

|  |  |  |
| --- | --- | --- |
| **Islamic University of Madinah Faculty of Engineering Department of Electrical Engineering** |  | **الجامعة الإسلامية كليه الهندسة**  **قسم الهندسة الكهربائية** |

**EE 3701 – Artificial Intelligence (Special Topic)**

**Spring Semester 2025 – Term 462**

**Project #2**

**(Neural Networks)**

**[CO\_2, PI\_1\_3, SO\_1]**

**Project Title**

**Facial Expression Recognition**

**Submitted To**

**Prof. Adel Abdennour**

**Student Name,ID**

**Mohammed Badawi Mukhtar,431017333**

**Date: April 9th, 2025**

**1. Introduction**

The healthcare industry is undergoing significant transformation, influenced by technological progress and demographic trends, notably the increasing elderly population worldwide. Market analyses project substantial growth, particularly in remote patient monitoring services [4] [5]. Addressing the need for non-intrusive monitoring solutions is critical, especially for elderly or disabled individuals who may resist wearable devices or constant visual oversight. Artificial Intelligence (AI), specifically neural networks, provides robust tools for analyzing data acquired through contactless methods, such as video streams, offering potential for real-time well-being assessment and event detection.

This project contributes to non-contact monitoring by developing and evaluating a neural network system for Facial Expression Recognition (FER). While the broader scope of assistive AI includes diverse functionalities, this work concentrates on classifying human emotions from facial images. The ability to recognize emotional states (happiness, sadness, fear, anger, surprise, disgust, neutral) can serve as a valuable indicator within assistive technologies, offering insights into the psychological state of individuals in care environments without invasive procedures. Such information could signal distress, pain, or levels of social interaction.

The central objective was to design, implement, train, and evaluate a Convolutional Neural Network (CNN) model for classifying facial expressions using the publicly available FER2013 dataset [6]. The FER2013 dataset comprises static facial images, aligning with non-contact visual monitoring paradigms applicable in smart homes or healthcare settings. This project leverages deep learning, implemented via the PyTorch framework [11], focusing on image preprocessing, CNN architecture design, training methodologies, and comprehensive performance evaluation. The model's effectiveness is assessed using standard classification metrics, and its performance context is established by comparison with relevant benchmarks in FER literature.

**2. Literature Review**

Research in automatic FER has explored various approaches. Traditional machine learning methods often involved manually extracting geometric features or appearance features followed by a separate classification step [2][3].specifically investigated using Gabor filters for feature extraction prior to CNN classification on the JAFFE dataset, suggesting improved speed and accuracy [3].

However, deep learning approaches, particularly CNNs, have become dominant due to their ability to automatically learn hierarchical features directly from image data [1][2]. Various CNN architectures have been applied to FER. Standard architectures like VGGNet [1] and ResNet [2] have shown strong performance on FER2013. Other approaches include using Inception layers, attentional convolutional networks, combining CNN features with SVM classifiers, or modifying network components like pooling layers [2].

The FER2013 dataset [3] is a key benchmark introduced at ICML 2013. It presents challenges due to its "in-the-wild" nature, containing variations in pose, lighting, and occlusions, making it more difficult than datasets captured under controlled lab conditions like CK+ or JAFFE, used in [2] and [3]. Human accuracy on FER2013 is estimated to be around 65.5% [1]. Reported single-network CNN accuracies on FER2013 ranged roughly from 62% to 72.7% [1], with VGGNet achieving 72.7% and ResNet 72.4% in one study [1].

Effective training of these deep models involves careful selection of optimizers, momentum was found effective by [1], learning rate schedules like ReduceLROnPlateau used by [4] and regularization techniques like Batch Normalization [9], Dropout [10], and extensive data augmentation to combat overfitting [1][2].

**3. Theory/Modelling**

This project employs deep learning, specifically Convolutional Neural Networks (CNNs), tailored for image classification. The task of Facial Expression Recognition from static images is framed as a multi-class classification problem, mapping input images to one of seven emotion categories.

