INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact Factor: 6.078
(Volume 7, Issue 4 - V7I4-1865)
Available online at: https://www.ijariit.com

Image classification

Guru S. S.
saiguru0321@gmail.com
Prathyusha Engineering College, Thiruvallur, Tamil Nadu

Giridharan K.
giridharankumar05@gmail.com
Prathyusha Engineering College, Thiruvallur, Tamil Nadu

Mohammed Fahad
mohammedfahadshahulhameed@gmail.com
Prathyusha Engineering College, Thiruvallur, Tamil Nadu

Fahid Akthar A.
fahidakthar@gmail.com
Prathyusha Engineering College, Thiruvallur, Tamil Nadu

IMAGE CLASSIFICATION USING DEEP LEARNING

Submitted by

FAHID AKTHAR A

GIRIDHARAN K

GURU SS

MOHAMMED FAHAD

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

PRA THYUSHA ENGINEERING COLLEGE

TABLE OF CONTENTS

ABSTRACT

PROBLEM STATEMENT

More than 25% of the entire revenue in E-Commerce is attributed to apparel & accessories. A major problem they face is categorizing these apparels from just the images especially when the categories provided by the brands are inconsistent.

There are potentially *n* number of categories in which a given image can be classified. Manually checking and classifying images is a very tedious process. The task becomes near impossible when we're faced with a massive number of images, say 10,000 or even 100,000. How useful would it be if we could automate this entire process and quickly label images per their corresponding class?

WE DO THIS USING IMAGE CLASSIFICATION...BUT WHAT IS IMAGE CLASSIFICATION?

Consider the following example.....

You are given an image of a car...and asked what image it is? A car...isn't it obvious. Well then how do you come to this conclusion ...? Take a step back and analyze how you came to this conclusion – you were shown an image and you classified the class it belonged to (a car, in this instance). And that, in a nutshell, is what image classification is all about.

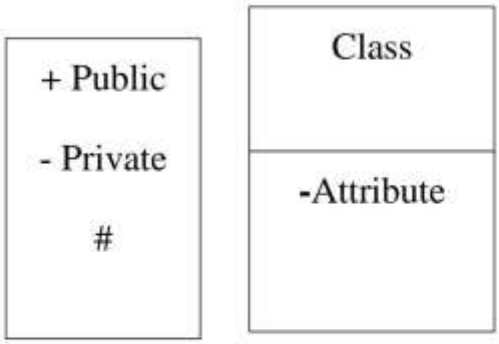
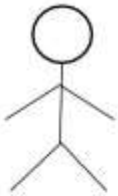
Image classification is the process of categorizing and labeling groups of pixels or vectors within an image based on specific rules. The categorization law can be devised using one or more spectral or textural characteristics. Two general methods of classification are 'supervised' and 'unsupervised'.

LIST OF FIGURES

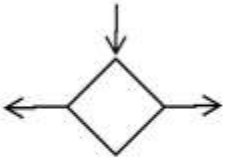
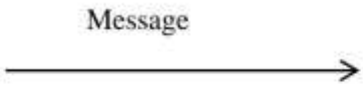
LIST OF ABBREVIATION

IDE	**I**ntegrated **D**evelopment **E**nvironment
SQL	**S**tructured **Q**uery **L**anguage
API	**A**pplication **P**rogram **I**nterface
SDK	**S**oftware **D**evelopment **K**it
H2	**H**ypersonic 2
HTML	**H**yper **T**ext **M**arkup **L**anguage
CSS	**C**ascading **S**tyle **S**heets

LIST OF SYMBOLS

SL.NO	NOTATION NAME	NOTATION	DESCRIPTION
1	Class		Represents a collection of similar entities grouped together.
2	Association		Associations represent static relationship between classes
			Roles represent the way the two classes see each other.
3	Actor		Specifies a role played by a user that interacts with the subject.
4	Communication		Communication between various use case.
5	Initial State		Initial state of the object.
6	Final State		Final state of the object.

x

7	Control flow		Represents various control flow between the states.
8	Decision Box		Represents decision making process from a constraint.
9	Use case		Interaction between the system and external environment.
10	External entity		Represents external entities such as keyboard, sensors, etc.
11	Transition		Represents communication that occurs between Process.
12	Object Lifeline		Represents the vertical dimensions that the object communication.
13	Message		Represents the message exchanged.

CHAPTER 1

INTRODUCTION

CHAPTER 1

INTRODUCTION

1.1. OVERVIEW

1.1.1. ARTIFICIAL INTELLIGENCE

Artificial intelligence is the simulation of human intelligence processes by machines, especially computer systems. Specific applications of AI include expert systems, natural language processing, speech recognition and machine vision.

Image classification: Visual Search

Current visual search technologies use **Artificial Intelligence (AI)** to understand the content and context of these images and return a list of related results. It is applied in more and more industries. One of them is e-commerce.

1.1.2 MACHINE LEARNING

Machine learning is a method of data analysis that automates analytical model building. It is a branch of artificial intelligence based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention.

The use of machine learning algorithms for image classification. The article presents a way of using machine learning algorithms to recognize objects in images. To implement this task, an artificial neural network was used, which has a high adaptability and allows work with a very large set of input data.

1.1.3 DEEP LEARNING

Deep learning is an artificial intelligence (AI) function that imitates the workings of the human brain in processing data and creating patterns for use in decision making. ... Also known as deep neural learning or deep neural network.

The deep learning model has a powerful learning ability, which integrates the feature extraction and classification process into a whole to complete the image classification test, which can effectively improve the image classification accuracy.

1.1.4 GOOGLE COLABS

Colaboratory, or "Colab" for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education.

With Colab you can import an image dataset, train an image classifier on it, and evaluate the model, all in just a few lines of code.

Colab notebooks execute code on Google's cloud servers, meaning you can leverage the power of Google hardware, including GPUs and TPUs, regardless of the power of your machine.

1.2 OBJECTIVE

Our objective is to make e-commerce shopping easier and agile so that customers could search via images to get specific product matches which we will create e commerce database which we will add in future enhancements.

1.3 LITERATURE SURVEY

[1] Jiin-Chiou Cheng, Narn-Yih Lee, Chien Chi, and Yi-Hua Chen 'Image Processing and Smart Contract for Digital Document'; IEEE International Conference on Applied System Invention (ICASI), vol. 4, no. 4, pp. 882 – 885, 2018.

