King Saud University
College of Computer and Information Sciences
Department of Information Technology


IT461- Practical Machine Learning

# Facial Expression Recognition (FER)

Project Final Report

Prepared by:



| Name | ID |
|---|---|
| Tarfah Bin Moammar | 44420011 |
| Ghadeer Alnuwaysir | 444200420 |
| Lana Albogami | 444201031 |
| Noor algumlas | 444200811 |
| Dalal Alyousef | 444203019 |

## Table of Content



## Table of Tables



## Table of Figures





1

## Introduction

Emotions play an important role in human communication and often reveal deeper meanings than spoken words. Facial expressions in particular are one of the most powerful ways people express their feelings. However, unlike humans, machines lack the ability to understand these signals. This creates a gap in human-computer interaction, particularly in applications where user experience and decision-making can be improved. We seek to fill this gap by enabling computers to automatically identify emotional states such as happiness, sadness, anger, surprise, and neutral from facial images or videos using facial expression recognition (FER).

FER is widely used. In healthcare it can help in diagnosing psychological conditions or monitor patient health, in customer service it can measure customer satisfaction, in education it can help monitor students and provide better learning experiences, and in security it can help detect stress or suspicious behavior. Our project idea came from noticing the growing demand for intelligent systems that are not only practical but also responsive. As society becomes reliant on technology, the ability of machines to understand human emotions has the potential to positively transform and improve human experience and its interaction. We seek to develop a model that improves communication, increases security, and offers more personalized experiences. Developing reliable emotional response systems is not just a technical challenge, it is also a step toward creating machines that better complement human life [1] [2].

The intended task of this project is Facial Expression Recognition (FER)**,** which can be formulated as a multi-class supervised learning classification problem**.** The main objective is to automatically classify human facial expressions into distinct emotional categories.

- **Input:** A grayscale facial image with a resolution of 48×48 pixels.
- **Output:** An emotion label from one of the seven categories: *anger, disgust, fear, happiness, sadness, surprise, or neutral.*


*Figure 1: Project Lifecycle*

## Background

Facial Expression Recognition (FER) is an active field in computer vision and artificial intelligence. It focuses on teaching machines to recognize human emotions by analyzing facial features. FER has many real-world applications, such as monitoring patients' conditions in healthcare, improving student engagement in education, measuring customer satisfaction in service industries, and detecting unusual behavior in security systems.

Some of the key concepts related to this task are:

- **Supervised Learning:** A method where the model is trained on labeled data. Each image has a known emotion label, and the model learns the relationship between the input image and the correct output.
- **Classification Problem:** A machine learning task where each input is assigned to one category from several possible classes. In this project, the classes are the seven emotions: *anger, disgust fear, happiness, sadness, surprise, and neutral.*
- **Convolutional Neural Networks (CNNs):** CNNs are a type of deep learning model highly effective for analyzing images. They can automatically extract important features from facial images, such as edges, shapes, and textures, which makes them very suitable for Facial Expression Recognition tasks.

  Recent research shows that CNN-based models significantly improve the accuracy of Facial Expression Recognition compared to older machine learning methods. CNNs can recognize both static and dynamic emotions with high reliability, and many studies highlight their effectiveness in handling complex image data [3] [4].

## Related works:

1- In 2013, Goodfellow and colleagues organized the Facial Expression Recognition (FER-2013) Challenge to benchmark approaches for emotion recognition from images of faces. The dataset introduced in this contest contained 35,887 grayscale images, each resized to 48×48 pixels and labeled into seven emotion categories: Angry, Disgust, Fear, Happy, Sad, Surprise, and Neutral. Human accuracy on this dataset was measured at approximately 65–68%, reflecting the task's difficulty. A total of 56 teams participated, with the best-performing model, a convolutional neural network trained with an SVM loss function, achieving 71% accuracy. Other strong approaches included convolutional networks with image transformations and feature-based methods such as SIFT + MKL, which performed competitively but generally below CNNs. The

contest demonstrated that deep learning methods outperformed traditional handcrafted features, establishing CNNs as a strong baseline for facial expression recognition tasks [5].