**3.1 Convolutional Neural Networks (CNNs)**

CNNs represent a specialized class of deep neural networks highly effective for processing grid-like data structures, most notably images [7]. Their architecture facilitates automatic learning of hierarchical feature representations directly from pixel data. Key components utilized in this project's model include convolutional layers for spatial feature extraction, ReLU activation functions for introducing non-linearity [8], MaxPooling layers for dimensionality reduction and spatial invariance [7], Batch Normalization for stabilizing training [9], fully connected layers for high-level classification, and Dropout for regularization [10]. The final classification probabilities are obtained via a Softmax function, integrated within the chosen loss function.

**3.2 Model Architecture Used**

The CNN architecture implemented comprises three convolutional blocks followed by two dense (fully connected) layers before the final output classification layer. It accepts 48x48 pixel grayscale images as input. The specific layer sequence is as follows:

1. Input: 48x48x1 grayscale image tensor.
2. Conv Block 1: 64-filter 3x3 convolution, Batch Normalization, ReLU activation.
3. Conv Block 2: 128-filter 3x3 convolution, Batch Normalization, ReLU activation, 2x2 MaxPooling.
4. Conv Block 3: 256-filter 3x3 convolution, Batch Normalization, ReLU activation, 2x2 MaxPooling.
5. Flatten: Reshapes the 3D feature maps into a 1D vector.
6. Dense Block 1: Linear layer (512 neurons), Batch Normalization, ReLU activation, Dropout (p=0.4).
7. Dense Block 2: Linear layer (256 neurons), Batch Normalization, ReLU activation, Dropout (p=0.3).
8. Output Layer: Linear layer (7 neurons), corresponding to the seven emotion classes.

**4. Simulation and Design Methods**

This section outlines the dataset, data handling techniques, training regimen, and evaluation strategy used. The implementation utilized Python and the PyTorch deep learning library [11].

**4.1 Dataset**

The FER2013 dataset [6] served as the foundation for training, validation, and testing. It contains 35,887 48x48 pixel grayscale face images across seven standard emotion categories (Angry, Disgust, Fear, Happy, Neutral, Sad, Surprise). The official training partition (28,709 images) was loaded and then split stratifiedly into an 80% training set (22,967 images) and a 20% validation set (5,742 images). The official Test Set, comprising 7,178 images, was used for the final model performance evaluation reported in Section 4.

**4.2 Data Preprocessing and Augmentation**

Data preparation involved several steps defined via image transformation utilities:

1. Base Preprocessing : Images were converted to grayscale (single channel), resized to 48x48 pixels, and transformed into framework-compatible tensors with pixel values scaled to the [0.0, 1.0] range.
2. Training Set Augmentation: To enhance model robustness and reduce overfitting, the following random transformations were applied exclusively to the training data during loading: random rotations (±25 degrees), random affine transformations (shifts up to 20%, zoom between 80-120%, shear up to 20 degrees), and random horizontal flips (50% probability).
3. Validation/Test Set Preprocessing: These sets used only the base preprocessing steps (grayscale, resize, tensor conversion) without augmentation to ensure evaluation reflects performance on unmodified data.

**4.3 Training Procedure**

The CNN model was trained using a specific configuration to optimize its performance. The Categorical Cross-Entropy loss function, suitable for multi-class classification, was employed to measure the difference between predicted and true labels. The Adam optimization algorithm [12] was utilized to update model weights, starting with an initial learning rate of 0.0001. Data was processed in batches of 32 images. Training proceeded for a maximum of 20 epochs. To adapt the learning rate during training, a scheduling mechanism monitored the validation loss; if the loss did not improve for 5 consecutive epochs, the learning rate was automatically reduced by a factor of 0.2 (down to a minimum value of 1e-6). Furthermore, an early stopping mechanism was implemented to prevent excessive training and potential overfitting; training halted if the validation loss failed to improve for 10 consecutive epochs. The model weights associated with the epoch yielding the lowest validation loss were saved and subsequently used for the final evaluation phase.

