Journal of the Brazilian Computer Society, 2026, 32:1, doi: 10.5753/jbcs.2026.5370

Intelligent Emotion Tracking System VIRE: Evaluation of Neural Network Architectures in Facial Emotion Recognition

Nathan Ferraz da Silva [**University of São Paulo** | ***nathansilva@usp.br***]
Geraldo Pereira Rocha Filho [**State University of Southwest Bahia** | ***geraldo.rocha@uesb.edu.br***]
Roger Immich [**Federal University of Rio Grande do Norte** | ***roger@imd.ufrn.br***]
Vinícius Pereira Gonçalves [**University of Brasilia** | ***vpgvinicius@unb.br***]
Rodolfo Ipolito Meneguette [**University of São Paulo** | ***meneguette@icmc.usp.br***]

Institute of Mathematical and Computer Sciences, University of São Paulo (USP) Av. Trabalhador São Carlense, 400 - Centro, São Carlos - SP, 13566-590, Brazil

Received: 23 December 2024 • **Accepted:** 03 September 2025 • **Published:** 09 March 2026

Abstract. This work proposes an emotional monitoring system called Visual Identification of Recognition of Emotions (VIRE), based on convolutional neural networks (CNNs) to analyze facial expressions. Using the six basic emotions proposed by Paul Ekman as a reference, which can be identified from the composition of various facial muscle states, VIRE aims to assist in the diagnosis of mental health conditions. While emotional expressions are communicated in various ways, this research focuses primarily on facial expressions due to their expressiveness resulting from the mobility of facial muscles. The methodology involved collecting data from the FER2013 dataset, preprocessing the images, hyperparameter tuning, and training three different architectures: AlexNet, DenseNet, and a custom CNN. The research will classify expressions into basic emotions and evaluate the models' performance in terms of accuracy and other metrics. VIRE has demonstrated potential, achieving an accuracy of about 60%, although improvements are needed for practical application. The ultimate goal is to create a tool that integrates technology and health, facilitating the identification of emotional states that may indicate mental health issues, thereby contributing to more accurate and effective diagnoses.

Keywords: Convolutional Neural Networks, Emotion Recognition, Artificial Intelligence, Deep Networks

1 Introduction

Emotional expressions of individuals manifest in various ways, including variations in tone of voice, body posture, and, indirectly, in clothing. The ability to recognize these signals is fundamental, as it can provide significant insights into an individual's emotional state; however, most research focuses on facial expressions [Ekman and Friesen, 1971]. This is due to the presence of a large number of facial muscles, a result of the need for chewing, which gives the face a broad capacity for mobility in various directions. The expression of happy, for example, is commonly characterized by the elevation of the zygomatic major muscle, which lifts the area between the lips and the cheek, resulting in a smile. Additionally, facial expressions are often used as indicators of emotional states in mental health research.

Diagnosing mental disorders presents a particularly complex challenge. Unlike other medical conditions that can often be confirmed through biomarkers or imaging, psychiatric diagnoses rely primarily on clinical interpretation of behavioral and emotional symptoms [Dalgalarrondo, 2019]. This process is inherently subjective and can vary across professionals, especially when supported by diagnostic criteria that may be limited or culturally sensitive [American Psychiatric Association, 2014]. The predominance of interpretative judgment can result in delayed or inconsistent diagnoses, highlighting the need for complementary tools that enhance the reliability and objectivity of the process.

In this context, emotion recognition emerges as a promising strategy. Emotional expressions are often used as observable indicators of mental states, and their analysis could support mental health assessments, especially facial expressions. This is particularly relevant in depressive disorders, where feelings of sadness and emotional flatness are common and may manifest through facial cues [Solomon, 2014]. An automated system capable of identifying these emotional signals can aid clinicians by offering additional evidence during the diagnostic process.

Artificial Intelligence, especially Convolutional Neural Networks (CNNs), offers strong potential to support this task. CNNs are proficient at extracting spatial patterns from visual data and have achieved remarkable results in facial analysis tasks [Chollet, 2018]. By recognizing subtle combinations of facial features, such as muscle movements and expressions, these networks can provide consistent and scalable insights into emotional states.

Despite the significant advances achieved by CNN, they are still subject to factors that can affect their training and interpretative capability [Szegedy *et al.*, 2015]. Models trained for emotional recognition require a large volume of labeled data, which may be subject to errors and biases, as this labeling is performed by humans [Frenay and Verleysen, 2014]. Moreover, the high demand for data and the computational complexity of convolutional networks necessitate the use of high-performance hardware. Finally, understanding the processes that lead a neural network to a particular outcome is

an extremely complex task, especially when it comes to deep networks. Although CNNs have these peculiarities, they can become powerful tools if configured properly.

