Avestia Publishing
Journal of Machine Intelligence and Data Science (JMIDS)
Volume 5, Year 2024
ISSN: 2564-3282
DOI: 10.11159/jmids.2024.005

Optimising Real-Time Facial Expression Recognition with ResNet Architectures

Haoliang Sheng[1], MengCheng Lau[1]
[1]Laurentian University
School of Engineering and Computer Science
935 Ramsey Lake Rd, Sudbury, Ontario, Canada
hsheng@laurentian.ca; mclau@laurentian.ca

Abstract - *This study explores the performance of various ResNet architectures, namely ResNet18, ResNet34, and ResNet50, in the task of recognizing facial expressions using the FER-2013 dataset. By implementing transfer learning and fine-tuning these models, we rigorously evaluate their effectiveness through both controlled tests and real-time applications using webcam inputs. Our findings reveal that ResNet18, despite its simpler structure, achieves the optimal balance between accuracy and efficiency. This underscores its potential for practical deployment in real-time scenarios. The research emphasizes the critical importance of comprehensive datasets in enhancing model generalization, especially for expressions that are underrepresented in the training data. Additionally, it highlights the necessity of striking a careful balance between model complexity and computational efficiency to ensure suitability for real-world applications. This work significantly advances the field of real-time facial expression recognition, offering valuable insights into the deployment of deep learning models in practical, dynamic environments.*

Keywords**:** Facial Expression Recognition, ResNet Architectures, Transfer Learning, Real-time Webcam.

1. Introduction

Facial expression recognition is crucial in fields such as human-computer interaction, security, and health assessment. The use of deep learning, particularly Convolutional Neural Networks (CNNs), has greatly enhanced its capabilities, providing major improvements over older methods. For instance, Li et al. developed a multi-stage convolutional neural network cascade that significantly improves detection accuracy while maintaining computational efficiency [1]. Ming et al. introduced FaceLiveNet, an end-to-end network that integrates face verification and liveness detection using facial expressions, enhancing the security of facial recognition systems [2]. Bhatti et al. developed a method using deep features extracted by CNNs and classified with extreme learning machines, improving accuracy and speed in educational settings [3]. Gan proposed a CNN model specifically designed for facial expression recognition, leading to improved performance in recognizing emotions from facial images [4].

Among these advancements, Deep Residual Networks (ResNets) stand out for their ability to tackle deep network training challenges, offering more

accurate and efficient recognition solutions. He et al. introduced deep residual learning, which allows the training of very deep neural networks by using residual connections that bypass one or more layers, significantly improving the accuracy of image recognition tasks [5]. Durga and Rajesh proposed a ResNet-based facial recognition system for multimedia applications, enhancing the accuracy and robustness of facial recognition technology [6]. Li and Lima utilized the ResNet-50 architecture for facial expression recognition, leveraging its deep learning capabilities to accurately identify and classify facial expressions, thus improving performance and reliability [7]. The use of ResNets has particularly been beneficial in recognizing subtle changes in facial expressions, which is essential for seamless human-computer interactions.

Date Received: 2024-03-09
Date Revised: 2024-07-12
Date Accepted: 2024-07-30
Date Published: 2024-08-23

The need for real-time and dynamic facial expression recognition has grown with advancements in technology, especially in surveillance and real-time communications, where high accuracy and low latency are paramount. Hajarolasvadi and Demirel proposed a method for deep facial emotion recognition in video sequences using eigenframes, which improves the accuracy and efficiency of emotion recognition in dynamic video environments [8]. Li and Deng provided a comprehensive survey of deep learning techniques for facial expression recognition, summarizing various approaches, datasets, and evaluation metrics, which underscores the breadth and depth of research in this field [9]. ResNets have been instrumental in addressing these needs due to their superior performance and computational efficiency.

The success of facial expression recognition also depends on advanced feature extraction methods like higher-order statistical features and Local Binary Patterns. Ali et al. employed higher-order spectra and support vector machines (SVM) for facial emotion recognition, demonstrating improved accuracy and robustness [10]. Sawardekar and Naik combined efficient Local Binary Patterns (LBP) with CNNs for facial expression recognition, enhancing the feature extraction process [11]. However, integrating these into ResNet architectures yielded minimal performance gains in our trials. Lv et al. implemented deep learning techniques for facial expression recognition, showcasing the potential of neural networks in automatically identifying and classifying facial expressions [12]. Debnath et al. proposed a four-layer ConvNet for facial emotion recognition, emphasizing the importance of data diversity and achieving high accuracy with minimal training epochs [13]. Zeng et al. combined hand-crafted features with deep learning models to enhance the performance of facial expression recognition systems, integrating traditional feature extraction methods with modern neural networks [14]. Although this exploration did not significantly impact our study, it indicates our attempts to refine recognition models.

The application of these technologies in areas such as education and social cognition highlights their broad utility. For instance, Bhatti et al. developed a method using deep features extracted by CNNs and classified with extreme learning machines, significantly improving accuracy and speed in educational settings by effectively recognizing instructors' facial expressions and providing real-time feedback to students [3]. This demonstrates the potential of advanced facial expression recognition systems to enhance interactive learning environments. Additionally, Graumann et al. investigated the impact of acute psychosocial stress on facial emotion recognition in borderline patients, finding that their recognition ability remains unaffected by stress, which suggests the robustness of facial recognition technologies in varying psychological conditions [15].

Our study begins with an analysis of the FER-2013 dataset [16], emphasizing its diversity and the challenges it presents, which are pivotal for evaluating advanced deep learning models in facial expression recognition. The methodology section outlines our approach using ResNet architectures and transfer learning, detailing the fine-tuning necessary for emotion recognition tasks. Experimental results compare model performances, assessing accuracy, efficiency, and real-time applicability through webcam tests. We conclude by highlighting the balance between model complexity and training data comprehensiveness, proposing future research avenues for improving model sensitivity and real-time performance.