Developed a decentralized application and designed a document system based on Ethereum Image Processing. This technology was selected because it is incorruptible, encrypted, and track able and permits data synchronization. By integrating the features of Image Processing, the system improves the efficiency operations at each stage. The system saves on paper, cuts management costs, prevents document forgery, and provides accurate and reliable information on digital documents.

[2] Murat YasinKubilay, Mehmet Sabır Kiraz and Hacı Ali Mantar 'CertLedger A New PKI Model with Certificate Transparency Based on Image Processing'; International Journal of Computer Science and Mobile Computing IJCSMC, vol. 6, no. 3, pp. 9 – 13, 2019.

The current trust model, CAs has the absolute responsibility to issue correct certificates for the designated subject. However, CAs can still be compromised and fake but valid certificates can be issued due to inadequate security practices or non-compliance with the Certificate Policy (CP) and Certificate Practice Statement (CPS).

[3] Marco Valdi, Franco Chiaraluce, Emanuele Frontoni, Luca Spalazzi 'Document verification through public ledger and Image Processing' International Journal of Advanced Networking and Applications, vol. 6, no. 8, pp. 23 – 29, 2019.

The project is proposed for the solution of addressing the issues of reliability and security of document revocation information. Its main advantages are in removing any single POF and being relatively simple to implement by leveraging existing open source platforms.

[4] Rujia Li, Yifan Wu 'Image Processing based academic document authentication system' Journal of Engineering, Computers & Applied Sciences (JEC&AS), vol. 2, no. 6, pp. 69-75, 2019.

The project consists in designing and implementing by conflating the hash value of local files to the Image Processing but remains numerous issues, did an effective technological approach protecting authentic credential certification and reputation appear.

[5] Ze Wang, Jiwu Jing, Daren Zha, Jingqiang Lin 'Image Processing based document transparency and revocation transparency' ; International Journal of Advanced Networking and Applications, vol. 4, no. 6, pp. 256-260, 2020.

In this they maintain a database to record the documents and revocation status information in the global document Image Processing which is inherently append only, to achieve document transparency and limited grained revocation Transparency.

CHAPTER 2

SYSTEM ANALYSIS

CHAPTER 2

SYSTEM ANALYSIS

The system study is to provide the description about the existing system, its limitation and proposed system, its advantages of the project.

2.1 EXISTING SYSTEMS

Lakhs of people getting certifications/degrees year after year, due to the lack of effective anti-forge mechanism, events that cause the certifications to be forged often get noticed. The certificates are stored in centralized server, so it takes too much time to verify. So it is easily possible to the certificate with fake data.

2.1.1 DISADVANTAGES

- NOT 100% ACCURATE or Precise
- Large time is required in training phase.
- One of the most controversial aspects of computer vision is the potential for invasion of privacy.
- Many people consider having the ability to move freely or search the internet privately to be basic rights that computer vision could infringe upon.
- Another downside of this approach is feature engineering became a real burden, as there were so many inputs to tweak.
- For a cat classifier, which colors were most relevant?
- How flexible should the shape definitions be? Because features needed to be tuned so precisely, building robust models was quite challenging, and accuracy suffered.
- This technology is being used in Google Lens and Google Photos as well.

2.2 PROPOSED SYSTEMS

Due to the high complexity of the existing deep networks with numerous parameters, a large number of training samples are usually required. For HSI classification, the size of labeled samples is relatively small, and deep learning based methods may suffer from over-fitting. In order to overcome the above difficulties, we propose a robust semi supervised deep learning framework for HSI classification by using limited labeled data and exploitation of the abundant unlabeled data.

2.2.1 ADVANTAGES

- Deliver fast results.
- Reduce cost.
- Provides unbiased results.
- Understandable Interface and Environment.

CHAPTER 3

SYSTEM REQUIREMENTS

CHAPTER 3

SYSTEM REQUIREMENTS

Since we are running our Mini Project in an online compiler. We recommend systems that meet or exceed the following specifications:

3.1 HARDWARE

- **Processor (CPU):** Intel Core i5 (sixth generation or newer) (MINIMUM REQUIREMENT)
- **Operating System**

  WINDOWS: ANY OS WILL BE OK BUT **Microsoft Windows 10 Professional x64** (PREFERRED) **.**

  IOS: mac OS Catalina (12.0 OR HIGHER)
- **Memory:** 4 GB RAM (MINIMUM REQUIREMENT)
- **Storage:** 30 GB internal storage drive (MAXIMUM)
- **Monitor/Display:** 14" LCD monitor, resolution of 1600 x 900 or better.
- **Network Adapter:** 802.11ac 2.4/5 GHz wireless adapter

3.2 SOFTWARE REQUIREMENTS

Software Description is a technical specification of requirement of software product. This specifies the environment for development, operation and maintenance of the product**.**

3.2.1 BROWSER

A web browser (commonly referred to as a browser) is application software for accessing the World Wide Web. When a user requests a web page from a particular website, the web browser retrieves the necessary content from a web server and then displays the page on the user's device.

A web browser is not the same thing as a search engine, though the two are often confused. A search engine is a website that provides links to other websites. However, to connect to a website's server and display its web pages, a user must have a web browser installed.

Mozilla firefox : version 90.0.2 (OR LOWER)

Google chrome : version 92.0.425 (OR LOWER)

safari : version 5.1.10 (OR LOWER)

3.2.1.1 GOOGLE COLABS

Colaboratory, or "Colab" for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education. With Colab you can import an image dataset, train an image classifier on it, and evaluate the model, all in just a few lines of code. Colab notebooks execute code on Google's cloud servers, meaning you can leverage the power of Google hardware, including GPUs and TPUs, regardless of the power of your machine.

CHAPTER 4

SYSTEM DESIGN

CHAPTER 4

SYSTEM DESIGN

System design is the process of planning a new system or to replace the existing system. Simply, system design is like the blueprint for building, it specifies all the features that are to be in the finished product.

4.1 SYSTEM ARCHITECTURE

System architecture is the conceptual model that defines the structure, behaviour and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.