The overall objective of this work is to propose VIRE, a CNN-based emotion monitoring system capable of analyzing facial expressions. To achieve this goal, two specific objectives were defined: (1) to find a model that can classify a facial capture into one of the following basic emotions: happy, angry, sad, fearful, disgusted, surprised and neutral; (2) to analyze the performance of three distinct models (Densenet, Alexnet, and a custom CNN) in the correct classification of images, based on their accuracy, loss, and confusion matrix metrics achieved through hyperparameters suggested by Hyperopt. The main contribution of this work is the development and evaluation of VIRE, an intelligent system for emotion recognition to support mental health monitoring through facial expression analysis.

The remainder of this article is organized as follows: Section 2 presents related work, exploring the emotion recognition problem. Section 3 describes how the solution was modeled, while Section 4 presents the results obtained to validate our methodology. Finally, Section 5 presents the conclusions of this research and future work.

2 Related Works

The work of Zhang *et al*. [2020] proposes a multimodal approach for the recognition of mental disorders, such as bipolar disorder and depression. The developed model combines acoustic, visual, and textual features through a deep learning framework, considering the correlation between these different modalities. This integration is crucial, as each modality, when considered in isolation, may not be able to capture the complexity of the signals associated with mental disorders, making their identification more difficult. Unlike multimodal approach that combines acoustic, visual, and textual modalities for mental disorder detection, VIRE focuses solely on facial expressions in visual data. This image-based strategy allows for lower complexity and easier deployment in home environments, such as elderly care and remote monitoring.

The research of Minaee *et al*. [2021] proposes a framework for emotion recognition using an attentional CNN. This type of network is capable of identifying regions within images that contain crucial information for detecting facial expressions. This attention mechanism allows a network with fewer than ten layers to perform comparably to deeper networks in emotion recognition. The tests were conducted on various datasets, such as FER-2013, CK+, FERG, and JAFFE, and the results were quite promising. In contrast to Minaee et al.'s attentional CNN, which focuses on identifying salient regions of facial images to enhance classification, VIRE employs standard CNN architectures without attention mechanisms, prioritizing simplicity and interpretability in its evaluation of different models.

Transfer learning also stands out as an effective approach in the context of emotion recognition. The research by Albraikan *et al*. [2022] proposes the IFER-DTFL technique, which first detects the face and then identifies the corresponding facial expression. The IFER-DTFL process is structured in three main stages: face detection, feature extraction, and expression classification. For face detection, the Mask R-CNN model is used, allowing for precise face identification. In the feature extraction phase, the DenseNet121 model, combined with the Adam optimizer, is used to extract relevant information. Finally, for facial expression classification, the Weighted Kernel Extreme Learning Machine (WKELM) model is used. While the IFER-DTFL technique integrates face detection, feature extraction with DenseNet, and classification using WKELM, VIRE adopts a more streamlined pipeline by evaluating end-to-end CNN architectures trained directly on the FER2013 dataset. Moreover, VIRE emphasizes the impact of hyperparameter tuning and data augmentation on performance.

There are works that focus on emotion recognition based on speech. In Filho *et al*. [2024], the authors propose the DEEP architecture (Detection of Voice Emotion in Portuguese Language) for emotion recognition, focusing on the specialization of models to identify emotions with greater precision and adaptability. The methodology is based on the use of the sound spectrum to extract acoustic features, including MFCCs, chromatic features, and prosodic features, from the VERBO database, which contains recordings of Brazilian speakers. These features feed specialized CNN, each trained to recognize a specific emotion. The main limitation of the work lies in linguistic and cultural variabilities, such as accents, dialects, and regionalisms, which can alter the sound patterns used to identify emotions. Moreover, emotions conveyed through speech often depend on the semantic content and context, making the interpretation more ambiguous, as the same tone of voice can represent different emotions depending on the words spoken. In contrast, facial expressions, although also culturally influenced, tend to be more universal and independent of verbal context.

