# FARZI

## Introduction

The rise of counterfeit currency poses a significant challenge to financial systems worldwide. To address this issue, an effective currency note authentication system is essential. This report outlines the design and implementation of such a system, focusing on the methods used to distinguish between genuine and counterfeit notes. The system utilizes advanced image processing techniques and machine learning models to achieve accurate and reliable authentication.

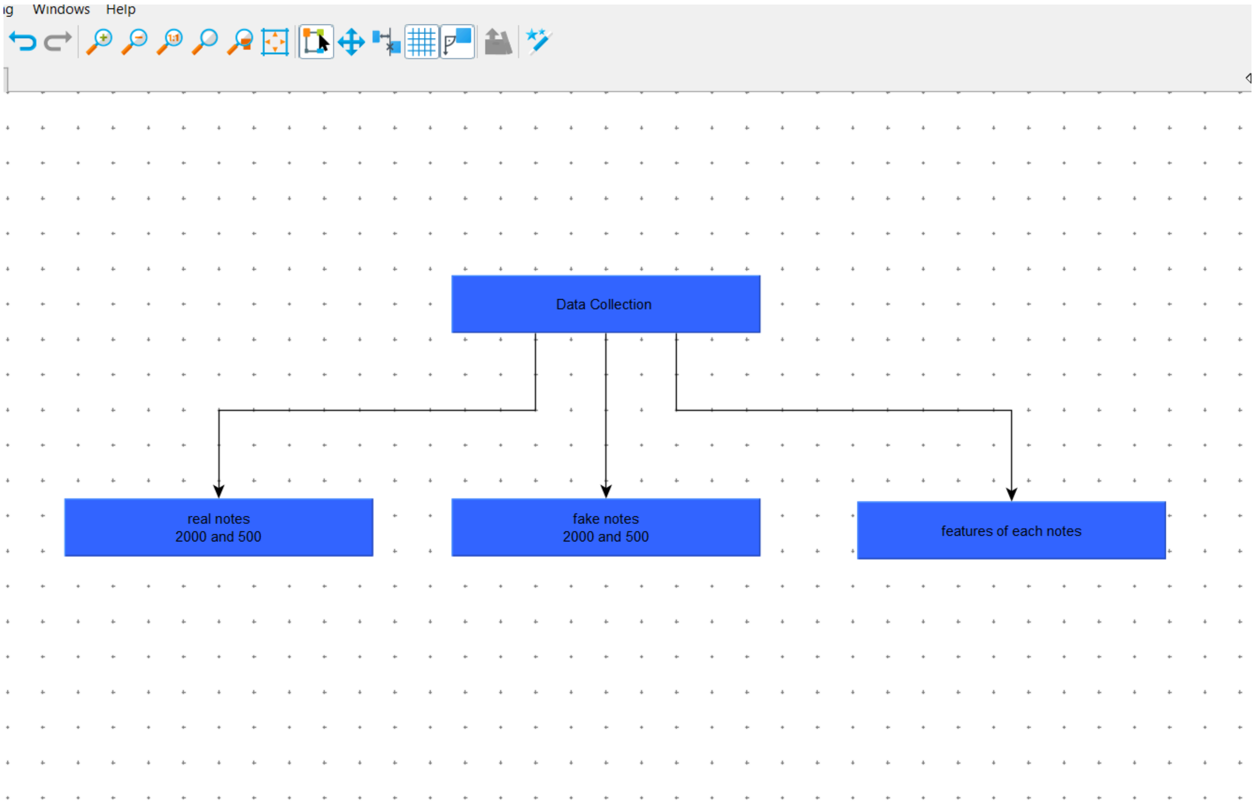
## Contents

## 1. Objective

The primary goal of the currency note authentication system is to differentiate between genuine and counterfeit notes. By leveraging image processing and machine learning techniques, the system aims to provide accurate, real-time validation of currency notes, enhancing security and reducing the circulation of fake currency.

