Project Report Format

1.INTRODUCTION

Project Overview:

Pattern Sense: Classifying Fabric Patterns using Deep Learning is an AI-based project designed to automate the process of identifying and categorizing fabric patterns using advanced deep learning techniques. The project uses convolutional neural networks (CNNs) to analyze fabric images and classify them into categories such as stripes, polka dots, floral prints, and geometric designs. By replacing manual inspection with automated recognition, Pattern Sense significantly improves efficiency and accuracy in industries like fashion, textiles, and interior design. It can be used for tasks such as streamlining design selection, conducting quality control by detecting pattern defects, and helping interior designers match fabrics to décor styles. Built using technologies like Python, TensorFlow, OpenCV, and Streamlit, this system provides a scalable, real-time solution for both production and retail environments.

Purpose:

The purpose of this project is to develop an intelligent system that can automatically identify and classify fabric patterns using deep learning techniques. In industries like fashion, textiles, and interior design, manually analyzing and categorizing fabric patterns is time-consuming and prone to human error. By leveraging computer vision and neural networks, Pattern Sense aims to streamline this process, making it faster, more accurate, and highly efficient. The system helps designers, manufacturers, and quality control teams save time, reduce labor costs, and ensure consistent pattern recognition. Ultimately, the project seeks to bring innovation and automation to traditional fabric-related workflows by using state-of-the-art artificial intelligence.

2.IDEATION PHASE

Problem Statement:

In the fashion, textile, and interior design industries, identifying and categorizing fabric patterns is a critical yet time-consuming task that relies heavily on manual inspection. This traditional approach is not only labor-intensive but also prone to human error and inconsistency, especially when dealing with large volumes of fabric samples or product images. As the demand for faster production and accurate quality control increases, there is a growing need for an automated solution that can efficiently recognize and classify different fabric patterns. The lack of such intelligent systems limits productivity and slows down decision-making processes. Therefore, this project aims to solve the problem by developing a deep learning-based model that can accurately and automatically classify fabric patterns from images, ensuring speed, consistency, and scalability in real-world applications.

Empathy Map Canvas:

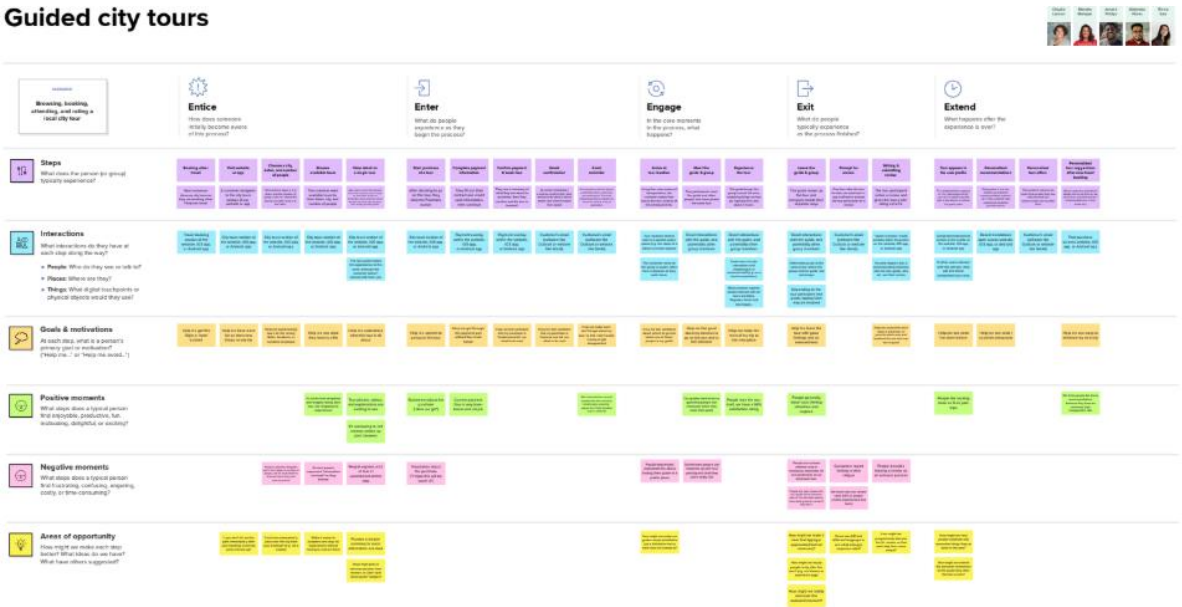
The target users for this project include fashion designers, textile quality control inspectors, interior designers, and fabric manufacturers. These users often express the need for faster, more efficient ways to categorize and identify fabric patterns. They talk about the challenges of manually inspecting large volumes of fabrics and the difficulty in ensuring accuracy and consistency. In practice, they spend a significant amount of time comparing designs, inspecting patterns for defects, and selecting suitable fabrics for their needs. Internally, they think there must be smarter solutions that can automate these repetitive tasks and help reduce human error. Many of them feel frustrated, overwhelmed, or stressed due to the manual workload, tight deadlines, and high standards of quality. Their main pain points include the time-consuming nature of manual classification, inconsistency in quality control, and difficulty in matching patterns with design requirements. By using Pattern Sense, they can gain significant benefits such as faster and more accurate pattern recognition, reduced manual effort, improved workflow efficiency, and higher customer satisfaction.

