Technical Report: Age-Invariant Face Recognition System

Table of Contents

---

1. Dataset Choice and Rationale

Datasets Used

This project utilizes four major age-invariant face recognition datasets for training and threshold tuning:

1. **MORPH (Craniofacial Longitudinal Morphological Face Database)**
   - Large-scale longitudinal aging dataset
   - Contains multiple age samples per subject
   - Age range: 16-77 years
   - Provides realistic aging progression patterns
2. **CACD (Cross-Age Celebrity Dataset)**
   - Celebrity faces across different ages
   - Real-world unconstrained images
   - Age range: 16-62 years
   - Captures natural aging variations in diverse conditions
3. **AgeDB (Age Database)**
   - Specifically designed for age-invariant face verification
   - Large age gaps between image pairs
   - Contains both in-the-wild and controlled images
   - Ideal for evaluating cross-age matching performance
4. **FG-NET (Face and Gesture Recognition Network Aging Database)**
   - Longitudinal aging database
   - Multiple images per subject across ages
   - Age range: 0-69 years
   - High-quality age progression sequences

Rationale for Dataset Selection

Why These Datasets?

- **Complementary Coverage**: Each dataset provides unique characteristics:
  - MORPH: Controlled, high-quality aging samples
  - CACD: Celebrity faces with diverse poses and expressions
  - AgeDB: Extreme age gaps for robust evaluation
  - FG-NET: Complete aging trajectories
- **Age-Invariance Focus**: All four datasets are specifically designed for studying age variations in face recognition, making them ideal for training age-adaptive thresholds

   - Patch size: 16×16
   - Number of patches: (224/16)² = 196 patches
   - Each patch is linearly embedded to D-dimensional space

2. **Transformer Encoder**

   ```
   Input: Sequence of embedded patches + [CLS] token
   ↓
   Multi-Head Self-Attention (MSA)
   ↓
   Layer Normalization
   ↓
   MLP (Feed-Forward Network)
   ↓
   Layer Normalization
   ↓
   Repeat for N layers
   ```

3. **Classification Head**
   - Extracts [CLS] token representation
   - Fully connected layer for age regression
   - Fully connected layer for gender classification

Model Specifications

```
Model: Vision Transformer (ViT)
Parameters: ~86M
Input Resolution: 224×224
Patch Size: 16×16
Number of Transformer Layers: 12
Hidden Dimension: 768
MLP Dimension: 3072
Attention Heads: 12
Output: Age (regression) + Gender (classification)
```

Why Vision Transformer for Age Prediction?

1. Global Context Understanding

- Unlike CNNs that process local features, ViT captures global facial relationships
- Aging affects multiple facial regions simultaneously (wrinkles, face shape, skin texture)
- Self-attention mechanism allows the model to learn age-relevant correlations across the entire face

2. Superior Feature Representation

- Transformer architecture excels at learning hierarchical representations
- Better at capturing subtle aging patterns compared to traditional CNNs
- Pre-trained on large-scale datasets, providing robust initial features

3. Robustness to Variations

- Handles pose variations, lighting conditions, and partial occlusions effectively
- Age estimation remains consistent across different image qualities

4. State-of-the-Art Performance

- ViT-based models achieve competitive results on age estimation benchmarks
- Lower Mean Absolute Error (MAE) compared to CNN-based approaches

- Facial landmark detection for alignment
- High recall rate even on small faces

2. Face Recognition (ArcFace Backbone)

```
Detected Face (112×112)
↓
ResNet-100 Backbone
  ├─ Conv Layers (Feature Extraction)
  ├─ Residual Blocks (Deep Feature Learning)
  └─ Global Average Pooling
↓
Embedding Layer (512-dim)
↓
L2 Normalization (Unit Hypersphere)
↓
Normalized Embedding Vector
```

Why InsightFace (buffalo_l)?

