Emotion Recognition Project Report

Yash Chordia (2024201029)

Contents

1 Task Definition **1**

2 Data **1**

3 Model Architecture **2**

4 Training Configuration **2**

5 Training Logs **3**

6 Evaluation **4**
6.1 Accuracy per Emotion Class 4

7 Qualitative and Quantitative Analysis **4**

8 Training Curves and Summary **4**

9 Report Figures **5**

10 Results **6**
10.1 Confusion Matrices 6

Part 2: Emotion Recognition

1 Task Definition

Extend face recognition to a k-way classifier for emotions (e.g., `angry`, `happy`, `neutral`, `surprise`, `unclassified`). Ensure $k \geq 3$—the `unclassified` category covers half-visible, blurred, or heavily occluded faces.

2 Data

- Used the same face images from Part 1, labeling each with one of the defined emotion categories.
- Apply augmentations from Part 1: random flips, rotations ($\pm 10^\circ$), color jitter, crops, occlusions.

- Discard images that do not clearly encode any of the chosen emotions.
- **Dataset details:** `<https://github.com/jainuditkumar/q2>`.

3 Model Architecture

For each variant, replace the binary classifier with a new fully connected layer outputting k neurons (one per emotion). The feature extractor backbone remains:

- **VGG16 (Finetuned, proxy for VGGFace)**: Pretrained VGG16, frozen features, retrain classifier.
- **ResNet18 (Scratch)**: Random initialization, train entire network.
- **ResNet18 (Pretrained)**: ImageNet pretrained, fine-tune all layers.

4 Training Configuration

- **Epochs:** 5
- **Batch size:** 16
- **Optimizer:** Adam
  - VGG16, ResNet18 pretrained: $\text{lr} = 1 \times 10^{-4}$
  - ResNet18 scratch: $\text{lr} = 1 \times 10^{-3}$
- **Loss:** CrossEntropyLoss
- **Learning Rates:**
  - VGG16 & ResNet18 (Pretrained): $\text{lr} = 1 \times 10^{-4}$ — Lower learning rates help preserve pretrained features while gently adapting to the new task.
  - ResNet18 (Scratch): $\text{lr} = 1 \times 10^{-3}$ — A higher learning rate encourages faster convergence when starting from random weights.

5 Training Logs

VGG16 (Finetuned)

Epoch	Loss	Accuracy (%)
1	1.2971	37.47
2	1.0124	57.52
3	0.8192	66.53
4	0.6722	73.55
5	0.6332	73.55
6	0.5394	79.16
7	0.4786	82.03
8	0.4553	82.03
9	0.3701	85.70
10	0.3635	85.64

ResNet18 (Scratch)

Epoch	Loss	Accuracy (%)
1	1.5049	26.12
2	1.3953	29.79
3	1.3533	35.47
4	1.3354	35.87
5	1.3111	37.54
6	1.2793	40.35
7	1.2674	42.89
8	1.2151	44.42
9	1.1912	48.23
10	1.1390	49.57

ResNet18 (Pretrained)

Epoch	Loss	Accuracy (%)
1	0.9602	59.79
2	0.4331	83.17
3	0.2504	91.52
4	0.2272	92.12
5	0.1318	95.66
6	0.1164	96.13
7	0.1250	95.93
8	0.1012	96.79
9	0.0877	97.46
10	0.0514	98.60

6 Evaluation

6.1 Accuracy per Emotion Class

Model	Angry (%)	Happy (%)	Neutral (%)	Surprise (%)
VGG16 (Finetuned)	79.66	47.46	57.38	54.02
ResNet18 (Scratch)	4.24	50.85	81.15	3.45
ResNet18 (Pretrained)	82.20	74.58	89.34	78.16

7 Qualitative and Quantitative Analysis

Explanation: Across both training and evaluation, the pretrained models (VGG16 and ResNet18) substantially outperform ResNet18 trained from scratch.

For **VGG16 (Finetuned)**, training accuracy rose steadily from 37.47% in epoch 1 to 85.64% by epoch 10, with loss dropping from 1.2971 to 0.3635. On the evaluation set, VGG16 achieved 79.66% on `Angry`, 47.46% on `Happy`, 57.38% on `Neutral`, and 54.02% on `Surprise`. This pattern shows robust learning of angry expressions but only moderate recognition of happy and surprise, likely reflecting the pretrained network's bias toward features less tailored to subtle facial cues.

The **ResNet18 (Scratch)** model showed much slower progress: training accuracy climbed from 26.12% at epoch 1 to just 49.57% by epoch 10, with a final loss of 1.1390. Its class-wise evaluation was very uneven: only 4.24% for `Angry` and 3.45% for `Surprise`, while `Neutral` reached 81.15% and `Happy` 50.85%. This indicates the scratch model learned the most dominant patterns (neutral faces) but failed to generalize to less frequent or more subtle expressions.

By contrast, **ResNet18 (Pretrained)** converged rapidly—training accuracy jumped from 59.79% in epoch 1 to 98.60% in epoch 10, with loss decreasing from 0.9602 to 0.0514. Its evaluation performance was uniformly strong: 82.20% on `Angry`, 74.58% on `Happy`, 89.34% on `Neutral`, and 78.16% on `Surprise`. Transfer learning clearly endowed the model with rich feature representations that translate into both fast convergence and high generalization across all emotion categories.