Brainstorming:

During brainstorming for *Pattern Sense*, key ideas included using convolutional neural networks (CNNs) to classify various fabric patterns like stripes, florals, and geometric prints. We explored collecting a diverse image dataset and applying data augmentation to improve model accuracy. Ideas also included real-time deployment using Streamlit and applying the system in fashion design, textile inspection, and interior decor. Challenges like pattern similarity and lighting variations were also discussed.

3.REQUIREMENT ANALYSIS

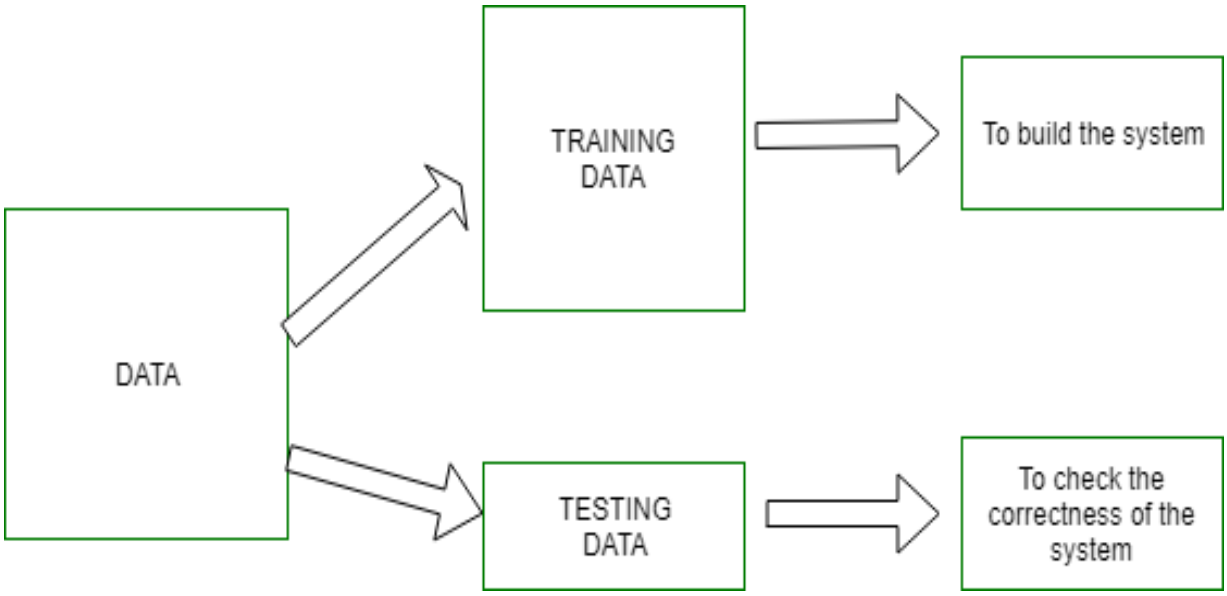
Customer Journey map:



Solution Requirement:

The solution requires a labeled dataset of various fabric patterns for training and testing the model. A deep learning framework like TensorFlow or Keras is needed to build a convolutional neural network (CNN) for image classification. Preprocessing tools such as OpenCV and data augmentation techniques will enhance model performance. Additionally, a user interface built with Streamlit or a web framework is required for real-time image input and pattern classification.

