ENEL 645 Final Project: Facial Recognition using Machine Learning Algorithms

Anish Patel
Department of Electrical and Software Engineering
anish.patel1@ucalgary.ca

Mohammed Atifkhan Pathan
Department of Electrical and Software Engineering
mohammedatifkhan.pat@ucalgary.ca

Momin Muhammad
Department of Electrical and Software Engineering
momin.muhammad@ucalgary.ca

Group 23

Satchy Karalasingham
Department of Electrical and Software Engineering
satchytan.karalasing@ucalgary.ca

Balkarn Gill
Department of Electrical and Software Engineering
balkarn.gill1@ucalgary.ca

Abstract

This project focuses in the field of Human-Computer Interaction (HCI) and uses the Facial Expression Recognition 2013 (FER-2013) dataset to classify five distinct emotions: Anger, Fear, Happiness, Sadness, and Surprise, using deep learning techniques. The objective was to develop a model that surpasses human performance in interpreting these emotional states from facial images. The methodology encompassed data preprocessing, selection of suitable machine learning algorithms, and optimization techniques such as fine-tuning, data augmentation, and oversampling. The project successfully met its goal, achieving an accuracy of approximately **71% on the test set**, which exceeds the reported human accuracy of about 65%[2]. This work contributes to the broader scope of affective computing and its applications in various industries, demonstrating the efficacy of the proposed approach in emotion recognition from facial images.

1. Introduction

Emotion recognition is a frontier in artificial intelligence research that bridges human emotion and computer comprehension. With facial recognition technology becoming increasingly integral in consumer devices like the Apple Vision Pro headset, the need for machines to engage empathetically with humans is apparent[1]. However, recognizing emotions from dynamic and natural facial expressions poses challenges due to the nuance and variety of human expressions.

The FER 2013 dataset, containing over 35,000 annotated facial images, serves as an excellent resource for training deep learning models. Automatic emotion detection from facial images holds practical significance, enriching user experience and aiding mental health diagnosis.

Our goal wasn't just high accuracy, but to surpass human performance, which averages around 65%[2]. Achieving this ambitious goal contributes to advancing affective computing theory and has many practical implications, such as enhancing user interfaces, improving mental health assessment tools, and creating immersive gaming experiences. Revolutionizing how machines interpret and respond to human emotions holds promise for empathetic and responsive technology.

2. Related Work

Emotion recognition from facial expressions is a rapidly evolving field within computer vision and machine learning, incorporating a variety of approaches to improve accuracy and efficiency. The advent of deep learning has significantly shifted the focus towards end-to-end learning techniques, particularly Convolutional Neural Networks (CNNs), due to their ability to automatically learn discriminative features directly from raw pixel data. CNN architectures like VGGNet, ResNet, and DenseNet have been adapted to emotion recognition tasks, showing superior performance on benchmark datasets.

Khanzada et al. (2023)[5] achieved a 75.8% accuracy on the FER2013 test set using advanced deep learning techniques like transfer learning, data augmentation, class weighting, and ensembling, showcasing their models in real-time mobile web applications. Białek, Matiolański, and Grega (2023)[6] focused on comparing CNN-based FER methods, highlighting the importance of dataset modifications, binary classification, and the use of efficient CNN architectures and ensemble models to improve speed and accuracy without extensive computational resources. They also tackled dataset imbalances and labeling inaccuracies by introducing improved datasets and employing binary models for emotion classification[6].

These contributions underline the importance of continuous innovation in dataset preparation, model architecture, and the application of ensemble and binary classification techniques[4][8]. The advancements in FER techniques not only enhance the performance of emotion recognition systems but also broaden their applicability in real-world settings, from mobile applications to supportive tools for human-computer interaction[10]. Future work in this area may explore further integration of multimodal data, attention mechanisms, and the development of more robust models capable of handling diverse and challenging real-world conditions[9].