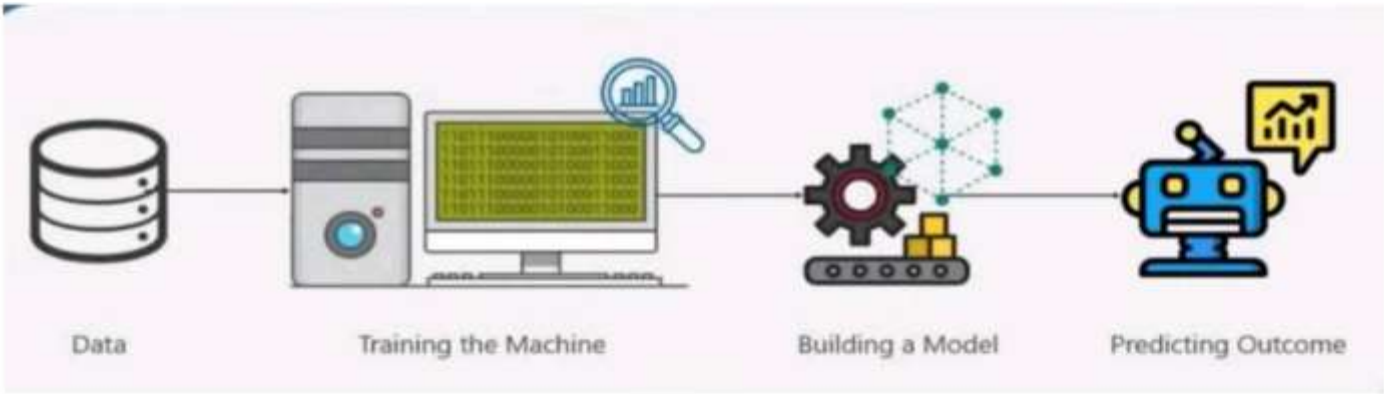


Figure 4.1 Architecture Diagram for Image Classification Using Deep Learning

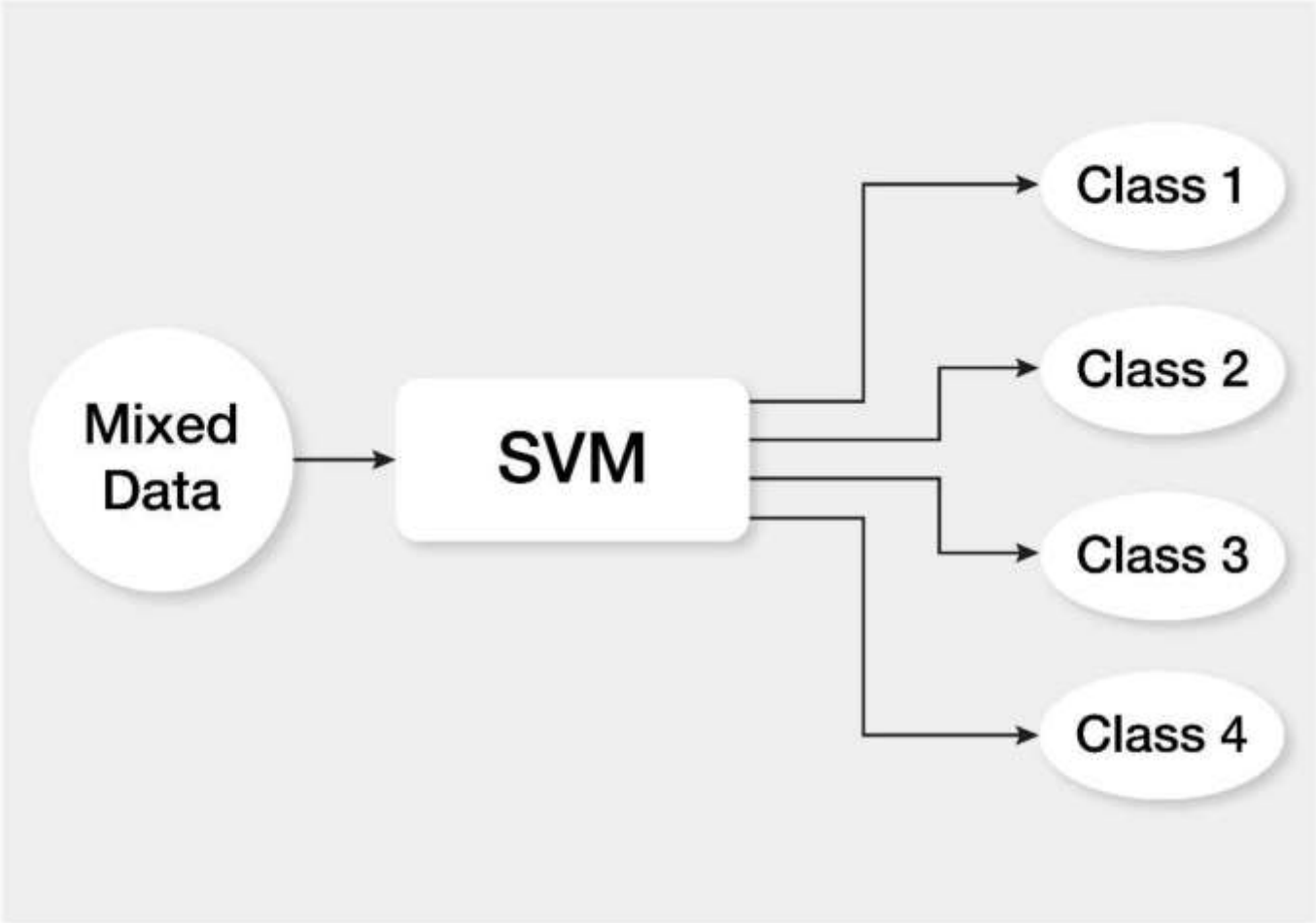


Figure 4.1.1 Architecture Diagram Using SVM Algorithm.

4.2 UML DIAGRAM

4.2.1 USECASE DIAGRAM

Use Cases are typically used to describe the typically visible interactions that the system will have with users and external systems.

