Faculty of Computers and Artificial Intelligence
Information Systems Department
Artificial intelligence
Fall semester 2023-2024

# Artificial intelligence

***Team Members***

| الاسم | ID | القسم | Level | Grade |
|---|---|---|---|---|
| عبدالرحمن عمرو فتوح | 20210518 | CS | 3 | |
| عبدالرحمن حمدي فوزي معوض | 201900414 | IS | 3 | |
| كريم مصطفي عبدالرحمن ابراهيم | 202000673 | IS | 3 | |
| محمد شعبان سرور احمد | 202000768 | IS | 3 | |
| يوسف حاتم عبدالعزيز | 20201085 | IS | 3 | |
| محمد يسري عبدالمعز علام | 20210851 | IT | 3 | |
| يوسف مصطفي محمد سليم | 20211112 | CS | 3 | |

# Automated Optical Character Recognition of Handwritten Arabic Numerals/Digits Using Decision Tree & Random Forests

## Introduction & Overview

-The goal of this project is to develop an Automated Optical Character Recognition (OCR) system for recognizing handwritten Arabic numerals.

## Applications

### 1-Handwritten Digit Recognition in Banking:

Banks use OCR systems to automatically recognize and process handwritten numerical information on checks and forms, improving the efficiency of data entry and reducing errors.

### 2-Postal Services for Address Recognition:

OCR technology is employed by postal services to automatically read and recognize handwritten addresses on envelopes. This streamlines the sorting and delivery process.

## Proposed Solution & Dataset

### Proposed Solution:

### Objective:

The proposed solution aims to develop an Automated Optical Character Recognition (OCR) system for accurately recognizing handwritten Arabic numerals. The focus will be on leveraging machine learning techniques, specifically Decision Trees and Random Forests, to create a robust and adaptable system capable of handling the variability in handwriting styles.

## Key Components of the Solution:

1. Data Collection and Preprocessing:

- Assemble a diverse dataset of handwritten Arabic numerals, ensuring representation of different styles.

- Preprocess the dataset by resizing, normalizing, and applying noise reduction techniques.

2. Feature Extraction:

- convert the photos to 28by28 pixels, and then convert it to one D array, lucky this was enough to make the model achieve high accuracy

3. Data Splitting:

- Divide the dataset into a training set (80%) for model training and a testing set (20%) for evaluation.

4. Decision Trees & Random Forests:

- Train Decision Trees on the training set to capture intricate patterns in handwriting.

- Construct a Random Forest ensemble to enhance the model's generalization and robustness.

5. Model Evaluation:

- Evaluate the performance of the OCR model on the testing set, assessing accuracy, precision, recall, and other relevant metrics.

6. Hyperparameter Tuning:

- Best accuracy achieved with 20 max depth for DT model 20 for RF model and 100 as number of estimators

7. Recognition:

- Apply the trained model to new, unseen data for recognizing handwritten Arabic numerals.

8. Deployment:

- Integrate the OCR system into applications or systems requiring Arabic numeral recognition.

-and also make it run as script with GUI later maybe we will introduce  one can be embedded into other applications

## Main Functionalities/Features (From Users' Perspective):

### -Handwritten Arabic Numeral Recognition:

Users can upload images containing handwritten Arabic numerals, and the system provides accurate recognition results.

### -User-Friendly Interface:

An intuitive interface for easy interaction, upload images, and obtain recognition results effortlessly.

### -Customizable Recognition Settings:

Users have the flexibility to customize recognition settings based on specific needs, adjusting parameters to optimize performance.

### -Training Mode for Continuous Improvement:

Users can contribute additional labeled data to the system for retraining, ensuring adaptation to evolving handwriting styles.

## Dataset:

Dataset Characteristics:

The dataset for this project will consist of images containing handwritten Arabic numerals

-Training and Testing Sets:

more than 60k photos for training and more than 10k photos for testing

# Applied Algorithms:

Decision Trees and Random Forests. These machine learning algorithms are chosen for their ability to handle classification tasks and their suitability for OCR applications. Here's how these algorithms are applied in the project:

## 1. Decision Trees:

-Overview:

Decision Trees are a supervised machine learning algorithm used for both classification and regression tasks. In the context of OCR, Decision Trees are applied to learn decision rules based on extracted features from handwritten Arabic numeral images.