1. Unified Pipeline

- Single framework handles both detection and recognition
- Seamless integration between components
- Optimized end-to-end performance

2. State-of-the-Art Accuracy

- Buffalo_l is one of InsightFace's most accurate models
- Trained on massive datasets (millions of identities)
- Achieves >99% accuracy on LFW benchmark

3. Efficient Inference

- ONNX-optimized models for fast inference
- CPU-compatible (important for Streamlit deployment)
- Low memory footprint

Face Verification Process

Step 1: Face Detection

```
# Detect single face using InsightFace
faces = app.get(image_bgr)

# Validation
if len(faces) == 0:
    return "No face detected"
if len(faces) > 1:
    return "Multiple faces detected"

face = faces[0]
bbox = face.bbox  # Bounding box [x1, y1, x2, y2]
```

- Superior accuracy across all age gaps
- Best balance of detection and recognition performance
- Unified pipeline reduces complexity
- Production-ready with minimal setup required

---

4. Loss Function Selection and Rationale

Overview of Experimental Loss Functions

During our experiments, we evaluated two prominent loss functions for age-invariant face recognition:

1. **ArcFace Loss** (Additive Angular Margin Loss) - Used for R100 fine-tuning
2. **Age-Aware Triplet Loss** - Used for FaceNet fine-tuning

4.1 ArcFace Loss (R100 Experiments)

Mathematical Formulation

Standard Softmax Loss (Baseline):

```
L_softmax = -log( e^(W_y^T * f) / Σ_j e^(W_j^T * f) )
```

ArcFace Loss (Our Implementation):

```
L_ArcFace = -log( e^(s * cos(θ_y + m)) / (e^(s * cos(θ_y + m)) + Σ_{j≠y} e^(s * cos(θ_j))) )
```

Where:

- `θ_y = arccos(W_y^T * f)`: angle between embedding and true class weight
- `m = 0.5`: additive angular margin (in radians, ~28.6°)
- `s = 30`: feature scale (controls gradient magnitude)
- `f`: L2-normalized embedding
- `W_y`: L2-normalized weight vector for true class

ArcFace Margin Visualization

Figure 5: Angular margin in ArcFace forces greater separation between identities

- Robust to appearance changes over time

4. Optimal Hyperparameters

- `s = 30.0`: Standard scale factor for face recognition
- `m = 0.5`: Empirically validated angular margin (28.6°)

4.2 Age-Aware Triplet Loss (FaceNet Experiments)

Mathematical Formulation

Base Triplet Loss:

```
L_triplet = max(0, ||f_a - f_p||² - ||f_a - f_n||² + margin)
```

Age-Aware Triplet Loss (Our Enhancement):

```
L_age_aware = L_triplet + α * (age_gap / 60) * ||f_a - f_p||²
```

Where:

- `f_a, f_p, f_n`: Anchor, positive, negative embeddings (L2-normalized)
- `margin`: Distance margin between positive and negative pairs
- `α = 0.1`: Age penalty weight
- `age_gap`: Absolute age difference between anchor and positive
- `60`: Normalization constant (max expected age gap)

Triplet Mining Strategy

Why Age-Aware Triplet Loss for FaceNet?

1. Explicit Age-Gap Modeling

- Age penalty term directly addresses age variation
- Encourages model to maintain small distances despite large age gaps
- More intuitive for age-invariant tasks than standard triplet loss

2. Flexible Training

- No need for fixed class labels (unlike ArcFace)
- Works with online triplet mining
- Adapts to dataset characteristics during training

3. Hard Example Mining

- Dynamic selection of challenging positive pairs (large age gaps)
- Progressive hard negative mining
- Accelerates convergence on difficult cases

4.3 Loss Function Comparison

Aspect	ArcFace Loss	Age-Aware Triplet Loss
Type	Classification-based	Metric learning

- Small fine-tuning datasets rarely improve upon large-scale pretraining

4.5 Final System Choice: InsightFace (buffalo_l)

Given the experimental results, we chose **InsightFace (buffalo_l)** for the deployed system because:

1. Superior Baseline Performance

- ROC AUC: 0.98 on FG-NET (highest among all models)
- No fine-tuning required
- Immediately production-ready

2. Pretrained on Massive Scale

- Trained on millions of identities
- Robust to age, pose, expression variations
- Generalizes well to unseen data

3. Unified Detection + Recognition

- RetinaFace detection + ArcFace recognition in single framework
- Optimized end-to-end pipeline
- Fewer failure points

4. Practical Advantages

- No training infrastructure needed
- Lower deployment complexity
- Consistent performance across datasets

5. Performance Analysis and Evaluation Metrics

5.1 Evaluation Protocol

All models were evaluated on the **FG-NET dataset** to ensure fair comparison:

FG-NET Test Set:

- 82 unique identities
- 1,002 total images
- Age range: 0-69 years
- Genuine pairs: 5,808 (same identity, different ages)
- Impostor pairs: 495,693 (different identities)

Evaluation Metrics:

1. **ROC AUC** (Receiver Operating Characteristic - Area Under Curve)
2. **EER** (Equal Error Rate)
3. **TPR @ FPR** (True Positive Rate at fixed False Positive Rates)
    - TPR @ FPR = 0.01% (high security)
    - TPR @ FPR = 0.1% (balanced)
    - TPR @ FPR = 1% (high recall)
4. **Verification Accuracy** at optimal threshold
5. **Age-Gap Specific Performance** (0-5, 5-10, 10-20, 20-30, 30+ years)

FaceNet Base Model (VGGFace2)

```
Performance by Age Gap on FG-NET:
Optimal Threshold: 0.3350

Age Gap 0-5 years:
  Pairs: 3,041 (Genuine: 1,655, Impostor: 1,386)
  ROC AUC: 0.9576
  Accuracy: 89.02%

Age Gap 5-10 years:
  Pairs: 2,822 (Genuine: 1,590, Impostor: 1,232)
  ROC AUC: 0.9333
  Accuracy: 85.44%

Age Gap 10-15 years:
  Pairs: 2,118 (Genuine: 1,082, Impostor: 1,036)
  ROC AUC: 0.8976
  Accuracy: 82.72%

Age Gap 15-20 years:
  Pairs: 1,319 (Genuine: 618, Impostor: 701)
  ROC AUC: 0.8642
  Accuracy: 80.14%

Age Gap 20+ years:
  Pairs: 2,316 (Genuine: 863, Impostor: 1,453)
  ROC AUC: 0.8775
  Accuracy: 82.60%
```

Key Observation: FaceNet shows gradual performance degradation as age gap increases, typical of age-invariant systems.

Optimized Thresholds (Target FAR = 5%)

```
age_adaptive_thresholds = {
    '0-5 years':   0.3083,  # Small age gap → high threshold
    '5-10 years':  0.2565,  # Medium age gap → moderate threshold
    '10-20 years': 0.2130,  # Large age gap → lower threshold
    '20-30 years': 0.1394,  # Very large age gap → low threshold
    '30+ years':   0.1381   # Extreme age gap → lowest threshold
}
```

Rationale Behind Decreasing Thresholds

As age gap increases:

1. **Facial appearance changes more dramatically**
   - Skin texture (wrinkles, age spots)
   - Facial structure (bone density, muscle tone)
   - Overall proportions shift
2. **Embedding similarity naturally decreases**
   - Same person at ages 20 and 70 has lower cosine similarity
   - Than same person at ages 20 and 22
3. **Lower threshold compensates for appearance drift**
   - Maintains consistent verification performance
   - Prevents false rejections for legitimate same-person pairs

5.5 Training Repository Reference

All fine-tuning experiments were conducted using the **AQUAFace** framework:

Repository: https://github.com/sadiqebrahim/AQUAFace

Key Features:

- ArcFace loss implementation for R100
- Age-aware triplet loss for FaceNet
- Validation on FG-NET, AgeDB, MORPH, CACD
- ROC curve generation and threshold optimization
- Comprehensive logging and visualization

Training Notebooks:

- `1-morph-cacd-agedb-fgnet-dataset-preprocessing.ipynb`: Dataset preparation
- `2-create-pairs-dataset-processing.ipynb`: Pair generation for verification
- `3-AQUAFace-Training.ipynb`: Model training and evaluation

5.6 Key Findings

✅ What Worked

1. **Pretrained Models Outperformed Fine-tuned Variants**
   - InsightFace (buffalo_l): ROC AUC 0.98
   - R100 Base: ROC AUC 0.9444
   - Large-scale pretraining provides robust age-invariant features
2. **Age-Adaptive Thresholds Crucial**
   - Fixed thresholds fail at large age gaps

2. **Age-Adaptive Threshold Optimization**
    - Developed age-gap-specific thresholds (0-5 to 30+ years)
    - Maintained consistent 5% FAR across all age bins
    - Simple yet effective approach for age-invariant verification
3. **Production-Ready Deployment**
    - Selected InsightFace (buffalo_l) for superior performance (AUC 0.98)
    - Integrated Vision Transformer for age prediction
    - Real-time CPU inference with unified detection + recognition

Reproducibility

All experiments, code, and notebooks are available:

- **Training Framework**: AQUAFace Repository
- **Deployment Code**: Streamlit application in `streamlit_app/`
- **Preprocessing Notebooks**: `assets/*.ipynb`