Machine Learning for Physicists

Final Report

DenseNet for Facial Emotion Recognition

2024

Department of Physics

Technische Universität Dortmund

Rene-Marcel Lehner
Department of Physics
TU Dortmund
Germany
rene-marcel.lehner@udo.edu

Laurin Hagemann
Department of Physics
TU Dortmund
Germany
laurin.hagemann@udo.edu

Abstract

This study explores the use of DenseNet architectures for facial emotion recognition (FER) using the FER2013 dataset. The DenseNet model achieved an accuracy of 57%, outperforming a traditional CNN model's 34%. While DenseNet improved feature reuse and mitigated overfitting, challenges with data quality and the reliance on static images persisted. Future work should integrate temporal data and advanced augmentation techniques to enhance model performance. Ethical considerations, including transparency and fairness, are also crucial for real-world applications.

Contents

1 Introduction

Human-machine interaction has become increasingly significant. Facial expressions serve as a strong indicator of emotions, making facial emotion recognition (FER) an effective method for accurately detecting human emotions. FER can assist in decision-making, enhance customer service through automated mood detection, and be used in security for border control and terrorism prevention [1]. As noted by [2], "detecting and classifying human emotional expressions [...] [is] used in a vast range of applications, such as education, healthcare, or public safety". Additionally, [3] emphasizes that "human-machine communication can be substantially enhanced by the inclusion of high-quality real-time recognition of spontaneous human emotional expressions".

Research Objective

Despite advancements in neural network architectures, facial emotion recognition (FER) remains a challenging field with relatively low accuracy rates, typically around 80% depending on the dataset and label quality [4]. This study aims to train a DenseNet architecture on FER data and compare its performance with a simpler alternative method. By doing so, we seek to gain insights on the improvements offered by DenseNet and reveal potential strategies for enhancing FER accuracy and reliability.
Using a dataset comprised of low-quality images with various face orientations (frontal, side, and top views), we implemented a convolutional neural network to classify individual facial images into one of seven emotional expressions: angry, disgust, fear, happy, neutral, sad, and surprise.

2 Dataset

The dataset consists of 48x48 pixel grayscale images categorized into 7 classes. The distribution of the images across these classes is as follows:

Angry	Disgust	Fear	Happy	Neutral	Sad	Surprise	Total
7532	815	7705	13739	9490	9242	5717	54240

Table 1: Distribution of images across different emotional classes. A visualization can be seen in Figure 1.

The dataset is divided into a training set and a test set with an 80/20 split. The images show high variability in age, gender, ethnicity, facial features such as beards and glasses, non-realistic faces, and the presence of hands (as shown in Figure 11). Despite the relatively poor label quality, larger and better-labeled datasets are not publicly accessible. The dataset is licensed under the CC BY-NC-SA 4.0 license.

Pre-processing involves standardizing the image size, converting to grayscale, and normalizing pixel values to a range of 0–1.

Tackling the imbalanced dataset and overfitting

To address the imbalanced dataset and varying image quality, several strategies are used. Class labels are encoded into a one-hot format with `to_categorical` from `tensorflow.keras.utils` [5]. To balance the class distribution, `class_weights` from `sklearn.utils` [6] is applied.

The Adam optimizer is chosen for its adaptive learning rate properties and is used alongside the `categorical_crossentropy` loss function for multi-class classification. `EarlyStopping` and `ReduceLROnPlateau` callbacks are employed to prevent overfitting and dynamically adjust the learning rate, respectively, thereby enhancing the model's ability to generalize to unseen data.

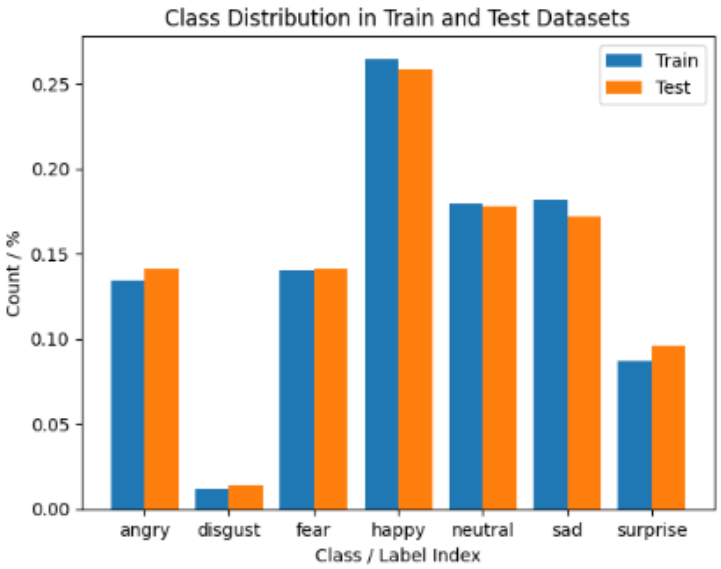

Figure 1: Class distribution of the FER 2013 dataset from [7].

3 Main method

Facial emotion recognition often involves dealing with faces that are not oriented directly towards the camera. DenseNets are particularly effective in such scenarios due to their ability to enhance feature propagation and reuse, which is beneficial when working with low-resolution images. Traditional CNNs, while powerful for image recognition tasks, lack the ability to reuse and retain features across layers, which appear to be crucial for capturing the nuanced details necessary for accurate emotion detection. Given the strengths of CNNs in image recognition, using a DenseNet leverages these strengths while addressing their limitations, making it a fitting choice for this complex task.