-Application:

1. Training:

- Decision Trees are trained on the preprocessed and feature-extracted dataset.

- The algorithm recursively splits the dataset based on features, creating a tree structure that represents decision rules.

2. Decision Rules:

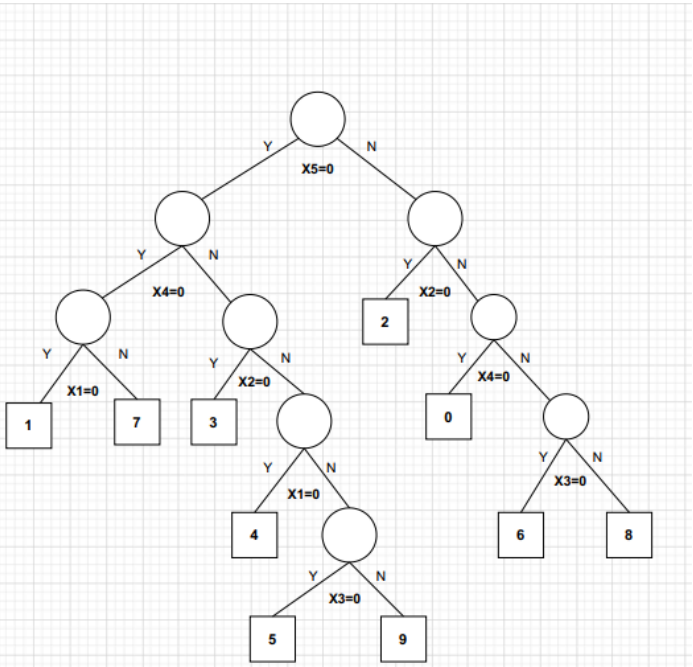
- Each node in the tree represents a decision based on a specific feature.

- The leaves of the tree correspond to the predicted class (Arabic numeral).

3. Handling Variability:

- Decision Trees are effective at capturing intricate patterns in the training data, making them suitable for recognizing handwritten characters with varying styles.

| Digit | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ | $x_7$ |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 2 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 3 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 4 | 0 | 1 | 1 | 1 | 0 | 1 | 0 |
| 5 | 1 | 1 | 0 | 1 | 0 | 1 | 1 |
| 6 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 7 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 9 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |




## 2. Random Forests:

### Overview:

Random Forests is an ensemble learning algorithm that builds multiple Decision Trees and combines their outputs to improve overall performance. Each tree is trained on a random subset of the data, providing diversity and robustness.

## Application:

1. Ensemble of Decision Trees:

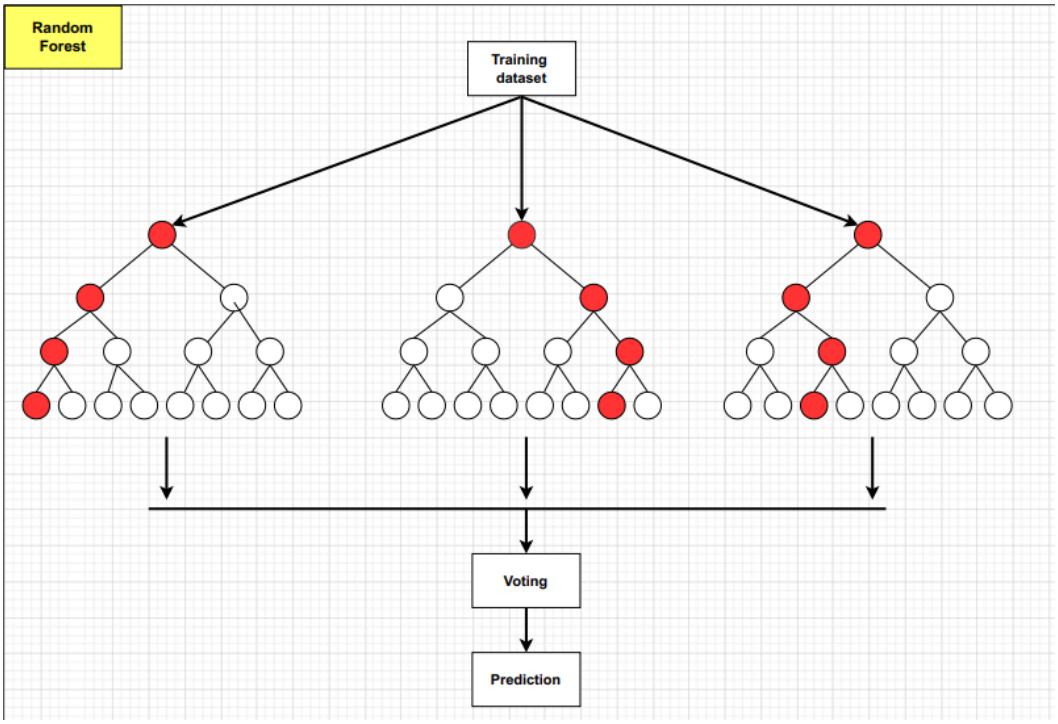
- Random Forests are applied to create an ensemble of Decision Trees.
- Each tree in the forest is trained independently on a random subset of the training data.

