1. Problem/Task/Application Statement

- Clearly describe the problem you are addressing. (2-3 sentences)

The goal of this project is to classify facial expressions using images from the FER2013 dataset. The binary classification task focuses on distinguishing between positive (happy) and negative (angry, fearful, sad) emotions.

- Explain why the problem is important and relevant. (2-3 sentences)

This problem is relevant for building emotion-aware applications, enhancing human-computer interaction, and supporting mental health tools that rely on emotional state recognition.

- Fill the following table about the dataset used (add column for any additional information that you would like to provide about the dataset)

Dataset Name	**Source**	**Number of Samples**	**Number of Features**	**Data Types**
FER2013	Kaggle (fer2013.csv)	17000 (balanced)	48x48 pixel images	Text (pixel values), Integer (labels)

2. Techniques and Methodologies Used

- **Approach Chosen:** Transfer learning using InceptionResNetV2 for feature extraction, followed by a custom CNN classifier head for binary emotion classification.
- **Tools, Techniques and Technologies:** Mention the software, programming languages, frameworks, or platforms used.

Name of the Tool / Technique / Technology	**Explanation**	**Importance/need in the project (Why is it used?)**	**Sources / References Used**
Python, TensorFlow, Keras	Programming language and deep learning libraries	For building, training, and evaluating the CNN model	Official TensorFlow & Keras docs
InceptionResNetV2	Pre-trained CNN base model	For leveraging pretrained image features on FER2013	Keras Applications
Matplotlib	Visualization tool	Used for displaying sample images	matplotlib.org
Scikit-learn	Data splitting and evaluation metrics	Stratified train/test split, confusion matrix	scikit-learn.org

- **Implementation Details:** Provide a brief explanation of how the techniques were applied.

Name of the Tool / Technique / Technology	**Implementation details**	**Sources / References Used**
InceptionResNetV2 (transfer learning)	Used with include_top=False and frozen weights to extract features	Keras Applications
CNN Layers (Conv2D, MaxPooling, Dense)	Additional conv-pooling block, 3 dense layers with applied ridge regularization	Keras layers and regularizers
FER2013 dataset preprocessing	Resizing grayscale images to RGB (139x139), no normalization needed	FER2013 CSV format + TF.image.resize
Binary Crossentropy loss & Adam optimizer	Used for binary classification training	keras.losses.BinaryCrossentropy

4. Evaluation Metrics and Approaches

- Fill the below table for metrics and approaches. Consider the following aspects while filling:
    - Define the key performance indicators (KPIs) or metrics used for evaluation.
    - Explain the experimental setup and validation methods.
    - Describe any benchmarking approaches used for comparison.

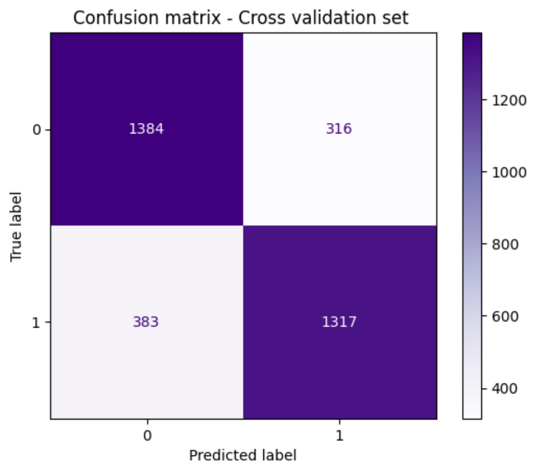
Name of the Evaluation Metric / Approach	**Description**	**Importance/need in the project (Why is it used?)**	**Sources / References Used**
Accuracy	% of correct predictions	Indicates model performance in balanced dataset	sklearn.metrics.accuracy_score
Confusion Matrix	TP, FP, FN, TN overview	For identifying bias toward any class	sklearn.metrics.confusion_matrix
Precision	Proportion of correctly predicted positive observations to the total predicted positive observations	Used to evaluate the model's ability to avoid false positives	sklearn.metrics.precision_score
Recall	Proportion of correctly predicted positive observations to all actual positive observations	Used to evaluate the model's ability to capture all relevant cases	sklearn.metrics.recall_score
F1-score	Harmonic mean of precision and recall	Balances precision and recall	sklearn.metrics.f1_score
Learning Curves	Graphs showing model performance (e.g., loss and accuracy) over training epochs for both training and validation sets.	Help identify underfitting, overfitting, and convergence issues by comparing how well the model learns over time.	Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow*. O'Reilly Media.