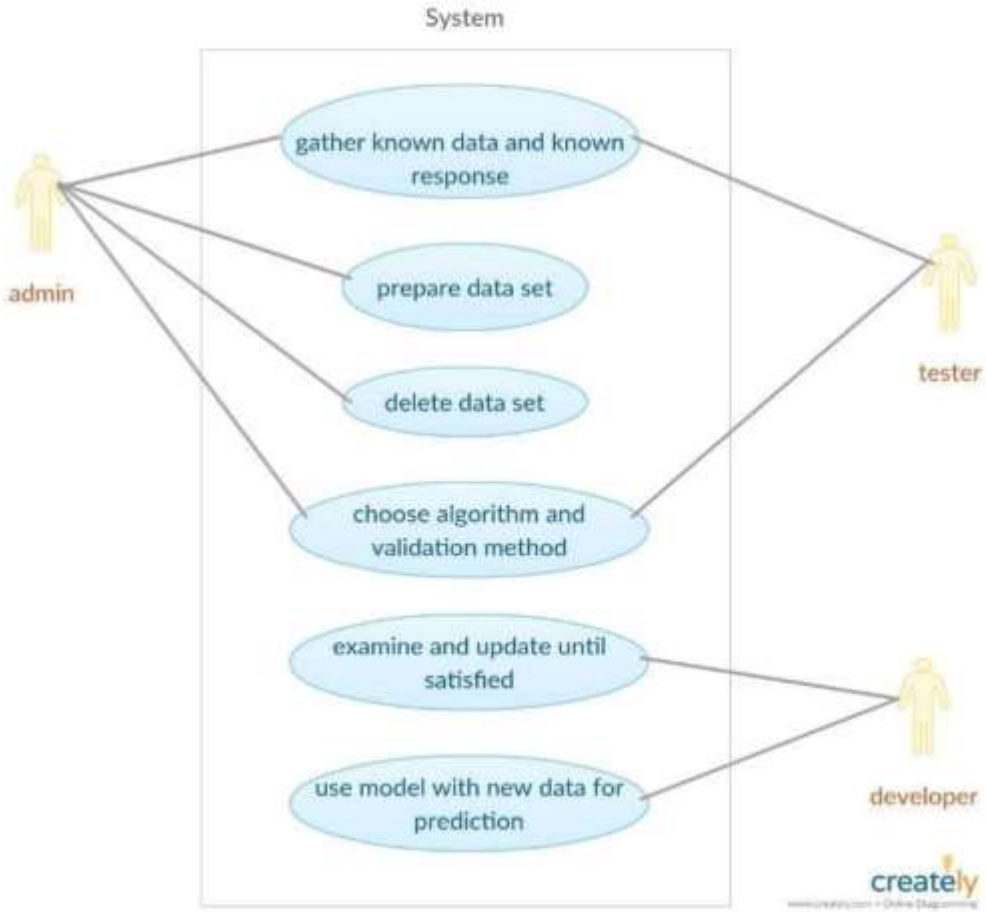


Figure 4.2.1 Use case Diagram for Image Classification Using Deep Learning

4.2.2 CLASS DIAGRAM

A class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among, objects.

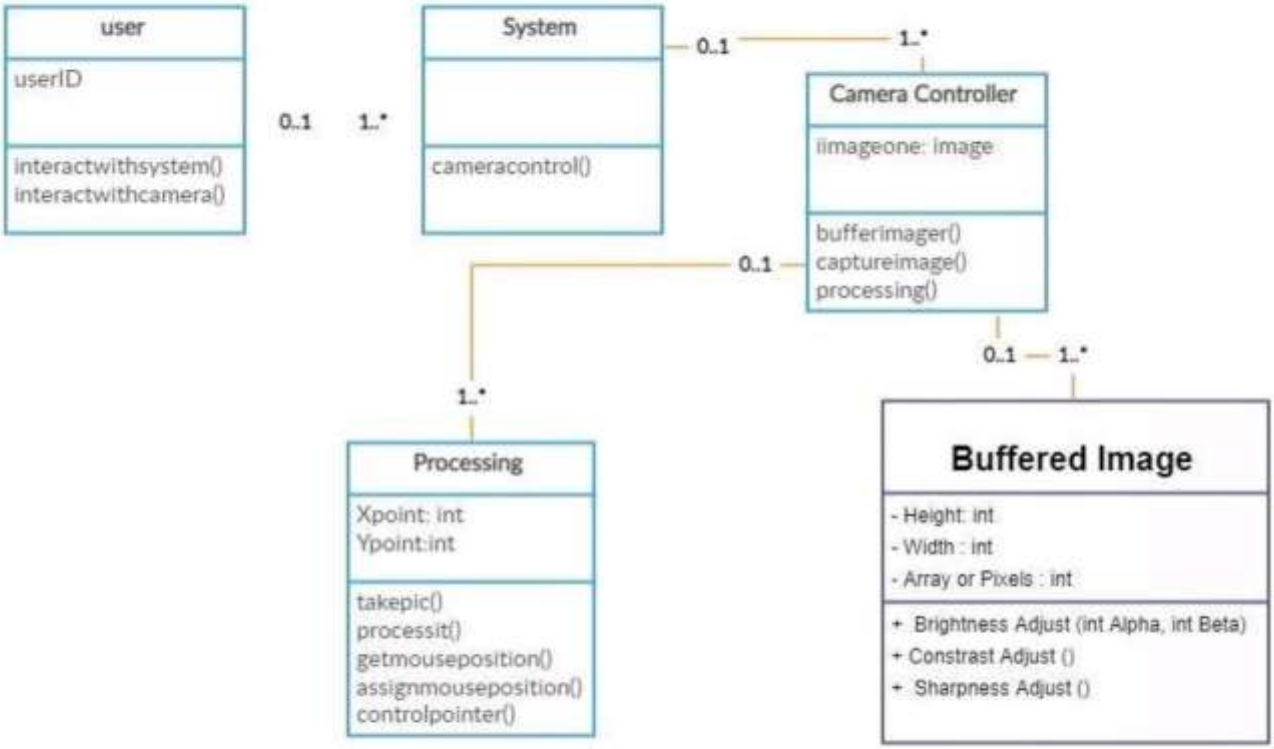


Figure 4.2.2 Class Diagram for Image Classification Using Deep Learning

4.2.3 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency.