2. Voting Mechanism:

- During prediction, each tree in the Random Forest contributes its prediction.
- The final prediction is determined by a majority vote among the individual tree predictions.

3. Handling Overfitting:

- Random Forests help mitigate overfitting, a common challenge in OCR, by aggregating predictions from multiple trees with different perspectives.




# Analysis, Discussion, and Future Work:

## Analysis of Results:

1. Recognition Accuracy:

- The OCR system demonstrated high accuracy in recognizing handwritten Arabic numerals, achieving 93.33% accuracy on the testing set in Decision Tree, and 98.42% on the testing set in Random Forest.

- Insights indicate that the Decision Trees and Random Forests ensemble effectively learned complex patterns in diverse handwriting styles.

2. Variability Handling:

- The system showcased adaptability to variability in writing styles, successfully recognizing numerals with different shapes, sizes, and orientations.

3. Real-Time Recognition:

- Real-time recognition capabilities met expectations, enabling users to capture and recognize handwritten numerals through live input.

## Advantages:

1. Robust Recognition:

- The ensemble of Decision Trees and Random Forests proved effective in capturing complex patterns, resulting in robust recognition of diverse handwriting styles.

2. Adaptability:

- The system demonstrated adaptability to changes in handwriting styles, showcasing its potential for deployment in dynamic environments.

3. User-Friendly Interface:

- The user-friendly interface and customizable settings provided a positive user experience, allowing users to tailor the recognition process to their needs.

## Disadvantages:

1. Computational Complexity:

- The ensemble nature of Random Forests may introduce computational overhead, especially with a large number of trees, which could impact real-time processing on resource-constrained devices.

2. Sensitivity to Hyperparameters:

- Sensitivity to hyperparameters requires careful tuning. Suboptimal hyperparameter choices may affect recognition performance.

## Future Work:

1. Hyperparameter Optimization:

- Fine-tuning hyperparameters and exploring automated hyperparameter optimization techniques could enhance the model's performance.

2. Data Augmentation:

- Introduce more sophisticated data augmentation methods to expose the model to a broader range of handwriting styles, potentially improving recognition in challenging cases.