3.1 DenseNet

DenseNet, or Densely Connected Convolutional Network, stands out from traditional Convolutional Neural Networks (CNNs) by introducing direct connections between all layers within a dense block. This architecture ensures that each layer receives the feature maps from all preceding layers, promoting feature reuse and improving gradient flow. Unlike standard CNNs, which only connect layers sequentially, DenseNets mitigate the vanishing gradient problem and enhance feature propagation, making them particularly effective for tasks involving low-resolution images and complex feature extraction [8]. The extensive reuse of features helps to:

- **Enhanced Gradient Flow**: The direct connections between layers facilitate improved gradient propagation during back-propagation, which helps mitigate the vanishing gradient problem that is particularly prevalent in deeper networks.
- **Improved Feature Extraction**: By merging feature maps from all preceding layers, DenseNets can extract and utilize a comprehensive set of features from low-quality images, capturing fine details that might be missed by conventional convolutional networks.
- **Diverse Feature Access**: The dense connectivity enables each layer to access features extracted at multiple levels of abstraction. This feature allows the network to identify facial expressions from various angles and orientations by using a broad spectrum of features.
- **Enhanced Generalization**: The feature reuse mechanism in DenseNets reduces the risk of overfitting, which is essential when the training data includes faces with diverse orientations. By utilizing features from multiple perspectives, DenseNets can generalize more effectively to unseen poses and lighting conditions.

Thus, in this report, we use a Densely Connected Convolutional Network (DenseNet) as described by [8]. The DenseNet design is visualized in Figure 2.

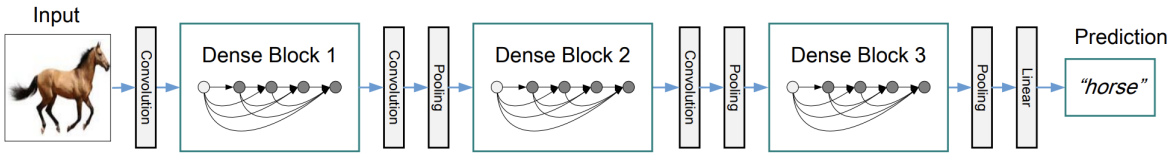

Figure 2: A deep DenseNet with three dense blocks. The layers between two adjacent blocks are referred to as transition layers and change feature-map sizes via convolution and pooling. [9]

3.2 Hyperparameter Optimization

In this project the Optuna framework [10] is used for hyperparameter optimization, making use of the Tree-structured Parzen Estimator method. Each set of hyperparameters is evaluated

over 20 epochs, and the loss is calculated based on the test dataset. A total of 40 optimization steps are conducted.
The optimization process does not identify optimal values for the regularization parameters, as indicated by the contour plots in Figures 3 and 4 in the appendix. These figures show that the process primarily explores lower values for the regularization parameters without converging on a specific optimal point.

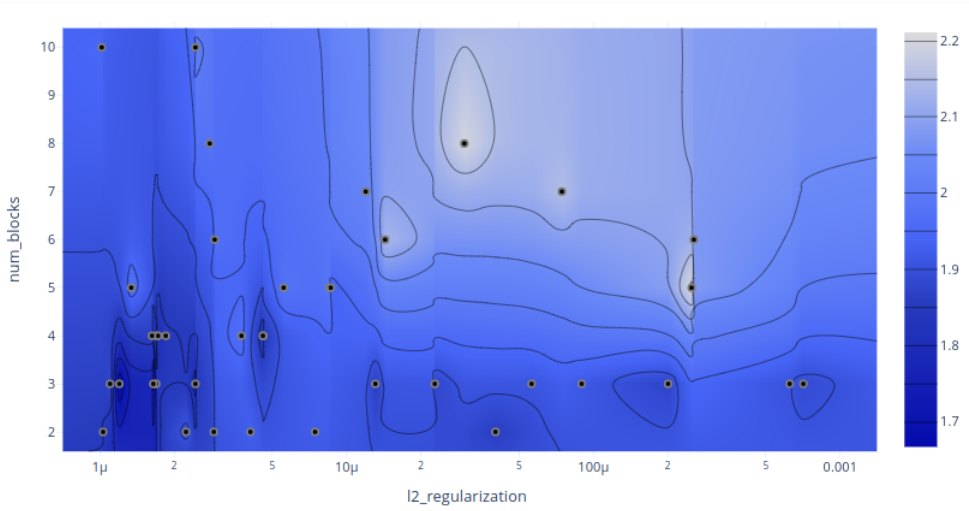

Figure 3: Contour plot of the number of blocks of our DenseNet (`num_blocks`) vs L2-regularization parameter.

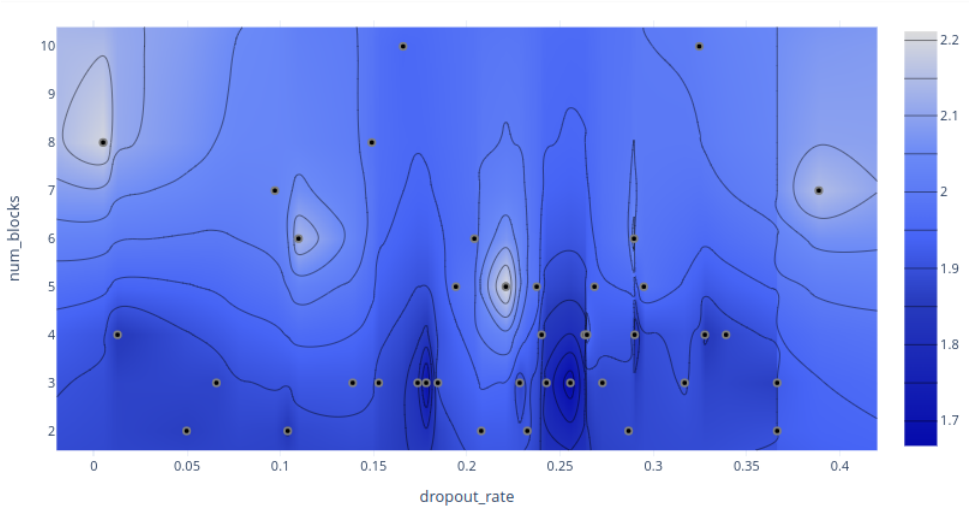

Figure 4: Contour plot of the number of blocks of our DenseNet (`num_blocks`) vs Dropout rate after each `DenseBlock`.