2- Mollahosseini, Chan, and Mahoor (2016) proposed a deep convolutional neural network architecture designed to improve facial expression recognition, especially for images captured "in the wild." Their model incorporated Inception layers inspired by GoogLeNet, which allowed the network to capture multi-scale features more effectively than standard CNNs. They evaluated performance across several public databases, including FER-2013, CK+, MMI, DISFA, and SFEW, using both subject-independent and cross-database testing. The network achieved higher accuracy than traditional feature-based approaches such as LBP or HOG, and demonstrated strong generalization to unseen data. Their results highlighted the benefit of deeper network designs and data augmentation strategies for advancing emotion recognition systems [6].

3- Mollahosseini, Hasani, and Mahoor (2017) introduced AffectNet, a large-scale facial expression dataset collected from the Internet to address limitations of previous databases in uncontrolled (in-the-wild) settings. They gathered over 1,000,000 facial images using emotion-related keywords in six different languages and had 450,000 of those manually annotated for both categorical emotions (like Happy, Sad, Anger, etc.) and dimensional affect (valence and arousal). The work provided baseline models using deep CNNs for both categorical classification and continuous prediction of valence/arousal. They also compared those baselines with traditional methods such as SVM/SVR applied to hand-crafted features (e.g. HOG), showing the modern CNN models generally outperform the classical ones with **58–60%** for (baseline deep CNN) [7].

4- Li and Deng (2020) conducted a comprehensive survey of deep learning approaches for facial expression recognition, emphasizing the shift from controlled lab settings to "in the wild" environments. The survey reviewed major datasets, preprocessing techniques such as face alignment and data augmentation, and deep architectures including CNNs and RNNs. It also highlighted challenges such as limited labeled data, identity bias, and variations in pose and illumination. The authors concluded that deep learning methods outperform traditional handcrafted approaches, but robust preprocessing and augmentation strategies remain essential for improving generalization in real-world conditions [8].

5- Minaee, Minaei, and Abdolrashidi (2021) proposed an attentional convolutional network for facial expression recognition that focuses on the most informative regions of the face (eyes,

mouth, etc.) rather than treating all pixels equally. Their model is relatively shallow (fewer than 10 layers) but uses a spatial transformer attention mechanism to learn where to look. They tested it on multiple datasets including FER-2013, CK+, JAFFE, and FERG, achieving ~70% test accuracy on FER-2013. The work also included visualizations of which parts of the face are most salient for different emotions (e.g. mouth region for "happy", eyes/eyebrow region for "angry") [9].

## Data

The dataset chosen for this project is a Face Expression Recognition Dataset. It contains facial images, each labeled with one of seven emotion categories: Angry, Disgust, Fear, Happy, Sad, Surprise, and Neutral. The dataset is specifically designed for facial expression recognition tasks and is suitable for training supervised learning models due to its large size and balanced coverage of multiple emotion classes.

the **Facial Expression Recognition Dataset (FER-2013)** [10], originally introduced during the ICML 2013 Challenges in Representation Learning. It is publicly available on Kaggle where it has been widely used in research and practice for benchmarking facial expression recognition models.

We selected this dataset because:

- **Relevance:** It includes labeled photos of human facial expressions spanning the seven main emotion categories; anger, disgust, fear, happiness, sadness, surprise, and neutral. Which directly relates to our problem statement.
- **Size and diversity:** It offers sufficient examples for training strong supervised learning models, with more than 35,000 photos of faces gathered in a variety of settings.
- **Benchmark use:** The dataset is a well fit benchmark for assessing the efficacy of various machine learning techniques because it has been extensively used in research, including the FER-2013 challenge.
- **Accessibility:** It is simple to incorporate into machine learning workflows, freely available, and well-documented. This guarantees reproducibility and enables us to evaluate our findings in relation to another published research.

Commented [GU1]: will we use supervised learning?

To better understand the dataset, we provide summary statistics that describe its overall structure. As we said the dataset consists of **35,887 grayscale facial images**, each with a resolution of **48 × 48 pixels**. The

images are labeled into **seven distinct emotion classes**. Table 1 presents a concise overview of these statistics and Table 2 presents the number of the training and validating sets for each label.

| Statistic | Value |
|---|---|
| Total Number of Examples | 35,887 |
| Training Set Size | 28,709 |
| Validation Set Size | 7,178 |
| Image Dimension | 48 × 48 (grayscale) |
| Number of Classes | 7 |
| Data Type | Grayscale images |