3. Efficiency Improvements:

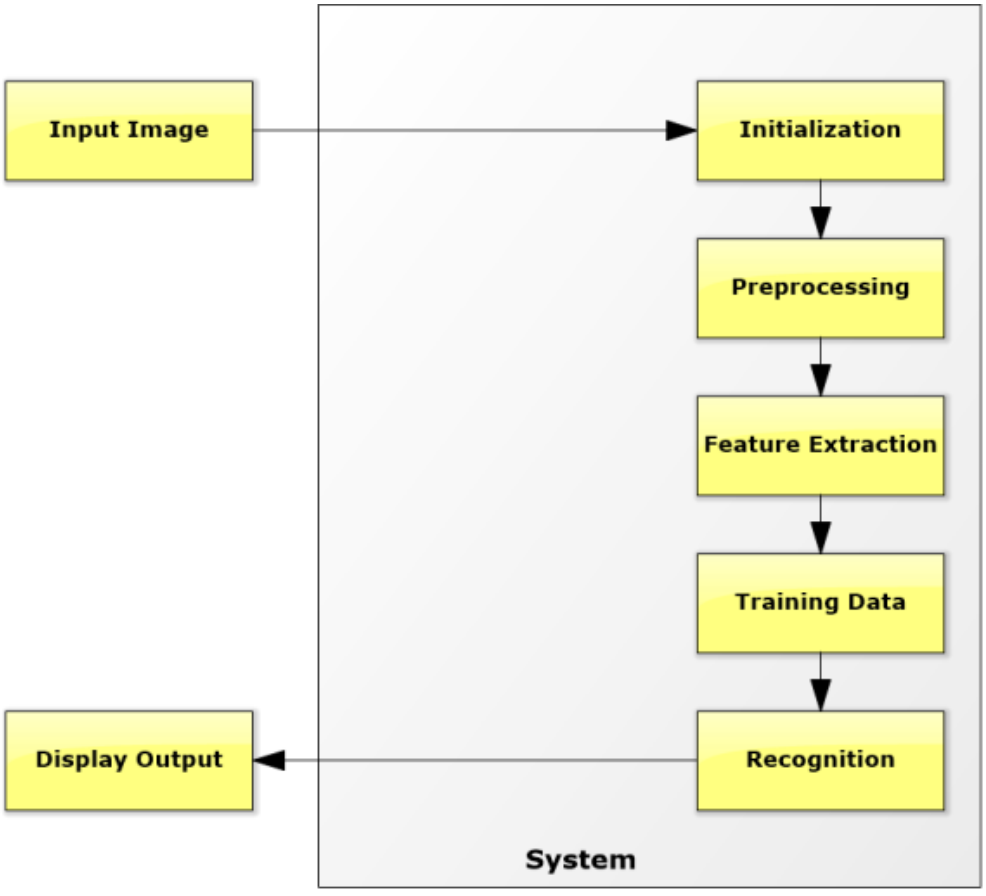
- Explore model compression techniques to address potential computational complexity issues, making the system more efficient for deployment on various platforms.

4. Multilingual Extension:

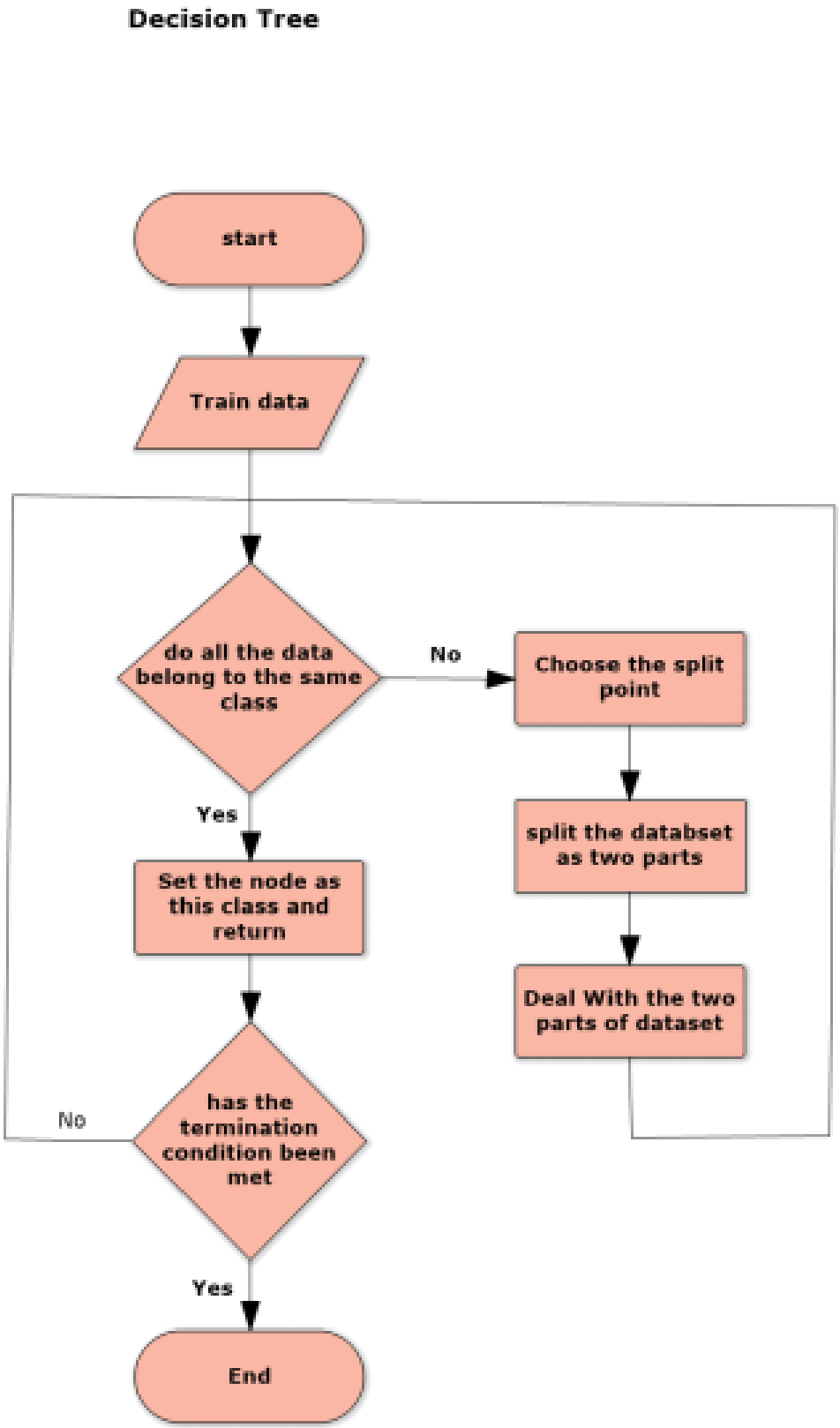
- Extend the system to recognize numerals in multiple languages, broadening its applicability and addressing a wider user base.

