**Face Recognition System for Secure Access Control**

**1. Abstract**

Secure access control is critically important for protecting sensitive environments. Traditional authentication methods like passwords are vulnerable to breaches, while biometric systems offer more robust security. We approach access control as a face recognition problem, implementing and benchmarking multiple deep learning architectures including inception , and resnet on a celebrity face dataset. While individual models perform well, each has limitations: CNNs may under-model global facial structure, while transformers might under-weight local facial features. We address this with a. Our system achieves high accuracy with minimal overfitting and provides clear decision explanations. This document details our dataset, methodology, results, and implementation in a reproducible format.

**2. Introduction**

Facial recognition for access control requires analyzing features at multiple scales: local details like eye shape, nose structure, and skin texture (micro-features), and global characteristics like face shape, symmetry, and spatial relationships between features. An effective security system must capture both types of features. Our goal is to build a reliable face recognition system using only the provided dataset, with clear performance metrics and explanations for its access decisions.

**3. Problem Statement and Proposed Solution**

Problem: Accurately verifying identities from limited, heterogeneous facial images without external datasets, while maintaining security against unauthorized access.

\*\*Solution\*\*: Transfer learning with pretrained backbones, careful preprocessing and data splits, and a final vgg 16 that combines local and global facial representations before classification and access decision.

**4. Dataset Description**

The dataset contains approximately 100 celebrity classes , with 20,000-25,000 images total. Images vary in pose, lighting, scale, and background. The dataset presents challenges due to non-uniform framing (not always face-forward or centered) and varying acquisition conditions. We honor the constraint not to use external datasets; all results reflect performance on this corpus only.

**5. Data Quality and Ethics**

Images may include variations in lighting, angles, occlusions, and image quality. We assume the dataset is properly de-identified. We avoid over-processing that might remove important facial features. Ethical considerations: The system is an access control tool and does not replace human security oversight.

**6. Preprocessing Pipeline**

All images are resized to (224×224) and normalized with mageNet mean/std to match pretrained network expectations. We apply data augmentation including random rotations, flips, and brightness adjustments to improve generalization.

**7. Train/Validation Splits**

We partition the data (80/20) at the image level, maintaining per-class proportions. Training data is shuffled each epoch; validation set remain fixed. This approach stabilizes variance while providing sufficient validation signal for early stopping.

**8. Reproducibility Settings**

We fix random seeds for Python/NumPy/PyTorch and enable deterministic cuDNN when using GPU. Exact transform parameters and split ratios are documented. Best checkpoints are saved based on lowest validation loss.

**9. Transfer Learning Strategy**

We initialize all backbones with pretrained ImageNet weights. We employ fine-tuning (updating all weights with a small learning rate) which performs better due to the domain gap .

**10. Models Under Study (Overview)**

- ResNet-18 - Compact residual CNN; reliable baseline with skip connections

- EfficientNet-B0 - Compound scaling balances depth/width/resolution efficiently

**11. ResNet-18: Behavior and Limitations**

ResNet-18 trains stably and quickly. Residual connections help with gradient flow, but the relatively shallow depth may not capture fine inter-class nuances needed for highly accurate face recognition. Validation accuracy typically saturates in the mid-80s, indicating either under-capacity or insufficient global context modeling.

**12. EfficientNet-B0: Behavior and Limitations**

EfficientNet-B0 is parameter-efficient and adept at capturing local facial features. We observe training accuracy around 95-96% and validation accuracy around 93-94%. Limitations include implicit modeling of global structure and sensitivity to lighting variations.

**15. Proposed Architecture: inception Feature Fusion**

We feed the same facial image into EfficientNet-B0 and ViT-Base/16 with their classification heads removed. The resulting feature vectors are concatenated and passed to a lightweight head: LayerNorm → Linear(→512) → ReLU → Dropout(0.3) → Linear(→num\_classes). LayerNorm stabilizes scales after fusion; the shallow MLP avoids over-parameterization.

**16. Training Configuration**

Optimizer: Adam with LR=1e-4 and weight decay=1e-4. Batch size: 32. Up to 30 epochs with early stopping (patience=5) on validation loss. Mixed precision is used for speed on GPU. The best checkpoint by validation loss is evaluated on the test set.

**18. Optimization Dynamics**

Loss declines smoothly for both training and validation, with a small generalization gap at the selected epoch. Early stopping prevents late-epoch overfitting. Weight decay and dropout curb memorization while preserving capacity for challenging cases.

20. Baseline Results (Illustrative)

- ResNet-18: validation ≈85%, test ≈84%

- inception: validation ≈94%, test ≈93%

**22. Error Analysis**

Misclassifications typically involve subjects with similar facial features or challenging acquisition conditions (extreme angles, poor lighting).

**24. Ablation Studies**

- Feature extraction vs. fine-tuning: Full fine-tuning outperforms head-only training

- Head depth: Deeper heads yield diminishing returns and higher overfit risk

- Regularization: Removing dropout or weight decay increases the train-val gap

**25. Robustness Checks**

We test sensitivity to input variations (lighting changes, occlusions) and observe stable predictions, suggesting the model isn't over-relying on spurious cues. However, evaluation on completely unseen identities is necessary to certify robustness.

**26. Limitations**

Dataset size and class imbalance may constrain statistical power. No external validation with completely unseen identities was performed. The model is computationally heavy which may affect real-time performance.

**27. Future Directions**

Evaluate on standard face recognition benchmarks; explore lighter backbones for edge deployment; integrate liveness detection to prevent spoofing; add temporal analysis using video sequences.

**28. Practical Deployment Notes**

Batching and mixed precision reduce inference latency on GPU. For resource-constrained environments, an inception can provide strong performance with lower compute requirements.

**29. Reproducibility Checklist**

- Fixed seeds for Python/NumPy/PyTorch; deterministic cuDNN when feasible

- Exact transforms and splits documented

- Best checkpoint saved by validation loss.

**30. Environment & Tools**

Software: Python, PyTorch, facenet\_pytorch , faiss

Hardware: NVIDIA GPU recommended; CPU support available

Development: Google Colab-compatible implementation

**32. Implementation Summary**

1. Load and preprocess facial image data (224×224 normalization)

2. Split data 70/30 (train/validation)

3. Instantiate resnet and inception without classification heads

4. Concatenate feature vectors from both models

5. Train lightweight MLP head with Adam+weight decay, early stopping

6. Report best-epoch test accuracy.

7. Implement security decision logic with adjustable confidence threshold

## 33. Security Integration

The trained model is integrated into an access control system that:

- Grants access for recognized faces above confidence threshold

- Denies access for unrecognized faces or low-confidence matches

- Logs all access attempts with timestamps and confidence scores

- Alerts security personnel for repeated failed attempts

**35. Team Members**

* Asma abu odeh
* Maha al qaddomi
* Mohammad osama