3. Materials and Methods

3.1. Dataset and Preprocessing:

The FER-2013 dataset was employed, which consists of over 35,000 labeled grayscale facial images categorized into five emotion categories. Noted for its complexity and real-world application viability, this dataset presents a challenging but valuable resource for emotion recognition due to its somewhat imbalanced class distribution and the naturalistic setting of the images. The dataset used includes happy, sad, angry, and surprised classification images.

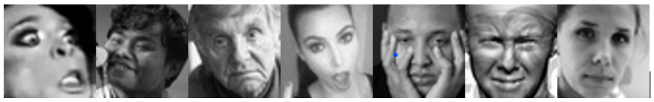


Figure 1: Photos Extracted from the FER dataset.

Splitting of the data into 'test' and 'validation' sets based on the original dataset filenames, with files with 'Private' in their filename assigned to the train set. The resulting split was **Train: 80%, Test: 10%, Validation: 10%**. For PyTorch compatibility, images were converted to grayscale (num_channel = 1), then to tensors. Data augmentation included scaling (as per model requirements), rotation (±7-15%), and random horizontal mirroring. Normalization was applied selectively, using ImageNet values for transfer learning models and dataset-specific values for Model #7. For transfer learning, 1-channel data was often converted to 3 channels by replication.

Two emotions were excluded due to an early project aiming to merge textual data, leaving 5 emotions: ‘0’: ‘Angry’, ‘1’: ‘Fear’, ‘2’: ‘Happy’, ‘3’: ‘Sad’, ‘4’: ‘Surprise’. This decision was retained despite a shift in project focus due to time constraints.

3.2. Feature Extraction and Augmentation:

For deep learning models, specifically CNNs, no explicit feature extraction step was required as these models inherently learn feature representations during training.

Due to the ‘Happy’ class having almost double the data as compared to the other 4 emotions, which were all fairly close in number of samples, we employed some oversampling to the rest of the data to have the same amount of samples as the highest sampled class.

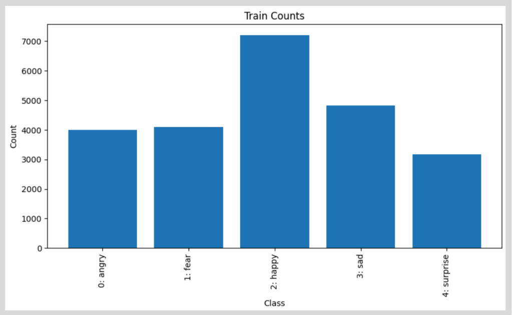


Figure 3: Proportion of data in each class prior to oversampling.

Data augmentation techniques like horizontal flipping, random rotations, and shifts were utilized to artificially expand the training dataset[11][12]. Creating a balanced set included performing much of the same augmentations as described in section 3.1, however, only the train set was balanced in such a way. The test and validation sets remained the same to preserve originality and gain true insights into model performance and generalizability. However, the models were run using both balanced and unbalanced dataset to validate gains due to oversampling.

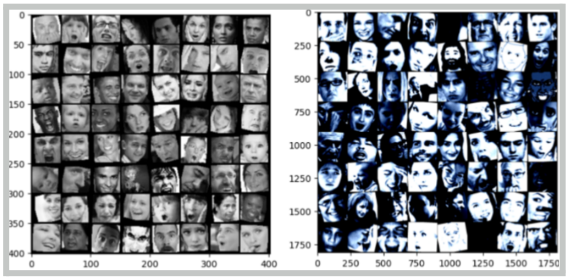


Figure 4: Augmented datasets. Without (left) and with Normalization (right)

3.3. Model Development and Selection:

Since the dataset was images and the problem focused on detection, the best models based on the literature suggested Convolutional Neural Networks. They can extract and leverage local correlation unlike FCNNs and hence are great for facial emotion detection that relies on these local correlations. Thus the focus was on **custom CNN architectures**, ranging from simple models to more complex (few convolutional layers to more convolutional layers).

