Project Title: Hybrid Deep Learning Pipelines for Face Classification & Verificatio

---

🔹 Problem Statement

Develop two systems:

- **Task A:** Classify face images among known identities.
- **Task B:** Verify whether two face images belong to the same person, even under distortion.

🔹 Task A – Face Classification

Approach: Multiclass classification using deep feature fusion + traditional ML.

Pipeline:

- **Preprocessing:** Resize (224×224), normalize, label from folders.
- **Feature Extraction:**
  - ○ **ResNet-50** (2048D), **EfficientNet-B3** (~1536D), **DeepFace (VGG-Face)** (2622D).
- **Feature Fusion:** Concatenated features → normalized via StandardScaler.
- **Classification Models:**
  - ○ Logistic Regression (scikit-learn)
  - ○ Support Vector Machine (SVM with linear kernel)
- **Evaluation:** Accuracy, Precision, Recall, F1-score (weighted).

🔹 Task B – Face Verification

Approach: Binary verification using a Siamese neural network + hard negative mining.

Pipeline:

- **Face Detection:** DeepFace.extract_faces() (OpenCV backend), normalized & cached.
- **Embeddings:** DeepFace.represent() using **ArcFace** (512D).
- **Pair Generation:**
  - ○ Positive: same identity;
  - ○ Hard Negative: cosine similarity with other identities.
- **Model:**
  - ○ Siamese Net: [Embedding A + B] → FC(1024→256→1) → Sigmoid.
- **Training:** Binary Cross Entropy loss, Adam optimizer.
- **Prediction:** Match if score > threshold; else predict **“unknown”**.

- **Evaluation:** Accuracy, Precision, Recall, F1-score.

◆ Key Innovations

- **Hybrid Feature Fusion:** Combines CNNs + DeepFace for rich representations.
- **Hard Negative Mining:** Ensures challenging training pairs.
- **Open-Set Support:** Threshold-based unknown face detection.
- **Efficient Caching:** For embeddings, faces, and labels.

◆ Tools & Technologies

- Python, PyTorch, DeepFace, scikit-learn, OpenCV
- Pretrained models: ResNet50, EfficientNet-B3, ArcFace, VGG-Face