**4.4 Evaluation Metrics**

The performance of the trained model on the Test Set was quantified using several standard metrics derived from common machine learning evaluation utilities. These included the overall accuracy, representing the percentage of correctly classified images, and the average cross-entropy loss on the test data. A detailed classification report was generated, providing per-class precision, recall, and F1-scores, which offer insights into the model's effectiveness for each specific emotion category. Additionally, a confusion matrix was computed to tabulate the counts of correct and incorrect predictions between each pair of classes, typically visualized as a heatmap for easier interpretation of misclassification patterns.

**5. Results**

The model training completed the specified 20 epochs. The lowest validation loss observed during training was 1.1079, occurring at epoch 20. The model state corresponding to this minimum validation loss was used for the final evaluation performed on the Test Set.

Figure 1 presents the training and validation loss and accuracy curves across the 20 epochs.

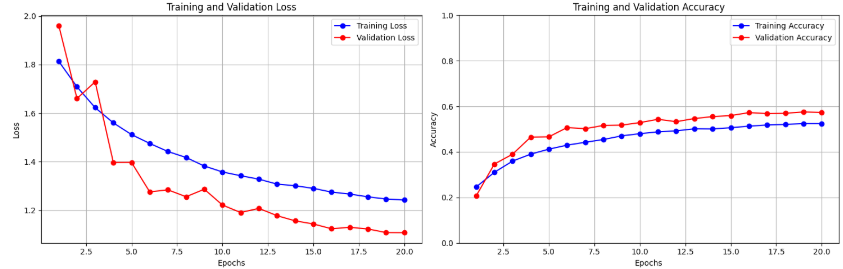
*Figure 1: Training and Validation Loss and Accuracy Curves over Epochs.*

Figure 1 illustrates the learning dynamics throughout the training process. Analysis of these curves provides insight into model convergence and the effectiveness of regularization techniques in mitigating overfitting.

The final evaluation on the Test Set resulted in an overall accuracy of 57.80% and an average loss of 1.1015.

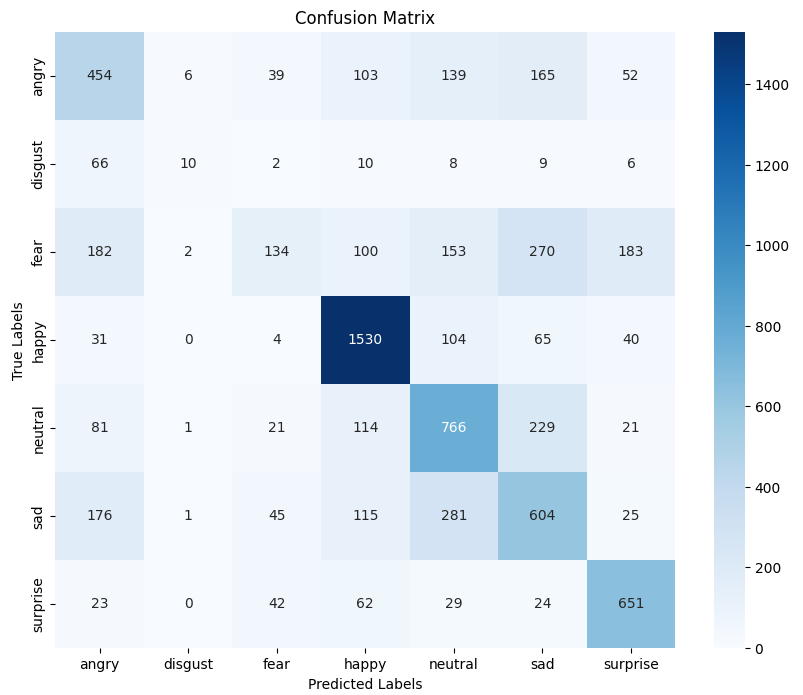
Detailed performance metrics for each emotion class on the Test Set are shown in Table 1.

