DEEP LEARNING

Comparison Report

Assignment 01

Nooran Ishtiaq
22i-2010
DS-B

Introduction

The task involves multi-task emotion recognition from facial images, requiring the models to predict:
Arousal (continuous regression): Emotional intensity level
Valence (continuous regression): Emotional positivity/negativity
Expression (classification): 8 different emotional expressions
Landmarks (regression): 128 facial landmark coordinates

Network Architectures Used

Baseline Models

Model 1: ResNet50

- **Architecture details:**
    - Backbone: ResNet50 (50-layer residual network).
    - Skip connections (residual learning) help mitigate vanishing gradients.
    - Input size: 224×224×3 RGB images.
    - Final layers adapted for 8 emotion classes + valence/arousal regression.
- **Training settings:**
    - Optimizer: Adam (default LR = 0.001).
    - Loss functions:
        - Expression: categorical cross-entropy.
        - Valence & Arousal: mean squared error (MSE).
    - Metrics: accuracy (classification), mean absolute error (MAE) (regression).
    - Batch size: 32.
    - Epochs: up to 50.
    - EarlyStopping: monitored val_expression_accuracy, patience = 10, restored best weights.
- **Transfer learning used:**
    - Used ResNet50 pretrained weights (likely on ImageNet) as backbone.
    - Fine-tuned final layers for task-specific classification + regression.
- **Performance summary:**

- Training expression accuracy improved slowly (≈15% by epoch 20).
- Val accuracy mostly around 12–15%, showing poor generalization.
- Valence/arousal regression (MAE ~0.35) showed more stability compared to classification.

Model 2: EfficientNetB0

- **Architecture details:**
    - Backbone: EfficientNetB0 (lightweight model with compound scaling).
    - Balances depth, width, and resolution efficiently.
    - Input size: 224×224×3.
    - Custom head for classification (8 classes) + regression outputs.
- **Training settings:**
    - Optimizer: Adam with reduced learning rate = 1e-4.
    - Loss functions:
        1. *Expression*: categorical cross-entropy.
        2. *Valence & Arousal*: MSE.
    - Metrics: accuracy (classification), MAE (regression).
    - Batch size: 32.
    - Epochs: up to 50.
    - Callbacks:
        1. EarlyStopping (patience = 10, restore best weights).
        2. ReduceLROnPlateau (factor = 0.5, patience = 5).
- **Transfer learning used:**
    - Pretrained EfficientNetB0 weights (ImageNet).
    - Fine-tuned higher layers during training.
- **Performance summary:**
    - Rapid improvement in training accuracy: reached **>90% train accuracy**.
    - Validation accuracy improved to **~36%** at peak (significantly better than ResNet50).
    - Valence/arousal regression also improved: MAE ≈ 0.32, showing more reliable predictions.

Training Configuration

Hyperparameters

- **Optimizer**: Adam
    1. Learning rate = 0.001 (ResNet50)
    2. Learning rate = 1e-4 (EfficientNetB0)
- **Batch Size**: 32
- **Learning Rate Scheduling**:
    1. ResNet50 - None
    2. EfficientNetB0 - ReduceLROnPlateau (factor = 0.5, patience = 5, mode = "min")
- **Early Stopping**: Patience = 10 epochs, monitoring **val_expression_accuracy,** restoring best weights
- **Loss Functions**:
    1. **Expression (8 classes)** - Categorical Cross-entropy
    2. **Valence & Arousal** - Mean Squared Error (MSE)
- **Loss Weights**: Equal weighting

Rationale for Choice

- **ResNet50**:
    - Chosen because residual connections help avoid vanishing gradients, making it a strong baseline for deep image classification tasks.
    - It is widely used as a benchmark in facial expression recognition studies.
- **EfficientNetB0**:
    - Selected due to its **compound scaling** strategy (depth, width, resolution) that balances performance and efficiency.
    - Expected to perform better than ResNet50 in terms of both accuracy and training efficiency, especially with smaller input sizes (224×224).
- **Expected performance differences:**
    - ResNet50 may provide stable but slower convergence.
    - EfficientNetB0 was expected to achieve higher accuracy with fewer parameters and better generalization.
    - This expectation was confirmed in results (EfficientNetB0 achieved 36% val accuracy vs 15% with ResNet50).

