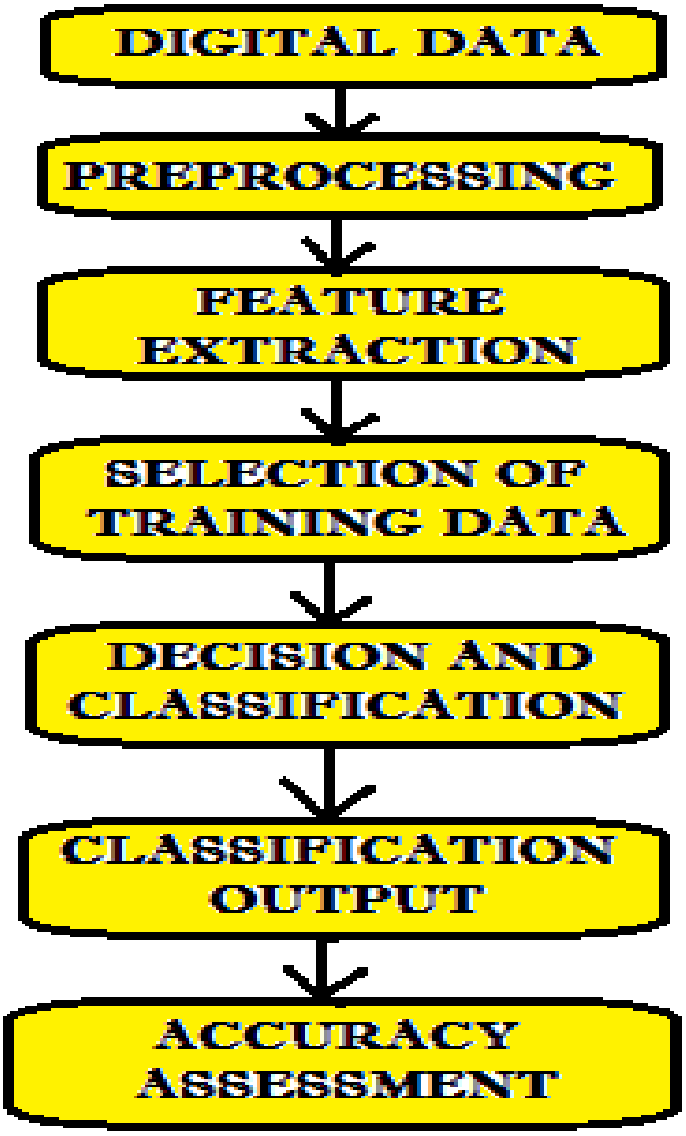


Figure 2

4. Implementation and Result

4.1. Modules/Classes of Implemented Project

Dataset Used:

Dataset used in this project to train the model is manually downloaded from the web and stored in google drive. There are 3 categories of image classes in the training dataset namely, 'pretty sunflower', 'rugby ball leather', 'ice cream cone'. Each image class has 40 images each and total images in the dataset is 40*3 (i.e.,120).

Libraries used:

- OS - The OS module in Python provides functions for interacting with the operating system. OS comes under Python's standard utility modules. 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.

- Matplotlib – It is a cross-platform, data visualization and graphical plotting library for Python and its numerical extension NumPy. As such, it offers a viable open-source alternative to MATLAB. Developers can also use matplotlib's APIs (Application Programming Interfaces) to embed plots in GUI applications.
  - matplotlib.pyplot- It is a plotting library used for 2D graphics in python programming language. It can be used in python scripts, shell, web application servers and other graphical user interface toolkits.

- NumPy- It stands for 'Numerical Python'. It is an open-source Python library used to perform various mathematical and scientific tasks. It contains multi-dimensional arrays and matrices, along with many high-level mathematical functions that operate on these arrays and matrices.

- skimage - scikit-image (a.k.a. skimage) is an image processing Python package that works with NumPy arrays which is a collection of algorithms for image processing.
  - skimage.io.imread - Image reading and writing via imread.
  - skimage.transform - Resize image to match a certain size.