Optimal Hyperparameters

num_blocks	min_layers	max_layers	growth_rate
3	4	26	16
reduction	dropout_rate	l2_regularization	
0.4264	0.2554	1.1993e-06	

Table 2: Optimal hyperparameters for the DenseNet model obtained using Optuna.

- num_blocks: The number of dense blocks in the DenseNet architecture. Dense blocks are the core components that connect layers densely.
- min_layers: The minimum number of layers within each dense block. This parameter defines the lower bound for the number of layers in a block.
- max_layers: The maximum number of layers within each dense block. This parameter sets the upper limit for the number of layers in a block.
- growth_rate: The number of filters added per dense block layer. It determines how the number of channels increases in each block.
- reduction: The reduction factor used in transition layers, which reduces the number of feature maps between dense blocks to control model complexity.
- dropout_rate: The dropout rate applied to each layer, which helps prevent overfitting by randomly setting a fraction of input units to zero during training.
- l2_regularization: The L2 regularization factor applied to the weights to prevent overfitting by penalizing large weights.

Initial training without dense layers or regularization shows a strong tendency for overfitting (Figure 5).
Figure 6 is a visualization of all the hyperparamters used for optimization.

3.3 Results

The training history with the optimal set of hyperparameters is shown in Figure 7. The model state corresponding to the highest validation accuracy (0.35) is restored for evaluation purposes. Figure 8 displays several sample images from the test dataset, along with their predicted classifications and true labels.
The confusion matrix for the optimal hyperparameters is presented in Figure 9. It is observed that the model tends to classify expressions such as anger, fear, and neutral as happy, with a particularly poor performance in detecting angry expressions.
To further illustrate the classification tendencies for happy and angry expressions, Figure **??** shows their probability distributions as generated by the model's one-hot encoding. True happy expressions are classified accurately, but many other emotions are also misclassified as happy, especially angry expressions, which are often confused with happy. Angry expressions are the worst classified, as they are not distinctly recognized and are mostly confused with other emotions.

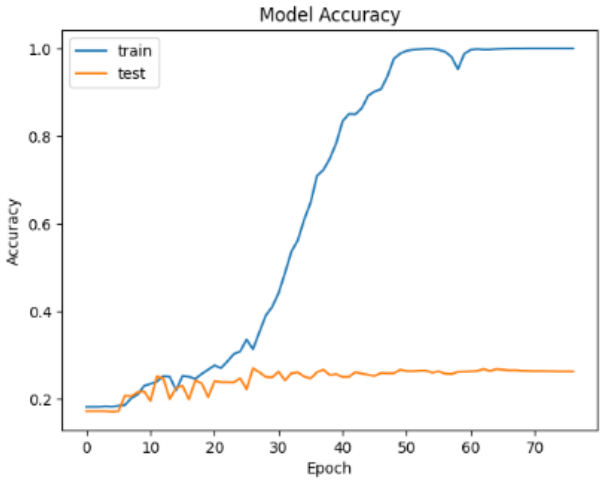

Figure 5: Custom DenseNet implementation trained on FER dataset without added dense layers, and with no regularization.

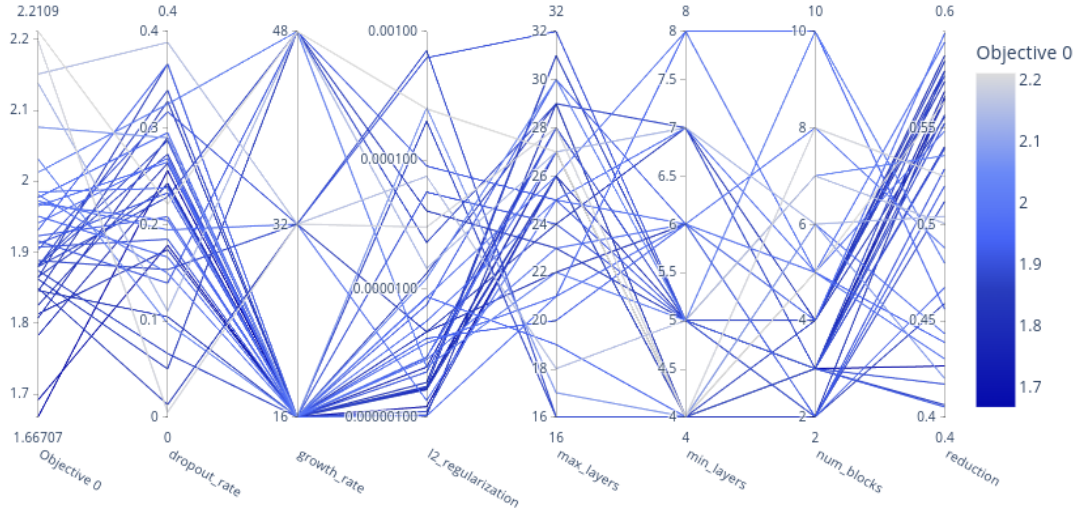

Figure 6: Plot of all hyper-parameters we optimized for, where "Objective 0" is the loss from misclassification.