2. Dataset

The FER-2013 dataset [16] is an extensive and well-curated collection specifically tailored for facial expression recognition tasks. This dataset encompasses seven distinct emotion categories: Angry, Disgust, Fear, Happy, Sad, Surprise, and Neutral. It contains a total of 35,886 images, with 28,708 images allocated for training and 3,589 images reserved for testing purposes. Each image in the dataset is a grayscale image with dimensions of 48x48 pixels, providing a consistent format for processing and analysis. However, the relatively low resolution of these images, when compared to the high-resolution inputs typically obtained from webcams, presents a substantial challenge in maintaining the model's efficacy in real-world applications.

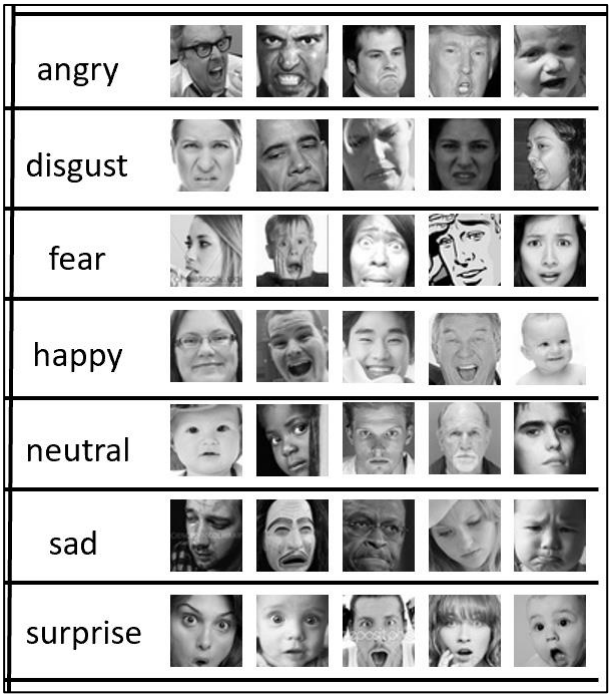

Figure 1. FER-2013 dataset

To address this challenge and enhance the model's robustness, we have decided to utilize grayscale images during the training process. This approach serves two primary purposes: first, it simplifies the model by focusing on structural features rather than colour information, which is less critical for the accurate recognition of facial expressions. Second, it mirrors the conditions of real-time webcam inputs, where variations in lighting conditions and background can significantly affect colour information. By employing this grayscale strategy alongside various data augmentation techniques—such as random flips, rotations, and adjustments in brightness, contrast, and affine transformations—we aim to develop a model that is not only simplified but also robust and capable of accurately classifying facial expressions under a wide range of dynamic real-world conditions. This combination of grayscale imaging and augmentation techniques ensures that our model can handle the variability and unpredictability of real-time facial expression recognition effectively.

3. Methodology

3. 1. Residual Network

ResNet, short for Residual Network, is a type of deep neural network that has been highly effective in various tasks such as image recognition and classification. Introduced by Kaiming He and his colleagues in 2015 [5], ResNet revolutionized the way deep networks are trained by addressing some of the critical issues associated with training very deep neural networks. The core idea behind ResNet is the use of residual learning, which simplifies the training of networks that are substantially deeper than those previously used.

A crucial component of ResNet is the use of skip connections, also known as residual connections. These connections work by allowing the input of a layer to be added directly to the output of a deeper layer, effectively skipping one or more layers in between. This mechanism helps to combat the vanishing gradient problem, a common issue in very deep networks where the gradients used for training can become exceedingly small, making it difficult for the earlier layers in the network to learn effectively. By providing an alternative pathway for the gradient to flow through the network, skip connections ensure that gradients remain robust even in deep layers, facilitating better learning and improving the overall training process. This innovative approach enables the construction of networks with hundreds or even thousands of layers, significantly enhancing their performance on complex tasks.

3. 2. Transfer Learning Approach

In our research, we have implemented the concept of transfer learning to harness the sophisticated capabilities of pre-trained models for feature extraction. We specifically selected the ResNet18, ResNet34, and ResNet50 architectures—different variants of the Residual Network—based on our objective to strike a careful balance between model depth and computational efficiency. This balance is crucial for ensuring that our models can be effectively deployed in real-time applications. By leveraging these pre-trained ResNet models, we aim to utilize their pre-established feature extraction abilities, which are essential for enhancing the accuracy and performance of our facial expression recognition system. This approach not only accelerates the training process but also enables our models to achieve high performance with fewer data, making them well-suited for practical, real-world deployment scenarios where both speed and accuracy are paramount.