## 2. Data Collection

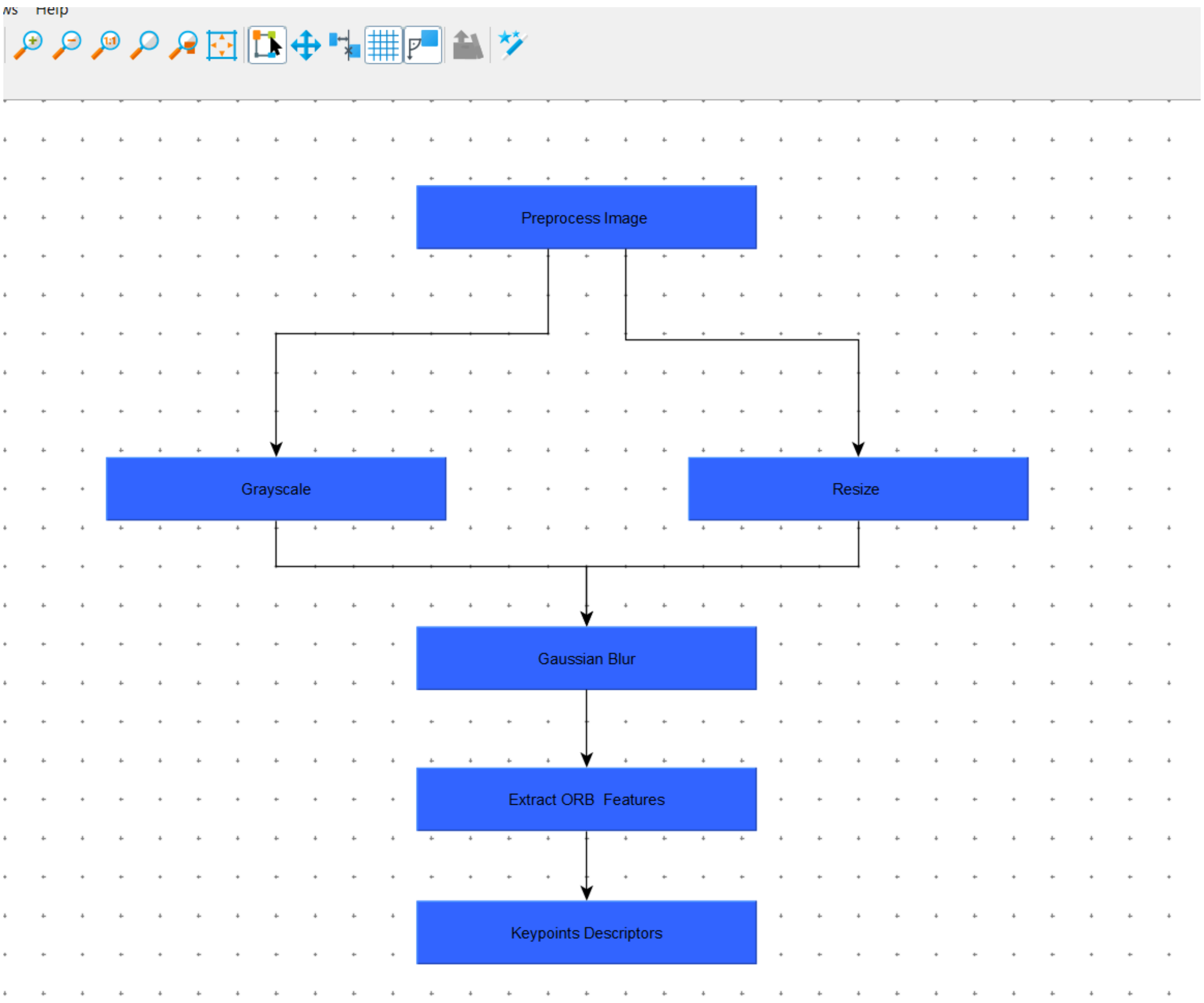
### Description



Data collection involves acquiring images of both genuine and counterfeit currency notes. This data serves as the foundation for training and evaluating the authentication system. The dataset includes images of real notes and various fake notes, categorized by denomination (e.g., 2000 INR and 500 INR).

## Process

- **Real Notes**: Collect high-resolution images of genuine currency notes.
- **Fake Notes**: Gather images of counterfeit notes, ensuring a diverse set of examples for each denomination.



# 3. Feature Extraction

## Description

Feature extraction transforms raw images into a format suitable for analysis. This process involves identifying key characteristics or features from the currency note images that are crucial for classification.