*Table 1: Summary Statistics*

*To further illustrate the dataset, we present visual examples and the class distribution. Figure 1 shows one representative grayscale face image from each of the seven emotion categories. This helps to convey the variability of facial features and expressions across the dataset. In addition, Figure 2 presents the distribution of training and validation examples per class, highlighting the imbalance between emotions such as* **Happy** *(the most frequent) and* **Disgust** *(the least frequent).*

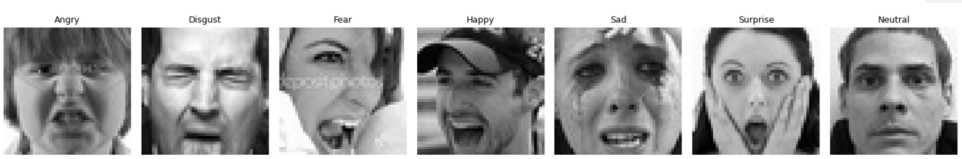


*Figure 2: Representative examples of facial expressions*

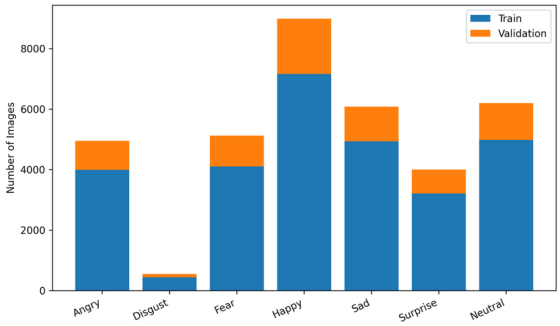


*Figure 3: Class Distribution by split (Train vs Validation)*

## Methods

We evaluated four models starting from a simple fully-connected neural network (NN), moving to a baseline CNN, then an enhanced CNN with deeper blocks and stronger regularization, and finally a fine-tuned VGG-16 model.

**1. Baseline Neural Network (NN)**

**Why we used it:** This model acts as a minimal baseline. It doesn't take advantage of image structure and therefore sets the "floor" of expected performance. Comparing every other model to this one helps quantify how much spatial modeling actually matters.

**How it works:**

- Flatten the 48×48 image into a 1D vector
- Dense layer with 64 units
- Dropout & Softmax output for 7 classes

**Why it fits / doesn't fit the data:** Since facial expressions rely on spatial patterns (eyes, mouth, eyebrows), flattening the image removes the structure the model needs. As expected, this model performs poorly but serves its purpose as a baseline.

**Feature engineering / dimensionality reduction:** Flattening is the only transformation.

**2. Baseline Convolutional Neural Network (CNN)**

**Why we used it:** CNNs are the standard approach for image tasks. They capture edges, textures, and local structure automatically, all essential for distinguishing subtle facial expressions. This model tests whether a straightforward CNN is enough for FER-2013.

**How it works:**

- Three convolution → pooling → dropout blocks
- A dense layer after feature extraction & Softmax output

**Why it fits the data:** The model extracts low-level facial cues (eye slants, mouth shape, wrinkles) that the NN cannot. Its size is appropriate for small images, making it a solid middle-ground between simplicity and performance.

**Feature engineering / dimensionality reduction:** Pooling layers progressively reduce spatial size while keeping important features.

**3. Enhanced CNN**

**Why we used it:**The baseline CNN improves performance, but the dataset is messy enough to justify a deeper and better-regularized model. The enhanced CNN was built to push accuracy further while stabilizing training.

**How it works:**

- More convolutional layers with higher filter counts
- Batch Normalization after most layers
- Strong dropout & AdamW optimizer
- Global Average Pooling instead of flattening

**Why it fits the data:** Deeper layers allow the model to capture more detailed facial features.

**Feature engineering / dimensionality reduction:** BatchNorm acts as feature normalization.

**4. VGG-16 Fine-Tuned**

**Why we used it:** VGG-16 provides strong pretrained features from ImageNet. The idea was to test if transfer learning could outperform custom CNNs, despite the dataset being grayscale and low-resolution.

**How it works:**

- Convert grayscale → 3-channel RGB
- Resize to 224×224 for VGG
- Freeze early VGG layers & add a lightweight classifier

**Why it fits & and why it falls short:** VGG's lower layers already know how to detect edges and textures, so the model trains quickly. However, resizing 48×48 images to 224×224 introduces blur and reduces fine details, which caps performance.
The model achieves high training accuracy but slightly worse validation accuracy than the enhanced CNN.

