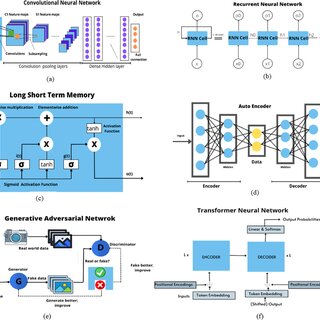
**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| Date | 18june 2025 |
| Team ID | LTVIP2025TMID43376 |
| Project Name | Pattern sense: classifying fabrics using deep learning |
| Maximum Marks | 4 Marks |

**Technical Architecture:**

A deep learning-based fabric pattern classification system uses a CNN (Convolutional Neural Network) to extract and learn spatial features from fabric images. The architecture includes image preprocessing, a trained CNN model for feature extraction and classification, and an output layer for pattern type prediction (e.g., striped, floral, plaid)

**Example: **

**Table-1 : Components & Technologies:**

|  |  |  |
| --- | --- | --- |
| S. No | Component / Technology | Description / Role |
| **1** | Image Dataset | Collection of fabric images with labeled patterns (e.g., striped, floral, plain, etc.) |
| **2** | Image Preprocessing Tools | Resizing, normalization, augmentation (rotation, flipping, zooming) |
| **3** | Deep Learning Framework | TensorFlow or PyTorch for model creation and training |
| **4** | Convolutional Neural Networks (CNNs | Core model architecture for pattern recognition in images |
| **5** | Transfer Learning Models | Pre-trained models like ResNet, VGG, or EfficientNet to improve accuracy and speed |
| **6** | GPU Hardware (e.g., NVIDIA GPU) | Accelerates training and inference of deep learning models |
| **7** | Model Evaluation Metrics | Accuracy, Precision, Recall, F1-score, Confusion Matrix |
| **8** | Dataset Split Strategy | Training, Validation, and Test sets (e.g., 70/15/15 split) |
| **9** | Annotation Tools | Tools like LabelImg for manually labeling fabric patterns if dataset isn’t labeled |
| **10** | Deployment Platform | Web or mobile app interface for real-time fabric classification |
| **11** | Backend Technologies | Python (Flask/Django) or Node.js for API integration |
| **12** | Cloud/Storage Services | Google Cloud, AWS S3, or Firebase for hosting models and storing images |
| **13** | Database | SQL or NoSQL (like MongoDB) to store metadata and results |
| **14** | User Interface (UI) | React, Flutter, or HTML/CSS for user-friendly input/output interaction |

**Table-2: Application Characteristics:**

|  |  |  |
| --- | --- | --- |
| **Characteristic** | **Description** | **Technology** |
| Input Type | Accepts fabric images as input | Camera, Image Upload, Mobile/Web UI |
| Pattern Recognition | Identifies and classifies fabric patterns (e.g., floral, geometric, plain) | CNN, Transfer Learning (e.g., ResNet) |
| Real-time Processing | Offers quick prediction results after image input | GPU Acceleration, Flask/Streamlit API |
| Scalability | Handles large datasets and supports future model updates | Cloud Platforms (AWS, GCP, Azure) |
| Accuracy & Precision | High accuracy in classifying complex patterns | Deep CNN Models, Data Augmentation |
| User Accessibility | Easy-to-use interface for both technical and non-technical users | Responsive UI (React, Flutter) |
| Model Reusability | Modular design allows reuse in related textile applications | Python Modules, Pre-trained Model APIs |
| Explainability | Provides confidence score or class activation maps for transparency | Grad-CAM, Softmax Confidence Scores |
| Data Security & Privacy | Ensures user-uploaded images are securely stored or processed | HTTPS, Cloud Storage Encryption |
| Offline Capability\* | Limited classification can be done without internet (if embedded) | TensorFlow Lite, ONNX Runtime |
| Feedback & Improvement | Accepts user feedback to improve future predictions | Backend Logging, Active Learning Pipelines |

**References:**

[**https://c4model.com/**](https://c4model.com/)

[**https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/**](https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/)

[**https://www.ibm.com/cloud/architecture**](https://www.ibm.com/cloud/architecture)

[**https://aws.amazon.com/architecture**](https://aws.amazon.com/architecture)

[**https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d**](https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d)