5. Results and Analysis

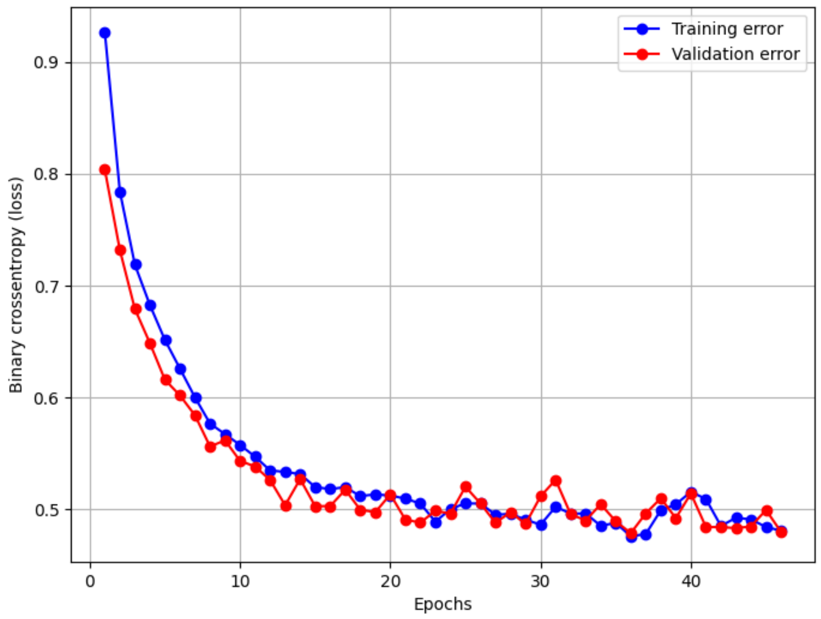
Test Approach: The dataset was split into 60% training, 20% validation (cross-validation), and 20% test. The InceptionResNetV2 base was frozen, and additional dense layers were trained using the Adam optimizer with a learning rate of 0.001. The input images were scaled to the [0, 255] range as expected by InceptionResNetV2 preprocessing in Keras.

Numerical Results:

• Confusion Matrix (Validation Set):
True Negative (TN): 1384
False Positive (FP): 316
False Negative (FN): 383
True Positive (TP): 1317



Learning curves:



- Accuracy = 0.7944
  Indicates that ~79.4% of predictions were correct.
- Precision = 0.8065
  ~80.7% of predicted positives were actual positives
- Recall = 0.7747
  ~77.5% of actual positives were correctly identified.
- F1 Score = 0.7903
  Reflects solid overall classification.

Analysis of the Results:

The model is performing well with balanced precision and recall, and the learning curves indicate effective training without overfitting. The confusion matrix shows a moderate class imbalance in prediction, but performance across both classes is still strong.F1 score close to 0.80 is considered quite good for this classification problem.