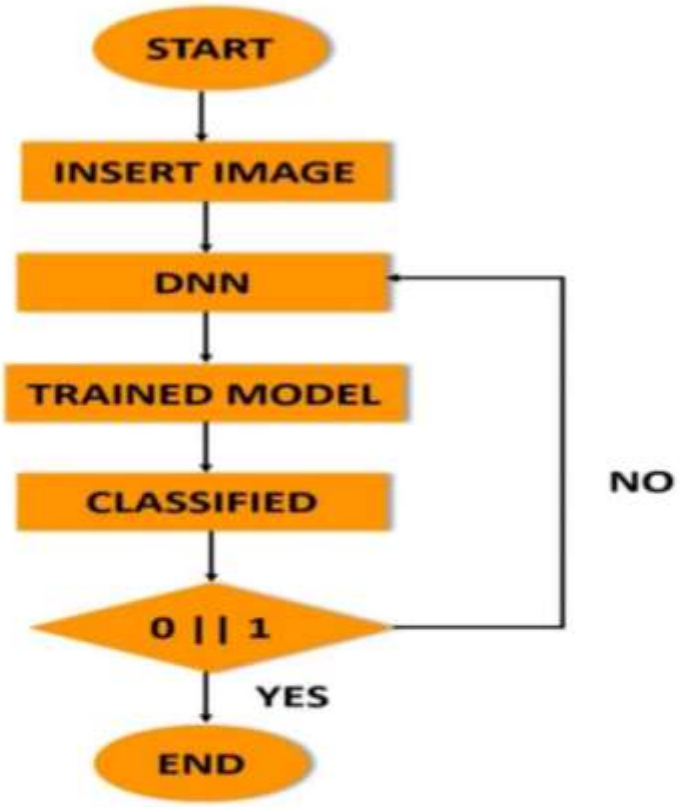


Figure 4.2.3 Activity Diagram for Image Classification Using Deep Learning

4.2.4 SEQUENCE DIAGRAM

A Sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart.

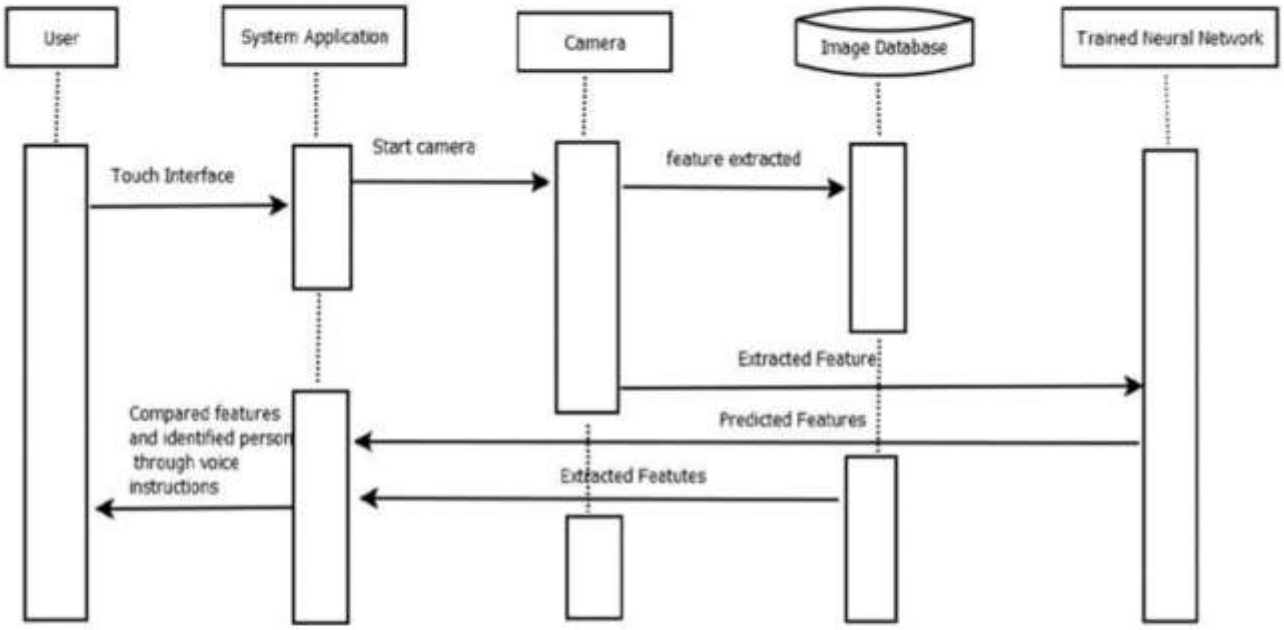


Figure 4.2.4 Sequence Diagram for Image Classification Using Deep Learning

4.2.5 COLLABORATION DIAGRAM

Another type of interaction diagram is the collaboration diagram. A collaboration diagram represents a collaboration, which is a set of objects related in a particular context, and interaction, which is a set of messages exchange among the objects within the collaboration to achieve a desired outcome.