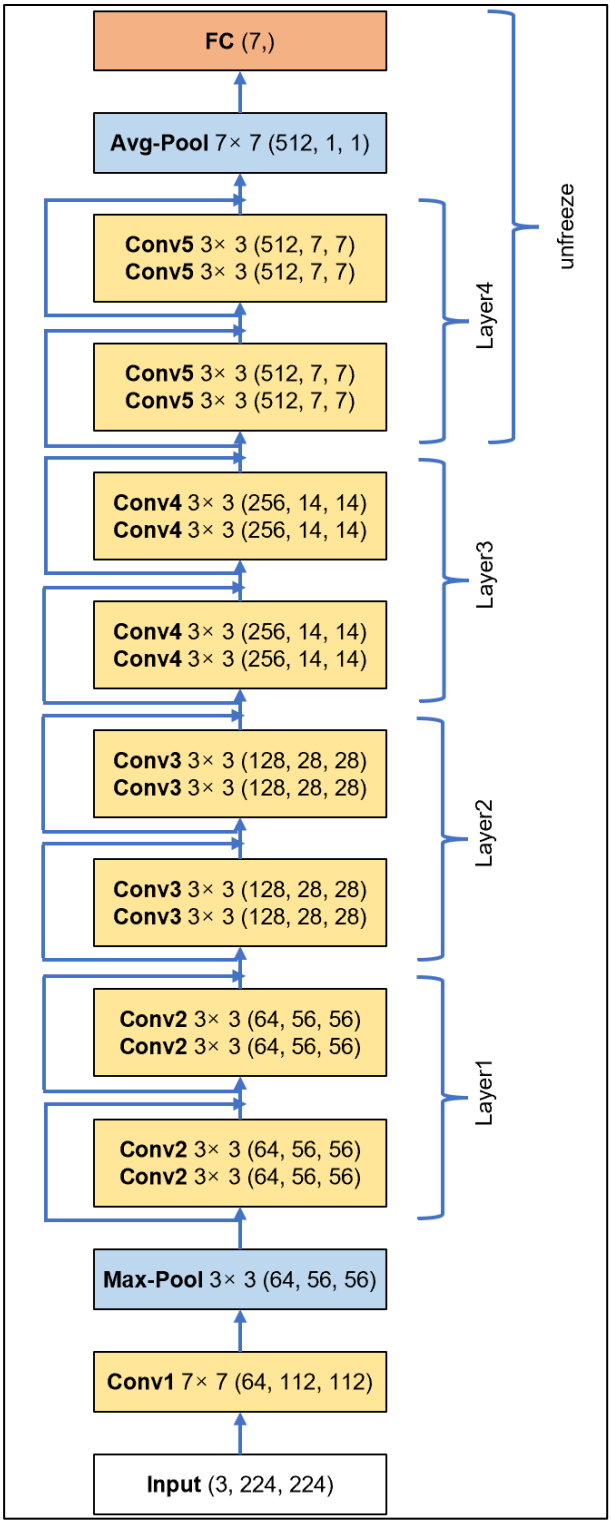

Figure 2. Architecture of ResNet18 model.

The illustration in Figure 2 details the architectural configurations of the ResNet18 model. Each box in the figure includes the module name in bold, the kernel size, and the output dimensions in brackets. To tailor these ResNet models specifically for our study, we customized the architecture by adjusting the final fully connected (fc) layer to classify images into seven distinct categories: Angry, Disgust, Fear, Happy, Sad, Surprise, and Neutral. This adjustment was essential to adapt the models to the unique requirements of our task, particularly for the classification challenges presented by the FER-2013 dataset, which is known for its complexities in facial expression recognition. Our goal was to create a framework that clearly demonstrates the modifications made to the network to meet the specific demands of our study.

To fine-tune the models optimally for our classification task, we undertook a strategic fine-tuning process. This process involved freezing the initial layers of the models to preserve the features learned from the ImageNet dataset, while retraining the later layers to adapt to our specific classification needs. This strategy significantly reduced training time and resource consumption, all while maintaining high levels of accuracy and efficiency for real-time facial expression classification. A novel aspect of our approach was the decision not only to modify the last fully connected layer but also to unfreeze and train the penultimate layer, known as layer4. This decision was based on our findings that allowing layer4 to learn from our specific dataset could substantially enhance the model's classification accuracy. Implementing this method led to a remarkable improvement in training accuracy, increasing from 41.6% to an impressive 91.2%. This careful application of transfer learning, coupled with selective fine-tuning of specific network layers, underscores our commitment to developing an efficient and highly accurate system for real-time facial expression recognition. Through this methodology, we have successfully adapted existing deep learning architectures to tackle the particular challenges inherent in our research domain.

3. 3. Adam Optimizer

We used the Adam optimizer [17] for training our models. Adam, short for Adaptive Moment Estimation, is an optimization algorithm that combines the benefits of two other extensions of stochastic gradient descent: AdaGrad and RMSProp. It computes adaptive learning rates for each parameter. The Adam optimizer is known for its efficiency and effectiveness in training deep learning models, particularly in handling sparse gradients on noisy problems.

The Adam optimizer updates parameters using the following equations:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$$
$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2$$
$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}$$
$$\hat{v}_t = \frac{v_t}{1-\beta_2^t}$$
$$\theta_t = \theta_{t-1} - \frac{\alpha \hat{m}_t}{\sqrt{\hat{v}_t + \epsilon}}$$

Where:

- m_t and v_t are the first and second moment estimates.
- β_1 and β_2 are the exponential decay rates for the moment estimates, typically set to 0.9 and 0.999, respectively.
- g_t is the gradient at time step t.
- α is the learning rate.
- ϵ is a small constant to prevent division by zero, usually set to 10^{-8}.

In this project, we use the default configuration of *torch.optim.Adam* in PyTorch with a learning rate (α) of 0.001.

3. 4. Real-Time Processing

The core of our methodology for real-time prediction involves deploying a pre-trained model to detect and classify facial expressions in real time, utilising webcam input. This section describes the technical implementation designed to enable live classification of facial expressions, fostering interactive user engagement. We have constructed the system within a Python environment, utilising the OpenCV library for video capture and face detection, alongside PyTorch for deploying the trained model. The process initiates with activating the user's webcam, which continuously streams live images into the system. We then employ OpenCV's `*CascadeClassifier*`, specifically the `haarcascade_frontalface_default.xml`, to process these images and detect the presence of a face. Upon detecting a face within the video frame, the system extracts the region of interest (ROI), the detected face. We subsequently preprocess and transform this face ROI to meet the input specifications of the pre-trained model. This transformation typically includes resizing the image, normalising pixel values, and converting the image into a tensor format apt for model input. After preprocessing, we feed the face ROI tensor into the trained ResNet model, set in evaluation mode to ensure batch normalisation and dropout layers function in inference mode. The model then predicts the facial expression by outputting a class label corresponding to the detected expression.