- sklearn - Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib.
  - sklearn.model_selection.train_test_split - Split arrays or matrices into random train and test subsets.
  - sklearn.model_selection.GridSearchCV - 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.

    - Exhaustive search over specified parameter values for an estimator.
    - Important members are fit, predict.
    - GridSearchCV implements a “fit” and a “score” method. It also implements “score_samples”, “predict”, predict_proba”, “decision_function”, “transform” and “inverse_transform” if they are implemented in the estimator used.
    - The parameters of the estimator used to apply these methods are optimized by cross-validated grid-search over a parameter grid.

  - sklearn.metrics.accuracy_score - Accuracy classification score. In multilabel classification, this function computes subset accuracy: the set of labels predicted for a sample must exactly match the corresponding set of labels in y_true.
  - sklearn.metrics.confusion_matrix - Compute confusion matrix to evaluate the accuracy of a classification.

- Pickle - The pickle module implements binary protocols for serializing and de-serializing a Python object structure. “Pickling” is the process whereby a Python object hierarchy is converted into a byte stream, and “unpickling” is the inverse operation, whereby a byte stream (from a binary file or bytes-like object) is converted back into an object hierarchy.

4.2. Implementation Detail

1. Gathering Dataset - Dataset used in this project to train the model is manually downloaded from the web and stored in google drive. There are 3 categories of image classes in the training dataset namely, ‘pretty sunflower’, ‘rugby ball leather’, ‘ice cream cone’. Each image class has 40 images each resulting total images in the dataset is 40*3 (i.e., 120).

2. Label encoding the values for CATEGORIES – Indexing all the three classes in CATEGORIES i.e., 0 for ‘pretty sunflower’, 1 for ‘rugby ball leather’, 2 for ‘ice cream cone’ respectively.

3. Creating Path – using os library a path is created to access all the images in the categories by performing join operation

```
for category in CATEGORIES:
  class_num = CATEGORIES.index(category)  # label encoding the values
  path = os.path.join(DATADIR,category)   # creating path to use all the images
  for img in os.listdir(path):
   img_array = imread(os.path.join(path,img))
```

4. Preprocessing –
    - Resizing – images are resized into dimension of 150,150,3 and pixel values gets normalized between 0 to 1
    - Flattening – resized images are converted from 2D to 1D vector and appended into flat data

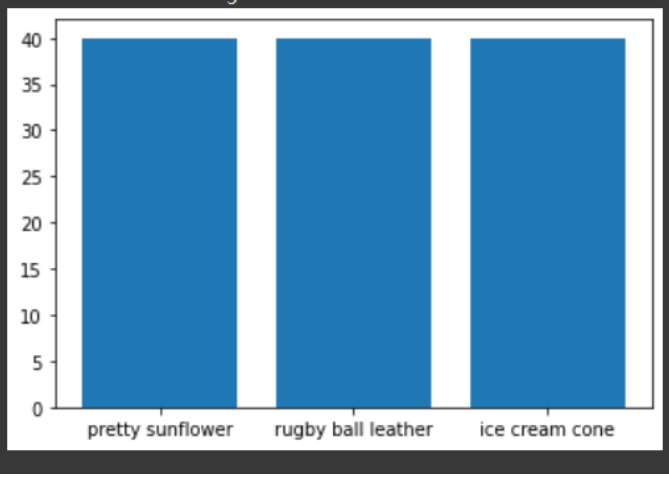
```
#preprocessing
  # 1. Resizing
  # 2. Flattening-->to convert image ie 2D into a vector 1D
   img_resized = resize(img_array,(150,150,3))   #Normalizes pixel values of imgs b/w 0 to 1
   flat_data.append(img_resized.flatten())
   images.append(img_resized)
   target.append(class_num)

flat_data = np.array(flat_data)
target = np.array(target)
images = np.array(images)
```