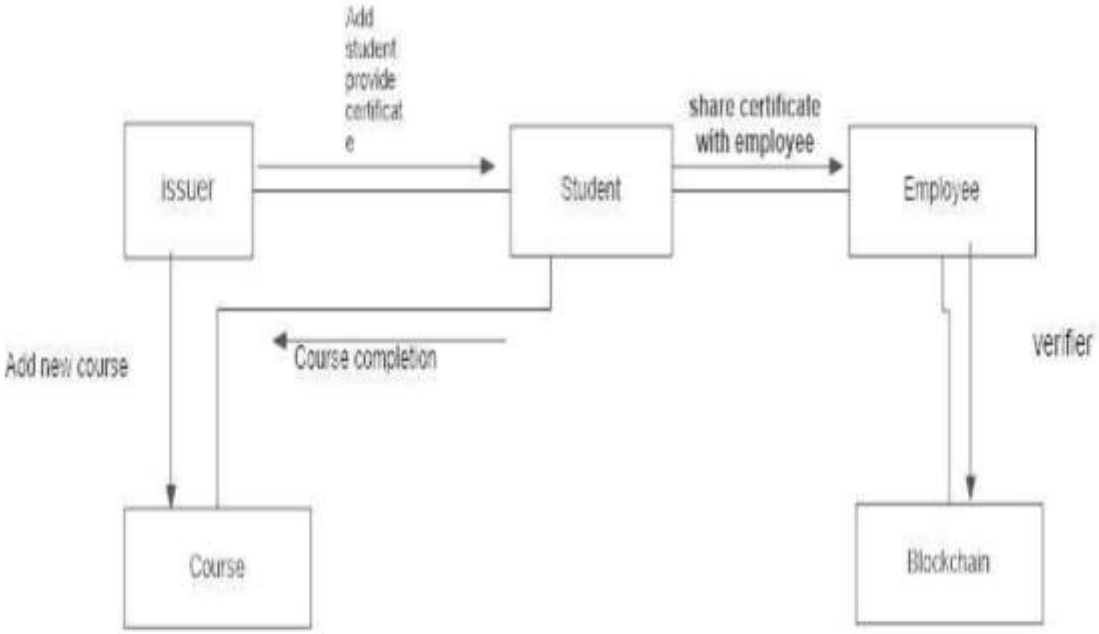


Figure 4.2.5 Collaboration Diagram for Image Classification Using Deep Learning

CHAPTER 5

SYSTEM IMPLEMENTATION

CHAPTER 5

SYSTEM IMPLEMENTATION

5.1 LIST OF PACKAGES

5.1.1 NUMPY

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.The ancestor of NumPy, Numeric, was originally created by Jim Hugunin with contributions from several other developers. In 2005, Travis Oliphant created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. NumPy is open source software and has many contributors.

5.1.2 MATPLOTLIB

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK. There is also a procedural "pylab" interface based on a state machine (like OpenGL), designed to closely resemble that of MATLAB, though its use is discouraged.SciPy makes use of Matplotlib.

Matplotlib was originally written by John D. Hunter.Matplotlib 2.0.x supports Python versions 2.7 through 3.10. Python 3 support started with Matplotlib 1.2. Matplotlib 1.4 is the last version to support Python 2.6.Matplotlib has pledged not to support Python 2 past 2020 by signing the Python 3 Statement.

5.1.3 PYNGROK

It is a python wrapper for ngork.Ngork is a cross-platform application that enables developers to expose a local development server to the Internet with minimal effort. The software makes your locally-hosted web server appear to be hosted on a subdomain of ngrok.com, meaning that no public IP or domain name on the local machine is needed.

5.1.4 BING IMAGE DOWNLOADER

Python library to download bulk of images form Bing.com. This package uses async url, which makes it very fast while downloading..Get the free access key for Bing Search APIs from here and enter it into the "access key" field of the copied code.This program lets you download tons of images from Bing. Please do not download or use any image that violates its copyright terms.

Any image downloader will do actually.

5.1.5 SCIKIT

Scikit-learn (formerly scikits.learn and also known as sklearn) is a free software machine learning library for the Python programming language.It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

5.1.5.1 SCIKIT IMAGE

scikit-image (formerly scikits.image) is an open-source image processing library for the Python programming language.It includes algorithms for segmentation, geometric transformations, color space manipulation, analysis, filtering, morphology, feature detection, and more.It is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

5.1.5.2 SCIKIT LEARN

The scikit-learn project started as scikits.learn, a Google Summer of Code project by David Cournapeau. Its name stems from the notion that it is a "SciKit" (SciPy Toolkit), a separately-developed and distributed third-party extension to SciPy.The original codebase was later rewritten by other developers. In 2010 Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort and Vincent Michel, all from the French Institute for Research in Computer Science and Automation in Rocquencourt, France, took leadership of the project and made the first public release on February the 1st 2010. Of the various scikits, scikit-learn as well as scikit-image were described as "well-maintained and popular" in November 2012.Scikit-learn is one of the most popular machine learning libraries on GitHub.

5.1.6 PILLOW

Pillow is built on top of PIL (Python Image Library). PIL is one of the important modules for image processing in Python. However, the PIL module is not supported since 2011 and doesn't support python 3.

Pillow module gives more functionalities, runs on all major operating system and support for python 3. **It supports wide variety of images such as "jpeg", "png", "bmp", "gif", "ppm", "tiff". You can do almost anything on digital images using pillow module.** Apart from basic image processing functionality, including point operations, filtering images using built-in convolution kernels, and color space conversions.

5.1.7 STREAMLIT

Stream lit is an open-source Python library that makes it easy to create and share beautiful, custom web apps for machine learning and data science. In just a few minutes you can build and deploy powerful data apps.

5.2 LIST OF MODULES

- Data Input
- Data Preprocessing
- Building Models
- Predicting Outcomes

5.3 MODULE DESCRIPTION

5.3.1. DATA INPUT

5.3.1.1 Bing Image Downloader

Bing image downloader is our input dataset that is fed to our model. It is not necessary that we need to use only bing image downloader any downloader will be able to do the job.

5.3.2. DATA PREPROCESSING

Data preprocessing does the job of adjusting the input image into different matrices in order for the algorithm to run and process.

5.3.2.1 Image Resize

We have used Skimage.transform library to resize our input image to constant which is essential , if not resized to constant pixels the algorithm would not be able to process them in various different pixels.

5.3.2.2 Flatten

Flatten is our second preprocessing **we need to do which would be** Converting 3d images in one dimension that is converting matrix to vector format. This is done using Numpy library. Flattening is essential so that the image converted to matrix would be harder than the vector format.

5.3.3. BUILDING MODELS

Algorithm:

- **Hyperparameter Tuning using GridSearchCV**

While building a Machine learning model we always define two things that are model parameters and model hyperparameters of a predictive algorithm. Model parameters are the ones that are an internal part of the model and their value is computed automatically by the model referring to the data like support vectors in a support vector machine. But hyperparameters are the ones that can be manipulated by the programmer to improve the performance of the model like the learning rate of a deep learning model. They are the one that commands over the algorithm and are initialized in the form of a tuple.

Hyperparameter tuning is the process of tuning the parameters present as the tuples while we build machine learning models. These parameters are defined by us which can be manipulated according to programmer wish. Machine learning algorithms never learn these parameters. These are tuned so that we could get good performance by the model. Hyperparameter tuning aims to find such parameters where the performance of the model is highest or where the model performance is best and the error rate is least. We define the hyperparameter as shown below for the random forest classifier model. These parameters are tuned randomly and results are checked.