6. State of the Art

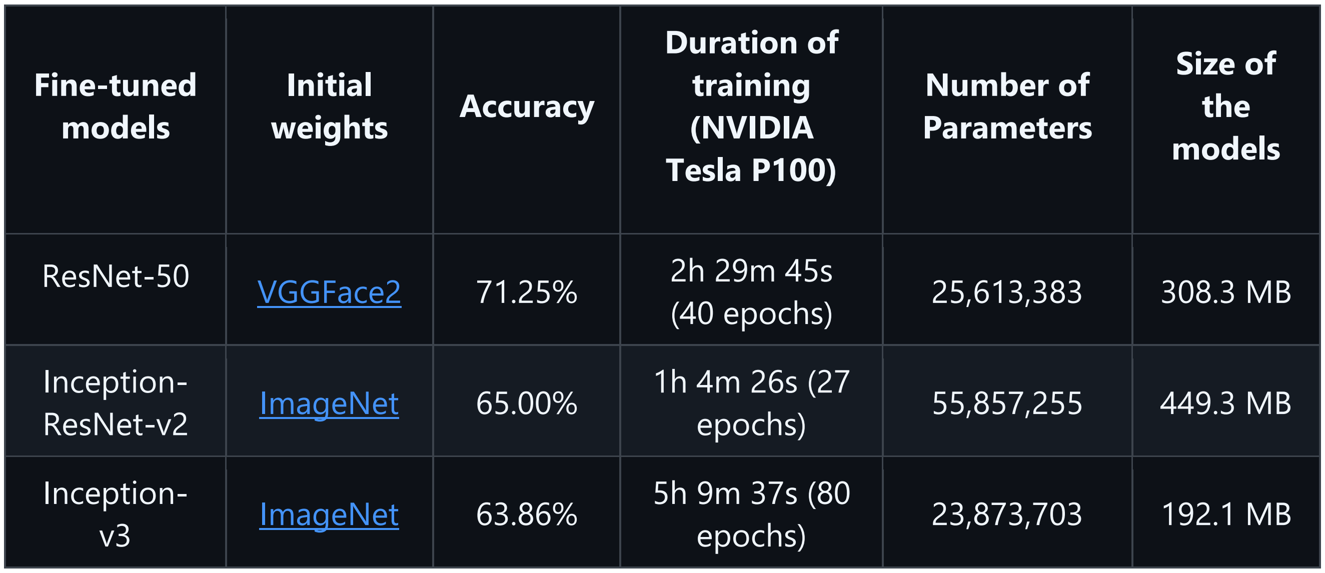
- DataSet application

In this project, the classification task was reduced to a binary problem, meaning the FER2013 dataset was transformed and divided into two emotion classes instead of the commonly used seven. This modification allowed for a more focused analysis tailored to the specific research objective. However, as a result of this transformation, direct comparison with results reported in the scientific literature is challenging, since most state-of-the-art approaches are based on the original seven-class structure of the FER2013 dataset.

A review of available publications did not reveal any studies applying a similar binary classification setup using FER2013, which limits the ability to benchmark the model's performance against existing solutions. Therefore, this project presents a novel approach that may serve as a foundation for future research in the area of binary emotion classification using FER2013.

- Architecture

Source: https://github.com/ivadym/FERehab/tree/master



Fine-tuned models	Initial weights	Accuracy	Duration of training (NVIDIA Tesla P100)	Number of Parameters	Size of the models
ResNet-50	VGGFace2	71.25%	2h 29m 45s (40 epochs)	25,613,383	308.3 MB
Inception-ResNet-v2	ImageNet	65.00%	1h 4m 26s (27 epochs)	55,857,255	449.3 MB
Inception-v3	ImageNet	63.86%	5h 9m 37s (80 epochs)	23,873,703	192.1 MB

The table above confirms the consistency between the experimental assumptions made in the context of FER (Facial Expression Recognition) and the methodology applied in this project. Specifically, the use of transfer learning with well-established architectures such as ResNet-50, Inception-ResNet-v2, and Inception-v3 aligns with the approach taken in this work. However, there are two notable differences that distinguish these state-of-the-art models from the one developed in this project.

Firstly, the fine-tuning process in the referenced models involved unfreezing and training several top layers of the network, which can lead to improved performance by allowing the model to better adapt to the target dataset. In contrast, in this project, fine-tuning was not applied, and the pre-trained weights were used as fixed feature extractors. Secondly, the best-performing model (ResNet-50) was initialized with weights pre-trained on VGGFace2, a dataset focused on human facial features. This likely contributed to its superior performance in FER tasks. In this project, such a specialized pre-training approach was not utilized, which may partly explain the performance gap.