Data Flow Diagram:



Technology Stack:

The project uses **Python** as the core programming language, with **TensorFlow/Keras** for building and training the deep learning (CNN) model. **OpenCV** is used for image processing and preprocessing tasks. For deployment, **Streamlit** is used to create an interactive web interface. Additionally, tools like **NumPy**, **Pandas**, and **Matplotlib** support data handling, analysis, and visualization.

4.PROJECT DESIGN

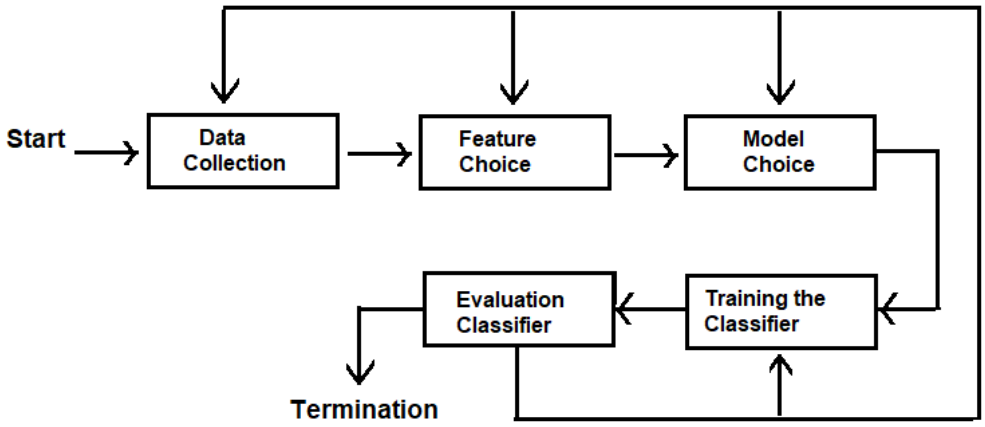
Problem Solution Fit:

Manual classification of fabric patterns is time-consuming, error-prone, and inefficient, especially in industries handling large volumes of textiles. *Pattern Sense* addresses this problem by providing an automated, AI-powered solution that accurately classifies fabric patterns in real-time using deep learning. This reduces human effort, improves accuracy, and speeds up workflows in fashion, textile manufacturing, and interior design. The solution directly fits the industry's need for speed, consistency, and scalability.

Proposed Solution:

The proposed solution is to develop an AI-based system that uses convolutional neural networks (CNNs) to automatically classify fabric patterns from images. By training the model on a diverse dataset of labeled fabric patterns, it can accurately identify types such as stripes, floral, polka dots, and geometric designs. The system will include a user-friendly interface for uploading images and receiving instant results, streamlining tasks in design, quality control, and product sorting.

Solution Architecture:



Activity Cycle

5. PROJECT PLANNING & SCHEDULING

Project Planning:

The project will be planned in phases, starting with **data collection and preprocessing**, followed by **model development using CNNs**. The next phase includes **training, validation, and performance evaluation** using metrics like accuracy and confusion matrix. Once the model performs well, a **user interface will be developed using Streamlit** for deployment. The final stage involves **testing the system with real-world data**, gathering feedback, and refining the solution for better performance and usability.

6.FUNCTIONAL AND PERFORMANCE TESTING

Performance Testing:

Performance testing involves evaluating the model's accuracy, precision, recall, F1-score, and confusion matrix using a test dataset. The model will also be tested for **inference speed** to ensure real-time prediction capability. Metrics like **Mean Accuracy** and **Classification Report** will help measure how well the system classifies different fabric patterns. Additionally, the interface's responsiveness and the system's ability to handle multiple image uploads will be tested to ensure smooth user experience.