The real-time prediction functionality resides within a user-friendly interface that exhibits the live video feed from the webcam. When the system identifies and classifies a facial expression, it outlines the detected face with a bounding box and displays the predicted expression label on the screen. This enables users to witness the classification results in real time as they exhibit various facial expressions to the webcam. The real-time prediction system's implementation is encapsulated in the '*realtime_prediction'* function, which accepts several parameters including the model, transformation procedures, display duration for the classification label, and the preferred computational device (CPU or GPU). This function controls the entire procedure from video capture, face detection, and expression classification to result display, ensuring a smooth and interactive user experience.

4. Experimental Setup and Results

This section provides an in-depth analysis of the performance and effectiveness of various ResNet architectures, specifically ResNet18, ResNet34, and ResNet50, in the context of facial expression recognition tasks. We meticulously document and analyse the training performance of each model, taking into account both training accuracy and the time required for training. This detailed examination allows us to assess the computational efficiency and learning capabilities of each architecture.

In addition to training performance, we thoroughly evaluate the testing performance of each model. This evaluation includes a comprehensive analysis across different models and within the specific classes of facial expressions, demonstrating how each architecture performs overall and in accurately identifying the seven distinct facial expressions: Angry, Disgust, Fear, Happy, Sad, Surprise, and Neutral.

To extend our analysis beyond theoretical performance metrics, we assess the real-world applicability of these models by testing them in a live setting using a webcam. This practical evaluation offers insights into the models' performance in dynamic, real-

time environments, thereby providing a complete picture of each model's practical value. Through this extensive evaluation, we aim to present a thorough understanding of the strengths and limitations of ResNet18, ResNet34, and ResNet50 in facial expression recognition, highlighting their potential for real-world deployment.

4. 1. Experimental Setup

In this section, we outline the experimental setup used to evaluate the performance of various ResNet architectures (ResNet18, ResNet34, and ResNet50) in facial expression recognition tasks. The setup details the hardware and software configurations, as well as the parameters used for training and testing the models.

Camera

- Model: C505e HD Camera
- Photo Quality: 0.9MP, 16:9 (1280x720)
- Video Quality: 720p, 16:9, 30fps

The C505e HD Camera was used to capture real-time facial expressions for live testing. The camera provides high-definition video quality at 720p resolution with a frame rate of 30 frames per second, ensuring clear and smooth video input for our models.

GPU

- Model: NVIDIA GeForce MX250
- Driver Version: 546.17
- CUDA Version: 12.3

The NVIDIA GeForce MX250 graphics card, equipped with driver version 546.17 and CUDA version 12.3, was utilized to accelerate the training and inference processes of our deep learning models. This setup ensures efficient computation and enhances the performance of the models during training and testing phases.

Software

PyTorch Version: PyTorch 2.3.1+cu118

The models were implemented and trained using PyTorch version 2.3.1+cu118. This version of PyTorch is compatible with CUDA 11.8, allowing us to leverage GPU acceleration for faster training times and improved computational efficiency.

The experimental setup was designed to ensure a robust and efficient environment for evaluating the performance of our ResNet models. The high-definition camera ensured high-quality input data for real-time testing, while the powerful NVIDIA GeForce MX250 GPU facilitated rapid model training and inference. PyTorch 2.3.1+cu118 provided a flexible and efficient framework for implementing and fine-tuning our deep learning models.

This setup allowed us to comprehensively assess the capabilities of ResNet18, ResNet34, and ResNet50 in recognizing facial expressions both in controlled testing environments and in real-time applications, thereby providing valuable insights into their practical deployment.

4.2 Hyperparameters and Training Setup

The training setup involves several critical hyperparameters and configurations, which are detailed below:

Layer Unfreezing: 'layer4'

- We unfreeze the layer4 of each ResNet model, allowing the weights in this layer to be updated during training. This selective unfreezing is aimed at enhancing the models' ability to learn task-specific features.

Number of Epochs (NUM_EPOCHS): 20

- The number of times the entire training dataset passes through the model during training.

Batch Size (BATCH_SIZE): 64

- The number of training samples used in one iteration of model training.

Learning Rate (LAYERS_LR):

- Different learning rates are assigned to different layers to fine-tune the model more effectively.
- layer4: 0.001
- fc (fully connected layer): 0.01

Class Names (CLASS_NAMES):

- The seven facial expression categories used for classification: 'angry', 'disgust', 'fear', 'happy', 'neutral', 'sad', 'surprise'.

Model Types (MODEL_TYPE):

- The ResNet architectures explored in this study: 'resnet18', 'resnet34', 'resnet50'.

We have configured the above hyperparameters and settings in a YAML file, enabling flexible and convenient modifications for future experiments. This approach allows for easy adjustments and combinations of different parameters, facilitating comprehensive analysis in subsequent studies.

4. 3. Training Performance

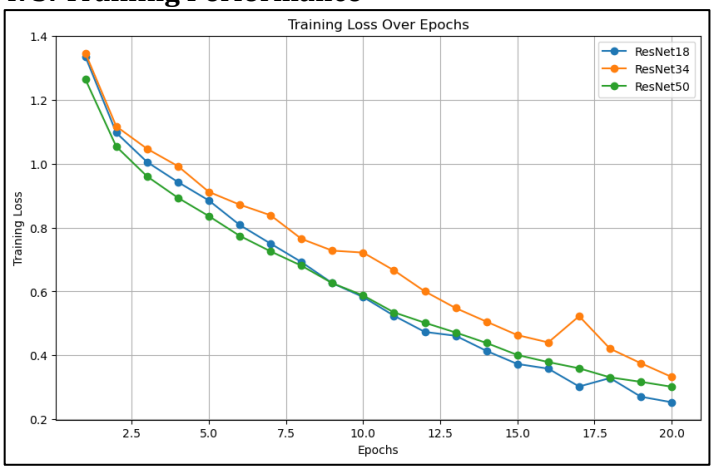

Figure 3. Train Loss over Epochs for ResNet models.