Figure 11 highlights the top 20 worst predictions by the model, identified by the lowest probability for the true label. Some labels are clearly questionable. Additionally, duplicates of slightly different images yield varying predictions, including instances where comic figures are present, indicating challenges related to the dataset's quality.

Classification Report

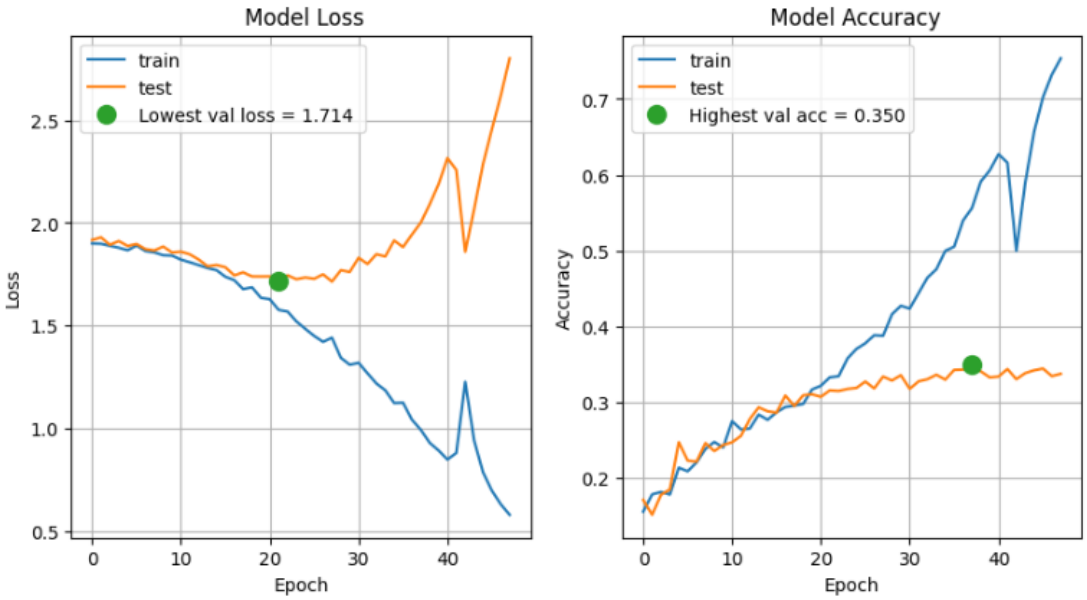

Figure 7: Model training history with optimal hyper-parameters.

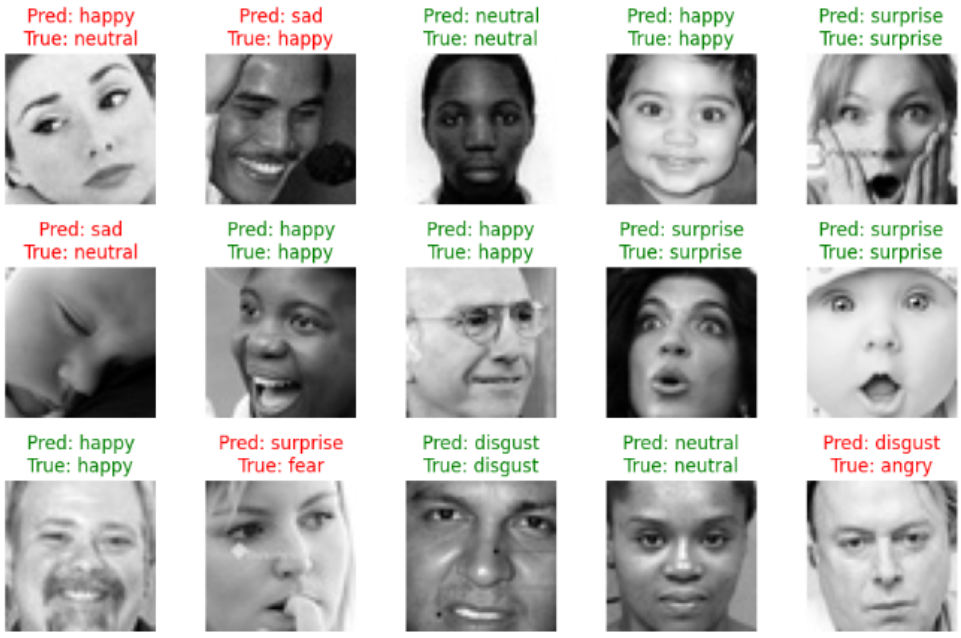

Figure 8: Classifications and true labels for randomly chosen sample images from the test-dataset.

4 Alternative Method

4.1 CNN

In addition to the primary DenseNet architecture, an alternative method using a traditional Convolutional Neural Network (CNN) architecture is explored. CNNs are very effective in