## Steps

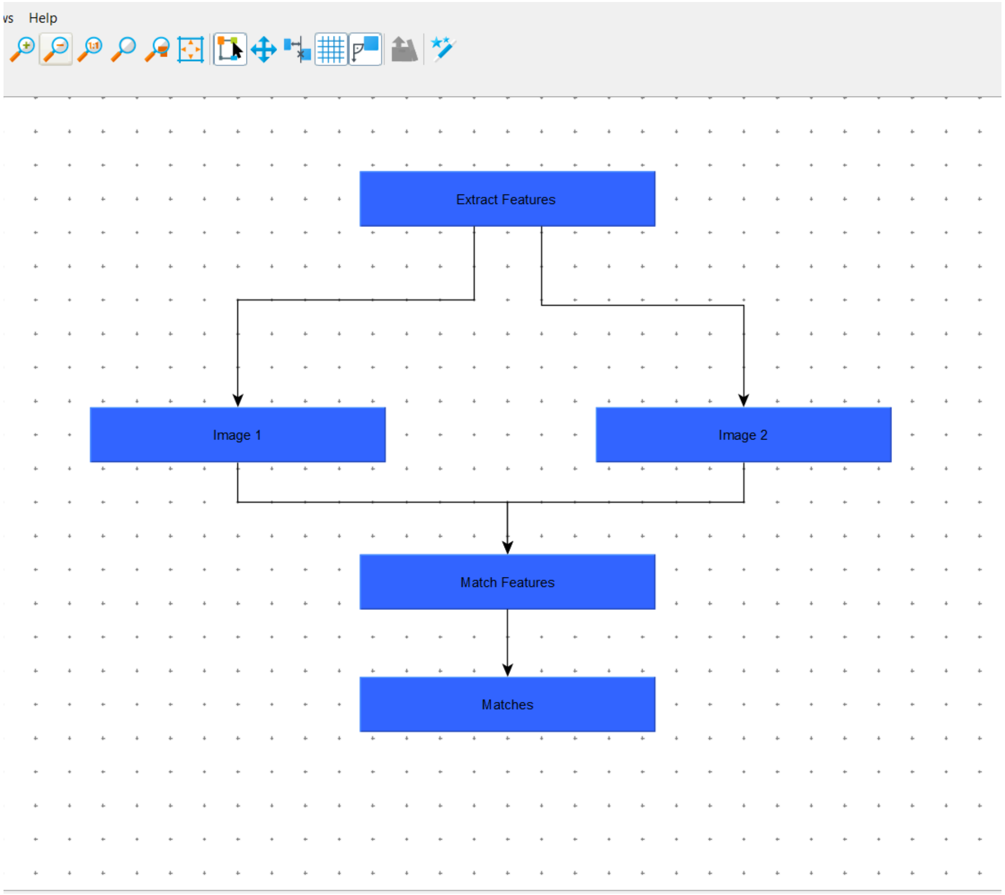
1. **Load Image**: Read the image file into the system.
2. **Preprocess Image**:
   - Convert the image to grayscale to simplify processing.

    - Resize the image to ensure uniformity.
    - Apply Gaussian blur to reduce noise.
3. **Extract Features Using ORB**:
    - Utilize ORB (Oriented FAST and Rotated BRIEF) to detect keypoints and compute descriptors from the processed image.

# 4. Feature Matching

## Description

Feature matching involves comparing features extracted from the test image with those in a database of known templates to find similarities and identify the authenticity of the note.



## Steps

1. **Extract Features**: Obtain keypoints and descriptors for the test image.
2. **Match Features**:

    - Use a Brute-Force Matcher to compare descriptors from the test image with those from template images.
    - Identify the best matches based on descriptor similarity.

# 5. Template Matching

## Description

Template matching is used to identify the best matching template from a set of known templates. This process helps in recognizing and verifying the authenticity of currency notes.