Figure3 shows train loss over epochs for ResNet models. Each line represents the training loss metric for a corresponding Residual Network model as it is trained over 20 epochs. The X-axis is labeled "Epochs" and ranges from 0 to 20, indicating the progression of training time, while the Y-axis is labeled "Training Loss" and appears to range from approximately 0.2 to 1.4, reflecting the magnitude of the loss.

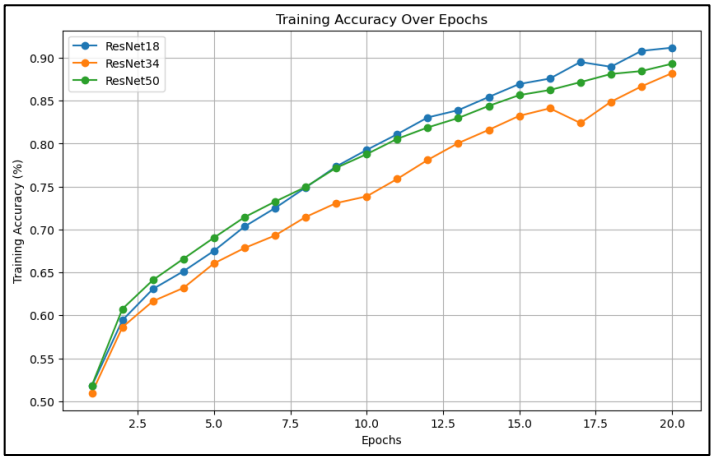

Figure 4. Train Accuracy over Epochs for ResNet models.

Figure 4 presents the validation loss of three different ResNet architectures—ResNet18, ResNet34, and ResNet50—over the course of 20 epochs during the validation phase of model training. The x-axis represents the epoch number, indicating the progression of the validation phase, while the y-axis denotes the validation loss, reflecting the models' performance on the validation dataset.

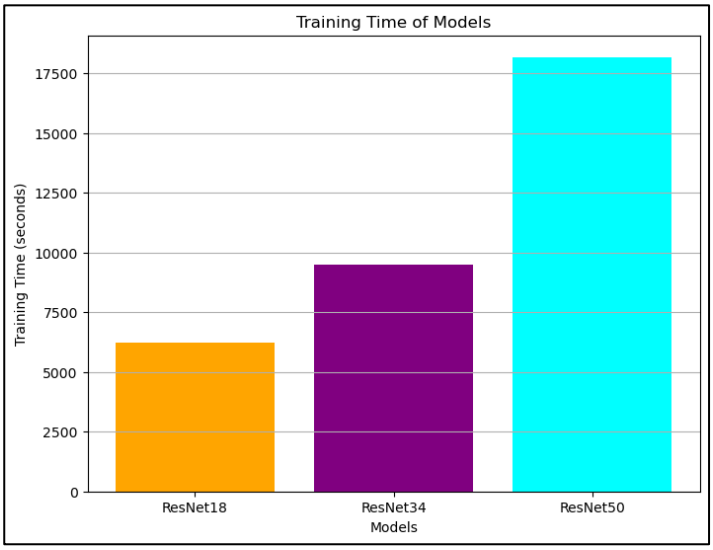

Figure 5. Train Time for ResNet models.

Figure 5 presents a bar graph illustrating the training time, measured in seconds, for three different models: ResNet18, ResNet34, and ResNet50. Each model's training time is represented by a coloured bar—orange for ResNet18, purple for ResNet34, and cyan for ResNet50. We can observe that:

i. **Performance Over Time**: Initially, ResNet50 exhibited the highest performance among the models. However, by the 9th epoch, ResNet18, which has the least complexity, surpassed ResNet50. This indicates that while deeper networks may have an initial advantage due to their increased capacity, simpler architectures like ResNet18 can catch up and even outperform them as training progresses, benefiting from faster convergence and potentially more effective generalization.

ii. **Consistency in Performance**: Throughout the training process, ResNet34 consistently lagged behind in performance. Despite its medium complexity, it did not achieve the higher accuracy rates seen in either ResNet18 or ResNet50 at any point during the epochs measured. This consistent underperformance suggests that ResNet34 may not be optimally

balanced for the task at hand, neither leveraging the simplicity of ResNet18 nor the depth of ResNet50 effectively.

iii. **Training Efficiency**: ResNet18 not only caught up with ResNet50 in terms of accuracy by the 9th epoch but also completed training in a significantly shorter amount of time. This demonstrates its higher training efficiency, making it a compelling choice for scenarios where quick model turnaround and deployment are crucial. The reduced training time directly translates into lower computational costs and faster iteration cycles.

iv. **Resource Utilization**: ResNet50 required the longest training time, which implies a higher computational cost. Given that its performance was matched and eventually surpassed by the less complex ResNet18, this extended training time may not be justified. The higher resource demands of ResNet50 make it less attractive for practical applications where computational efficiency and resource constraints are important considerations.

v. **Model Complexity vs. Performance Trade-Off**: The observations suggest a diminishing return on increased model complexity. In the case of ResNet50, the additional layers and parameters do not translate into a proportional increase in performance during later epochs. This highlights the importance of balancing model complexity with practical performance gains, particularly when simpler models can achieve comparable or superior results.

vi. **Practical Implications for Model Selection**: When selecting a model for real-world applications, it is crucial to balance accuracy, training time, and computational resources. ResNet18, with its efficient training process and strong performance, may offer the best balance for many applications, especially in environments where resource efficiency is paramount. This makes it a suitable choice for deployment in resource-constrained settings without sacrificing performance.

vii. **Potential for Further Improvement**: None of the models displayed a plateau in accuracy after 20 epochs, indicating that further training could lead to additional performance gains. This suggests that the models have not yet fully converged and that continued training could improve accuracy, provided that overfitting is managed effectively. This potential for further improvement highlights the importance of continued experimentation with training duration and techniques to maximize model performance.

