MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
(A constituent institution of MAHE, Manipal)

Project Report
On
WILDLIFE CLASSIFICATION

Fundamentals of Machine Learning Lab
Subject Code: DSE 2242

Names	Registration No
MANDAR CHAUDHARI	220968222
NAMAN MAHESHWARI	220968252
SHIVLI MATHUR	220968298

Department of Data Science & Computer Applications,

Manipal Institute of Technology,

Manipal

JAN -MAY 2024

Table of Contents

Abstract

1. Introduction
2. Methodology
    2.1. Flowchart/ block diagram of the proposed method
    2.2. Explain each phase of the block diagram.
3. Experimental Setup
4. Dataset
5. Results, and Discussion
6. Conclusion

ABSTRACT

Wildlife conservation is a critical area of research and application, often requiring the use of advanced technologies to monitor and protect endangered species. In this project, we explore the application of Machine Learning (ML) algorithms for the classification of two closely related species, **The Cheetah and The Tiger**, based on their images. The primary objective is to develop and compare the performance of three different ML models: **k-Nearest Neighbors (KNN), Random Forest, and Decision Tree**, for accurate and efficient classification of these species. By comparing the performance of KNN, Random Forest, and Decision Tree models in this context, we provide valuable insights into the strengths and limitations of each approach when applied to wildlife classification tasks.

We begin by collecting a dataset consisting of high-resolution images of cheetahs and tigers, ensuring a balanced representation of both species. Next, we preprocess the images to extract relevant features and prepare them for input into the ML models. We then train each model using a portion of the dataset and evaluate their performance using metrics such as accuracy, precision, recall, and F1-score. The results indicate that all three models can achieve high accuracy in distinguishing between cheetahs and tigers, with **Random Forest demonstrating the highest performance among the three algorithms**.

CHAPTER 1
INTRODUCTION

In this project, we explore the application of Machine Learning (ML) algorithms for the classification of two closely related species, **The Cheetah and The Tiger**, based on their images. The primary objective is to develop and compare the performance of three different ML models: **k-Nearest Neighbors (KNN), Random Forest, and Decision Tree.**

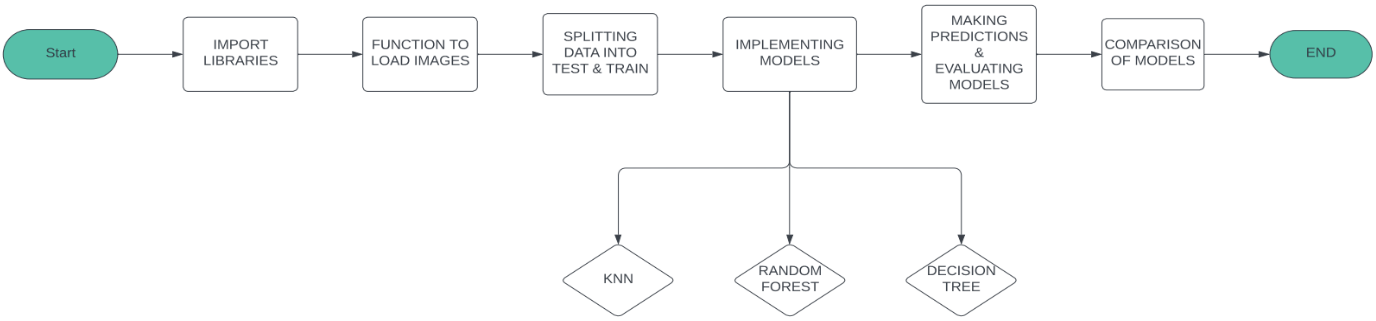
In the context of image classification of cheetah and tiger, the **K-Nearest Neighbors (KNN)** algorithm is utilized as a machine learning technique. KNN is a type of supervised learning algorithm that can be used for classification tasks. In the case of identifying big cats like cheetahs and tigers from images, KNN works by comparing the input image with known images in the dataset. The algorithm then classifies the input image based on the majority class of its k-nearest neighbors in the feature space.