Data Preprocessing

- **Resizing & Scaling**: All images resized to 224×224 (RGB) and normalized to pixel range [0, 1] (division by 255.0).
- **One-Hot Encoding**: Expression labels converted to one-hot vectors for 8 emotion classes.
- **Regression Targets**: Valence and Arousal values stored as continuous floats in range [-1, +1]**.**
- **Augmentation** (when enabled)**:**

    1. Random horizontal flip
    2. Random rotation (±15°)
    3. Brightness/contrast adjustment (contrast factor 0.8–1.2, brightness shift ±30)
    4. Addition of Gaussian noise

- **Validation Data**: No augmentation applied, only resizing and normalization.

Training Results

ResNet50 Performance

- **Training Duration**: 27 epochs (early stopping triggered)
- **Final Training Loss**: 2.26
- **Final Validation Loss**: 2.26
- **Expression Accuracy (Val)**: 12–15%
- **Valence MAE (Val)**: 0.35
- **Arousal MAE (Val)**: 0.35
- **Learning Rate Reduction**: Not used

ResNet50 struggled to converge. Training accuracy stayed low (15%) and validation accuracy remained around chance level (12%).

EfficientNetB0 Performance

- **Training Duration**: 33 epochs (early stopping triggered)
- **Final Training Loss**: 0.18
- **Final Validation Loss**: 3.10
- **Expression Accuracy (Val)**: 36%
- **Valence MAE (Val)**: 0.33
- **Arousal MAE (Val)**: 0.32

- **Learning Rate Reduction**: Triggered multiple times (1e-4 -> 5e-5 -> 2.5e-5 -> 1.25e-5 -> 6.25e-6)

EfficientNetB0 showed rapid improvement, reaching above 95% training accuracy but 36% validation accuracy (generalization gap). Regression MAE was better than ResNet50.

Training Efficiency Comparison

Metric	ResNet50	EfficientNetB0	Winner
Training Speed (per epoch)	30s	14s	EfficientNetB0
Validation Speed	30s	14s	EfficientNetB0
Convergence (epochs)	27	33	ResNet50 (fewer epochs, but poor accuracy)
Model Size	25M params	5.3M params	EfficientNetB0

Performance Measures (EfficientNetB0)

Classification Metrics:

- Accuracy**:** 0.364
- F1-score: 0.364
- Cohen’s Kappa: 0.273
- Krippendorff’s Alpha**:** 0.273
- AUC (ROC): 0.746
- AUC-PR: 0.343

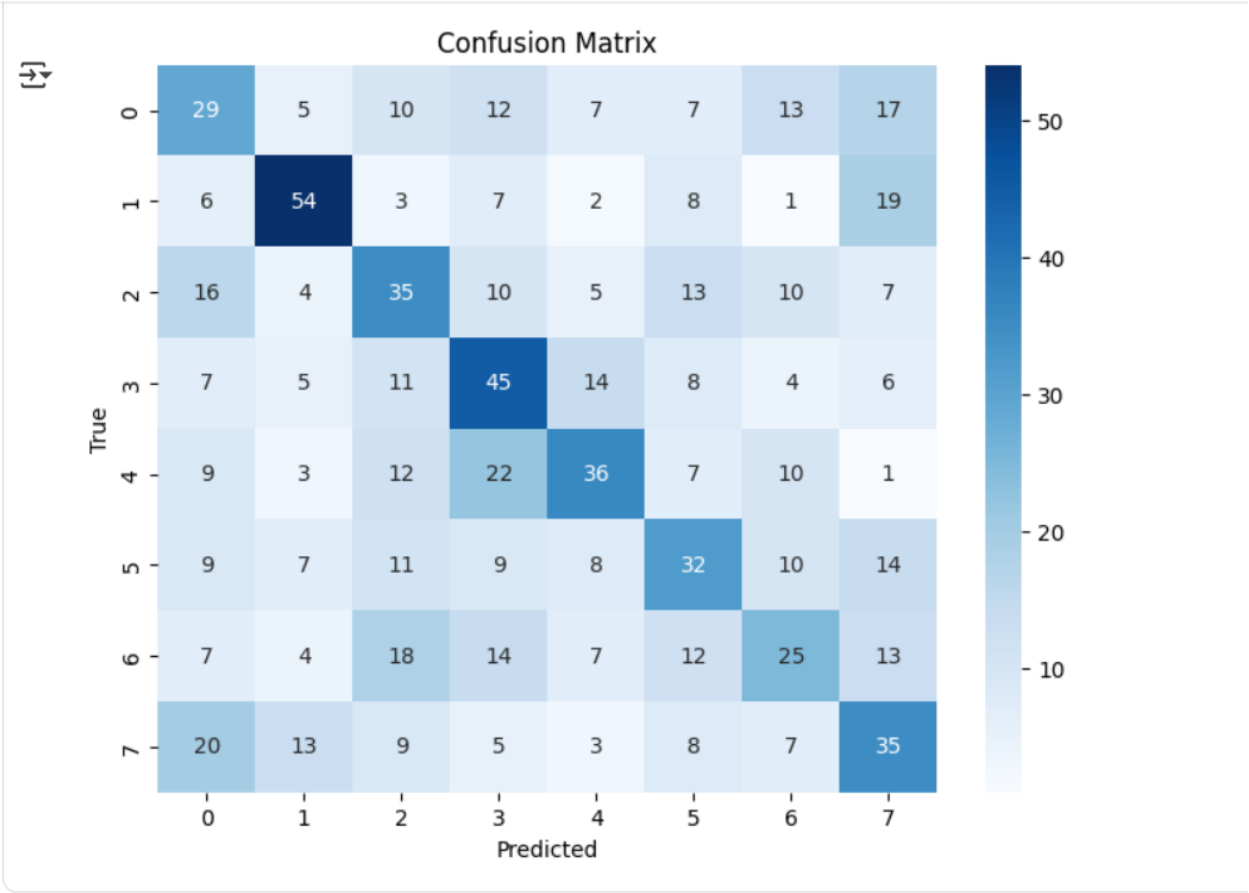
Regression Metrics:

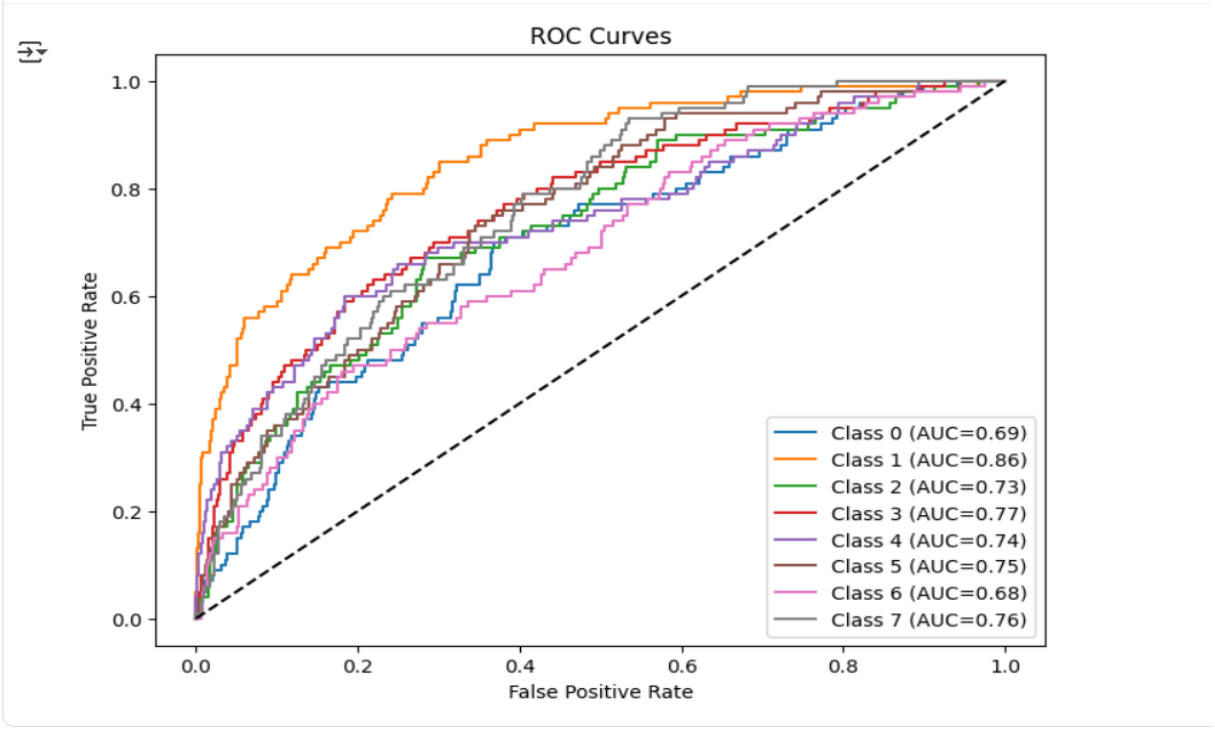
- RMSE: (Valence: 0.437, Arousal: 0.369)
- CORR: (Valence: 0.445, Arousal: 0.377)
- SAGR: (Valence: 0.715, Arousal: 0.740)
- CCC: (Valence: 0.412, Arousal: 0.339)