By taking these points into account, it becomes evident that ResNet18's simplicity and efficiency make it a strong candidate for real-world facial expression recognition tasks, balancing performance with practical constraints. During the training of the ResNet34 model, we observed a significant anomaly in the training accuracy at Epoch 17, depicted in Figure 4. Further analysis identified the 307th batch within this epoch as the source of the anomaly, with an exceptionally high batch loss. Figure 6 presents the batch losses for ResNet34 during the 17th epoch, highlighting the 307th batch as an outlier with a significantly higher loss. This visualization underscores the occasional irregularities that can occur during training and the need for robust analysis methods to understand their impact.

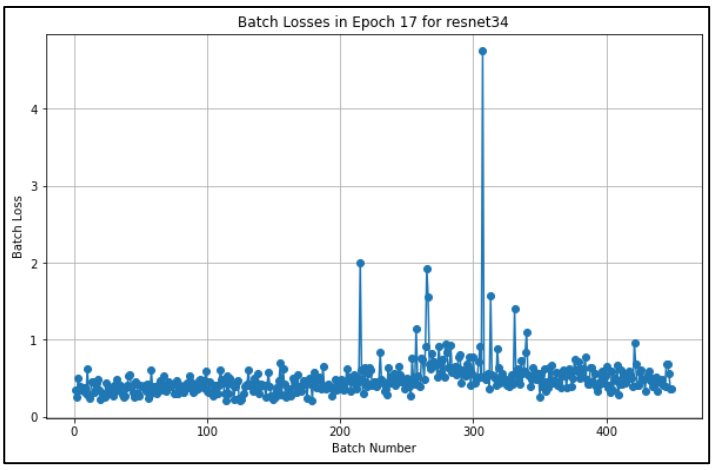

Figure 6. Batch Losses in Epoch 17 for ResNet34.

The following are some possible causes and explanations to provide a comprehensive explanation for the observed anomaly in ResNet34's training accuracy, enhancing the clarity and robustness of our findings.

i. **Consistency in Learning Rate:** Our training process maintained a consistent learning rate of 0.001 across all epochs, as we did not employ learning rate schedulers. Thus, fluctuations in learning rate are not a contributing factor to this anomaly.

ii. **Randomness in Code Execution:** The stochastic nature of neural network training introduces variability in outcomes. This includes random initialization of network weights, the order of batch processing, and random data augmentations. These factors collectively contribute to occasional irregularities, such as the one observed in the 307th batch. While we cannot pinpoint the exact images or transformations responsible, it is clear that such anomalies are part of the inherent randomness in model training.

iii. **Resilience of the Model:** Despite the anomaly, the training accuracy normalized in subsequent epochs, demonstrating the robustness of the model. This quick recovery suggests that such irregularities do not significantly impact the overall training process.

iv. **Relevance to Real-Time Predictions:** The ResNet34 model was not our primary choice for real-time facial expression recognition. Instead, we opted for ResNet18, which offers a better balance of accuracy and computational efficiency. Therefore, the observed anomaly, while notable, does not detract from the main conclusions of our study.

4. 5. Testing Performance

Following the training, the model was evaluated on a separate testing dataset to assess its generalization capabilities.

Evaluation Metrics:

To thoroughly assess our model's performance, we employed several key metrics [18], each offering unique insights into different aspects of the model's accuracy and efficiency.

i. **Precision**:
- Formula: Precision $= \frac{\mathrm{TP}}{TP+FP}$
- Function: Precision indicates the proportion of positive identifications that were actually correct. It is crucial for scenarios where the cost of false positives is high. High precision means that an algorithm returned substantially more relevant results than irrelevant ones.

ii. **Recall (Sensitivity)**:
- Formula: Recall $= \frac{TP}{TP+FN}$
- Function: Recall assesses the model's ability to find all relevant cases within a dataset. It is particularly important in situations where missing a positive (such as failing to identify a correct facial expression) is more problematic than falsely identifying one.

iii. **F1 Score**:
- Formula: $F1 = \frac{2\times(Precision\times Recall)}{Precision+Recall}$
- Function: The F1 Score is the harmonic mean of precision and recall, providing a balance between the two metrics. It is useful when you need to take both false positives and false negatives into account.

iv. **AUC Score**:
- Formula: $TPR = \frac{TP}{TP+FN}$; $FPR = \frac{FP}{FP+TN}$
- Function: The AUC score provides a measure of a model's ability to distinguish between the positive and negative classes. A higher AUC score means that the model is better at correctly classifying positive and negative examples. In the context of multi-class classification, AUC is often calculated for each class against all others and then averaged to provide a single score.

Table 1. Test performance for each Facial Expression.

Model	Precision	Recall	F1	Support	AUC
resnet18	0.648	0.650	0.646	7178	0.898
resnet34	0.652	0.651	0.650	7178	0.893
resnet50	0.664	0.654	0.658	7178	0.885

Table 1 provides a comprehensive overview of the testing performance for three distinct ResNet models: ResNet18, ResNet34, and ResNet50. Six key metrics are reported for each model, which include Precision, Recall, F1-Score, Support, Accuracy, and AUC (Area Under the Curve). Notably, the Precision, Recall, and F1-Score values correspond to the *'weighted avg'* output obtained from the `*classification_report*` function of `*sklearn.metrics*`, which accounts for the imbalance in the support of each class. The Support is consistent across all models, signifying the total count of test dataset instances used to evaluate each model. The Accuracy metric reported is the overall accuracy across all classes. The AUC values represent the arithmetic mean of the AUC for the seven different facial expressions, providing a consolidated view of the model's discriminative power.

Table 2: Test performance for each Facial Expression.