Random Forest is a machine learning algorithm commonly used in image classification tasks, including distinguishing between distinct species like cheetahs and tigers. This algorithm works by creating multiple decision trees during the training phase and then combining their outputs to improve accuracy and reduce overfitting. In the context of classifying cheetahs and tigers, random forest can analyze various features extracted from images, such as body covering patterns, to differentiate between the two species effectively.

In the context of image classification of cheetah and tiger, a **Decision Tree** is a machine learning model that uses a tree-like structure to make decisions based on the features extracted from the images. The decision tree algorithm works by splitting the data based on unique features at each node, leading to a classification decision at the leaf nodes. In the case of classifying images of cheetahs and tigers, the decision tree would analyze various visual features extracted from the images, such as patterns, colors, shapes, and textures, to make a decision on whether the image represents a cheetah or a tiger.

CHAPTER 2
METHODOLOGY

2.1 Flowchart of the proposed method



2.2 Explanation of each phase of the block diagram

- os: Provides a way to interact with the operating system, useful for file operations.
- cv2: OpenCV library for computer vision tasks like image processing.
- numpy as np: Imports the NumPy library for numerical computing, commonly aliased as `np` for easier access.
- matplotlib.pyplot as plt: Matplotlib library for creating visualizations like plots and charts, with `plt` as a common alias.
- seaborn as sns Seaborn library for statistical data visualization, often used to enhance the aesthetics of plots.
- from sklearn.metrics import ... : Imports specific metrics functions from scikit-learn for evaluating machine learning models.

- accuracy_score: Computes the accuracy of a classification model.

-confusion_matrix: Generates a confusion matrix to evaluate classification performance.

- precision_score: Calculates the precision of a classification model.

- f1_score: Computes the F1 score, a harmonic mean of precision and recall.

- recall_score: Calculates the recall of a classification model.

- from sklearn.model_selection import train_test_split* Imports the `train_test_split` function from scikit-learn to split data into training and testing sets.

FUNCTION TO
LOAD IMAGES

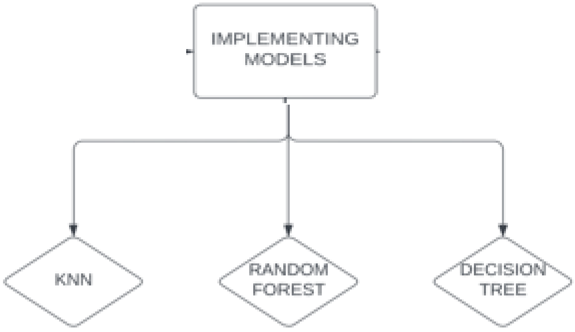
This Python function, `load_images`, takes a `folder_path` as input, which is expected to contain subfolders, each representing a category of images (e.g., different animal species). The function reads images from these subfolders, resizes them to a fixed size (100x100 pixels), flattens them into a 1D array, and stores them along with their corresponding labels. Here is a breakdown of each step:

1. Input: `folder_path` is the path to the main folder containing subfolders with images. For each subfolder, the name of the subfolder is used as a label for the images it contains.
2. Initialization: Two empty lists, `images` and `labels`, are created to store the flattened images and their corresponding labels, respectively.
3. Enumerating through subfolders: The `enumerate` function is used to iterate over each subfolder in the `folder_path`. The `label` variable is assigned the index of the subfolder (starting from 0).
4. Iterating through images in each subfolder: For each subfolder (animal category), the function iterates through all the images in that subfolder.
5. Reading and processing images: For each image, it constructs the full path (`image_path`) and reads the image using `cv2.imread`. It then resizes the image to a fixed size of 100x100 pixels using `cv2.resize`.
6. Flattening the image: The image is flattened into a 1D array using the `flatten` method, and the flattened image is appended to the `images` list.
7. Adding labels* The label (index of the current subfolder) is appended to the `labels` list.
8. Returning the data: Finally, the function returns two NumPy arrays: `images`, which contains the flattened image arrays, and `labels`, which contains the corresponding labels for each image.