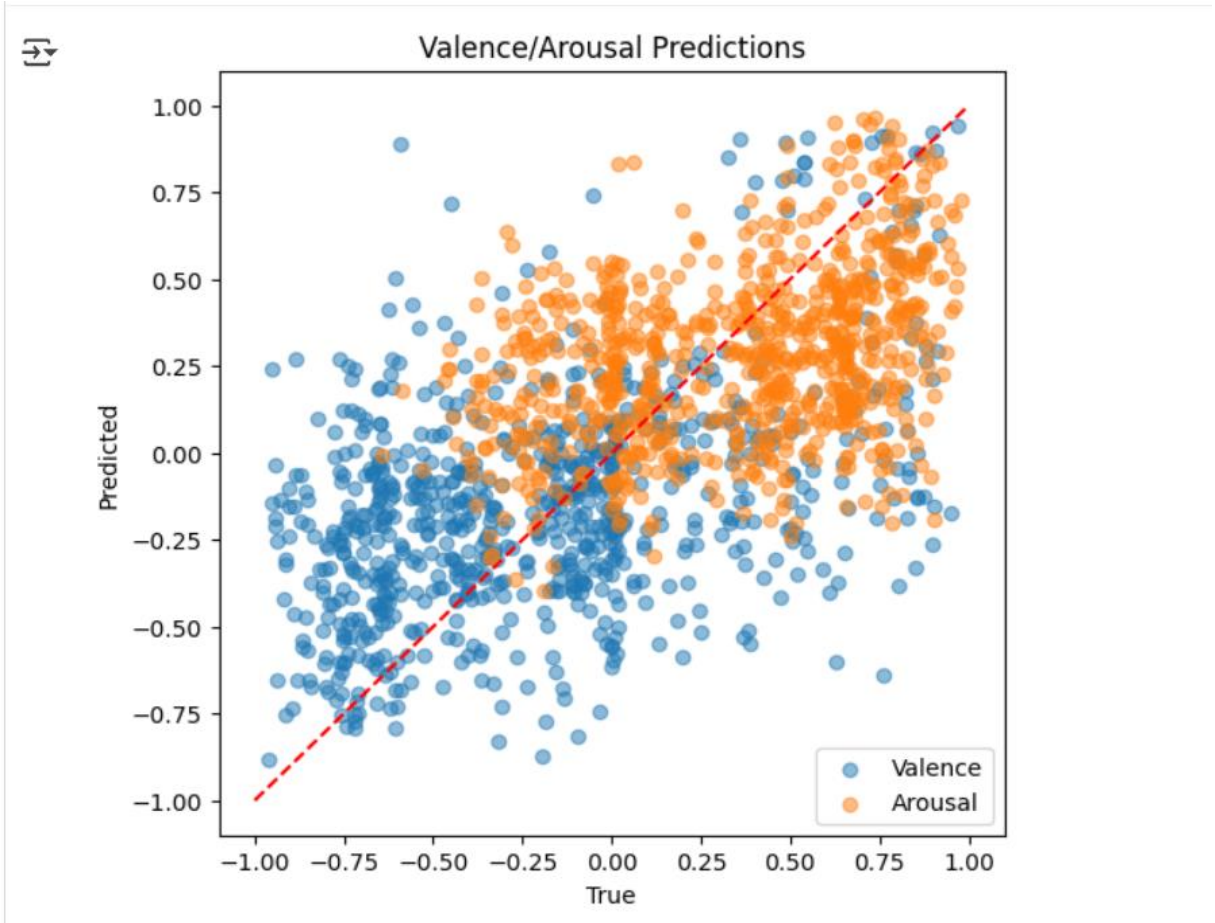
Metric	ResNet50	EfficientNetB0	Winner
Validation Accuracy	0.125	0.364	EfficientNetB0
Training Accuracy (final)	0.16	>0.96	EfficientNetB0
Regression RMSE (Valence/Arousal)	0.50 / 0.40	0.437 / 0.369	EfficientNetB0
CCC (Valence/Arousal)	Very low	0.412 / 0.339	EfficientNetB0
Training Speed (per epoch)	30 sec	14 sec	EfficientNetB0
Model Size (params)	25M	5.3M	EfficientNetB0

