Hackathon Model Report: Face Verification using Ensemble Siamese Networks

1. Project Overview

This project implements a face verification system using an ensemble of Siamese networks. The goal is to determine whether two given face images belong to the same identity. This task is particularly relevant for applications involving biometric verification under challenging environmental distortions.

---

2. Dataset Overview

- **Data Structure:**
  - **Training Pairs:** 1246 total (623 positive, 623 negative)
  - **Validation Pairs:** 280 total (140 positive, 140 negative)
  - **Class Balance:** 50% same-person pairs and 50% different-person pairs
- **Image Properties:**
  - **Shape:** Each image is resized to (224, 224, 3)
  - **Pixel Value Range:** [0.0, 1.0]

---

3. Model Architecture and Strategy

- **Base Architecture:** A diverse ensemble of 3 Siamese networks
- **Backbones Used:**
  - ensemble_siamese_0.h5: Likely ResNet-based

    - ensemble_siamese_1.h5: EfficientNet-based
    - ensemble_siamese_2.h5: Attention-based or augmented variant
- **Loss Function:** Combination of contrastive loss and center loss
- **Augmentation:** Advanced augmentations such as Cutout, Mixup were optionally applied (see script for details)

---

4. Training Details

- **Device:** Trained on Apple MPS (Metal Performance Shaders)
- **Epochs:** 25
- **Batch Size:** 16
- **Learning Rate Strategy:** Custom scheduler or fixed value per model
- **Reproducibility:** Random seed was manually set to ensure consistent runs

---

5. Results

Ensemble Performance

- **Final Ensemble Top-1 Accuracy: 0.9321**

Best Single Model Performance

- **Model:** ensemble_siamese_1.h5
- **Top-1 Accuracy: 0.9429**

Comparison

- **Improvement over Ensemble:** -0.0107 (i.e., ensemble underperformed slightly compared to the best individual model)

Note: Top-1 accuracy here refers to whether the predicted pair label (same/different) is correct. Since this is a binary classification task, Top-1 accuracy directly reflects verification performance.

---

6. Deployment and Evaluation

- **Evaluation Compatibility:**
  - All .h5 model files are saved and loadable via TensorFlow/Keras
  - A separate script can be provided for evaluation on test pairs
- **Cross-Device Note:**
  - Models were trained using MPS (Apple Silicon). Running the same code on CUDA (NVIDIA GPUs) **may yield slightly different results** due to backend-specific numeric behavior.
  - This should be mentioned in the README to ensure fair evaluation.

---

7. Recommendations for Evaluator

- Use the same preprocessing pipeline (image resizing and normalization)
- Load models using tf.keras.models.load_model(...)
- Evaluate using consistent positive/negative pair structure
- If exact reproduction is required, prefer running on Mac M1/M2 with MPS backend