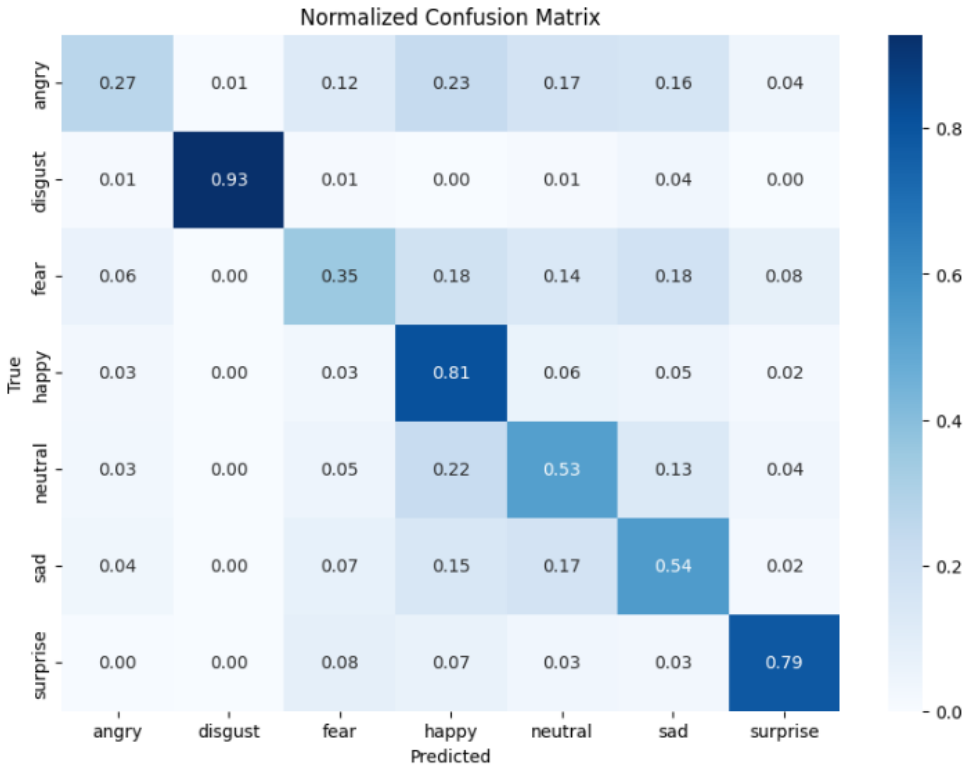

Figure 9: Confusion matrix with the optimal set of hyper-parameters.

various visual data tasks and serve as a natural predecessor to DenseNets. The objective is to determine whether a simpler CNN can perform competitively with the more complex DenseNet architecture.

The alternative CNN architecture consists of five convolutional layer blocks, starting with 16 filters and increasing to 256 filters in the fifth block, each block doubling the number of filters from the previous one. Each convolutional layer employs a 3x3 kernel, followed by batch normalization and max pooling. Following these layers, a fully connected layer with 512 neurons and dropout is applied, leading to the final dense layer for classification.

4.2 Results

The alternative CNN achieves an overall accuracy of 34%, which is significantly lower than the 57% accuracy achieved by the DenseNet. Despite the seemingly competitive overall accuracy, a closer look at the confusion matrix and class-specific accuracies reveal significant shortcomings. The confusion matrix shows poor diagonal alignment, indicating that the model struggles to correctly classify emotions. Furthermore, the model entirely fails to predict the "disgust" class, likely due to its sparse representation in the dataset. The higher accuracy is mainly attributed to the model's tendency to predict the most representative classes, such as "angry" and "surprise."

Despite achieving a somewhat competitive overall accuracy, the alternative CNN fails to generalize effectively across different classes. The lack of reliable predictions indicates that

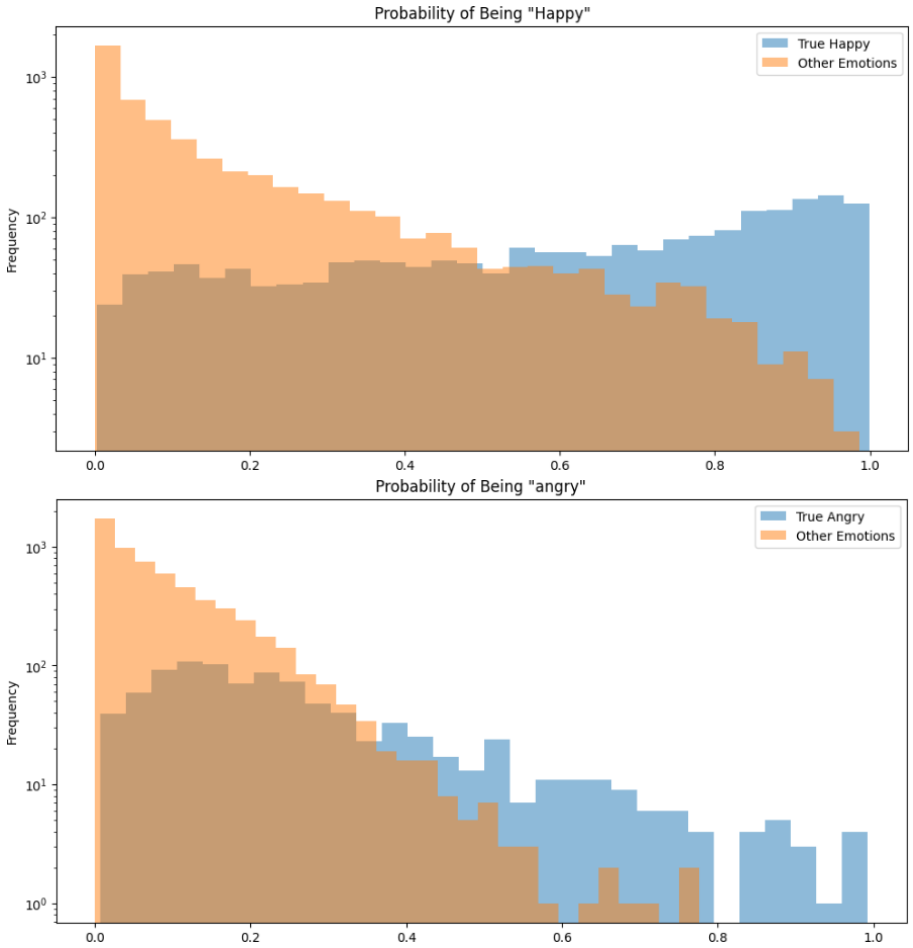

Figure 10: Probability distribution of "happy" classification (top) and "angry" classification (bottom).