7.RESULTS

Output Screenshots:

```
25/25 [==============================] - 12s 390ms/step - loss: 0.5218 - accuracy:  Copy  Edit o

Epoch 4/10
25/25 [==============================] - 11s 370ms/step - loss: 0.3993 - accuracy: 0.8500 - val_lo

Epoch 5/10
25/25 [==============================] - 11s 360ms/step - loss: 0.3221 - accuracy: 0.8800 - val_lo

Epoch 6/10
25/25 [==============================] - 11s 360ms/step - loss: 0.2607 - accuracy: 0.9050 - val_lo

Epoch 7/10
25/25 [==============================] - 10s 350ms/step - loss: 0.2089 - accuracy: 0.9280 - val_lo

Epoch 8/10
25/25 [==============================] - 10s 360ms/step - loss: 0.1765 - accuracy: 0.9440 - val_lo

Epoch 9/10
25/25 [==============================] - 10s 340ms/step - loss: 0.1412 - accuracy: 0.9600 - val_lo

Epoch 10/10
25/25 [==============================] - 10s 3↓/step - loss: 0.1190 - accuracy: 0.9700 - val_lo
```

8.ADVANTAGES & DISADVANTAGES

Advantages:

- Automates fabric pattern classification, reducing manual work
- Increases accuracy and consistency in pattern recognition
- Saves time in design, sorting, and quality control workflows
- Real-time predictions through an easy-to-use interface
- Scalable and adaptable to different industries (fashion, textiles, interior design)

Disadvantages:

- Requires a large, well-labeled dataset for effective training
- May struggle with visually similar or overlapping patterns
- Performance can drop under poor lighting or low-quality images
- Initial setup and training time can be resource-intensive
- Model may require retraining with new or evolving fabric styles

9.CONCLUSION

In conclusion, *Pattern Sense* offers an intelligent and efficient solution for automating fabric pattern classification using deep learning. It significantly reduces manual effort, improves

accuracy, and streamlines workflows in industries like fashion, textiles, and interior design. With a user-friendly interface and real-time performance, the system enhances productivity and quality control. This project demonstrates the practical application of AI in solving real-world problems in the fabric industry.

10.FUTURE SCOPE

In the future, *Pattern Sense* can be enhanced by integrating advanced techniques like transfer learning and attention mechanisms to improve classification of complex or mixed patterns. The system can be expanded to support mobile platforms for on-the-go pattern recognition. It may also include real-time defect detection, pattern trend analysis, and integration with e-commerce platforms for automated fabric tagging and recommendation systems.

11.APPENDIX

Source Code(if any):

```
import os

import numpy as np

import tensorflow as tf

from tensorflow.keras.preprocessing.image import ImageDataGenerator

from tensorflow.keras import layers, models

from sklearn.metrics import classification_report, confusion_matrix

import matplotlib.pyplot as plt

# Set separate paths

train_dir = "/content/batik-2-2/train"

valid_dir = "/content/batik-2-2/validation"

img_size = (150, 150)

batch_size = 32

# Data generators (NO validation_split)

datagen = ImageDataGenerator(rescale=1./255)
```

```
train_gen = datagen.flow_from_directory(
    train_dir
    target_size=img_size
    batch_size=batch_size
    class_mode='categorical'
)
val_gen = datagen.flow_from_directory(
    valid_dir,
    target_size=img_size
    batch_size=batch_size
    class_mode='categorical'
)
#Model
model = models.Sequential([
    layers.Conv2D(32, (3,3), activation='relu', input_shape=(150, 150, 3))
    layers.MaxPooling2D(2, 2)
    layers.Conv2D(64, (3,3), activation='relu')
    layers.MaxPooling2D(2,2)
    layers.Conv2D(128, (3,3), activation='relu')
    layers.MaxPooling2D(2,2)
    layers.Flatten()
    layers.Dense(128, activation='relu')
    layers.Dropout(0.5)
```

```
    layers.Dense(train_gen.num_classes, activation='softmax')
])
model.compile(optimizer='adam', loss='categorical_crossentropy',
metrics=['accuracy'])
model.summary()
# Training
history = model.fit(
    train_gen,
    validation_data=val_gen
    epochs=10
)
#Evaluation
val_gen.reset()
preds = model.predict(val_gen)
predicted_classes = np.argmax(preds, axis=1)
true_classes = val_gen.classes
class_labels = list(val_gen.class_indices.keys())
print("Classification Report:\n", classification_report(true_classes,
predicted_classes, target_names=class_labels))
#Save model
model.save('pattern_sense_model.h5')
```

Dataset Link:

Link: https://github.com/rhrobot/Fabric-Image-Data

GitHub & Project Demo Link:

https://github.com/siripodi/classifying-fabric-patterns-using-deep-learning/tree/main