**Feature engineering / dimensionality reduction:** Channel replication & Resizing

| Model | What It Is | Performance |
|---|---|---|
| Baseline NN | Fully-connected network | Lowest accuracy |
| Baseline CNN | 3-block convolutional model | ~60% val accuracy |
| Enhanced CNN | Deeper CNN with BatchNorm, GAP, AdamW | Best performance (~66%) |
| VGG-16 Fine-Tuned | Pretrained ImageNet model with new classifier | Slightly below Enhanced CNN (~65%) |

*Table 2 models performance summary*

## Experiment

### 1.Preprocessing and Dataset Organization

We began by preparing the FER-2013 dataset so it could be used directly in TensorFlow. The dataset was downloaded from Google Drive and extracted into two main directories: a training folder containing 28,821 images and a validation folder containing 7,066 images. Each folder included the seven emotion categories as separate subdirectories, which allowed TensorFlow to read the classes automatically. This structure is shown in the sample code in our notebook where the directory loader identifies all images and assigns corresponding labels.

All images were kept in their original 48×48-pixel resolution and grayscale format. Since the data came from various sources and had different lighting conditions, keeping preprocessing light helped preserve the dataset's original characteristics without removing facial details that may be useful for learning.

### 2.Normalization

Before training, we normalized all pixel values from the original range of 0–255 to the range 0–1. This was done through a simple transformation that divides every image tensor by 255. Normalization stabilizes gradient updates, speeds up convergence, and reduces the risk of exploding values during training. The mapping function applied to both training and validation sets ensured that all images followed the same scale before being fed to the model.

### 3.Train/Validation Split

The dataset we used already contained a predefined 80–20 split. The training folder holds the majority of examples (approximately 28,821 images) while the validation folder contains around 7,066 images. Using the provided split helped maintain consistency with other FER-2013 benchmarks. When loading the data, TensorFlow's image_dataset_from_directory function automatically shuffled the training set and produced batches of images and one-hot encoded labels for all seven emotion classes. This split allowed the models to learn from a diverse group of facial expressions while still reserving unseen examples for evaluating generalization.

## 4.Data Augmentation

The original dataset contains noticeable class imbalance, as shown in both the report and the notebook. For example, “happy” has more than 7,000 images while “disgust” has fewer than 500. Because of this imbalance, augmentation became important, especially when training the CNN models.

We introduced augmentation in the convolutional models through dropout and pooling operations, which acted as a form of regularization. For the enhanced CNN and VGG-16 models, augmentation was introduced implicitly through several architectural features:

- dropout layers with different rates across the network
- batch normalization to stabilize feature distributions
- random shuffling at each epoch through TensorFlow’s dataset pipeline
- repeated exposure to slightly different mini batches

These techniques helped reduce overfitting and improved the model’s ability to generalize facial expressions with variations in lighting, pose, or slight distortions. Although we did not apply explicit geometric augmentations such as flips or rotations, the use of strong regularization within the enhanced CNN architecture served a similar purpose and contributed to more stable validation performance.

## 5.Training Pipeline

After preparing the data, we built a training pipeline that supports multiple models. TensorFlow datasets were cached to avoid disk reads during every epoch, shuffled to ensure balanced batch ordering, and prefetched to keep the GPU fully utilized. This created a smooth data flow during training.

The training loop followed these steps:

1. Load a model architecture (NN, CNN, enhanced CNN, or VGG-16).
2. Compile the model with categorical cross-entropy and accuracy.
3. Feed normalized, batched datasets into the training step.
4. Train for a set number of epochs while tracking accuracy and loss.

Evaluate the validation set to measure how well the model generalizes.

For the enhanced CNN and VGG-16 models, we added callbacks such as learning-rate reduction and early stopping. These callbacks monitored validation loss and prevented unnecessary over-training. The

notebook shows how these callbacks helped the enhanced model reach stable performance while avoiding overfitting on the FER-2013 images

Overall, the training pipeline ensured that each model received clean, normalized, and consistently formatted data while taking advantage of TensorFlow's performance optimizations.

## 6.Model Architectures

This section describes all models implemented: Baseline Neural Network (NN), Original CNN, Enhanced CNN, and VGG-16 Fine-Tuned. The purpose is to document architectural differences that contributed to performance variations across models.