the simpler CNN architecture is insufficient for the complexity of facial emotion detection. It is important to note that no hyperparameter optimization was performed for this alternative method. Instead, hyperparameters were chosen based on experience, aiming for a balance between model size and complexity. The chosen complexity is close to the optimal hyperparameters for the DenseNet to ensure a fair comparison. Nonetheless, these results demonstrate that the DenseNet architecture provides significant improvements over a classical CNN in terms of feature extraction and generalization.
The appendix provides predictions of various emotions/faces, similar to the figures for the DenseNet model, as well as a non-normalized confusion matrix to display the discrepancies in class predictions.

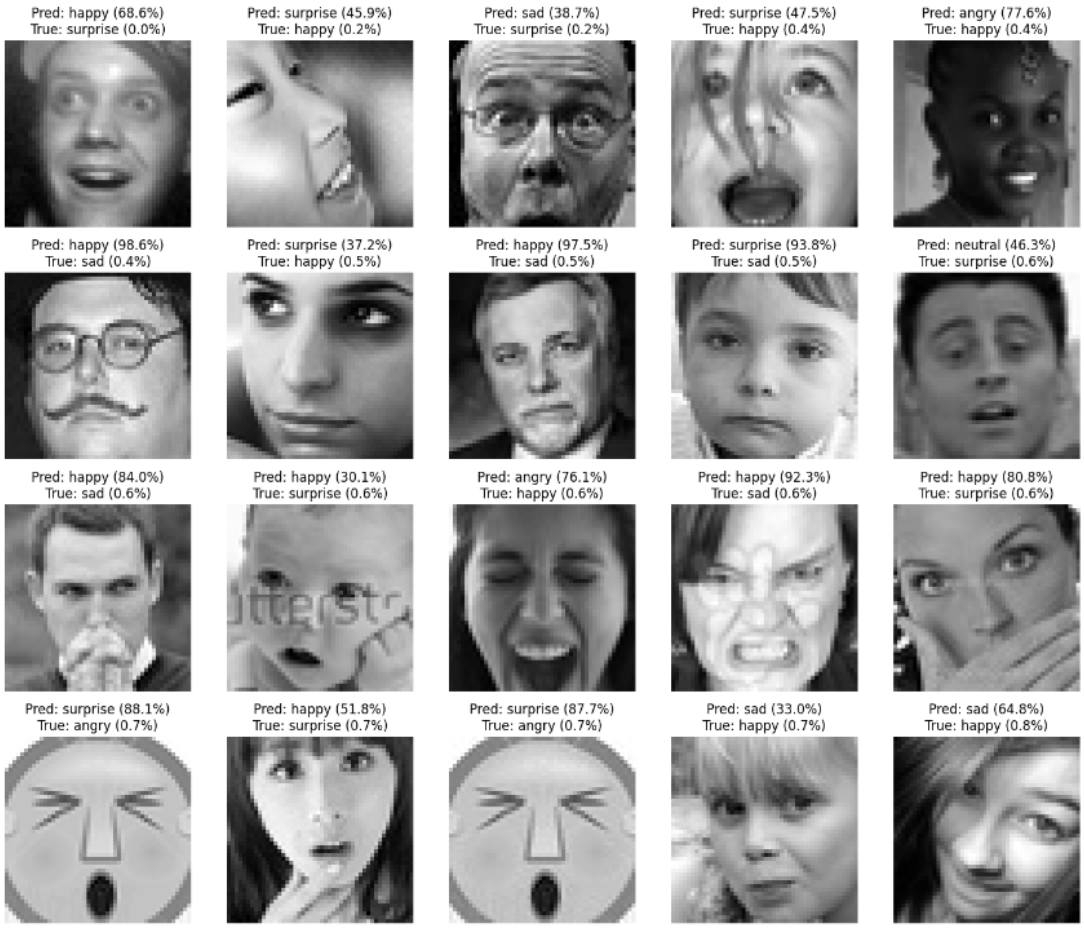

Figure 11: Top 20 worst predictions (lowest probability for the true label), with the predicted label and the true label and their respective percentages as classified by our model.

5 Discussion

The performance of the DenseNet model in our study, with an overall accuracy of 57%, demonstrates significant improvements over the simpler CNN architecture, which achieved an accuracy of 34%. These results highlight the advantages of DenseNet's dense connectivity, which enhances feature reuse and gradient flow, thereby improving the network's ability to capture and utilize features from low-resolution images. However, it is important to acknowledge that the model's accuracy, particularly for certain emotions like anger and fear, remains suboptimal. The confusion matrix indicates a tendency to misclassify these emotions as happy, which suggests that while DenseNet improves performance, there are still challenges in accurately distinguishing between subtle differences in facial expressions.

A notable observation is the divergence of training and test accuracy after 20 epochs, as shown in Figure 7. This suggests that with increased training epochs during the hyperparameter optimization process, better results might be achievable. The current optimization procedure,

Class	Precision	Recall	F1-Score
0 (Angry)	0.55	0.27	0.36
1 (Disgust)	0.85	0.93	0.89
2 (Fear)	0.48	0.35	0.41
3 (Happy)	0.62	0.81	0.71
4 (Neutral)	0.51	0.53	0.52
5 (Sad)	0.53	0.54	0.54
6 (Surprise)	0.68	0.79	0.73
Macro Avg	0.60	0.60	0.59
Weighted Avg	0.56	0.57	0.56
Accuracy		**0.57**	

Table 3: Classification Report: Precision, Recall, and F1-Score for each class.