- **Support Vector Machine**

1. Import Python libraries

2. Display image of each bee type, The file is loaded labels.csv into a data frame called labels, where the index is the image name and the genus column tells us the bee type. genus takes the value of either 0.0 (Apis or honey bee) or 1.0 (Bombus or bumble bee).The function get_image converts an index value from the data frame into a file path where the image is located, opens the image using the object in Pillow, and then returns the image as a numpy array.

3. Image manipulation with rgb2grey,Image data is represented as a matrix, where the depth is the number of channels. An RGB image has three channels (red, green, and blue) whereas the returned greyscale image has only one channel.

4. Histogram of oriented gradients,The images are converted into a format that a machine learning algorithm can understand. An image is divided in a grid fashion into cells, and for the pixels within each cell, a histogram of gradient directions is compiled. To improve invariance to highlights and shadows in an image, cells are block normalized, meaning an intensity value is calculated for a larger region of an image called a block and used to contrast normalize all cell-level histograms within each block.

5. Create image features and flatten into a single row,A function called create features that combines these two sets of features by flattening the three-dimensional array into a one-dimensional (flat) array.

6. Loop over images to preprocess, Above we generated a flattened features array for the bombus image. Now it's time to loop over all of our images. We will create features for each image and then stack the flattened features arrays into a big matrix we can pass into our model. In the create_feature_matrix function, we'll do the following:

- Load an image
- Generate a row of features using the create_features function above
- Stack the rows into a features matrix
- In the resulting features matrix, rows correspond to images and columns to features.

7. Scale feature matrix + PCA

The dataset have over 31,000 features for each image and only 500 images total. To use an SVM, our model of choice, the number of features needs to be reduced.PCA is a way of linearly transforming the data such that most of the information in the data is contained within a smaller number of features called components. from an image dataset containing handwritten numbers. The image on the left is the original image with 784 components.

8. Split into train and test sets,The data are converted into train and test sets. The training data will inculd 70% of images and the remaining 30% was used to test the model.

9. Train model,To classify as honey or bumble bee – Support Vector Classifier (SVC), a type of SVM is used.

10. Score model,Predictions are calculated on the test data, followed by the accuracy.

11. ROC curve + AUC,The svm.predict_proba was used to get the probability that each class is the true label. The model predicts there is a 64% chance the bee in the image is an Apis.

5.4 STEPS INVOLVED:

5.4.1. Google Colab:

Colaboratory, or "Colab" for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education. With Colab you can import an image dataset, train an image classifier on it, and evaluate the model, all in just a few lines of code. Colab notebooks execute code on Google's cloud servers, meaning you can leverage the power of Google hardware, including GPUs and TPUs, regardless of the power of your machine.

5.4.2. Demonstration of Teachable Machine

5.4.3. Bing Image Downloader

Python library to download bulk of images form Bing.com. This package uses async url, which makes it very fast while downloading..Get the free access key for Bing Search APIs from here and enter it into the "access key" field of the copied code. This program lets you download tons of images from Bing. Please do not download or use any image that violates its copyright terms. Any image downloader will do actually.

5.4.4. Data Pre-processing

5.4.4.1 RESIZE

5.4.4.2 FLATTEN

5.4.5. Hyper parameter Tuning using Grid Search

5.4.6. Support Vector Machine

5.4.7. Evaluation of model

5.4.8 Saving the model in a Pickle file

5.4.9 Checking for a new image from Google

5.4.10. Deployment using Stream lit

Stream lit is an open-source Python library that makes it easy to create and share beautiful, custom web apps for machine learning and data science. In just a few minutes you can build and deploy powerful data apps.

CHAPTER 6
TRAINING AND TESTING

CHAPTER 6

TRAINING AND TESTING

Testing is the process of executing a program or application with the intent of finding software bugs, and to verify that the software product is fit for use.

6.1 TESTING MODELS

```
[ ] from sklearn.metrics import accuracy_score,confusion_matrix

[ ] accuracy_score(y_pred,y_test) # Checking matching predicted vs orginal accuracy rate

    0.7037037037037037
```

Figure 6.1 TESTING Model's Accuracy

The evaluation model shows the accuracy rate of our trained models which is 70.37% which again proves the disadvantage that it is not 100% accurate.

6.2 TEST OBJECTIVES

- Train Datasets
- Image Resizing
- Image Flattening
- Evaluating Model Accuracy
- Testing a Random Image

6.3 TEST CASES OF IMAGE CLASSIFICATION USING DEEP LEARNING

ID	TEST CASES	PRE-CONDITONS	EXPECTED RESULTS	ACTUAL RESULTS	PASS/FAIL
TC001	Train Datasets	Input Details.	Successful Train Data set	Successful Train Data set	PASS
TC002	Image Resizing	Image /Pixel Details	Successful Resizing Of Image	Successful Resizing Of Image	PASS
TC003	Image Flattening	Image Dimension Details	Successfully Converted 3D to 1D	Successfully Converted 3D to 1D	PASS
TC004	Evaluating Model Accuracy	Accuracy of Model	70.3%.	70.3%	PASS
TC005	Testing a Random Image	Random URL of an image	Return Image Details	Return Image Details	PASS
TC006	Deployment	Deployment Details	Deployment Details	Deployment Details	FAIL

Table 6.1 TEST CASES OF IMAGE CLASSIFICATION USING DEEP LEARNING

CHAPTER 7
RESULTS AND DISCUSSION

CHAPTER 7
RESULTS AND DISCUSSION

7.1 RESULTS

Thus our model was executed successfully but we will develop our model I future by integrating with an e-commerce platform wherein we will be able to search for products via image search using this model.