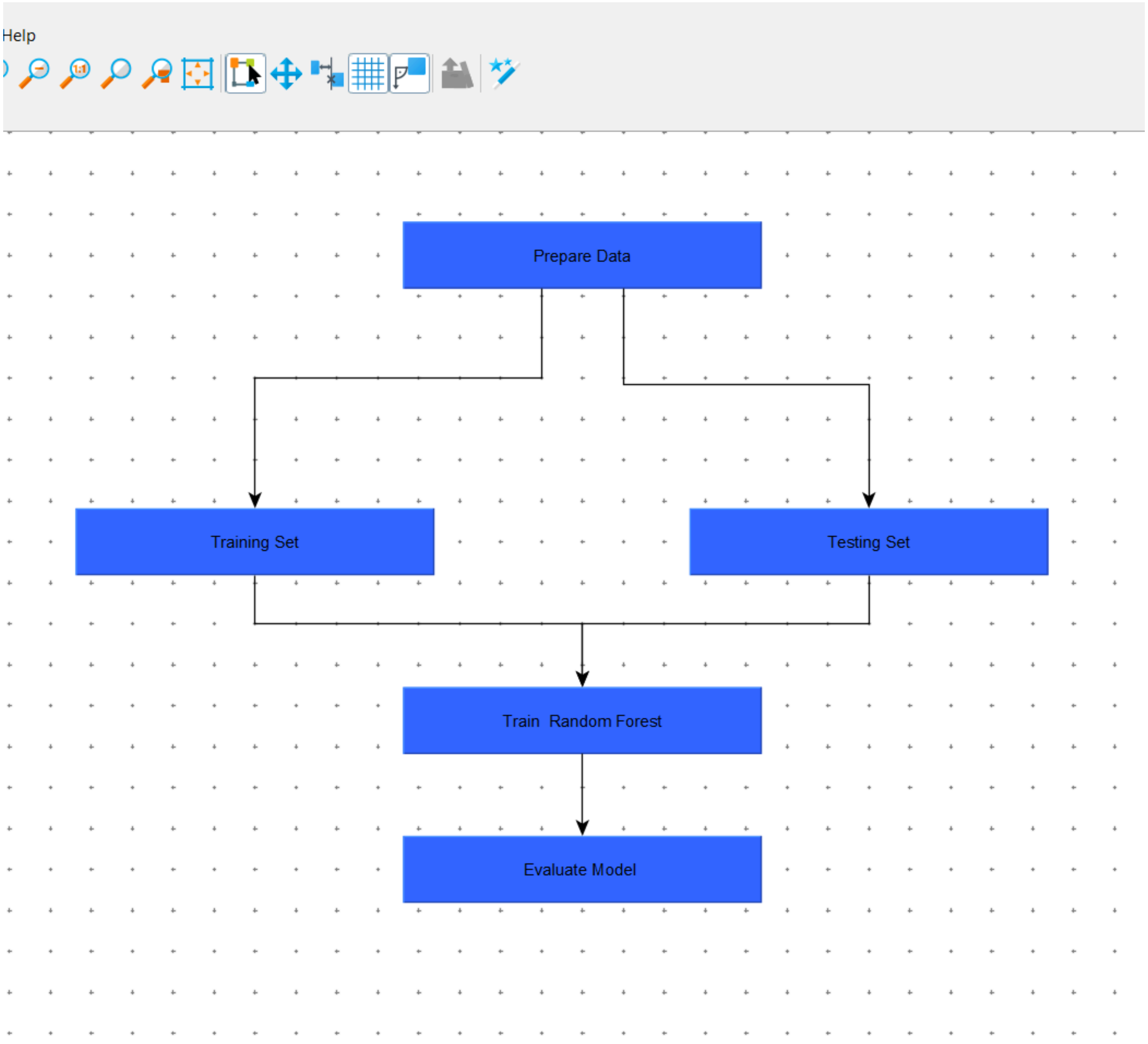
## Steps

1. **Feature Extraction**: Extract features from the test image.
2. **Template Comparison**:
    - Compare the extracted features from the test image to features in various templates.
    - Select the template with the highest similarity score as the best match.

# 6. Random Forest Classifier

## Description

The Random Forest classifier is a machine learning model used to classify currency notes based on their features. It builds multiple decision trees and aggregates their results to improve accuracy.