Class	Precision	Recall	F1-Score
0	0.38	0.15	0.22
1	0.00	0.00	0.00
2	0.19	0.47	0.27
3	0.73	0.49	0.59
4	0.30	0.13	0.18
5	0.25	0.41	0.31
6	0.53	0.32	0.40
Macro Avg	0.34	0.28	0.28
Weighted Avg	0.41	0.34	0.34
Accuracy		**0.34**	

Table 4: Classification report for the alternative CNN model.

limited to 20 epochs, may have prematurely converged on suboptimal hyperparameters. Figures 3 and 4 support this by indicating that only low block count optima were found, whereas higher block counts paired with greater regularization needs might yield better performance.

The dataset poses significant challenges due to its small size and variability, including cases where faces are half-concealed by hands, as seen in some of the worst predictions in Figure 11. Data augmentation techniques, such as mirroring, rotating, and adjusting brightness, could help generalize the model to these edge cases. However, certain augmentations, like cutting off the lower half of the image, cannot be automated and were beyond the scope of this report.

Our evaluation demonstrated that the DenseNet model, with the use of L2 regularization and dropout layers, performed better in overcoming overfitting compared to an unregularized version. The training history and validation metrics show that overfitting was controlled and mitigated, although the model still showed some challenges in accurately predicting certain emotions, such as "anger" and "fear" (refer to Figure 9). It is also important to address the discrepancy between the validation accuracy during training (35%) and the final classification report (57%). This could be attributed to several factors, including possible data leaks between training and validation sets due to the unknown state of the dataset.

A comparison with a traditional CNN model revealed that while the CNN achieved a close overall accuracy, it failed to generalize effectively across all classes. The confusion matrix of the CNN showed significant misclassifications, particularly for sparsely represented classes like "disgust." This underlines the advantage of using DenseNet architectures, which can appear to better handle the complexity and nuances of facial emotion detection tasks.
Furthermore, it is essential to consider aspects of transparency, control, fairness, and proportionality when deploying FER systems in real-world applications. According to the European Data Protection Supervisor [1], FER systems should be transparent about how data is collected, processed, and used, ensuring that users have control over their personal data. Ensuring fairness and avoiding biases in FER systems are crucial to prevent discrimination and uphold ethical standards.

6 Conclusion

In conclusion, this study demonstrates the potential of DenseNet architectures to enhance facial emotion recognition tasks by leveraging their dense connectivity to improve feature propagation and reuse. The DenseNet model outperforms a traditional CNN, achieving higher overall accuracy and better generalization across different facial expressions. However, the limitations related to static image data, dataset quality, and the inherent complexity of emotion recognition from facial expressions are evident.
Future research should focus on integrating temporal information to capture the dynamic nature of emotions, as well as improving dataset quality and labeling consistency. The inclusion of video data or sequences of images could provide a more accurate representation of emotional changes over time, addressing the current model's inability to detect relative changes in facial expressions. Additionally, leveraging advanced data augmentation techniques, such as generating synthetic images and varying face orientations, can help the model generalize better to diverse facial expressions.
Techniques like attention mechanisms and transformers could further enhance the model's ability to capture details of facial expressions. Expanding the hyperparameter optimization process to include a wider range of parameters and using cross-validation can also improve the model's performance and reduce overfitting. By addressing these aspects, the model's accuracy and robustness can likely be enhanced, making it more effective for real-world applications in emotion detection.
Overall, while significant progress has been made, the field of facial emotion recognition remains challenging, with ample opportunities for further improvements and innovations. By addressing these challenges, we can develop more reliable and accurate systems for a wide range of applications, from healthcare to security and beyond.

Reference

[1] E. D. P. Supervisor. *Facial Emotion Recognition*. Accessed: 2024-07-30. 2021. URL: https://www.edps.europa.eu/system/files/2021-05/21-05-26_techdispatch-facial-emotion-recognition_ref_en.pdf.

[2] A.-L. Cîrneanu, D. Popescu, and D. Iordache. "New Trends in Emotion Recognition Using Image Analysis by Neural Networks, A Systematic Review". In: *Sensors* **23**.16 (2023). ISSN: 1424-8220. DOI: 10.3390/s23167092. URL: https://www.mdpi.com/1424-8220/23/16/7092.

[3] M. Lukac et al. "Study on emotion recognition bias in different regional groups". In: *Scientific Reports* **13**.1 (2023), p. 8414. DOI: 10.1038/s41598-023-34932-z. URL: https://www.nature.com/articles/s41598-023-34932-z.

[4] Papers with Code. *Facial Expression Recognition on FER2013*. Accessed: 2024-07-30. 2024. URL: https://paperswithcode.com/sota/facial-expression-recognition-on-fer2013.

[5] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: http://tensorflow.org/.

[6] F. Pedregosa et al. "Scikit-learn: Machine learning in Python". In: *Journal of machine learning research* **12**.Oct (2011), pp. 2825–2830.

[7] Kaggle. *FER-2013 – Learn facial expressions from an image*. URL: https://www.kaggle.com/datasets/msambare/fer2013.

[8] G. Huang et al. *Densely Connected Convolutional Networks*. 2018. arXiv: 1608.06993 [cs.CV]. URL: https://arxiv.org/abs/1608.06993.

[9] G. Huang et al. "Densely Connected Convolutional Networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), pp. 4700–4708.