Expression	Precision	Recall	F1	Support	AUC
angry	0.560	0.590	0.571	958	0.872
disgust	0.727	0.583	0.647	111	0.958

fear	0.522	0.483	0.500	1024	0.805
happy	0.849	0.857	0.851	1774	0.965
neutral	0.589	0.619	0.601	1233	0.856
sad	0.516	0.507	0.511	1247	0.832
surprise	0.811	0.768	0.789	831	0.956

Table 2 offers a detailed analysis of classification performance metrics for seven different facial expressions: angry, disgust, fear, happy, neutral, sad, and surprise. These metrics encompass Precision, Recall, F1-Score, Support, and AUC, representing the arithmetic mean values from the performance of three ResNet models (ResNet18, ResNet34, and ResNet50). The following are our observations:

i. **Model Complexity vs. Performance**: The increased complexity associated with deeper networks, such as ResNet50, does not always translate to improved performance metrics. In many cases, ResNet18 either matches or exceeds ResNet50 in terms of accuracy and AUC. This observation underscores the importance of a balanced approach when selecting models, highlighting those simpler architectures can sometimes offer comparable or superior performance with lower computational costs.

ii. **Precision-Recall Trade-off**: The evaluation of the models reveals a nuanced trade-off between precision and recall across different facial expressions. No single model consistently outperforms the others in all categories, indicating the necessity of considering multiple metrics for a comprehensive assessment of model performance. This approach ensures a more rounded understanding of how each model handles various expressions.

iii. **Variability in Expression Recognition**: Significant variability is observed in the models' ability to recognize different facial expressions. For instance, expressions such as 'happy' and 'surprise' are classified with high precision and recall, whereas 'fear' and 'sad' are more challenging to recognize accurately. This suggests the need for model adjustments or targeted data augmentation strategies to improve performance on the more difficult expressions.

iv. **AUC**: The AUC scores exhibit differences among the various expressions, with 'happy' and 'disgust' achieving the highest scores. This indicates that the models are more adept at distinguishing these expressions, likely due to their more distinct or consistent features. Understanding these discrepancies can guide further refinements in model training and data collection.

v. **Impact of Support on Performance**: The support number, or the count of instances for each expression, does not directly correlate with performance. This is evident in the case of 'fear' and 'sad' expressions, which despite having ample instances, remain difficult to classify accurately. This highlights the importance of not only the quantity but also the quality and diversity of the training data in achieving robust model performance.

vi. **Opportunity for Model Optimization**: The relatively close range of metrics such as precision, recall, and F1-scores across the models suggests there is significant room for improvement. Potential enhancements could be achieved through hyperparameter tuning, extended training periods, or experimenting with alternative architectures. Such efforts could help to boost the overall performance of the models.

vii. **Importance of a Balanced Dataset**: The observed differences in performance across expressions highlight the critical need for a balanced and representative training dataset. Ensuring that the dataset adequately represents all expressions is crucial for enhancing the model's generalization capabilities and improving accuracy, especially for expressions that are currently less precisely classified.

These detailed observations provide a clear understanding of the strengths and limitations of different ResNet architectures in the context of facial expression recognition. They also point towards specific areas for improvement, guiding future research and development efforts in this field.

4. 6. Real-Time Webcam Performance

In this section on Real-Time Webcam, we delve into the practical application of our trained models within a live environment. Utilizing a standard webcam, we implemented real-time facial expression recognition to evaluate the models' abilities to interpret and classify

facial expressions as they naturally occur. This approach allows us to assess the practical viability and responsiveness of our models in dynamic, everyday settings, providing valuable insights into their performance outside of controlled testing conditions.

By conducting these real-time tests, we aim to determine how well the models handle the variability and unpredictability inherent in live human interactions. This evaluation is crucial for understanding the robustness and accuracy of our facial expression recognition system in real-world scenarios, ensuring that the models can deliver consistent and reliable results when deployed in practical applications.

Figure 7: Real-Time Facial Expression Detection Snapshots.

We captured the real-time performance of our trained models through a series of snapshots, each displaying the models' analytical interpretation of six discernible facial expressions: Angry, Fear, Happy, Sad, Surprise, and Neutral. Each snapshot corresponds to a detected facial expression, with the notable exception of 'Disgust', which was deliberately omitted from our real-time analysis. This omission is not due to oversight but stems from the initial training phase, where the FER-2013 dataset provided limited examples of the 'Disgust' expression. Consequently, our models were ineffective at recognizing and classifying 'Disgust' confidently in real-world scenarios. This limitation underscores a critical challenge in machine learning and deep learning projects: the performance of models is closely linked to the quality and breadth of the training data. The scarcity of 'Disgust' examples in the training set created a proficiency gap, which became evident during live webcam testing.

Our real-time analysis not only confirmed the models' ability to operate interactively but also revealed variable accuracy across different expressions. The models excelled in detecting 'Happy' and 'Surprise', which benefited from robust datasets during training. The correlation between the volume of expression data in the training set and the models' real-time detection accuracy was evident, reinforcing the maxim that 'data is king' in machine learning. Despite generally impressive performance under real-time conditions, the models showed limited adaptability to less represented expressions, such as 'Disgust', indicating a need for dataset enrichment and possibly model retraining. This analysis of real-time performance provides essential insights into the models' readiness for deployment in real-world scenarios, where a variety of expressions and unpredictability are standard.

In summary, while the models demonstrated notable real-time performance, their inability to detect 'Disgust' highlights an important area for future research and development. Enhancing the dataset to include underrepresented expressions and refining the models to improve their detection capabilities across a broader spectrum of expressions will be crucial. This approach will ensure that the models are more robust and capable of handling the diverse range of facial expressions encountered in real-world applications.

5. Conclusion

In this study, we explored the efficacy of various ResNet architectures—ResNet18, ResNet34, and ResNet50—in the context of facial expression recognition, leveraging the FER-2013 dataset for model training and evaluation. Our findings highlight several key insights into the performance, efficiency, and practical applicability of these models.

Firstly, the experimental results demonstrated that while deeper networks like ResNet50 initially showed superior performance, simpler architectures such as ResNet18 quickly caught up and often surpassed them in terms of accuracy and efficiency. This underscores the importance of balancing model

complexity with computational cost, especially for real-time applications where resource efficiency is paramount.