5. Plotting a Bar Graph – A bar Graph is plotted showing each category and numer of image data it contains

```
[ ]  unique,count = np.unique(target,return_counts=True)
     plt.bar(CATEGORIES,count)
```

```
<BarContainer object of 3 artists>
```



6. Splitting the data into training and testing - Image data is split into x_train , x_test and y_train, y_test using train_test_split from sklearn.model_selection library.test size is kept 30%.

```
#Split data into Training and testing
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(flat_data,target,test_size=0.3,random_state=109)
```

7. Model Training Using SVM – With the help of GridSearchCV we determined the parameters in param_grid for SVM. GridSearchCV 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.

   In param_grid:

   - if kernel is 'linear' then, 'C': [1,10,100,1000]
   - if kernel is 'rbf' then, 'C': [1,10,100,1000],'gamma': [0.001,0.0001]

   Now, model is trained using training data in this way

   → model.fit (training data, expected_output)

```
#Model Training
from sklearn.model_selection import GridSearchCV
from sklearn import svm
param_grid = [
              {'C':[1,10,100,1000],'kernel':['linear']},
              {'C':[1,10,100,1000],'gamma':[0.001,0.0001],'kernel':['rbf']},
]
svc = svm.SVC(probability=True)
clf = GridSearchCV(svc,param_grid)
clf.fit(x_train , y_train)
```

8. Model testing: Now the model is tested using testing data in this way

   → model.predict(testing data)

   The accuracy of the model can be calculated using the accuracy_score() method from sklearn.metrics.

```
#Model Testing
y_pred = clf.predict(x_test)
print("The predicted Data is :")
print(y_pred)
print("The actual data is:")
print(np.array(y_test))
print(f"The model is {accuracy_score(y_pred,y_test)*100}% accurate")
```

```
The predicted Data is :
[0 2 2 1 2 0 0 2 1 1 0 2 0 2 2 0 0 2 0 2 1 1 2 0 0 0 2 2 0 1 1 1 2 2 1 0]
The actual data is:
[0 2 2 1 2 0 0 2 1 2 0 2 0 2 2 0 0 2 0 2 1 1 2 0 0 0 2 2 2 1 1 1 1 2 2 0]
The model is 88.88888888888889% accurate
```

9. Saving and loading the model using Pickle library

```
# save and load the model using Pickle library
import pickle
model = pickle.load(open('/content/drive/MyDrive/Minor_project_model/img_model.p','rb'))
print("Pickle is loaded successfully")
```

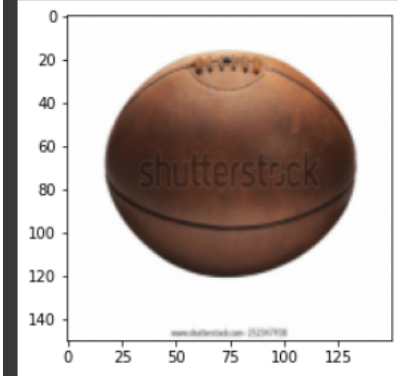
```
Pickle is loaded successfully
```

10. Model Evaluation: Finally, in the Model evaluation phase, the model generated can be used to evaluate new data.

```
#Evaluating a brand new Image
flat_data = []
url = input('Enter your URL')
img = imread(url)
img_resized = resize(img,(150,150,3))
flat_data.append(img_resized.flatten())
flat_data = np.array(flat_data)
print(img.shape)
plt.imshow(img_resized)
probability=model.predict_proba(flat_data)
for ind,val in enumerate(CATEGORIES):
    print(f'{val} = {probability[0][ind]*100}%')
print("The PREDICTED IMAGE is : "+CATEGORIES[model.predict(flat_data)[0]])
```

The final output would be like this:

```
Enter your URLhttps://image.shutterstock.com/image-photo/old-vintage-tan-rugby-ball-600w-252347938.jpg
(430, 600, 3)
pretty sunflower = 1.266147970241737%
rugby ball leather = 96.60033124084256%
ice cream cone = 2.133520788915681%
The PREDICTED IMAGE is : rugby ball leather
```



4.3 Results and Discussion

When the random live image is feeded into the model, it predicts the possible class for the image it would belong. Like for example if we feed the model an image of rugby ball leather with the help of URL, it will try to find the appropriate class for that image. The predicted output class is displayed along with its probability and dimension. The probability of classes other than predicted one is also displayed.