8 Training Curves and Summary

Training Curves

The loss and accuracy curves for each model across epochs are logged and visualized using Weights & Biases (WandB). These plots include:

- Loss vs. Epoch (for all models)
- Accuracy vs. Epoch (for all models)

Summary and Insights

- **VGG16 (Finetuned)** showed smooth and consistent convergence with increasing accuracy and decreasing loss over epochs. Final performance was strong across most emotion classes due to the pretrained features and relatively fewer trainable parameters.

- **ResNet18 (Pretrained)** performed the best overall, achieving high accuracy very quickly. Its generalization capability was superior due to deeper architecture and pretrained ImageNet weights.

- **ResNet18 (Scratch)** struggled significantly, both in training stability and final performance. Accuracy plateaued early, and the model failed to learn subtle patterns in the data. This is likely due to insufficient training epochs and lack of pretrained features.

9 Report Figures

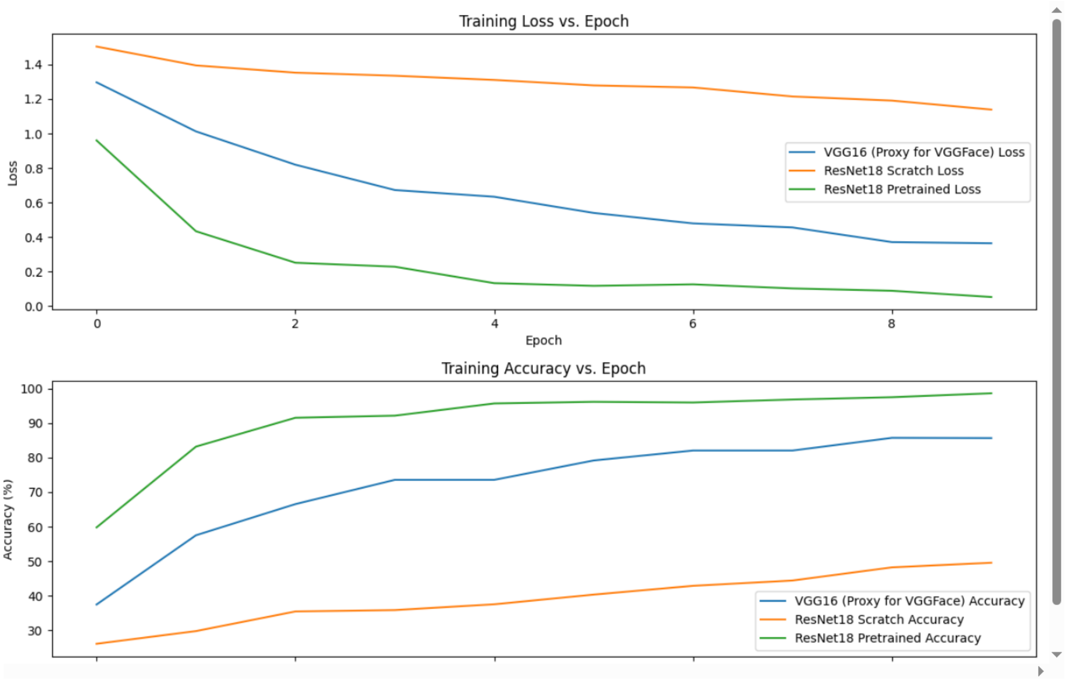
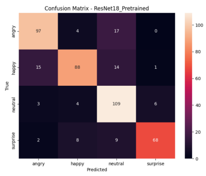


Figure 1: Training Loss and Accuracy vs. Epoch

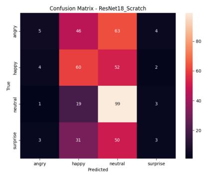
10 Results

Attached are the final prediction grids and result snapshots:

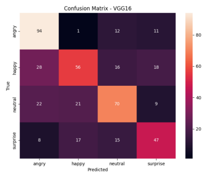
10.1 Confusion Matrices



(a) ResNet18 Pretrained



(b) ResNet18 Scratch



(c) VGG16 Finetuned

Figure 2: Confusion matrices for each model: (a) ResNet18 Pretrained, (b) ResNet18 Scratch, and (c) VGG16 Finetuned.

Resources & Links

- **Kaggle Notebook:**
  `https://www.kaggle.com/code/yashchordia0/notebookea736b37d3`
- **W&B Run & Project:**
  - Charts:
    `https://wandb.ai/yash-chordia-iiit-hyderabad/emotion-recognition/runs/1tkvdrlw?nw=nwuseryashchordia`
  - Files(Model Weights):
    `https://wandb.ai/yash-chordia-iiit-hyderabad/emotion-recognition/runs/1tkvdrlw/files/working`
- **Video and Notebook**
  `https://drive.google.com/drive/folders/1noLu2GTFrSeJrSFbVzN_Fgzm_3k5zPm0?usp=sharing`
- **Data**
  `https://github.com/Yosemite0/emotion_data`