ARTIFICIAL INTELLIGENCE &

MACHINE LEARNING

Project Documentation format

1. Introduction

- **Project Title:** “Pattern Sense: Classifying Fabric Patterns using Deep Learning”
- **Team Members:** K N Abdul Azeem(Deployment & Testing), K Venu(Designed UI), B Narasimhulu(Model Evaluation), J Chandureddy(Backend Development)

2. Project Overview

- **Purpose:** To automate the identification and classification of fabric patterns using deep learning.The project is designed to help industries like fashion,textiles and interior design by speeding up the pattern recognition process , reducing human error and enabling smart sorting and categorization of fabric types.
- **Features:** a. Automated Pattern Recognition
  b. Multi-Class Classification
  c. User-friendly Interface
  d. High Accuracy Model

3. Architecture

- **Frontend:** HTML, CSS
- **Backend:** Python, OpenCV, CNN
- **Database:** SQLite/MYSQL

4. Setup Instructions

- **Prerequisites:** Python, TensorFlow, Keras, Flask, Numpy, Pandas, Scikit-learn.
- **Installation:**

  i. Clone the Repository :

  ```
  git clone https://github.com/AbdulAzeem18/Pattern-Sense-Classifying-Fabric-Patterns-using-Deep-Learning
  ```

  ii. Create a Virtual Environment :

  ```
  python -m venv venv
  Venv\Scripts\activate
  ```

  iii. Install Dependencies :

  ```
  pip install -r requirements.txt
  ```

  iv. Setup the Environment Variables:

  ```
  touch.env
  ```

  v. Run the Application:

  ```
  Flask run
  ```

5. Folder Structure

- **Client:** It contains the user interface (index.html, style) in the frontend.
- **Server:** It loads the deep learning model, process the uploaded image and returns the predicted class and communicates with the frontend using Flask routes(app.py)

at the backend.

6. Running the Application

Install dependencies, setup the environment and run the application using flask.

7. Authentication

Authentication: Verifies who the user is (identity)

i. User Registration
ii. User Login
iii. Session Management

Authorization: Determines what the user can do (permissions)

i. Use Flask-Login’s @login_required decorator to protect sensitive routes.

8. User Interface

The Pattern Sense UI is a clean, simple, and user-friendly web interface designed allow to upload fabric images and view the predicted pattern type.

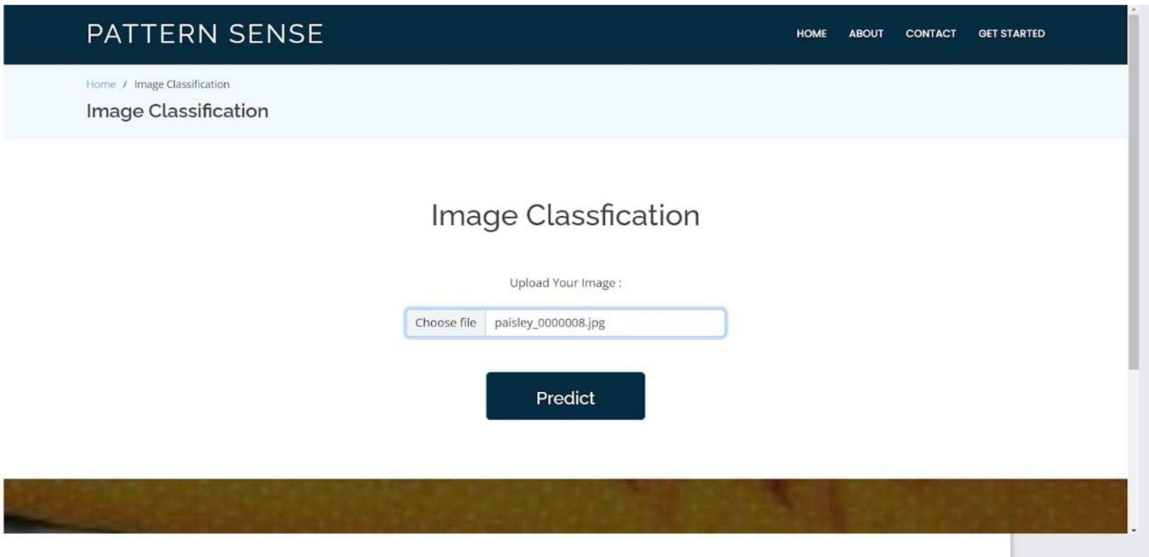
9. Testing

a. Model Testing(Deep Learning Pipeline)
Tools Used:
i. Python
ii. TensorFlow
iii. Scikit-learn
iv. Matplotlib/Seaborn

b. Backend Testing(Flask API)
Tools Used:
i. Flask-Testing

c. Frontend Testing(HTML- based Client)
Tools Used:
i. Manual Testing

10. Screenshots



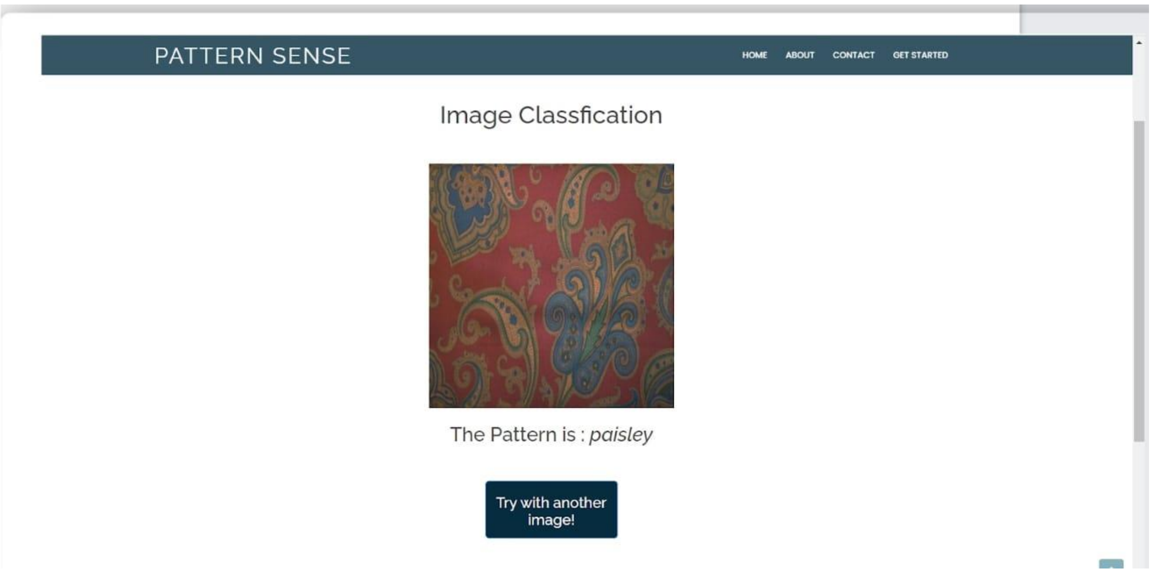


Image Classfication

The Pattern is : *paisley*

11. Known Issues

i. Limited Dataset Diversity
ii. Model Over fitting on Small Datasets
iii. Prediction Latency
iv. Error Handling for Invalid Inputs
v. Authentication Token Expiry(if implemented)
vi. No Offline Functionality
vii. Frontend Compatibility
viii. Lack of Real-Time Training Feedback

12. Future Enhancements

i. Dataset Expansion
ii. Mobile App Integration
iii. Offline Prediction Mode
iv. Pattern Detection and Segmentation
v. Pattern Similarity Search
vi. User Feedback Loop
vii. Multi-language Support
viii. Voice-Based Interface