Discriminative Project

CNN / Deep Learning Based Automated Class Attendance System

Milestone 1

Group VI
Gayatri Nair
Harish Padmanabhan
Harrish Ebi Francis Peter
Joshua
Priyanka Raj Rajendran

nair.gaya@northeastern.edu
padmanabhan.h@northeastern.edu
peterjoshua.h@northeastern.edu
rajendran.priy@northeastern.edu

Git Repo Link:
https://github.com/harish2412/CelebA_Face_Identification

Submission Date: 29 September 2025

CNN/DEEP LEARNING BASED AUTOMATED CLASS ATTENDANCE SYSTEM

OVERVIEW:

This project develops a deep learning–based system for automated celebrity identification using Convolutional Neural Networks (CNNs). The primary objective is to address the challenge of accurate facial recognition in both single-person and multi-person contexts, with potential applications such as automated attendance tracking. To achieve this, the project leverages the CelebA dataset to construct robust classification and object detection models that are optimized for real-time identification, ensuring both high accuracy and computational efficiency.

The system adopts a two-phase methodology. The first phase focuses on single-celebrity classification through the implementation and comparison of multiple CNN architectures. The second phase extends to multi-celebrity detection and localization by incorporating advanced object detection frameworks. This staged approach not only strengthens the theoretical understanding of discriminative deep learning techniques but also demonstrates their practical applicability in real-world scenarios where scalability and reliability are essential.

PROJECT SCOPE:

- Single-celebrity image classification using CelebA dataset
- Implementation and comparison of multiple CNN architectures (**ResNet-50, EfficientNet-B3, EfficientNet-B4, ConvNeXt-Base**)
- Transfer learning implementation with pre-trained ImageNet models
- Model performance evaluation using standard computer vision metrics
- Development of model interface for demonstration purposes
- Comprehensive documentation and analysis of results

DATASET:

- **Source:** CelebA (aligned images + identity file).
- **Files:**
  - img_align_celeba — images (.jpg)
  - identity_CelebA.txt — (image_id, celebrity_id)
  - list_eval_partition.csv
- **Curation & Splits:**
  - Keep identities with ≥12 images.
  - Splits: 10% validation, 10% test
  - Per-class caps: 15 train, 4 val, 5 test (caps are maxima).
  - Scale: K = 5000 identities (top-K subset), then apply the caps above.
  - Label remapping ensures only train identities are in val/test.

METHODS:

DATA PIPELINE:

- **Train:** Resize→CenterCrop(256) → RandomResizedCrop(224) → Flip → Normalize
- **Transforms:** Resize(256) → CenterCrop(224) → Normalize
- **Loader:** PyTorch DataLoader, batch size 64, AMP enabled

ARCHITECTURES:

- Backbones(timm): EfficientNet-B3/B4, ResNet-50, ConvNeXt-Base
- Heads:
    - ArcFace (margin 0.15→0.50, s=64)
    - Softmax (baseline)

OPTIMIZATION:

- Loss: Cross-entropy with ArcFace logits
- Optimizer: AdamW (differential LRs)
- Scheduler: CosineAnnealingLR (15 epochs)
- Regularization: Dropout in head, gradient clipping = 1.0
- Metrics: Top-1, Top-5 accuracy on validation and test.

EXPERIMENT SETUP:

- Epochs: 15
- Batch size: 64 (GPU)/16 (CPU fallback)
- Hardware: Colab GPU (A100)
- Seeds fixed at 42

MODELS TRAINED:

- EfficientNet-B3
- EfficientNet-B4
- ResNet-50
- ConvNeXt-Base

RESULTS:

Model	Val Top-1 (%)	Test Top-1 (%)	Test Top-5 (%)
convnext_base	92.89	92.36	96.05
tf_efficientnet_b4_ns	86.23	86.03	92.53
tf_efficientnet_b3_ns	85.11	84.80	92.09
resnet50	68.04	67.77	80.40

To illustrate inference, we report the Top-1 prediction from the best model (ConvNeXt-Base). In both examples, the Top-1 prediction correctly identifies the target celebrity with high confidence.