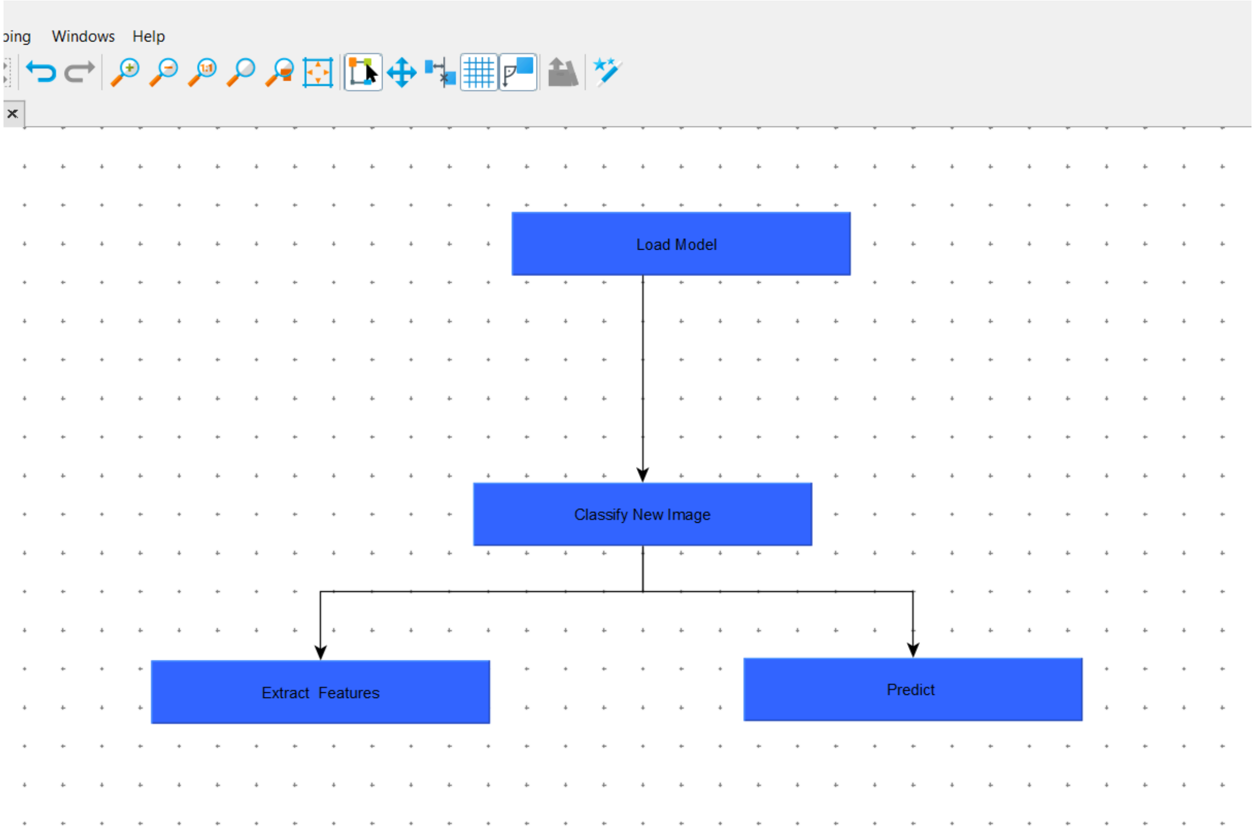
### Steps

1. **Prepare Data**:
   - Split the collected data into training and testing sets.
2. **Train Model**:
   - Use the training data to build and train the Random Forest classifier.
3. **Evaluate Model**:
   - Assess the model's performance using metrics such as accuracy, precision, and recall on the test data.