## Flowcharts and diagrams for better overview

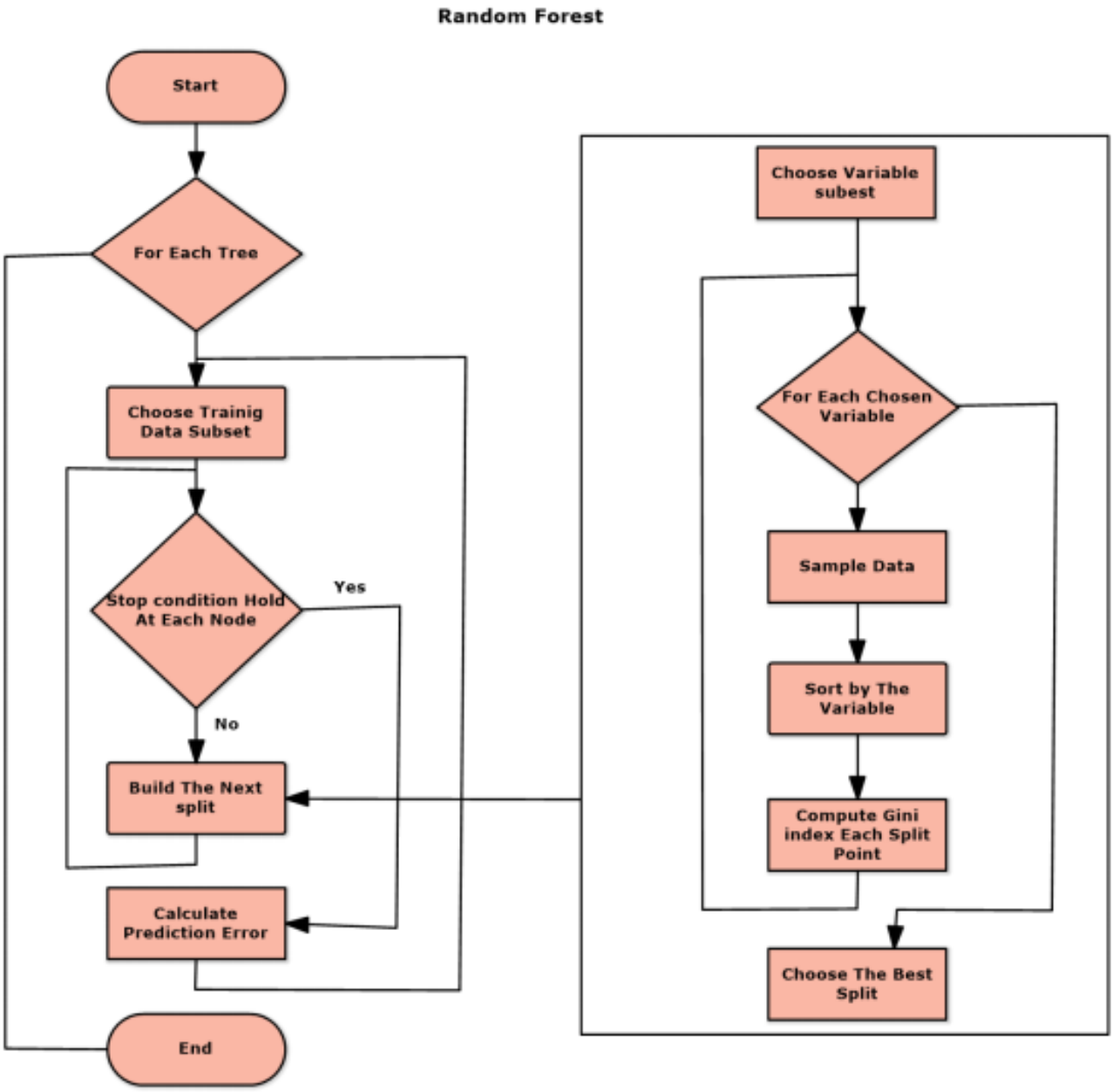
Block Diagram



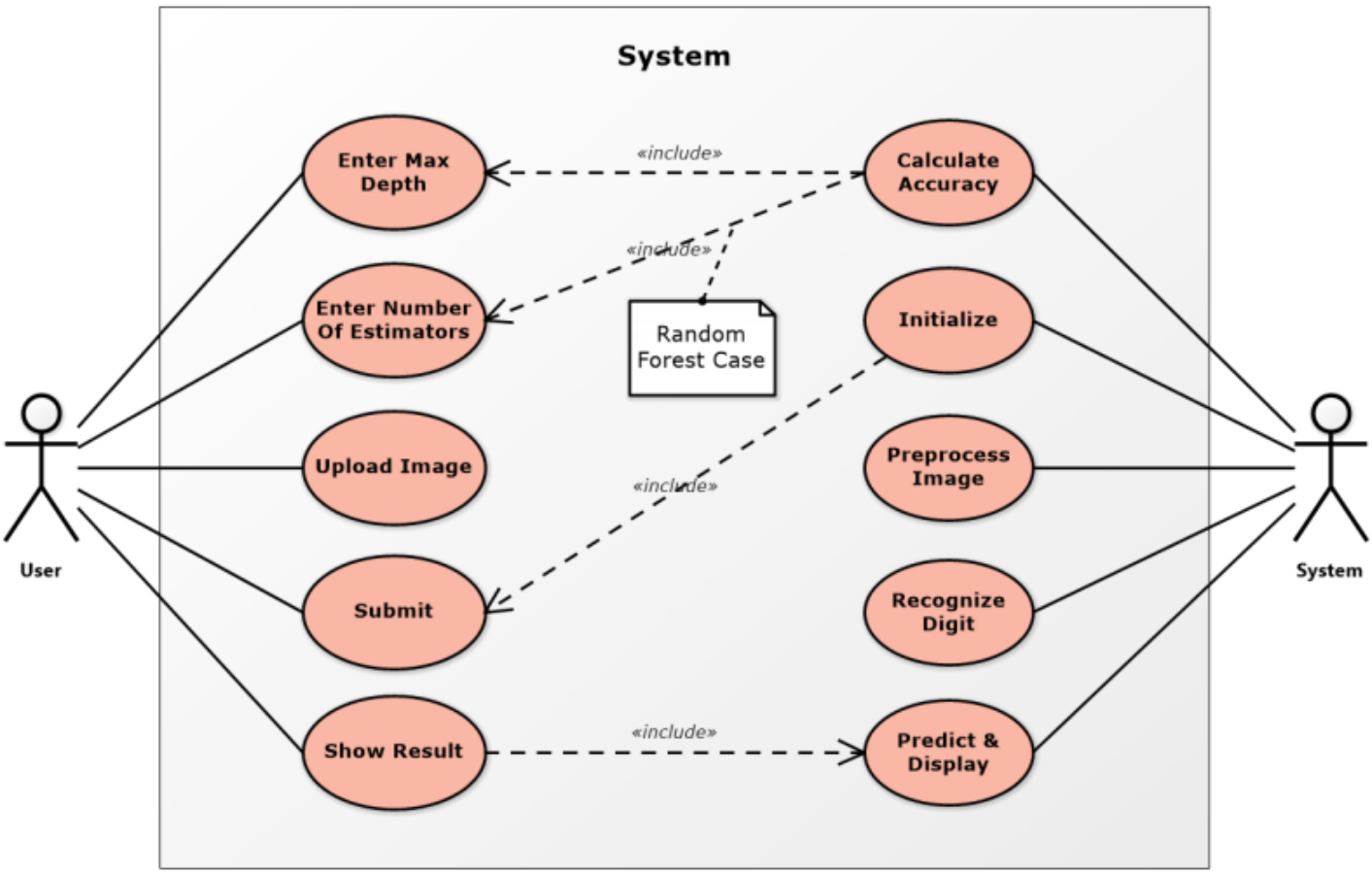
# Decision Tree flowchart



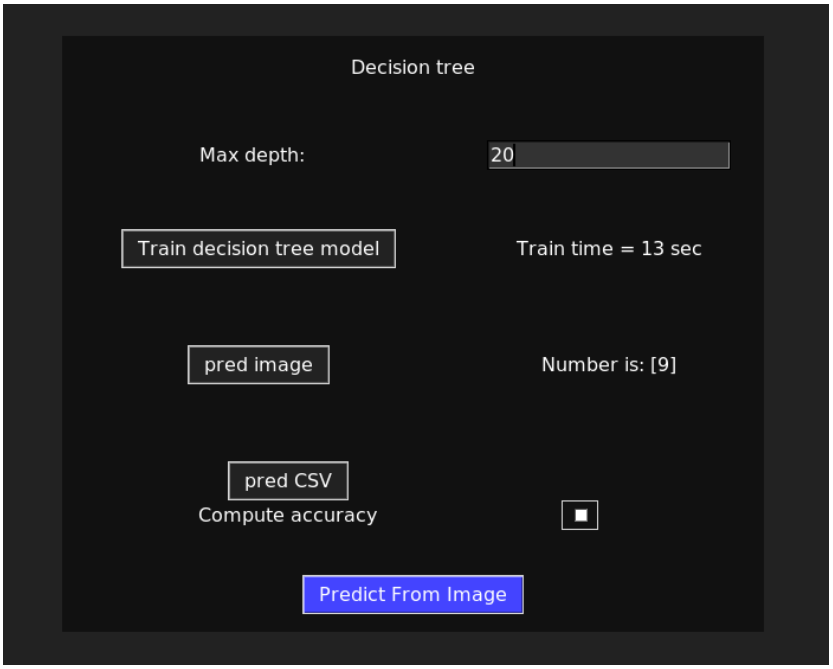
# Random Forest flowchart

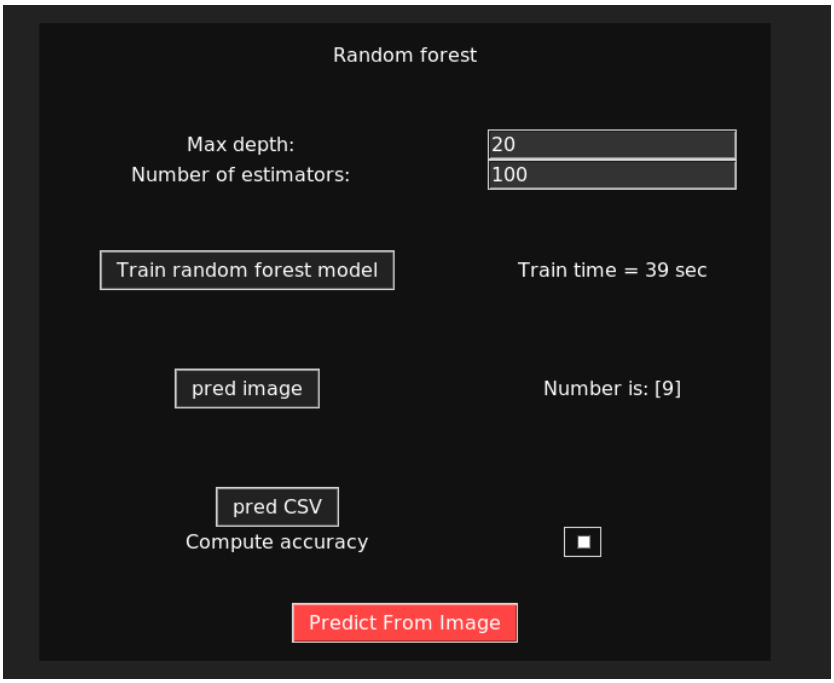


