**Title: Face Recognition System – Final Report**

### **1. Project Summary**

This project aimed to build a robust face recognition system divided into three phases: face detection, feature extraction, and classification. The final goal was to deploy the solution using a Streamlit web application.

* **Phase 1**: Detected and extracted faces from input images using Haar Cascade Classifier.
* **Phase 2**: Generated face embeddings using the VGG16 pre-trained model.
* **Phase 3**: Trained classifiers using the extracted embeddings and deployed the best-performing model via Streamlit.

### **2. What I Learned**

* **Understanding Face Recognition Systems**: Gained theoretical knowledge about the full pipeline, including detection, embedding, and classification.
* **Image Preprocessing**: Face detection using Haar Cascades and image resizing.
* **Transfer Learning**: Leveraging VGG16 for feature extraction without fine-tuning.
* **Feature Embedding**: Generating 4096-dimensional vectors representing facial features.
* **Classifier Evaluation**: Comparing Random Forest, SVM and KNN.
* **App Deployment**: Developing and deploying an interactive face recognition app with Streamlit.
* **Model Serialization**: Saving and loading models and embeddings using joblib and NumPy.
* **Version Control & Dependency Management**: Debugging dependency conflicts using requirements.txt.

### **3. Challenges & Solutions**

| **Challenge** | **Solution** |
| --- | --- |
| Understanding different components of face recognition systems | Invested time in learning the theory behind detection, embeddings, and classification |
| FaceNet model download issues or corruption | Switched to VGG16 which was more stable and compatible with the working environment |
| VGG16 struggles with side-view facial images | Focused on frontal face images only for better results |
| Streamlit app failing to install dependencies | Manually corrected and verified requirements.txt file |

### **4. Conclusion**

This project allowed me to gain hands-on experience with the full machine learning pipeline, from data preprocessing and model building to real-world deployment. I faced technical hurdles such as import errors, deployment issues, and classifier limitations, but resolved them with structured debugging and experimentation.

Moving forward, I aim to:

* Integrate real-time webcam input.
* Use more advanced models.