## 7. Model Deployment

### Description

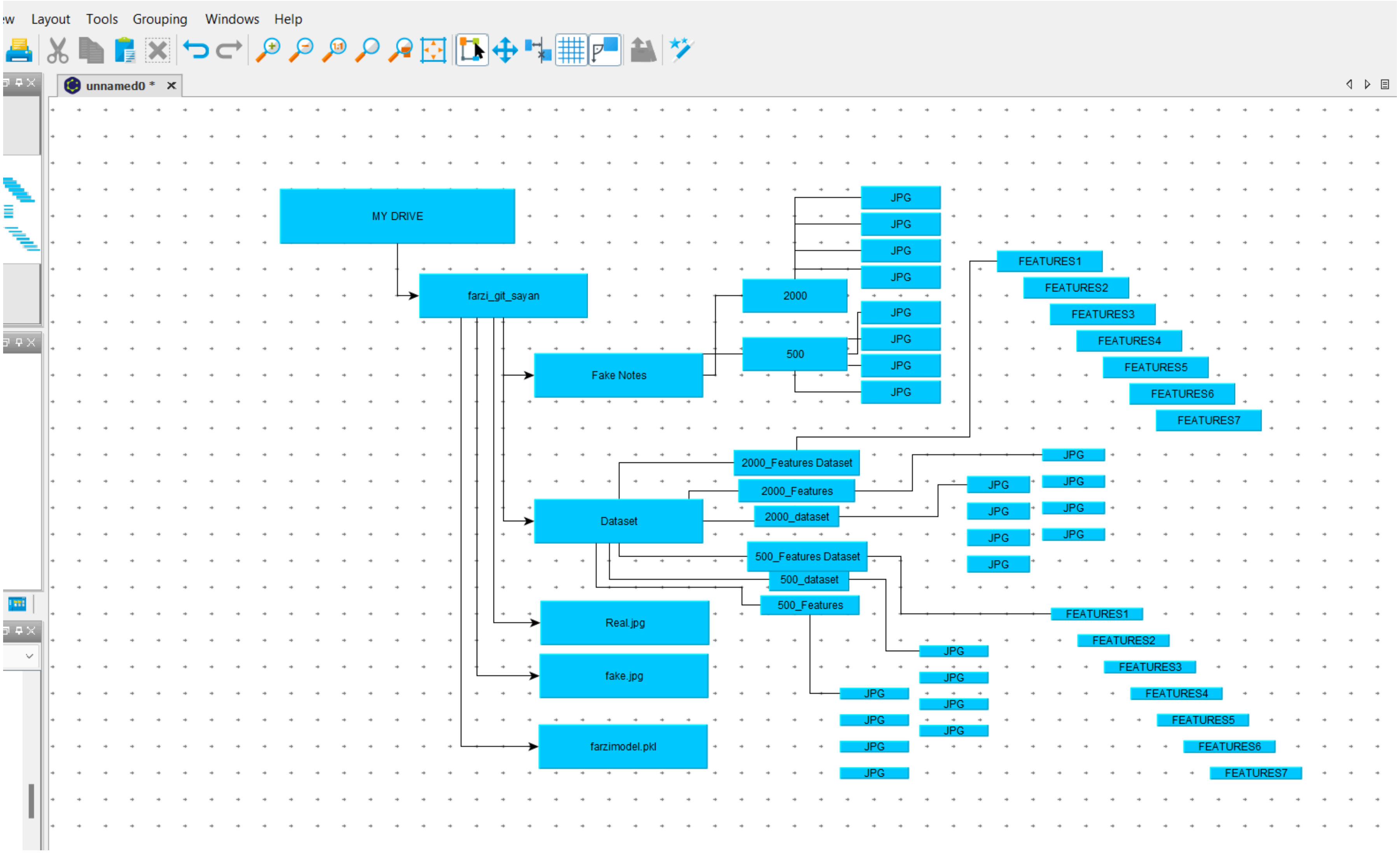
Model deployment involves integrating the trained machine learning model into a real-time system for classifying new images of currency notes.



## Steps

1. **Load Model**: Retrieve the trained model from storage.
2. **Classify New Image**:
   - Extract features from the new image.
   - Use the model to predict whether the note is real or fake based on the extracted features.

Folder structure is below:


ew Layout Tools Grouping Windows Help
unnamed0 *
MY DRIVE
farzi_git_sayan
Fake Notes
2000
500
JPG
JPG
JPG
JPG
JPG
JPG
JPG
JPG
FEATURES1
FEATURES2
FEATURES3
FEATURES4
FEATURES5
FEATURES6
FEATURES7
Dataset
2000_Features Dataset
2000_Features
2000_dataset
JPG
JPG
JPG
JPG
JPG
JPG
JPG
JPG
500_Features Dataset
500_dataset
500_Features
FEATURES1
FEATURES2
FEATURES3
FEATURES4
FEATURES5
FEATURES6
FEATURES7
JPG
JPG
JPG
JPG
JPG
JPG
JPG
JPG
Real.jpg
fake.jpg
farzimodel.pkl