**Baseline Neural Network (NN):**

A simple fully-connected network used to establish a baseline. The architecture consisted of:

• Input: Flatten(48×48)
• Dense(64, ReLU)
• Dropout(0.3)
• Output: Dense(7, Softmax)

**Original CNN:**

Designed to capture spatial facial features using convolutional layers.

• Conv Block 1: Conv2D(32, 3×3), BatchNorm, MaxPool, Dropout
• Conv Block 2: Conv2D(64, 3×3), BatchNorm, MaxPool, Dropout
• Conv Block 3: Conv2D(128, 3×3), BatchNorm, MaxPool, Dropout
• Global Average Pooling
• Dense(512, ReLU), Dropout(0.5)
• Dense(256, ReLU), Dropout(0.5)
• Output: Dense(7, Softmax)

**Enhanced CNN:**

An improved deeper architecture with stronger regularization and AdamW optimizer.

• Additional convolutional layers
• Increased filter counts
• Stronger dropout regularization
• Same classification head structure

**VGG-16 Fine-Tuned:**

Transfer-learning model using ImageNet-pretrained VGG-16 with unfrozen top layers.

• Base: VGG-16 (frozen/partially unfrozen)

• Flatten or GAP

• Dense(256, ReLU)

• Dropout

• Output: Dense(7, Softmax)

## 7.Hyperparameter Tuning

Hyperparameter tuning was conducted using GridSearchCV with a KerasClassifier wrapper.
The search explored different learning rates, dropout rates, neuron counts, batch sizes,
and epochs to determine an optimal configuration for stable and generalizable training for the **Baseline Neural Network (NN)** model.

The best performing configuration selected by GridSearch was RMSprop optimizer, dropout rate of 0.3, 128 neurons, batch size of 64, and 20 training epochs.

**Hyperparameter Search Table**

| Hyperparameter | Values Tested |
|---|---|
| Learning Rate | 0.001, 0.0005 |
| Dropout Rate | 0.2, 0.3, 0.5 |
| Dense Layer Neurons | 64, 128 |
| Batch Size | 32, 64 |
| Epochs | 10, 20 |

*Table 3 Hyperparameter Search Table*

## 8. Performance Metrics

**Baseline Neural Network (NN):**

• Training Accuracy: 0.2488

• Validation Accuracy: 0.2922

**After Hyperparameter Tuning (Final NN Model):**
• Best Validation Accuracy: 0.3462

**Original CNN:**

• Training Accuracy: 0.6164
• Validation Accuracy: 0.6039
• Best Validation Accuracy: 0.6050

**Enhanced CNN:**
• Training Accuracy: 0.7244
• Validation Accuracy: 0.6643
• Best Validation Accuracy: 0.6644

**VGG-16 Fine-Tuned:**
• Training Accuracy: 0.9723
• Validation Accuracy: 0.6524
• Best Validation Accuracy: 0.6571

### 9.Compute Resources

All models were trained using a Google Colab T4 GPU. The CNN models required 20–35 ms per step depending on depth, while VGG-16 required additional computation due to the pre-trained backbone. GPU acceleration significantly reduced training time and enabled multiple experimental runs, including hyperparameter tuning and fine-tuning.

## Results and Discussion

### Experimental Results and Model Performance Summary

A comprehensive evaluation was conducted across four neural network architectures, Baseline Neural Network (NN), Original CNN, Enhanced CNN, and Fine-Tuned VGG-16 to assess their effectiveness in facial expression recognition. The following table summarizes the key performance metrics obtained during experimentation, enabling direct comparison across models.

**Performance Metrics of Implemented Models**

| Model | Training Accuracy | Validation Accuracy | Best Validation Accuracy | Validation Loss |
|---|---|---|---|---|
| Baseline NN | 24.88% | 29.22% | 29.22% | 1.812 |
| Fine-tuned NN | 33,57% | 34,62% | 36,06% | 1.6632 |
| Baseline CNN | 61.64% | 60.39% | 60.50% | 1.100 |
| Enhanced CNN | 72.44% | 66.43% | 66.44% | 1.045 |
| VGG-16 Fine-Tuned | 97.23% | 65.24% | 65.71% | 1.158 |

*Table 4 Performance Metrics of Implemented Models*

The results highlight the advantage of convolutional architectures over simple dense networks. The fine-tuned NN struggled to extract meaningful spatial features from facial images, resulting in poor accuracy (36,06%) In contrast, CNN achieved a much higher validation accuracy (60.39%), representing a (31%) absolute improvement.

