MANIPAL INSTITUTE OF TECHNOLOGY

MANIPAL

(A constituent institution of MAHE, Manipal)

Project Report

On

Panda or Bear Image Classification

Fundamentals of Machine Learning Lab

Subject Code: DSE 2242

Names	Registration No
Martha Sri Manikanth	
Pratham Chirag Ghosh	
Mitwa Saraf	

Department of Data Science & Computer Applications,

Manipal Institute of Technology,

Manipal

JAN -MAY 2024

Table of Contents

CHAPTER 1: INTRODUCTION

This project aims to contribute to the efforts of wildlife monitoring by developing a robust image classification system for pandas and bears, utilizing a diverse range of classic machine learning algorithms.

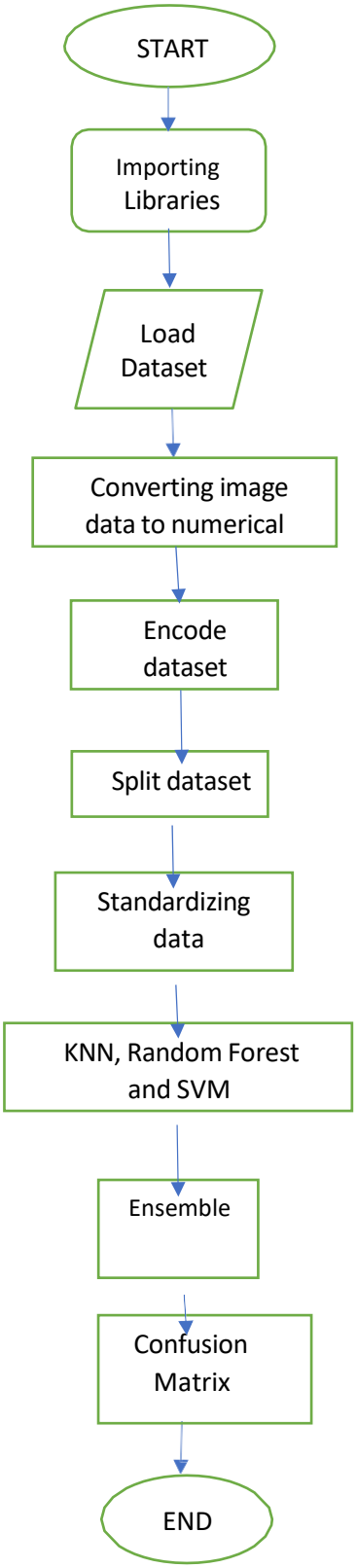
The classification of images of pandas and bears presents unique challenges due to factors like varying fur patterns, fur color, and poses. Therefore, employing multiple machine learning algorithms becomes imperative to ensure accurate classification across diverse scenarios.

This project will implement a suite of machine learning algorithms including linear Support Vector Machines (SVMs), Random Forests, and K-Nearest Neighbors (KNN). Each algorithm will be meticulously trained and fine-tuned using a comprehensive dataset of panda and bear images collected from various sources.

Furthermore, the performance of these algorithms will be compared based on key parameters such as accuracy, precision, recall, F1-score, and computational efficiency. Through this comparative analysis, we aim to identify the most suitable algorithm(s) for accurate and efficient classification of panda and bear images, ultimately aiding conservation efforts and wildlife monitoring initiatives.

CHAPTER 2: METHODOLGY

2.1 Flowchart/block diagram of the proposed method



Explain each phase of block diagram

1. Start/End
    - This step marks the beginning or end of the process.
    - It indicates where the execution of model begins and where it ends.
2. Import libraries
    - This step involves importing the necessary python libraries such as OpenCV, Pandas, NumPy, matplotlib, seaborn, scikit-learn etc.
3. Load dataset
    - This step involves importing dataset using pandas and os library
    - File paths are initialized.
    - Two Dataframes are created and concatenated for use.
4. Converting image data to numerical
    - This step uses NumpPy and PIL library.
    - Img_to_npy function is initialized which returns two NumPy arrays: images (a list containing NumPy arrays representing resized images) and labels(a list containing corresponding labels for each image).
    - os.listdir(path) constructs the full image path by joining the directory path (path) with the file name (i).
    - PIL's Image.open() function opens the image in the directory.
    - Resize() resizes the image to a fixed size of 200x200 pixels.
    - Np.array(img) converts the image to a NumPy array.
5. Encode dataset
    - For each image, the code flattens the image data by reshaping it into a 1D array of length 120000 (assuming the original image size is 200x200 pixels with 3 color channels).
    - The flattened image data is appended to the (imgdata) list, and the corresponding label is appended to the (img_labels) list.
6. Split dataset
    - train_test_split function from scikit-learn to split the dataset into training and testing set.
    - X represents the feature matrix containing the flattened image data).
    - Y represents the target variable containing the labels corresponding to each image in the feature matrix.
    - 30% of the data is used for testing, while the remaining 70% will be used for training.
7. Standardizing data
    - Feature scaling using the StandardScaler from scikit-learn.
8. KNN, Random Forest, Linear SVM
    - For each algorithm, the classifier is initialized, trained and evaluated.
    - The scaled training data (x_train_scaled) is used for training the classifier.
    - Evaluation is done based on accuracy of each classifier with *x_test_scaled*.