4.4 Month wise plan of work (Progress Chart/Time Line Chart)

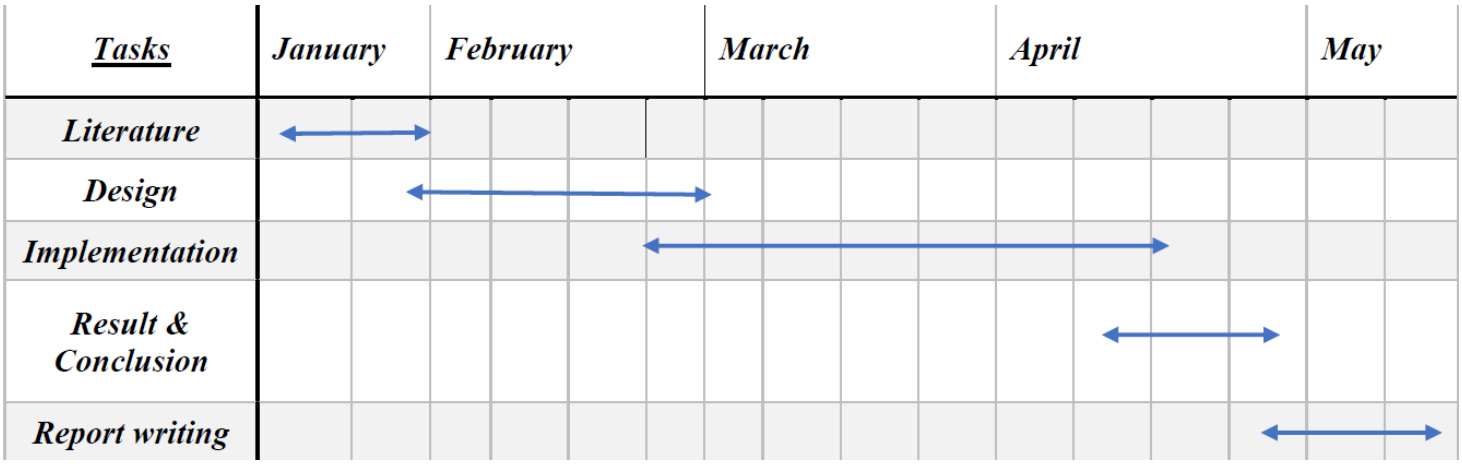


Figure 3

5. Conclusion and Future Plan

In this project work, I assembled and trained the SVM model to classify images of pretty sunflower, rugby ball leather and ice cream cones. I used GridSearchCV to find out the best parameters for SVM to classify the images and have measured the accuracy of the model.

My future plan is to work upon image recognition and processing in Autonomous driving vehicles further advancing my knowledge in Machine Learning and Deep Learning domain.

References

[1] Python: https://www.python.org/

[2] Javatpoint. “Best Python libraries for Machine Learning” https://www.javatpoint.com/best-python-libraries-for-machine-learning

[3] Vegi Shanmukh. “Image Classification Using MachineLearning-SVM”: https://medium.com/analytics-vidhya/image-classification-using-machine-learning-support-vector-machine-svm-dc7a0ec92e01

[4] Taru Jain. “Basics of Image Classification Techniques in Machine Learning”: https://iq.opengenus.org/basics-of-machine-learning-image-classification-techniques/

[5] The Python Standard Library: https://docs.python.org/3/library/

[6] Google Colab: https://colab.research.google.com/