Since the dataset is big for its kind, yet small in the world of big data, the next viable solution was using **Transfer Learning** as it is also standard practice when data is small. The models used ranged from models trained on ImageNet data and also included a model trained on the ‘vggface2’ dataset which is a model with over 3.3 million face images. There were a couple big face image datasets such as ‘vggface’ and ‘vggface2’ however, the pre-trained model weights for models trained on these datasets were difficult to acquire due to resources being expired or not being specifically for PyTorch. Nonetheless, models 10 and 11 (

using InceptionResNetV1) had a vggface2 pre-trained weight, which was the best transfer learning model we could find with relevant weights.

Table 1: Models used for training

Model	Description
Simple CNN (basic)	A basic Convolutional Neural Network (CNN) model used to establish a baseline for performance comparison.
Complex CNN (5 convolution layers)	Advanced CNN with five layers for enhanced feature extraction.
CNN (6 convolution layers)	CNN with six layers to test depth impact on performance.
ResNet50	Pre-trained model for emotion recognition. Use weights from its training on ImageNet
InceptionResNet	Another pre-trained model aimed at achieving superior performance as the weights used came from the model being trained on ‘vggface2’.
DenseNet	An additional model tested to explore different architectures.

Table 2: Hyperparameters used for tuning models

Hyperparameter	Values used
Image Dimensions	CNN: 48 x 48 x 1 ResNet50: 224 x 224 x 3 InceptionResNetV1: 160 x 160 x 3 ResNet50: 224 x 224 x 3 DenseNet: 64 x 64 x 3
Batch Sizes	32, 64, 128, 256
Epochs	20 to 80 (max)
Optimizers	Adam, SGD
Learning Rates	0.1, 0.01, 0.001, 0.0001
Step Sizes	5, 7, 9
Regularization	0.1, 0.01, 0.001
Activation function	ReLU after convolution layers, Softmax as last output layer
Loss Functions	Categorical Cross-Entropy, NLLLoss
Learning rate scheduler	StepLR, with and without ReduceLROnPlateau

After the selection process was completed the models were trained and tuned on various hyper parameters to improve them. The full breakdown of the models used and the best hyperparameters can be found under the model_info.xlsx file in the Github repository [15]. This file outlines the parameters for each model that resulted in the best outcome for training and validation. The file does not include every small change and combination tried however, the table below lists the hyper parameters used and different values tested with.

Due to limited gpu resources, even with cluster, we did not implement gridsearch or cross validation as the training times would greatly hinder progress.

3.4. Training and Validation Methodology:

The training loop for all models followed a consistent workflow: each epoch involved training on the dataset, followed by evaluation on both the training and validation sets. Losses calculated from the discrepancy between predictions and true labels guided parameter updates to enhance accuracy. The strongest model's weights were saved after each epoch, and upon completing all epochs, the top model was saved.

The CNN network architecture comprises multiple convolutional layers, usually followed by batch normalization, a rectified linear unit (ReLU) activation function, max pooling, and dropout for regularization. This is the standard flow but the values for each may vary. The depth of the network allows for complex feature extraction. The final layers consist of a pooling operation and a fully connected layer that maps the features to the five emotion classes. The forward function defines the data flow through the network, processing the input image through the convolutional layers and performing the necessary activation, pooling, and dropout steps. It then flattens the data and utilizes the fully-connected layer to generate the final class prediction.

For transfer learning models, pre-trained models were used which were initially trained on a vast dataset. The model is specialized by freezing the model's early layers to preserve their learned features, essential for generic image recognition. The model's final layers, specifically tailored to the original task's classes, are replaced with new layers designed for the new task's categories. These new layers are trainable, allowing the model to adapt to the nuances of the new dataset. The training process updates only these layers, utilizing the pre-trained features for improved performance.

Lastly, the loss function, optimizer and LR scheduler is defined for the training loop to run on the models defined. This methodology guided all the model training with fine tuning of the architecture along with hyperparameters.