Table 1: Classification Report on the Test Set.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Class | Precision | Recall | F1-Score | Support |
| Angry | 0.45 | 0.47 | 0.46 | 958 |
| Disgust | 0.50 | 0.09 | 0.15 | 111 |
| Fear | 0.47 | 0.13 | 0.20 | 1024 |
| Happy | 0.75 | 0.86 | 0.80 | 1774 |
| Neutral | 0.52 | 0.62 | 0.56 | 1233 |
| Sad | 0.44 | 0.48 | 0.46 | 1247 |
| Surprise | 0.67 | 0.78 | 0.72 | 831 |
| Accuracy |  |  | 0.58 | 7178 |
| Macro Avg | 0.54 | 0.49 | 0.48 | 7178 |
| Weighted Avg | 0.56 | 0.58 | 0.55 | 7178 |

The results in Table 1 indicate varied performance across classes. The model demonstrates strong performance for 'Happy' (F1=0.80) and 'Surprise' (F1=0.72). However, performance is significantly challenged for 'Disgust' (F1=0.15) and 'Fear' (F1=0.20), primarily due to very low recall values (0.09 and 0.13, respectively). This suggests difficulty in correctly identifying instances of these emotions, partly attributable to the low sample count for 'Disgust' and potential visual ambiguities. The weighted average F1-score (0.55) surpasses the macro average (0.48), highlighting the influence of the well-represented 'Happy' class on the overall weighted metric.

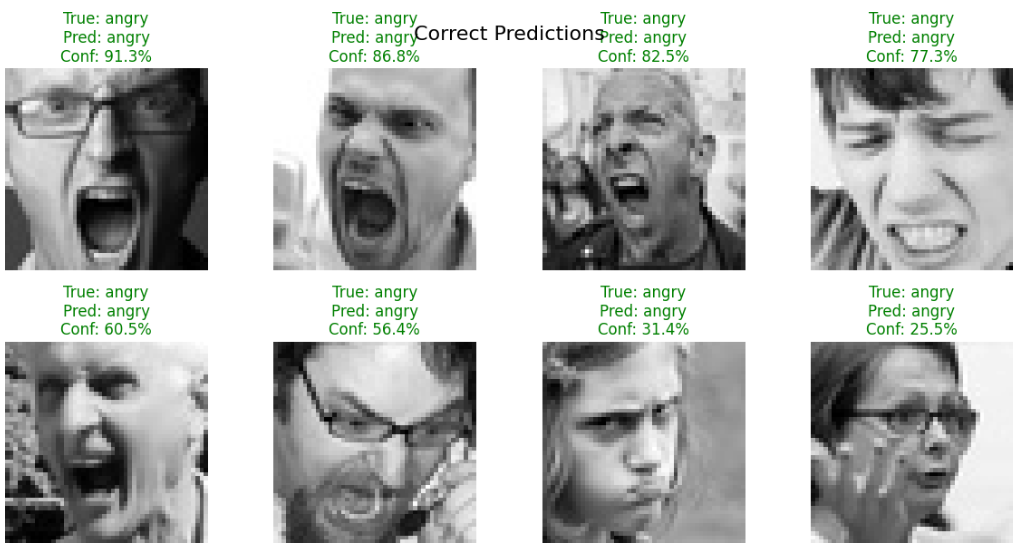
The confusion matrix (Figure 2) details the specific misclassification patterns between emotion classes.

**

*Figure 2: Confusion Matrix for Test Set Predictions.*

Analysis of the confusion matrix reveals common errors. Notably, 'Fear' was frequently misclassified as 'Sad' or 'Surprise'. 'Sad' was often confused with 'Neutral' or 'Angry'. 'Angry' was commonly misidentified as 'Sad'. The 'Disgust' class showed poor recognition, often being classified as 'Angry'. These confusions likely stem from visual similarities between the facial expressions associated with these emotions.

Sample predictions illustrating correct and incorrect classifications by the model are shown in Figure 3.



*Figure 3: Sample Predictions on Test Images.*

Visual inspection of sample predictions provides qualitative confirmation of the model's performance characteristics observed in the quantitative metrics.