9. Ensemble
    - A voting classifier is used.
    - The Voting Classifier combines the predictions of three different classifiers (Linear SVM, Random Forest, and KNN) to make a final prediction.
10. Confusion Matrix
    - This step involves visualizing the performance of each classifier using Confusion matrices.
    - Confusion matrix for predictions made by the Linear SVM, Random Forest, KNN and Voting Classifier is computed and plotted as heatmaps (seaborn library).

CHAPTER 3: EXPERIMENTAL SETUP

1. Hardware Configuration:

- Employ a computer system equipped with adequate computational resources, comprising CPU, RAM, and optionally, GPU capabilities, to support the training and assessment of machine learning models.

2. Software Requirements:

- Install necessary software packages and libraries for data preprocessing, model implementation, and evaluation. This includes:
  - Python programming language
  - Miscellaneous operating system interfaces (os library).
  - Data manipulation, computation and analysis libraries such as Pandas, Numpy, Seaborn and Matplotlib.
  - Machine learning libraries such as scikit-learn.
  - Image processing libraries such as OpenCV, and PIL (Python Imaging Library).
  - Jupyter Notebook or similar environments like VS Code for code development and experimentation

CHAPTER 4: DATASET

- The link to the image dataset used: https://www.kaggle.com/datasets/mattop/panda-or-bear-image-classification
- The dataset contains total of 600 images, 300 of each class: Bear and Panda.
- All images are reshaped to 200 X 200 X 3 in jpeg format.
- There are 420 training images, 180 test images.

Fig: Panda or Bear Image classification: Pandas Testing image

CHAPTER 5: RESULT AND DISCUSSION

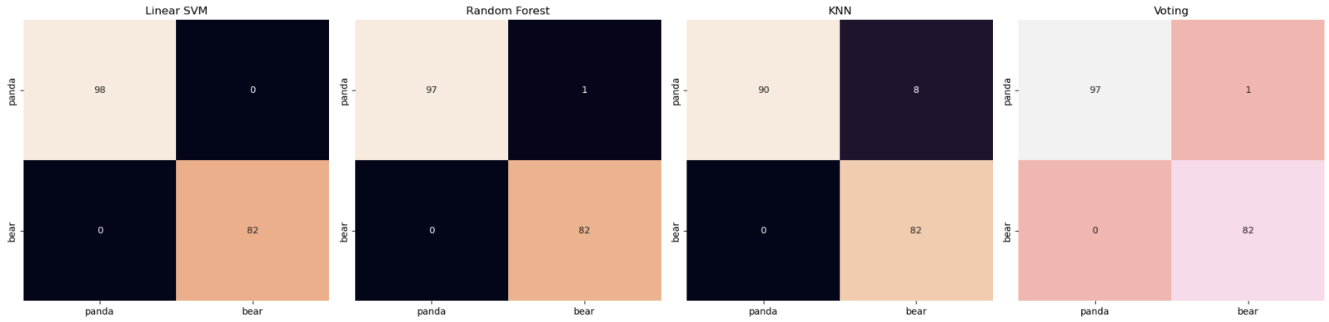


Fig: Confusion matrices of Linear SVM, Random Forest, KNN and Ensemble algorithms

Models	Linear SVM	Random Forest	KNN	Ensemble
Accuracy (%)	100.00	99.444	95.5556	99.444

Table: Accuracy (%) of the models implemented

```
print(classification_report(y_test,knn_y_pred))
              precision    recall  f1-score   support

           0       1.00      0.92      0.96        98
           1       0.91      1.00      0.95        82

    accuracy                           0.96       180
   macro avg       0.96      0.96      0.96       180
weighted avg       0.96      0.96      0.96       180

print(classification_report(y_test,rf_y_pred))
              precision    recall  f1-score   support

           0       1.00      0.99      0.99        98
           1       0.99      1.00      0.99        82

    accuracy                           0.99       180
   macro avg       0.99      0.99      0.99       180
weighted avg       0.99      0.99      0.99       180

print(classification_report(y_test,linear_svm_pred))
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        98
           1       1.00      1.00      1.00        82

    accuracy                           1.00       180
   macro avg       1.00      1.00      1.00       180
weighted avg       1.00      1.00      1.00       180
```

Fig: Classification report for KNN, Random Forest And Linear SVM:

Discussion:

- Linear SVM outperformed the KNN algorithm and Random Forest and demonstrated the accuracy of 100.00%, indicating the effectiveness in classifying images of pandas and bears.
- Model Complexity: KNN, being a simple algorithm, showed competitive performance, while SVM, known for handling high-dimensional data, also performed better than Random Forest.
- Scalability: Random Forest's ability to handle complex datasets and capture intricate patterns made it a robust choice for this image recognition task.
- Using a Voting Classifier aims to leverage the strengths of each individual classifier and potentially improve overall prediction accuracy.
- By evaluating the results and discussing the performance of each model, we can conclude that Linear SVM is the most effective algorithm for this specific image recognition task.

CHAPTER 6: CONCLUSION

In summary, the project successfully classified images of panda and bears using classic machine learning algorithms like KNN, SVM, and Random Forest.

Linear SVM proved to be the most effective algorithm showcasing superior accuracy. This highlights the value of traditional ML methods in image recognition tasks, pointing out the importance of algorithm selection tailored to the task at hand. Further optimization and exploration could enhance performance.