This function uses the OpenCV library (`cv2`) for reading and resizing images, and it assumes that the images are stored in subfolders within the specified `folder_path`.

Here we use the `train_test_split` function from the `sklearn.model_selection` module in Python. Here is a breakdown of what each part of the code does:
- `X_train`: This variable will store the training data for the features after splitting.
- `X_test`: This variable will store the testing data for the features after splitting.
- `y_train`: This variable will store the training data for the target variable after splitting.
- `y_test`: This variable will store the testing data for the target variable after splitting.
- `train_test_split`: This function is used to split the dataset into training and testing sets.
- `X`: This variable contains the features of the dataset.
- `y`: This variable contains the target variable of the dataset.
- `test_size: This parameter specifies the proportion of the dataset that should be included in the testing set.
- `random_state: This parameter sets the random seed for reproducibility.

Overall, this line of code is splitting the dataset into training and testing sets for both the features and the target variable.



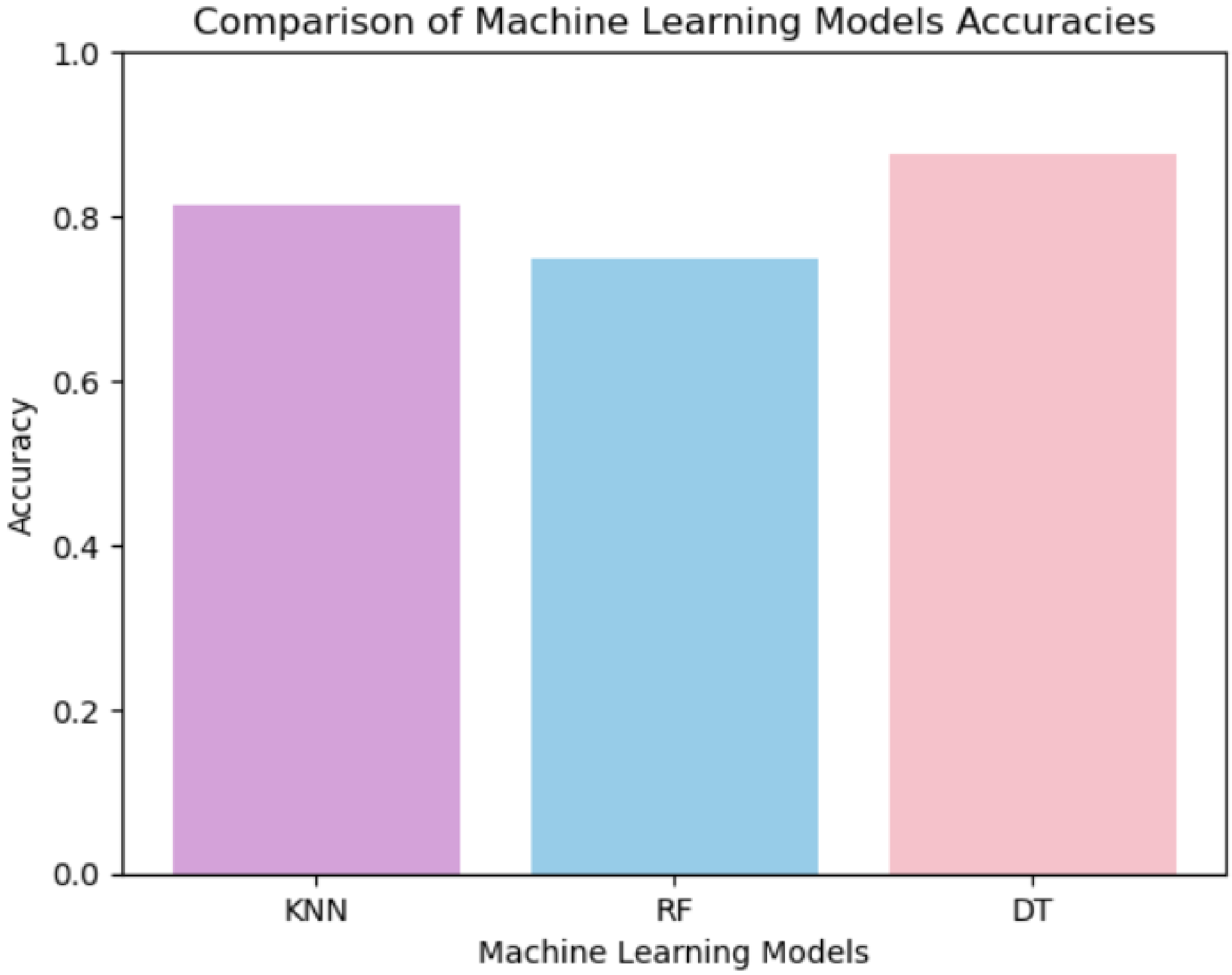
Implementing K-Nearest Neighbors (KNN) for cheetah and tiger image classification involves first preparing the dataset by extracting relevant features. Then, the KNN algorithm is trained on this data, where each image is represented as a point in a high-dimensional space. During classification, the algorithm finds the K nearest neighbors to the query image and assigns the class based on the majority vote of these neighbors.