7.2 OUTPUT SCREENSHOT

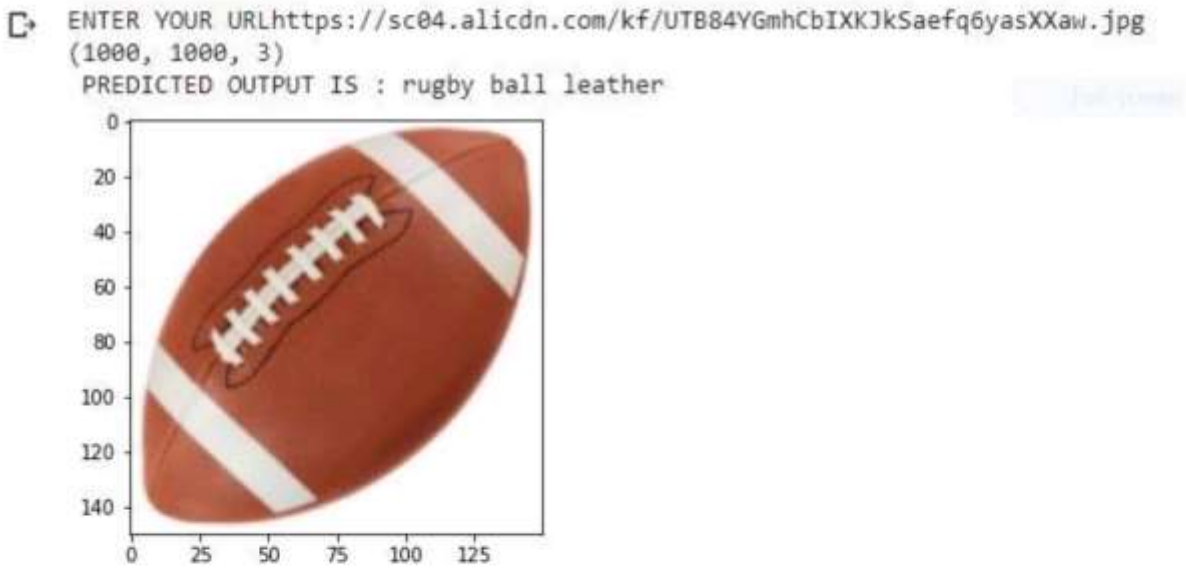


Figure 7.2 1 Output Screenshot of Demo Module

CHAPTER 8

CONCLUSION AND FUTURE ENHANCEMENT

CHAPTER 8

CONCLUSION AND FUTURE ENHANCEMENT

8.1 CONCLUSION

There are many ways in which these techniques are classified and categorizing these techniques into supervised and unsupervised is the most common way. The comparison of these techniques on the basis of efficiency largely depends on the type of data they are being used for. This paper summarizes the information about commonly used image classification techniques. This will help the researchers to select the most appropriate classification technique according to their requirements.

8.2 FUTURE ENHANCEMENT

- The future of image processing will involve improved accuracy and precision.
- We would integrate our services with E-commerce website in order to buy specific products via images.
- Irrigation monitoring and providing information can be made possible by tracking satellite imaging of the fields. Processing of infrared images can act as an additional means to monitor and analyze irrigation. This analysis can then be utilized in pre-harvesting operations for deciding whether to harvest or not. Growth of weeds can also be detected by using a combination of machine learning and image processing algorithms and techniques.
- Drone aircrafts monitoring environmental and traffic conditions can use image processing to capture high resolution real-time videos and photographs. In case of natural or other disasters like flood, earthquake, fire etc.
- 3D imaging is a process where a 2D image is converted into a 3D image by creating the optical illusion of depth. The next step is rendering where colors and textures are included in the 3D model to make it look realistic. With such 3D imaging and rendering, doctors can see extremely high quality 3D images of organs that they couldn't have seen otherwise. This, in turn, can help them carry out delicate surgeries and make accurate diagnoses.

CHAPTER 9

ANNEXURE

9.1 APPENDIX

SOURCE CODE

```
!pip install bing-image-downloader
!mkdir images
from bing_image_downloader import downloader
downloader.download("pretty
sunflower",limit=30,output_dir='images',adult_filter_off=True)
from bing_image_downloader import downloader
downloader.download("rugbyball
leather",limit=30,output_dir='images',adult_filter_off=True)
from bing_image_downloader import downloader
downloader.download("ice cream cone",limit=30,output_dir='images',adult_filter_off=True)
import os
import matplotlib.pyplot as plt
import numpy as np
from skimage.io import imread
from skimage.transform import resize
target=[]
images=[]
flat_data=[]
DATADIR = '/content/images'
CATEGORIES = ['pretty sunflower','rugby ball leather','ice cream cone']
for category in CATEGORIES:
  class_num = CATEGORIES.index(category)
  path = os.path.join(DATADIR,category)
  for img in os.listdir(path):
    img_array = imread(os.path.join(path,img))
```



```
    img_resized = resize(img_array,(150,150,3))
    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)
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)
from sklearn.model_selection import GridSearchCV #GridSearchCV is a model
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)
y_pred=clf.predict(x_test)
y_pred
y_test
from sklearn.metrics import accuracy_score,confusion_matrix
accuracy_score(y_pred,y_test)
import pickle
pickle.dump(clf,open('img_model.p','wb'))
model = pickle.load(open('img_model.p','rb'))
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)
y_out = model.predict(flat_data)
y_out = CATEGORIES[y_out[0]]
print(f' PREDICTED OUTPUT IS : {y_out}')
```

9.2 REFERENCE

[1] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An Algorithm for Designing Over complete Dictionaries for Sparse Representation. IEEE Transactions on Signal Processing, 54(11):4311-4322, 2006.

[2] L. Bo, X. Ren, and D. Fox. Kernel Descriptors for Visual Recognition. In NIPS, 2010.

[3] L. Bo and C. Sminchisescu. Efficient Match Kernel between Sets of Features for Visual Recog nition. In NIPS, 2009,

[4] O. Boiman, E. Shechtman, and M. Irani. In Defense of Nearest-Neighbor based Image Classi fication, In CVPR, 2008.

[5]Y. Boureau, F. Bach, Y. LeCun, and J. Ponce. Learning Mid-level Features for Recognition. In CVPR, 2010.

[6] A. Coates and A. Ng. The Importance of Encoding versus Training with Sparse Coding and Vector Quantization. In ICML, 2011.

[7] G. Davis, S. Mallat, and M. Avellaneda. Adaptive Greedy Approximations. Constructive Approximation, 13(1):57-98, 2012.

[8] P. Gehler and S. Nowozin. On Feature Combination for Multiclass Object Classification. InICCV, 2009.1

[9] G. Hinton, S. Osindero, and Y. Teh. A Fast Learning Algorithm for Deep Belief Nets. Neural Computation, 18(7):1527-1554, 2006.

[10] P. Jain, B. Kulis, and K. Grauman. Fast Image Search for Learned Metrics. In CVPR, 2008.