4. Results and Discussion

Through the exploration of the models, many models failed during the hyperparameter tuning phases. During successful runs of the models, we saw validation accuracy scores of as low as 11.34% with unoptimized hyperparameters and data augmentation on the Simple model, to 75.29% on the InceptionResNet model with hyperparameters dialed in and tuned to the maximum of what the timeline allowed for.

The results obtained from the best hyperparameters for the respective models are as shown in the following section.

4.1. Results:

4.1a: Results from CNN Models

CNN_Base: This initial model served as a baseline for our experiments. With no pre-trained weights and a batch size of 32, the model achieved a best validation accuracy of 55.73% over 30 epochs. Data augmentation techniques, including random horizontal flip and rotation, were applied to the training set to improve generalization.

CNN_Base2: An extension of our baseline model, CNN_Base2, was trained for a longer duration, totaling 80 epochs. Similar data augmentation strategies were employed. This model demonstrated a slight improvement, attaining a validation accuracy of 56.46%.

Complex CNN (5convolution layers): Building upon the insights gained from our baseline experiments, this model incorporated a more sophisticated architecture with five convolution layers. Despite being trained for only 25 epochs, it significantly outperformed the previous models by achieving a validation accuracy of 70.51%. The same data augmentation techniques were used to ensure consistency in training conditions. In the next iteration of the complex CNN model we extended training to 80 epochs. The validation accuracy was still 70.51%, suggesting the additional epochs did not benefit the model training. Following models were trained with early stopping.

CNN_Layer (6 layers, without normalization): This configuration achieved a validation accuracy of 71.4% over 45 epochs, showing a slight improvement over the 5-layer complex CNN. No normalization meant the data set was not normalized with the std and mean values for the FER dataset. The same model was run after balancing the class distribution and the accuracy remained almost identical. Suggesting that the oversampling did not provide much gains in accuracy. This is surprising and could be attributed to other things that might not be considered.

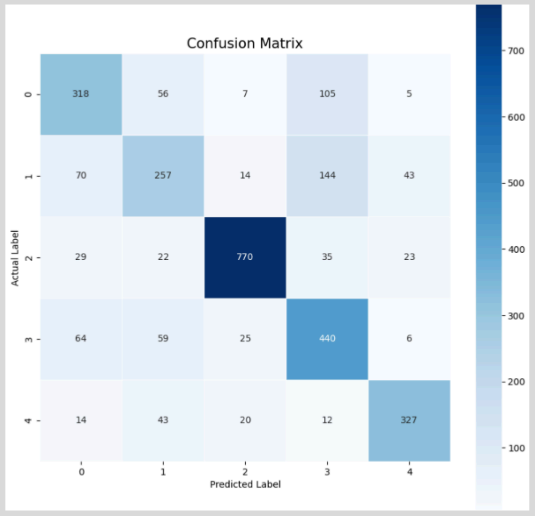
CNN_Layer (6 convolution layers, with normalization, unbalanced): Incorporating data normalization using the std and mean values calculated from the dataset with six convolution layers, this model achieved a validation accuracy of 70.4% over 80 epochs. This is very close to the models before this, hence suggesting that normalization might not be a big factor for improvement either.

4.1b: Results from Transfer Learning

ResNet50: Utilizing the ResNet50 architecture with transfer learning did not yield the expected improvements, achieving a lower validation accuracy of 45.72% over 30 epochs.

ResNet50_v2: A variant of the ResNet50 model, with different hyperparameters, as defined in the model_info.xlsx sheet[15], saw a slight improvement in validation accuracy to 49.07%.

InceptionResNet: Markedly, this model achieved the highest validation accuracy among those listed at 75.29% over 35 epochs, indicating a strong performance albeit with an overfitting issue.



.

Figure 5: Confusion Matrix for InceptionResNet

As seen in the confusion matrix, and pretty much for all of the models, the happy class since being the majority class usually has the highest accuracy, even after balancing since the test set remained with the original distribution.