The Enhanced CNN had the best overall performance, achieving (66.43%) validation accuracy, outperforming CNN by (6.04%). This improvement demonstrates the benefits of increased depth, stronger regularization, and optimized hyperparameters.

The VGG-16 Fine-Tuned model achieved extremely high training accuracy (97.23%), suggesting strong learning capability. However, its validation accuracy was lower than the Enhanced CNN, indicating heavy overfitting and potential domain mismatch between ImageNet pretraining and facial expression data.

## Generalization Capability Analysis

The generalization behavior of each model was assessed using learning curves that compare training and validation performance across epochs.

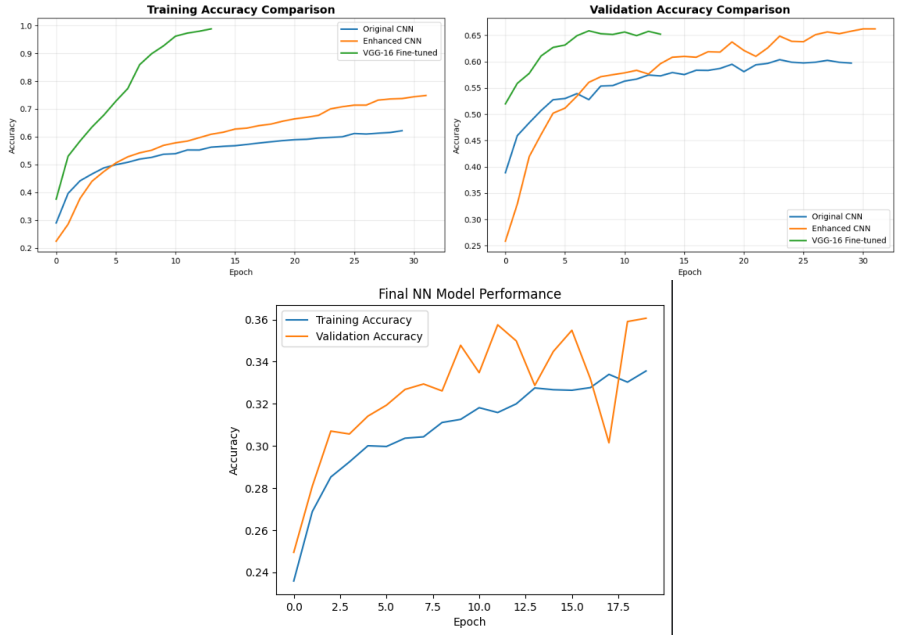


*Figure 4: Learning Curves Training vs Validation Performance*

Description:

- Fine-tuned NN: Training and validation accuracies remained low and parallel (30%-35%), indicating strong underfitting.
- Original CNN: Training and validation curves converged near 60%, showing balanced learning and reasonable generalization.
- Enhanced CNN: Mild divergence between training (72%) and validation (66%) curves indicated slight overfitting, but generalization remained strong.

- VGG-16: A large gap (97% vs 65%) reflected extreme overfitting, despite powerful pretrained feature extractors.

These observations align with the generalization gap values in Performance Metrics. While all CNNs learned meaningful features, only the Enhanced CNN struck an optimal balance between learning capacity and generalization.

### Interpretation of Findings and Key Insights

Learning curve analysis revealed that the Enhanced CNN demonstrated optimal training dynamics with smooth, consistent convergence and minimal overfitting, maintaining stable validation performance throughout training. In contrast, VGG-16 exhibited rapid initial convergence due to pretrained weights but showed early stability in verification, indicating overfitting despite its architectural advantages. The Original CNN displayed steady but slower improvement patterns, while the Baseline NN consistently underfits parallel training and validation curves. These learning behaviors establish that successful facial expression recognition requires architectures that balance learning capacity with regularization to achieve stable convergence and sustainable generalization performance.

### Summary of Key Insights

The Enhanced CNN achieved the best overall performance with 66.43% accuracy and showed the strongest generalization capability across the evaluated models. In contrast, the VGG-16 transfer-learning approach did not outperform the custom architectures, likely due to domain mismatch and increased susceptibility to overfitting. Learning-curve behavior showed that deeper CNNs with proper regularization generalize more effectively, whereas overly deep pretrained models such as VGG-16 tend to overfit. Overall, architectural depth, expanded filter capacity, and strong regularization emerged as key factors for improving facial expression recognition performance.