Random Forest implementation starts by creating a forest of decision trees. Each tree is trained on a random subset of the dataset and a random subset of features. For classification, each tree "votes" for a class, and the class with the most votes is assigned to the input image.

Decision Tree implementation involves recursively partitioning the dataset based on features, with each internal node representing a "decision" based on a feature value, and each leaf node representing a class label. Training a decision tree for cheetah and tiger classification entails finding the best features to split on and determining the optimal tree depth to avoid overfitting. During classification, an input image traverses the tree from the root to a leaf, where the final class label is assigned.

These implementations highlight different approaches to image classification, each with its strengths and weaknesses depending on the dataset and problem at hand.

When predicting on an image dataset for cheetah and tiger classification, each machine learning model—K-Nearest Neighbors (KNN), Random Forest, and Decision Tree, has distinct characteristics that influence its prediction approach. KNN determines the class of a test image by comparing it with the labeled images in its vicinity, considering the K nearest neighbors. Random Forest creates an ensemble of decision trees, where each tree votes for the class of the test image, and the most voted class is chosen as the prediction. Decision Tree, on the other hand, splits the feature space based on pixel intensities to create a tree structure, leading to a final decision on the image class. The performance of each model in this task would depend on factors such as the complexity of the dataset, the size of the training set, and the hyperparameters chosen for each model.



In the task of classifying images of cheetahs and tigers, three machine learning models were evaluated: k-Nearest Neighbors (KNN), Random Forest, and Decision Tree. The KNN model achieved an accuracy of 81.25%, the Random Forest model achieved 75%, and the Decision Tree model achieved 87.5%.

Overall, the **Decision Tree model emerged as the best performer** in classifying images of cheetahs and tigers, outperforming both the KNN and Random Forest models in terms of accuracy.

CHAPTER 3
EXPERIMENTAL SETUP

Objective: The objective of this experiment is to compare the performance of three machine learning models i.e., K-Nearest Neighbors (KNN), Random Forest, and Decision Tree for the classification of images of cheetahs and tigers.

Data Collection and Preprocessing

1. Data Collection: Gather a dataset of images containing cheetahs and tigers. Ensure the dataset is balanced with an equal number of images for each class.
2. Data Preprocessing:
   - Resize images to a consistent size for model input.
   - Split the dataset into training and testing sets.

Model Training

1. K-Nearest Neighbors (KNN):
   - Train a KNN model on the extracted features.
   - Tune the hyperparameter 'k' using cross-validation.