InceptionResNet (Balanced dataset): Adjusting for dataset balance, this variant of InceptionResNet achieved a validation accuracy of 73.99%, slightly lower than its counterpart but potentially more generalizable.

DenseNet: This model reported the lowest validation accuracy of 33.9% over 30 epochs, suggesting it might not be as effective for this particular task or that further optimization is needed.

4.2. Discussion:

The results of the facial expression recognition models present insightful outcomes and merit a comprehensive discussion on several fronts.

4.2a Model Architecture and Performance

The deployment of Convolutional Neural Networks (CNNs) yielded varying degrees of success in emotion recognition tasks. The baseline CNN model (CNN_Base) and its immediate successor (CNN_Base2), which differed primarily in the number of epochs, demonstrated modest validation accuracies of 55.73% and 56.46%, respectively. The relatively low performance of these models underscores the challenges inherent in facial expression recognition, particularly when dealing with nuances and variations in facial features and expressions.

In contrast, models with increased complexity, such as the Complex CNN with 5 convolution layers and the CNN_Layer with 6 convolution layers (both with and without normalization), exhibited superior performance, achieving validation accuracies above 70%. This increase underscores the hypothesis that deeper architectures can capture more abstract features critical for distinguishing subtle differences in facial expressions. These were the best models we were able to generate.

The InceptionResNetV1 model was trained on the ‘vggface2’ dataset, hence why it was able to do much better for this task. This highlights the strength of transfer learning when similar features can be leveraged across datasets. However, the performance was undermined by overfitting.

4.2b Data Augmentation and Preprocessing

Our experimentation with various data augmentation techniques, including random horizontal flips and rotations, played a significant role in enhancing model generalization. Notably, the implementation of normalization, particularly for models involving transfer learning, did not follow the conventional use of color channels. Instead, these models were adapted to grayscale inputs, which is an unconventional approach given the color-dependent pre-training of networks like ResNet50 and DenseNet. This unconventional approach may have contributed to the lower performance of the transfer learning models, with ResNet50 and ResNet50_v2 achieving validation accuracies under 50%. However, it also raises questions about the potential of transfer learning in grayscale domains, warranting further investigation.

4.2c Overfitting Concerns

The highest validation accuracy was observed with the InceptionResNet (Overfitting) model at 75.29%, which, as the name suggests, may have been overfitted to the training data. Overfitting was addressed in the subsequent InceptionResNet variant using a balanced dataset, resulting in a slight reduction in accuracy but potentially enhancing the model's ability to generalize to unseen data.

Table 3: Top 5 Validation Accuracies Achieved

Model	Validation Accuracy
InceptionResNet	75.29%
InceptionResNet (Balanced)	73.99%
CNN_Layer (without norm.)	71.40%
Complex CNN (5 layers)	70.51%
CNN_Layer (with norm.)	70.40%

Overall, our model training was successful, with our best validation accuracy reaching over 75%. The best validation accuracy, with no overfitting of the training data was 71.4% which was still above the 65%[2] that we were aiming for at the beginning of the study.

5. Conclusion

5.1 Implications for Affective Computing

The implications of these findings are broad and significant for the field of affective computing. A model that can accurately recognize human emotions from facial expressions holds promise for enhancing human-computer interaction, mental health monitoring, and providing nuanced feedback within AI-driven systems. There is an emphasis in the modern era to detect emotions of people wearing VR headsets, or using their phone’s internal camera.

The variation in model performance also highlights the critical importance of selecting suitable architectures and data processing techniques tailored to the specific characteristics of the task at hand.

5.2 Future Directions

Further research is necessary to optimize the balance between model complexity and generalization capabilities. Strategies might include exploring alternative preprocessing methods, such as employing different color spaces or combining grayscale and color data. Additionally, integrating attention mechanisms could provide models with the ability to focus on relevant features within facial expressions, potentially improving recognition accuracy. We could find pre-trained weights for models trained on facial images as that would greatly empower the transfer learning. We can apply fine-tuning to further refine the transfer learning models by unfreezing and adjusting the weights of the earlier layers with a very low learning rate, ensuring the model is finely tuned to the specific characteristics of the new task without overfitting.