## Conclusion

### Key findings

This project evaluated several deep learning models for facial emotion recognition using the FER-2013 dataset [10]. Among the tested architectures, the Enhanced CNN achieved the strongest performance with **66.43% accuracy**, which is notably close to the human-level accuracy reported in the original FER-2013 challenge (**65–68%)** and approaching the best performing model from that competition a deep CNN

trained with an SVM loss function that achieved approximately 71% accuracy [5]. The Enhanced CNN benefited from increased depth, expanded filter capacity, and regularization, all of which contributed to smoother learning curves and stronger generalization. In contrast, the VGG-16 transfer-learning model did not outperform the custom architecture, likely due to domain mismatch and its higher susceptibility to overfitting.

### challenges

Several challenges emerged throughout the development process. The FER-2013 [10] dataset's low resolution, grayscale format, and significant class imbalance made it difficult for the models to reliably distinguish subtle emotions, especially between visually similar categories such as *fear*, *sadness*, and *disgust*. These limitations are well known in the literature and are a major reason why achieving acceptable performance on FER-2013 [10] is inherently difficult even strong benchmark models rarely exceed 70% accuracy on this dataset. Training deeper architectures required longer computation time and careful management of preprocessing pipelines and input dimensions, particularly for pretrained models like VGG-16 that are more sensitive to domain mismatch. Additionally, hyperparameter tuning highlighted how neural networks can be highly sensitive to choices such as dropout rate, batch size, and optimizer configuration, all of which significantly impact stability and generalization.

### Improvements &Future direction

Future improvements could include applying more extensive data augmentation to increase model robustness, experimenting with modern architectures and incorporating domain-matched datasets for transfer learning. Overall, the results demonstrate that well-designed convolutional architectures with appropriate depth, filter design, and regularization play a central role in improving performance on challenging datasets like FER-2013 [10].

## Contributions

| Contributor | Task |
|---|---|
| Tarfah Bin Moammar | Motivation, related works 1, results and discussion |
| Ghadeer Alnuwaysir | Dataset, related works 2, methods |
| Lana Albogami | Background, related works 3, experiments |
| Noor algumlas | Background, related works 4, conclusion |
| Dalal Alyousef | Dataset, related works 5, experiments |

*Table 5 contributions*

## References


[1] Y. Wang, "A survey on facial expression recognition of static and dynamic emotions," 2024.

[2] "A comprehensive survey on deep facial expression recognition: challenges, applications, and future guidelines," *ScienceDirect,* 2023.

[3] Y. Li, J. Zeng, S. Shan and X. Chen, "Occlusion aware facial expression recognition using CNN with attention mechanism," *IEEE Transactions on Image Processing,* p. 2439–2450, 2019.

[4] G. Zhao and M. Pietikäinen, "Dynamic texture recognition using local binary patterns with an application to facial expression," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* p. 915–928, 2007.

[5] I. J. Goodfellow, D. Erhan, P. L. Carrier, A. Courville, M. Mirza, B. Hamner, W. Cukierski, Y. Tang, D. Thaler, D.-H. Lee, Y. Zhou, C. Ramaiah, F. Feng, R. Li, X. Wang, D. Athanasakis and J. Sha, "Challenges in Representation Learning: A report on three machine learning contests," : arXiv:1307.0414 [stat.ML], 2013.

[6] D. C. M. H. M. Ali Mollahosseini, "https://arxiv.org/pdf/1511.04110," arXiv:1511.04110 [cs.NE], 2015.

[7] A. Mollahosseini, B. Hasani and M. H. Mahoor, "A Database for Facial Expression, Valence, and Arousal Computing in the Wild," arXiv:1708.03985 [cs.CV], 2017.

[8] S. L. &. W. Deng, "Deep Facial Expression Recognition: A Survey," arXiv, 2018.

[9] S. Minaee, M. Minaei and A. Abdolrashidi, "Deep-Emotion: Facial Expression Recognition Using Attentional Convolutional Network," *Sensors,* vol. 21, no. 9, 2021.

[10] J. Oheix, "Face Expression Recognition Dataset," 2020. [Online]. Available: https://www.kaggle.com/datasets/jonathanoheix/face-expression-recognition-dataset.