Prediction 1

1. celeb_id=8656 (label 4347) — 99.64%
2. celeb_id=4891 (label 2505) — 0.13%
3. celeb_id=8659 (label 4350) — 0.09%
4. celeb_id=1448 (label 810) — 0.03%
5. celeb_id=8744 (label 4405) — 0.02%

Prediction 2

1. celeb_id=8656 (label 4347) — 100.00%
2. celeb_id=8191 (label 4102) — 0.00%
3. celeb_id=7380 (label 3723) — 0.00%
4. celeb_id=3483 (label 1845) — 0.00%
5. celeb_id=7894 (label 3972) — 0.00%

LEARNING CURVES (PER MODEL)

For each model we report training and validation loss, along with Val Top-1/Top-5 accuracy across 15 epochs.

- EfficientNet-B3 and B4 showed smoother convergence with higher Top-5 scores.
- ResNet-50 converged faster but plateaued earlier.
- ConvNeXt-Base jumps early and finishes highest; it converges the fastest and stays ahead throughout.

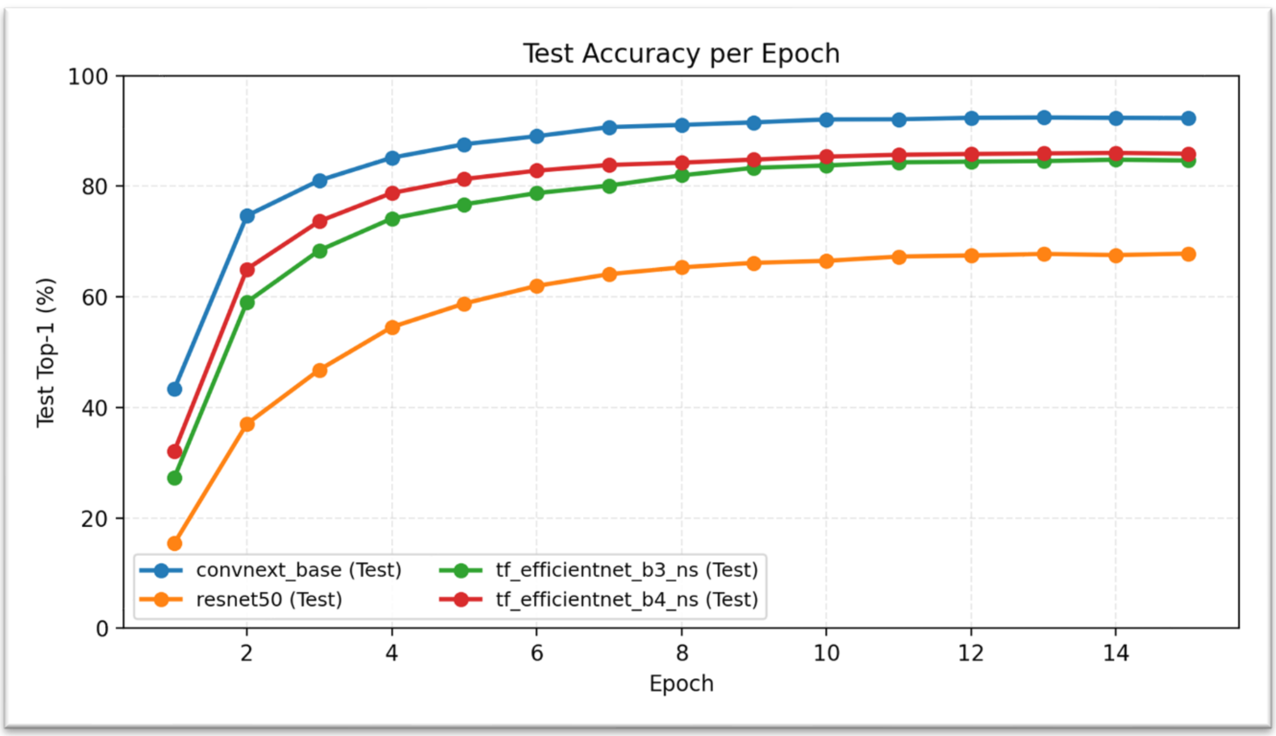

Figure 1. **Test Top-1 Accuracy vs Epoch**

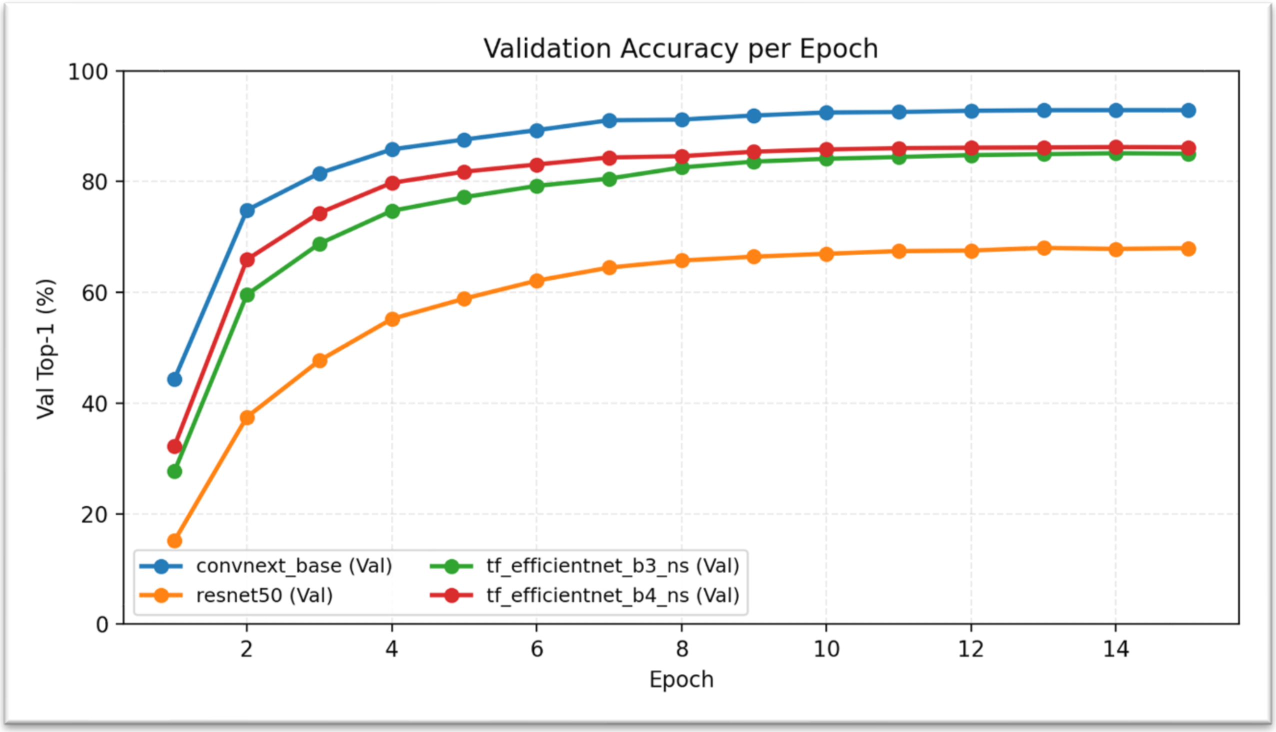

Figure 2. **Validation Top-1 vs Epoch**

Both demonstrate that ConvNeXt-Base and EfficientNet variants converge faster and reach higher accuracy compared to ResNet-50.