Secondly, our comprehensive analysis revealed significant variability in the models' ability to recognize different facial expressions. Expressions such as 'Happy' and 'Surprise' were classified with high precision and recall, benefiting from robust representation in the training dataset. Conversely, expressions like 'Fear' and 'Sad' posed greater challenges, indicating the need for targeted data augmentation and further model refinement to improve performance on these less represented expressions.

Furthermore, the real-time evaluation using a standard webcam confirmed the practical viability of our models. While the models performed well in dynamic, everyday settings, their inability to reliably detect the 'Disgust' expression highlighted a critical area for improvement. The limited representation of 'Disgust' in the FER-2013 dataset led to a proficiency gap, emphasizing the crucial role of a balanced and comprehensive training dataset in achieving robust model performance.

Our analysis suggests several avenues for future research. Enhancing the dataset to include more diverse and representative samples of underrepresented expressions is essential. Additionally, further experimentation with hyperparameter tuning, extended training periods, and alternative architectures could yield significant performance improvements.

In conclusion, our study demonstrates the potential of ResNet architectures for real-time facial expression recognition, while also identifying important areas for further development. By addressing these challenges, we can enhance the robustness and accuracy of facial expression recognition systems, making them more effective for a wide range of real-world applications, including human-computer interaction, security, and health assessment.

References

[1] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua, "A convolutional neural network cascade for face detection," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 5325–5334. doi: 10.1109/CVPR.2015.7299170.

[2] Z. Ming, J. Chazalon, M. Muzzamil Luqman, M. Visani, and J.-C. Burie, "FaceLiveNet: End-to-End Networks Combining Face Verification with Interactive Facial Expression-Based Liveness Detection," in *2018 24th International Conference on Pattern Recognition (ICPR)*, Aug. 2018, pp. 3507–3512. doi: 10.1109/ICPR.2018.8545274.

[3] Y. K. Bhatti, A. Jamil, N. Nida, M. H. Yousaf, S. Viriri, and S. A. Velastin, "Facial Expression Recognition of Instructor Using Deep Features and Extreme Learning Machine," *Computational Intelligence and Neuroscience*, vol. 2021, p. e5570870, May 2021, doi: 10.1155/2021/5570870.

[4] Y. Gan, "Facial Expression Recognition Using Convolutional Neural Network," in *Proceedings of the 2nd International Conference on Vision, Image and Signal Processing*, in ICVISP 2018. New York, NY, USA: Association for Computing Machinery, Aug. 2018, pp. 1–5. doi: 10.1145/3271553.3271584.

[5] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 770-778). doi: 10.1109/CVPR.2016.90.

[6] B. K. Durga and V. Rajesh, "A ResNet deep learning based facial recognition design for future multimedia applications," *Computers and Electrical Engineering*, vol. 104, p. 108384, Dec. 2022, doi: 10.1016/j.compeleceng.2022.108384.

[7] B. Li and D. Lima, "Facial expression recognition via ResNet-50," *International Journal of Cognitive Computing in Engineering*, vol. 2, pp. 57–64, Jun. 2021, doi: 10.1016/j.ijcce.2021.02.002.

[8] N. Hajarolasvadi and H. Demirel, "Deep facial emotion recognition in video using eigenframes," *IET Image Processing*, vol. 14, no. 14, pp. 3536–3546, 2020, doi: 10.1049/iet-ipr.2019.1566.

[9] S. Li and W. Deng, "Deep Facial Expression Recognition: A Survey," *IEEE Trans. Affective Comput.*, vol. 13, no. 3, pp. 1195–1215, Jul. 2022, doi: 10.1109/TAFFC.2020.2981446.

[10] H. Ali, M. Hariharan, S. Yaacob, and A. H. Adom, "Facial Emotion Recognition Based on Higher-Order Spectra Using Support Vector Machines," *Journal of Medical Imaging and Health Informatics*, vol. 5, no. 6, pp. 1272–1277, Nov. 2015, doi: 10.1166/jmihi.2015.1527.

[11] S. Sawardekar and S. R. Naik, "Facial Expression Recognition using Efficient LBP and CNN," vol. 05, no. 06, 2018.

[12] Y. Lv, Z. Feng, and C. Xu, "Facial expression recognition via deep learning," in *2014 International Conference on Smart Computing*, Nov. 2014, pp. 303–308. doi: 10.1109/SMARTCOMP.2014.7043872.

[13] T. Debnath, Md. M. Reza, A. Rahman, A. Beheshti, S.

S. Band, and H. Alinejad-Rokny, “Four-layer ConvNet to facial emotion recognition with minimal epochs and the significance of data diversity,” *Sci Rep*, vol. 12, p. 6991, Apr. 2022, doi: 10.1038/s41598-022-11173-0.

[14] G. Zeng, J. Zhou, X. Jia, W. Xie, and L. Shen, “Hand-Crafted Feature Guided Deep Learning for Facial Expression Recognition,” in *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, Xi'an, China: IEEE Press, May 2018, pp. 423–430. doi: 10.1109/FG.2018.00068.

[15] L. Graumann, M. Duesenberg, S. Metz, L. Schulze, O. T. Wolf, S. Roepke, C. Otte, and K. Wingenfeld, “Facial emotion recognition in borderline patients is unaffected by acute psychosocial stress,” *Journal of Psychiatric Research*, vol. 132, pp. 131–135, Jan. 2021, doi: 10.1016/j.jpsychires.2020.10.007.

[16] M. Sambare. (2020) “FER-2013 Learn facial expressions from an image” [Online]. Available: https://www.kaggle.com/datasets/msambare/fer2013/data

[17] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *International Conference on Learning Representations (ICLR)*, San Diego, CA, 2015.

[18] T. Saito and M. Rehmsmeier, "The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets," *PLoS ONE*, vol. 10, no. 3, pp. 1-21, Mar. 2015.