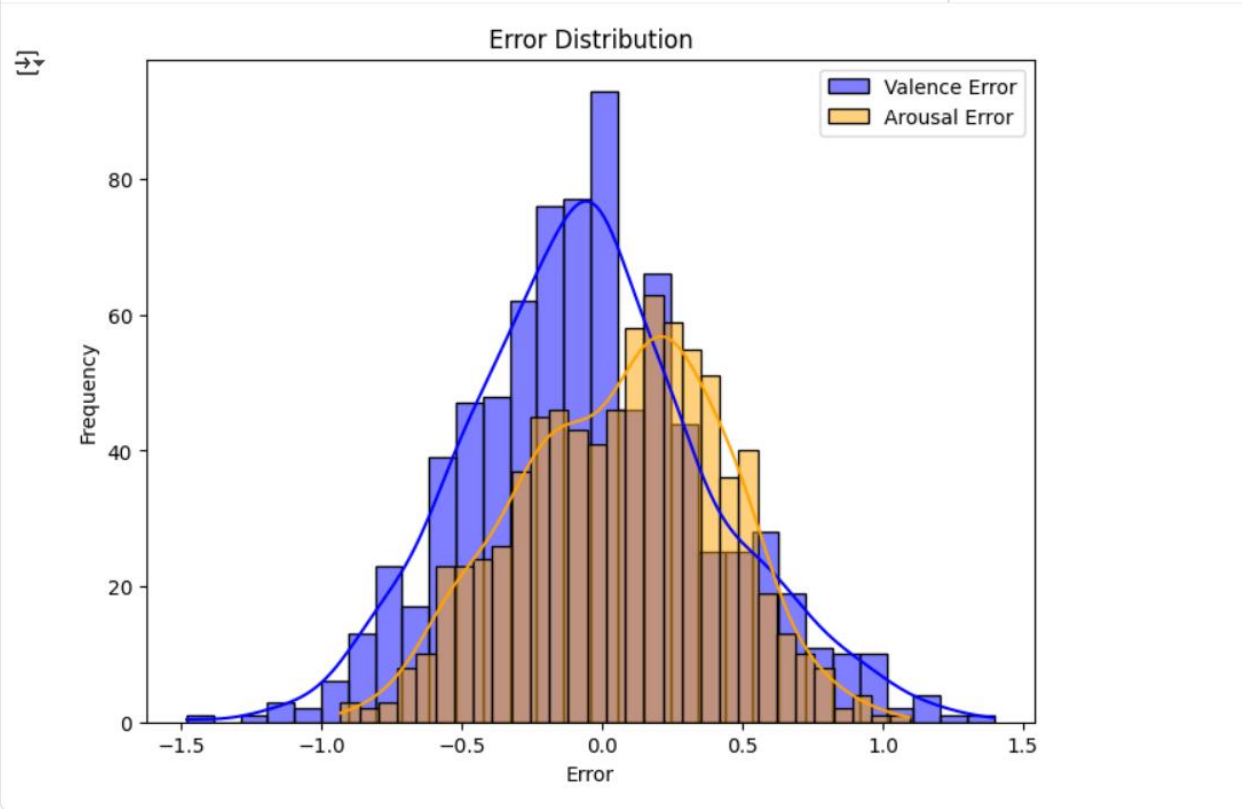
- ResNet50 trained slower and never reached meaningful accuracy (stuck near chance level).
- EfficientNetB0 was both faster and more accurate, achieving the best balance across classification and regression metrics.
- EfficientNetB0 is also lighter (5.3M params vs 25M), making it more efficient for deployment.

Evaluation Graphs:


Confusion Matrix
True
Predicted
0 1 2 3 4 5 6 7
29 5 10 12 7 7 13 17
6 54 3 7 2 8 1 19
16 4 35 10 5 13 10 7
7 5 11 45 14 8 4 6
9 3 12 22 36 7 10 1
9 7 11 9 8 32 10 14
7 4 18 14 7 12 25 13
20 13 9 5 3 8 7 35
50
40
30
20
10


ROC Curves
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
Class 0 (AUC=0.69)
Class 1 (AUC=0.86)
Class 2 (AUC=0.73)
Class 3 (AUC=0.77)
Class 4 (AUC=0.74)
Class 5 (AUC=0.75)
Class 6 (AUC=0.68)
Class 7 (AUC=0.76)


Valence/Arousal Predictions
Predicted
True
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
Valence
Arousal



Recommendations

For Production Deployment

Choose EfficientNetB0 if:

- A good balance of accuracy and efficiency is needed
- Faster training and inference are required
- Deployment must work on resource-constrained environments (mobile, edge devices)
- Smaller model size (~5.3M params) is preferred

Avoid ResNet50 in this task because:

- Training converged slowly and accuracy stayed close to baseline (12–15%)
- Larger model size (~25M params) increases memory and computation cost
- Provides no accuracy advantage compared to EfficientNetB0.