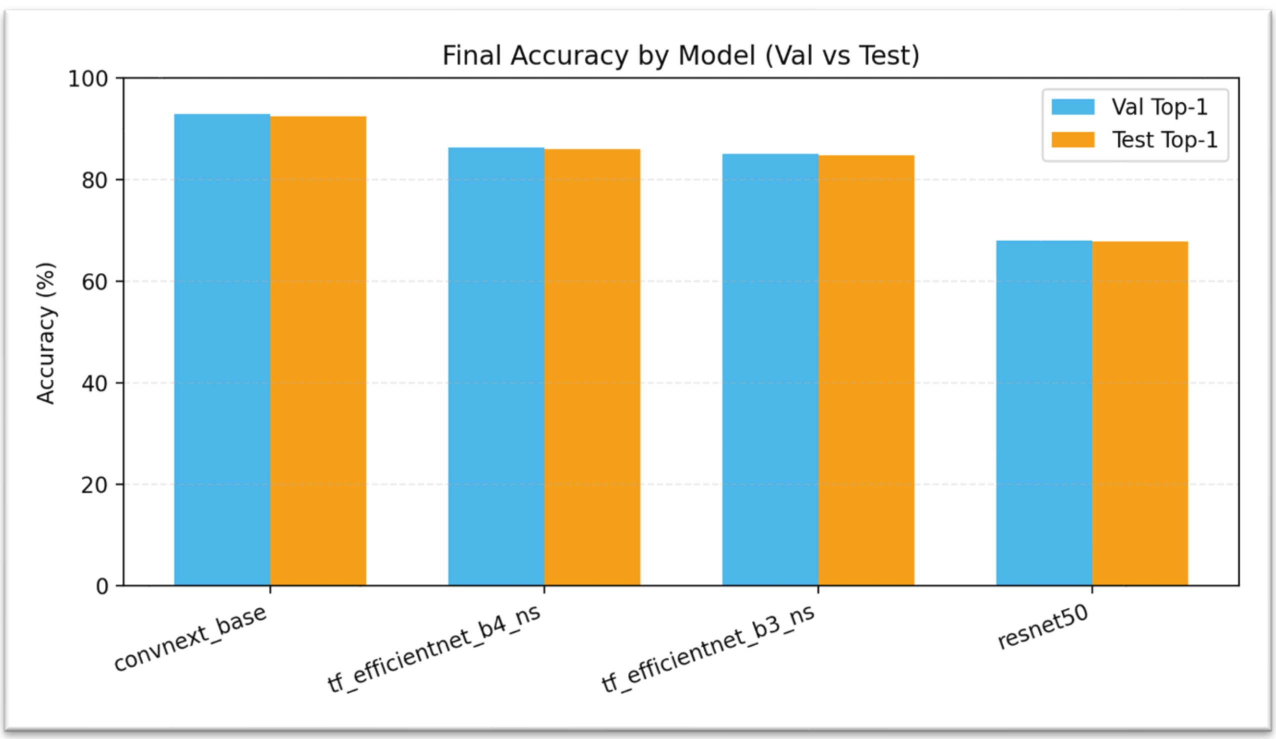

Figure 3. **Final Val/Test Top-1 across models**

Figure 3 summarizes the final validation and test Top-1 accuracy across all trained models, highlighting ConvNeXt-Base as the strongest performer, followed by EfficientNet variants.

OBSERVATION:

- ArcFace improved separability in identity embeddings.
- Balanced per-class quotas stabilized optimization.
- EfficientNets delivered stronger accuracy-per-FLOP trade-offs.
- Top-5 consistently exceeded Top-1, indicating close confusions among visually similar celebrities.

ABLATIONS & DESIGN RATIONALE:

- ArcFace vs. Softmax confirmed superiority of margin-based embeddings.
- Margin scheduling (0.15→0.50) mitigated instability in early epochs.
- Balanced quotas addressed long-tail identity distribution.
- Light augmentations preserved facial identity cues.

ERROR ANALYSIS:

- Frequent confusions between visually similar celebrities.
- Failures under occlusions (sunglasses, extreme poses).
- No leakage across splits (validation confirmed).

LIMITATIONS & RISKS:

- Raises privacy/ethics concerns.
- 224×224 crops may lose fine-grained facial details.

REPRODUCIBILITY:

- Fixed seeds (42).
- PyTorch + timm with AMP.
- Checkpoints saved per model.
- To reproduce: place archive.zip in Drive, extract, and run training loop unchanged.

FUTURE WORK:

- Open-set evaluation with unseen identities.
- Larger backbones (ViTs).
- Hard example mining.
- Advanced augmentations (RandAugment, erasing).
- Calibration for reliable probabilities.
- Export embeddings for retrieval and clustering tasks.