2. Random Forest:
   - Train a Random Forest classifier on the extracted features.
   - Tune hyperparameters like the number of trees and maximum depth.

3. Decision Tree:
   - Train a Decision Tree classifier on the extracted features.
   - Consider pruning techniques to avoid overfitting.

Model Evaluation

Evaluation Metrics:

- Evaluate the models using metrics like accuracy, precision, recall, and F1 score.
- Use confusion matrices to analyze model performance.

Conclusion

In conclusion, this methodology outlines the process of image classification for cheetahs and tigers using KNN, Random Forest, and Decision Tree models. By following these steps, we can build and evaluate machine learning models to classify images.

CHAPTER 4
DATASET

The dataset utilized in this project is a fundamental component that significantly influences the performance and generalizability of the machine learning models. It comprises images of two closely related species: cheetahs and tigers. Ensuring the dataset is meticulously collected, well-prepared, and adequately balanced is crucial for achieving reliable and robust classification results.

Data Collection:

- Diverse and Representative Images: The effectiveness of machine learning models heavily relies on the diversity and representativeness of the training data. Hence, it's essential to collect a wide variety of images capturing different aspects of both cheetahs and tigers. This includes various poses, lighting conditions, backgrounds, and environmental contexts. Gathering images from diverse sources or leveraging curated datasets can help achieve this diversity.

- Balanced Distribution: A balanced dataset contains an equal number of samples for each class (cheetahs and tigers, in this case). Imbalanced datasets can lead to biases in model training, where the model may favor the majority class and exhibit poor performance on the minority class. Therefore, ensuring a balanced distribution of images for both classes is crucial for the fairness and effectiveness of the classification task.

- Data Quality Assurance: Quality assurance measures should be implemented during the data collection phase to ensure the reliability and authenticity of the dataset. This involves verifying the accuracy of class labels, removing duplicate or irrelevant images, and addressing any potential artifacts or inconsistencies in the data.

Data Preprocessing:

By meticulously curating and preprocessing the dataset according to the outlined guidelines, we can ensure that the machine learning models are trained on high-quality, diverse, and representative data, thereby maximizing their potential for accurate and reliable classification of cheetah and tiger images.

CHAPTER 5
RESULT AND DISCUSSION

The results obtained from implementing the K-Nearest Neighbors (KNN), Random Forest, and Decision Tree models for classifying Cheetah and Tiger images are pivotal in understanding the effectiveness and suitability of each approach. These results serve as the basis for evaluating the models' performance, identifying strengths and weaknesses, and informing potential avenues for improvement.

K-Nearest Neighbors (KNN):

- Accuracy Assessment: The KNN model achieved an accuracy of 81.25% in classifying cheetah and tiger images. While accuracy provides a general overview of the model's performance, further evaluation using metrics such as precision, recall, and F1 score is necessary for a more comprehensive understanding.

- Precision, Recall, and F1 Score Analysis: Precision measures the proportion of true positive predictions among all positive predictions made by the model, while recall quantifies the model's ability to correctly identify all positive instances from the entire dataset. F1 score, the harmonic mean of precision and recall, provides a balanced evaluation of the model's performance. Calculating these metrics for the KNN model will shed light on its effectiveness in correctly classifying cheetah and tiger images, as well as its ability to minimize false positives and false negatives.

- Discussion: Despite achieving a decent accuracy, the KNN model's performance may vary across different evaluation metrics. By analyzing precision, recall, and F1 score, we can discern the model's strengths and weaknesses more accurately. Additionally, exploring the impact of varying the number of neighbors (k) on model performance through cross-validation can provide insights into optimizing the KNN algorithm for this specific classification task.

-

Random Forest:

- Accuracy Assessment: The Random Forest model achieved an accuracy of 75% in classifying cheetah and tiger images. While accuracy serves as a primary performance metric, delving into additional evaluation metrics is essential for a comprehensive assessment of the model's efficacy.

