# Project Title:

## Pattern Sense: Classifying Fabric Patterns Using Deep Learning

---

### 📂 Overview:

Fabric pattern classification plays a crucial role in textile, fashion, and e-commerce industries. This project proposes a deep learning solution to automatically classify fabric images into categories based on their visual patterns using Convolutional Neural Networks (CNNs).

---

### 🎯 Objectives

- Develop an image classification model using CNN to detect fabric patterns.
- Integrate the model with a user-friendly Flask-based web interface.
- Enable real-time prediction by uploading images through the browser.

---

### 🧠 Technologies Used

- Python 3.x
- TensorFlow / Keras
- OpenCV / PIL (Pillow)
- NumPy / Matplotlib
- Flask
- HTML5 / CSS3
- Bootstrap (Optional for UI)

---

### 🏗️ Project Structure

```
Pattern-Sense/
├── app.py                  # Flask app script
├── model.h5                  # Trained Keras model
├── requirements.txt            # Python dependencies
├── static/
│   └── styles.css            # Optional: CSS styles
├── templates/
```

```
|   ├── home.html          # Upload form
|   └── result.html        # Prediction result page
└── uploads/               # Temporarily stores uploaded images
```

---

## 📥 Setup Instructions

1. **Clone the project**
   Download or extract the ZIP into a local folder.
2. **Create a virtual environment**
3. python -m venv venv
4. source venv/bin/activate     # On Windows: venv\Scripts\activate
5. **Install dependencies**
6. pip install -r requirements.txt
7. **Ensure model file is present**
   The model.h5 file should be placed in the root project directory.
8. **Run the app**
9. python app.py
10. **Visit the web app**
    Open http://127.0.0.1:5000 in your browser.

---

## 🖼️ How It Works

1. **User uploads an image** of a fabric.
2. The server:
   - Preprocesses the image (resize to 224x224, normalize).
   - Loads the pretrained model (model.h5).
   - Runs prediction and maps it to the corresponding class name.
3. The prediction result is shown on the result.html page.

---

## 🔍 Key Functions (app.py)

- @app.route('/'): Loads the home page with the upload form.
- @app.route('/upload', methods=['POST']): Handles file upload, image preprocessing, prediction, and rendering results.

## 🔤 Model Details

- **Input Shape**: 224x224x3
- **Model Type**: Convolutional Neural Network (Keras Sequential)
- **Training**: (Optional step, as pretrained model is used)
- **Classes**: Pattern categories (e.g., stripes, polka dots, floral — based on training dataset)

---

## ✅ Features

- Simple and responsive UI
- Real-time fabric image classification
- Easy to deploy and extend
- Clean error handling for missing files and invalid formats