In conclusion, the exploration of emotion recognition through deep learning techniques has provided valuable insights into the capabilities and limitations of various CNN architectures and data augmentation techniques. While deeper and more complex networks tend to yield better results, they also pose risks of overfitting, emphasizing the need for careful model selection and validation. The continued refinement of these models will likely drive advancements in the field of affective computing, with wide-reaching implications for technology that is increasingly sensitive to human emotion.

6. References

[1] Z. Zhang, L. Giménez Giménez Mateu, and J. M. Fort, "Apple Vision Pro: a new horizon in psychological research and therapy," Front. Psychol., vol. 14, article 1280213, Nov. 2, 2023. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10651722/

[2]M. Mukhopadhyay, S. Pal, A. Nayyar, P. Choudhury, et al., "Facial Emotion Detection to Assess Learner's State of Mind in an Online Learning System," February 2020. https://www.researchgate.net/publication/339336771_Facial_Emotion_Detection_to_Assess_Learner's_State_of_Mind_in_an_Online_Learning_System

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, Las Vegas, NV, USA, 2016. https://arxiv.org/pdf/1512.03385.pdf

[4] I. Goodfellow, Y. Bengio, and A. Courville, "Deep Learning," MIT Press, 2016. https://www.deeplearningbook.org/

[5] A. Khanzada, C. Bai, F.T. Celepcikay, "Facial Expression Recognition with Deep Learning: Improving on the State of the Art and Applying to the Real World," Stanford University - CS230 Deep Learning, 2023. https://cs230.stanford.edu/projects winter 2020/reports/32610274.pdf

[6] C. Białek, A. Matiola ́nski, M. Grega, "An Efficient Approach to Face Emotion Recognition with Convolutional Neural Networks," Electronics 2023. https://doi.org/10.3390/electronics12122707.

[7] A. Dhall, R. Goecke, S. Lucey, and T. Gedeon, "Static facial expression analysis in tough conditions: Data, evaluation protocol and benchmark," in 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), Barcelona, Spain, 2011, pp. 2106-2112. https://ieeexplore.ieee.org/document/6130508

[8] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A Comprehensive Survey on Transfer Learning," arXiv:1911.02685 [cs.LG], Nov. 2019. Available: https://arxiv.org/abs/1911.02685

[9] S. Zafeiriou, C. Zhang, and Z. Zhang, "A survey on face detection in the wild: Past, present and future," Computer Vision and Image Understanding, 2015. https://www.sciencedirect.com/science/article/pii/S1077314215000727

[10] T. Kanade, J. F. Cohn, and Y. Tian, "Comprehensive database for facial expression analysis," in Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580), Grenoble, France, 2000, https://ieeexplore.ieee.org/document/840611

[11] C. Pramerdorfer and M. Kampel, "Facial Expression Recognition using Convolutional Neural Networks: State of the Art," arXiv:1612.02903 [cs.CV], Dec. 2016. Available: https://arxiv.org/abs/1612.02903

[12] M. S. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel, and J. Movellan, "Automatic recognition of facial actions in spontaneous expressions," Journal of Multimodal User Interfaces, vol. 3, no. 1-2, pp. 96-107, 2010. https://inc.ucsd.edu/mplab/46/media/Bartlett_JMM06.pdf

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in Advances in neural information processing systems, Lake Tahoe, NV, USA, 2012, pp. 1097-1105. https://papers.nips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[14] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436-444, 2015. https://www.nature.com/articles/nature14539

[15] A. Patel, A. Pathan, B. Gill, S. Karalasingham, M. Muhammad, "ENEL645-FinalProject," GitHub repository, 2023. [Online]. Available: https://github.com/anishpatel321/ENEL645-FinalProject