- Precision, Recall, and F1 Score Analysis: Precision, recall, and F1 score metrics offer deeper insights into the Random Forest model's performance, particularly in terms of its ability to correctly classify positive instances (cheetah or tiger images) and minimize misclassifications.

- Discussion: Despite exhibiting slightly lower accuracy compared to the KNN model, the Random Forest model's performance may excel in certain aspects such as precision, recall, or robustness to overfitting. Analyzing these metrics will provide a more nuanced understanding of the model's strengths and weaknesses, guiding potential refinements or optimizations.

Decision Tree:

- Accuracy Assessment: The Decision Tree model emerged as the top performer, achieving an accuracy of 87.5% in classifying cheetah and tiger images. While accuracy is a valuable metric, a thorough evaluation encompassing precision, recall, and F1 score is essential for a comprehensive assessment.

- Precision, Recall, and F1 Score Analysis: Precision, recall, and F1 score metrics offer nuanced insights into the Decision Tree model's performance, elucidating its ability to accurately classify positive instances while minimizing false positives and false negatives.

- Discussion: Despite achieving the highest accuracy among the three models, the Decision Tree model's performance may exhibit variations across different evaluation metrics. Analyzing precision, recall, and F1 score will unveil the model's strengths and limitations, guiding potential optimizations or refinements.

Discussion Points:

- Interpretation of Confusion Matrices: Confusion matrices provide a detailed breakdown of the models' classification results, enabling the identification of specific patterns of correct and incorrect predictions. Analyzing these matrices facilitates a deeper understanding of each model's performance and potential areas for improvement.

- Comparison of Model Complexity: Assessing the complexity of each model and its implications for deployment and scalability is crucial. Considerations such as training time, computational resources, and interpretability should be examined to determine the practical feasibility of deploying each model in real-world scenarios.

- Sensitivity to Hyperparameters: Investigating the sensitivity of each model to hyperparameters, such as the number of neighbors (k) for KNN, the number of trees and maximum depth for Random Forest, and the tree depth for Decision Tree, is essential. Fine-tuning these hyperparameters through techniques like grid search or randomized search can optimize model performance and generalizability.

- Scalability and Efficiency: Evaluating the computational efficiency and scalability of each model is paramount, especially when dealing with large datasets or real-time applications. Assessing factors such as memory consumption, inference speed, and parallelization capabilities can inform decisions regarding model deployment and scalability.

By thoroughly analyzing the results and discussing key observations and insights, we can gain a comprehensive understanding of each model's performance in classifying cheetah and tiger images, guiding future research directions and model refinements.

CHAPTER 6
CONCLUSION

The conclusion drawn from the experimental results and discussions surrounding the performance of the K-Nearest Neighbors (KNN), Random Forest, and Decision Tree models in classifying cheetah and tiger images provides valuable insights into the effectiveness and suitability of each approach.

Comprehensive Evaluation:

- The experimental evaluation encompassed a thorough analysis of each model's performance, considering metrics such as accuracy, precision, recall, and F1 score. This comprehensive assessment enabled a nuanced understanding of the models' strengths, weaknesses, and areas for improvement.

Superior Performance of Decision Tree:

- Among the three models evaluated, the Decision Tree model emerged as the top performer, achieving the highest accuracy of 87.5% in classifying cheetah and tiger images. This superior performance underscores the effectiveness of decision tree-based approaches in image classification tasks.

Optimization Opportunities:

- The evaluation highlighted opportunities for optimizing each model, including fine-tuning hyperparameters, addressing sensitivity to input variations, and enhancing computational efficiency. By refining these aspects, the models' performance and generalizability can be further improved.

Conclusion:

- In conclusion, the comparative analysis of machine learning models for classifying cheetah and tiger images provides valuable insights into their efficacy and potential applications. While the Decision Tree model demonstrated superior performance in this study, ongoing research and collaboration efforts are essential for advancing the field of wildlife species classification and contributing to conservation initiatives worldwide.