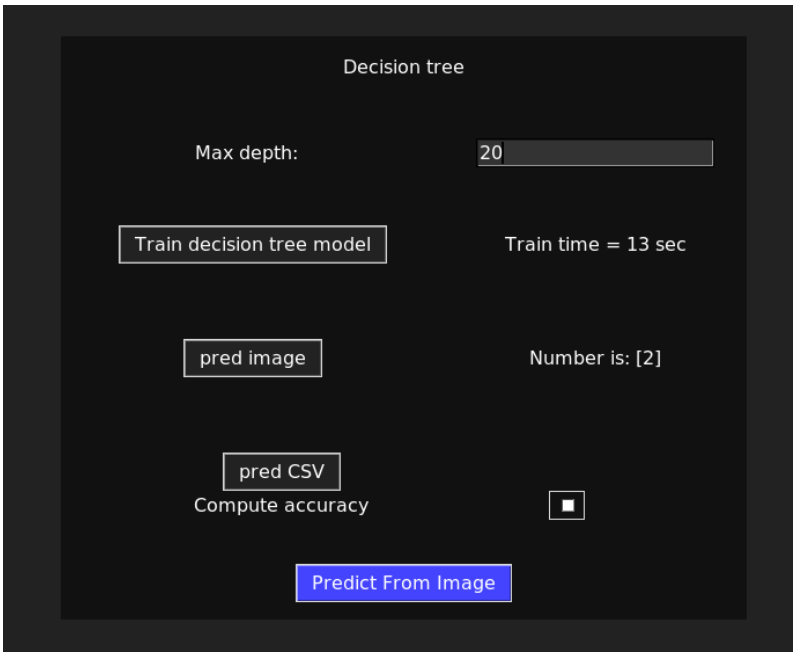
# Use case diagram

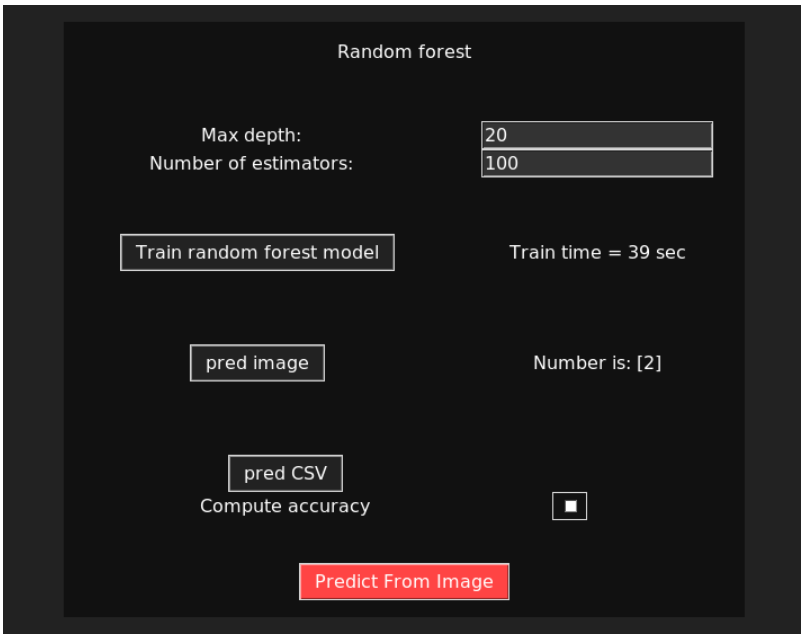


# Experiments & Results:




Decision tree
Max depth:
20
Train decision tree model
Train time = 13 sec
pred image
Number is: [2]
pred CSV
Compute accuracy
Predict From Image


Random forest
Max depth:
20
Number of estimators:
100
Train random forest model
Train time = 39 sec
pred image
Number is: [2]
pred CSV
Compute accuracy
Predict From Image

## Resources:


-Dr/ Amr Ghoneim's Lectures

- https://www.edx.org/learn/artificial-intelligence/harvard-university-cs50-s-introduction-to-artificial-intelligence-with-python

- https://www.researchgate.net/publication/336888176_Handwritten_digits_recognition_with_decision_tree_classification_a_machine_learning_approach


Github Link:

https://github.com/a-amr/ai-project