**6. Discussion/Conclusions**

This project involved the implementation and evaluation of a Convolutional Neural Network for Facial Expression Recognition using PyTorch on the FER2013 dataset. The developed model achieved a final test accuracy of 57.80%, confirming the capability of CNNs to learn discriminative features for this task, albeit with challenges.

The training strategy incorporated data augmentation and regularization (Dropout, Batch Normalization). The model trained for 20 epochs, with the best performance on the validation set occurring in the final epoch. The model state achieving this best validation performance was selected for final testing. The training dynamics observed suggest that the employed regularization techniques were partially effective in managing overfitting.

Performance analysis highlighted significant variability across emotion classes, a common finding for the imbalanced FER2013 dataset. While 'Happy' and 'Surprise' were recognized relatively well, the model exhibited considerable difficulty with minority or ambiguous classes like 'Disgust' and 'Fear', primarily due to low recall rates. Common confusions, such as between visually similar expressions like 'Fear'/'Surprise' or 'Sad'/'Angry', underscore the inherent difficulty of the task.

Key lessons learned include the necessity of data augmentation and regularization in CNN training, the essential role of validation set monitoring for model selection, and the importance of examining per-class metrics and confusion matrices for a comprehensive performance assessment beyond overall accuracy. The results indicate that while the implemented model provides a functional baseline, achieving high accuracy across all FER2013 classes necessitates addressing data imbalance and potentially exploring more sophisticated architectures.

Future work could focus on implementing class imbalance techniques (e.g., weighted loss, resampling), utilizing transfer learning from pre-trained face networks, performing systematic hyperparameter optimization, or extending the model to incorporate temporal information for video-based FER. Evaluating the model on different datasets would also be valuable for assessing generalization.

In conclusion, this project successfully demonstrated the application of CNNs to facial expression recognition, achieving reasonable baseline performance. It provides practical insights into the deep learning workflow for image classification and highlights specific challenges and areas for future improvement relevant to developing robust AI systems for non-contact well-being monitoring.

**7. References**

[1] B. S. and M. Tt, "Human facial emotion recognition using deep neural networks," *Int. Arab J. Inf. Technol.*, vol. 20, no. 3, 2023. doi: 10.34028/iajit/20/3/2.

[2] Y. Khaireddin and Z. Chen, "Facial emotion recognition: State of the art performance on FER2013," arXiv preprint arXiv:2105.03588, May 2021. doi: 10.48550/arXiv.2105.03588. [Online]. Available: <https://arxiv.org/abs/2105.03588>

[3] "Fast Facial emotion recognition Using Convolutional Neural Networks and Gabor Filters." Accessed: Apr. 10, 2025. [Online]. Available: <https://www.researchgate.net/publication/344190368_Fast_Facial_emotion_recognition_Using_Convolutional_Neural_Networks_and_Gabor_Filters>

[4] M. A. Malik, "IoT healthcare market," Allied Market Research, 2016. Accessed: Apr. 10, 2025. [Online]. Available: <https://www.alliedmarketresearch.com/iot-healthcare-market>

[5] Deloitte, "2023 Global Health Care Outlook," 2023. Accessed: Apr. 10, 2025. [Online]. Available: <https://www.deloitte.com/global/en/Industries/life-sciences-health-care/analysis/global-health-care-outlook.html>

[6] I. J. Goodfellow *et al.*, "Challenges in representation learning: A report on the ICML 2013 workshop," arXiv preprint arXiv:1312.6082, Dec. 2013. [Online]. Available: <https://arxiv.org/abs/1312.6082>

[7] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[8] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn. (ICML-10)*, 2010, pp. 807–814.

[9] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Mach. Learn. (ICML)*, 2015, pp. 448–456.

[10] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, 2014. [11] A. Paszke *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Adv. Neural Inf. Process. Syst. 32 (NeurIPS 2019)*, 2019, pp. 8024–8035.

[12] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, Dec. 2014. [Online]. Available: <https://arxiv.org/abs/1412.6980>