[10] T. Akiba et al. *Optuna: A Next-generation Hyperparameter Optimization Framework*. 2019. arXiv: 1907.10902 [cs.LG]. URL: https://arxiv.org/abs/1907.10902.

A Plots

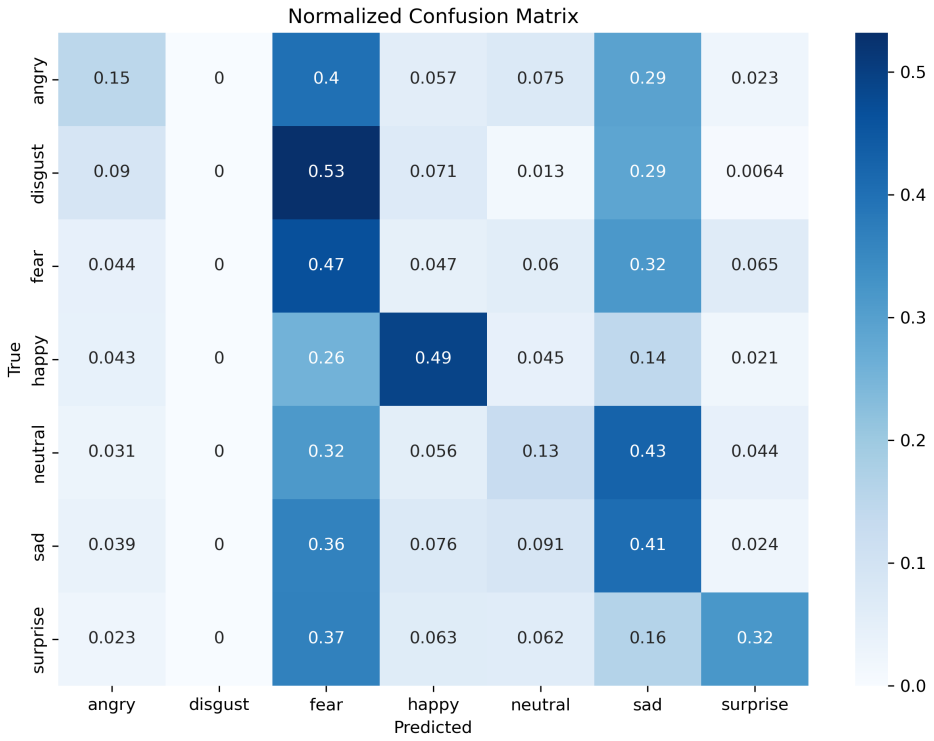

Figure 12: Normalized confusion matrix of the alternative CNN model. The matrix clearly lacks diagonal alignment and reveals prediction tendencies towards the most representative classes.

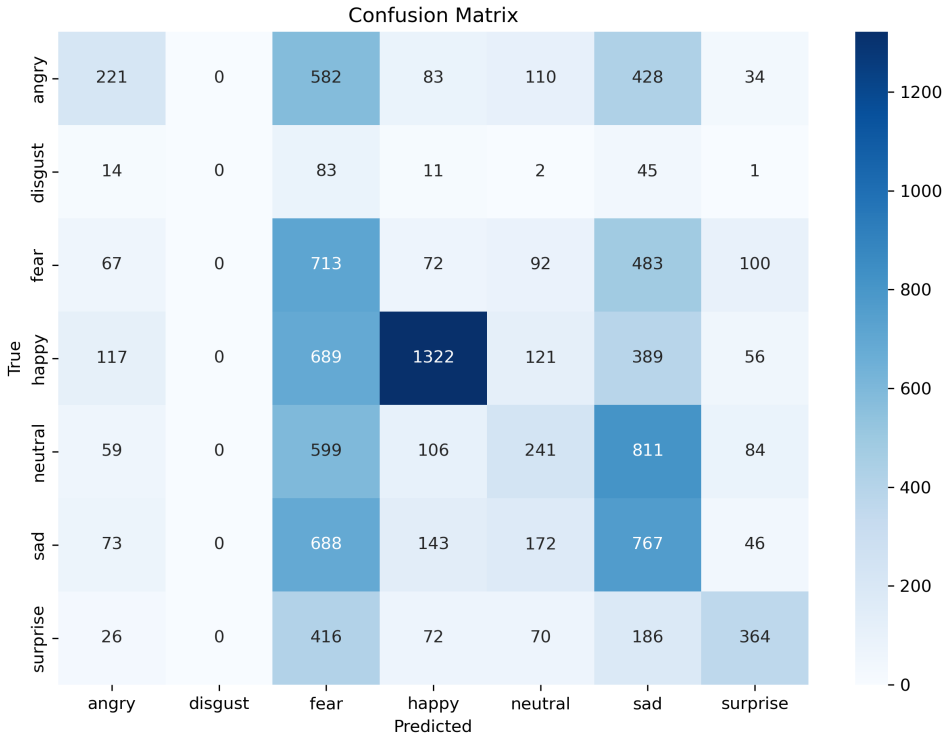

Figure 13: Confusion matrix of the alternative CNN model. The matrix clearly lacks diagonal alignment and reveals prediction tendencies towards the most representative classes. Without the normalization biases in the prediction become more apparent.

B Code & Dataset

The code can be accessed from: GitHub

The merged dataset used for this project can be downloaded from: Google Drive

Most notable files

- `main.ipynb` – Main notebook, using optimal hyperparameters from `optuna.db`.
- `densenet.py` – Implementation of the DenseNet class.
- `alternativeMethod_CNN.ipynb` – Notebook for the alternative method.
- `hyperparam_optimization.py` – Hyperparameter optimization using Optuna.