The related works present significant advances but also notable limitations. Multimodal systems such as Zhang *et al*. [2020]'s may yield richer representations but often require complex and costly data acquisition, limiting practical deployment. Attention-based models like the one proposed by Minaee *et al*. [2021] demonstrate that shallower architectures can perform competitively, yet they demand specific mechanisms for identifying salient regions, which may add implementation complexity. Transfer learning approaches, such as IFER-DTFL Albraikan *et al*. [2022], benefit from pre-trained models but may face generalization issues when transferred to domains with limited resources or different cultural contexts. In contrast, VIRE focuses on developing and evaluating comparatively simpler CNN architectures and optimized through systematic hyperparameter tuning, aiming to balance performance and practical deployment. Although models like those proposed by Minaee *et al*. [2021] show promise, the DenseNet used in our experiments presents certain limitations, particularly its high computational cost due to dense feature map concatenations and tendency to overfit in deep configurations. These challenges motivate the exploration of lightweight CNNs with fewer layers and attention mechanisms, which could maintain or even enhance performance while addressing efficiency and regularization issues.

The work in Qu *et al*. [2023] used a CNN with three con-

volutional layers and one fully connected layer to identify emotions, using the public FER-2013 dataset. The main objective of the study was to explore the CNN's ability to recognize patterns, by testing different parameter settings such as the number of epochs, the optimizer, and the learning rate. This allowed the authors to better understand how deep networks work and to analyze their application in a real-world scenario. The highest accuracy achieved, 60.20%, was obtained with 200 training epochs, the SGD optimizer, and a learning rate of 0.019.

The research by Oguine *et al*. [2022] proposed a Deep Convolutional Neural Network (DCNN) model for real-time emotional recognition. To optimize performance, the model followed these steps: high-quality databases were selected, the facial region was then detected and cropped, and subsequently converted to grayscale. Data augmentation techniques were applied to prevent overfitting, and hyperparameter tuning, such as the number of feature maps and the number of network layers, was performed, resulting in an accuracy of 70.04%. Additionally, a Haar Cascade model was used for real-time facial detection. Although the experiments demonstrated that the proposed architecture surpassed other state-of-the-art methods in terms of performance and generalization, the research still had limitations in predicting the emotions "disgust" and "anger," due to the scarcity of training data. Another observed challenge was the lower generalization of real-time predictions, influenced both by the posed nature of the images used in training and by ambient lighting conditions.

In their paper, Lonkar [2021] proposed an FER (Facial Emotion Recognition) system that utilized a customized CNN architecture. To achieve optimal performance, the author employed several strategies, such as: data augmentation to increase the diversity of the training set and batch normalization to stabilize the learning process. The paper also highlighted the use of dropout to prevent overfitting, the application of class weights to handle data imbalance in the FER2013 dataset, and experimentation with popular optimizers like Adam and NAdam. The combination of these techniques resulted in a 70.10% accuracy for a single network on the FER2013 dataset.

The study by Attrah [2025] proposed an LSTM model that used blendshapes extracted from the MediaPipe library from videos to aid in real-time classification. Unlike previous approaches that employed classical methods like Haar Cascade, HOG, or SIFT, the study focused on efficiency and cost-effectiveness. Due to the complexity of FER2013, the work applied rigorous preprocessing to remove irrelevant data. This process included cleaning, indexing, and data augmentation techniques, such as rotation and horizontal flipping, to improve the model's generalization. A notable contribution of the research was the selection of only 27 of MediaPipe's 52 blendshapes, which significantly reduced the computational load without compromising performance. The results achieved were a 71% precision and an F1-score of 62%, which met the FER2013 benchmark.

Table 1. Model Performance Comparison on the FER2013 Dataset

Work	Strategy	# Epochs	Acc
[Qu *et al*., 2023]	CNN + SGD and LearningRate 0.019	200	60,20%
[Oguine *et al*., 2022]	CNN + Haar Cascade	100	70.04%
[Lonkar, 2021]	CNN + DataAug and Dropout	100	70.10%
[Attrah, 2025]	LSTM	5000	71%
This (Best)	DenseNet	150	64.43%

3 Methodology

This section presents the *Visual Identification of Recognition of Emotions* (VIRE), an e-health solution for identifying emotions through image analysis. To achieve this, it was modeled based on the analysis of three convolutional neural networks designed to extract deep visual features from images. Thus, VIRE leverages the image processing capability of CNNs to detect subtle nuances in facial expressions that correspond to specific emotional states.

VIRE operates in a home environment, using cameras to monitor emotions and recommend care for the elderly, children, and individuals who require constant supervision, as illustrated in Figure 1. The process involves three main steps: capturing images of faces (Label A), analyzing and identifying emotions using a convolutional neural network (Label B), and sending important notifications regarding the emotional state of the individuals being monitored to a caregiver on their mobile device (Label C).

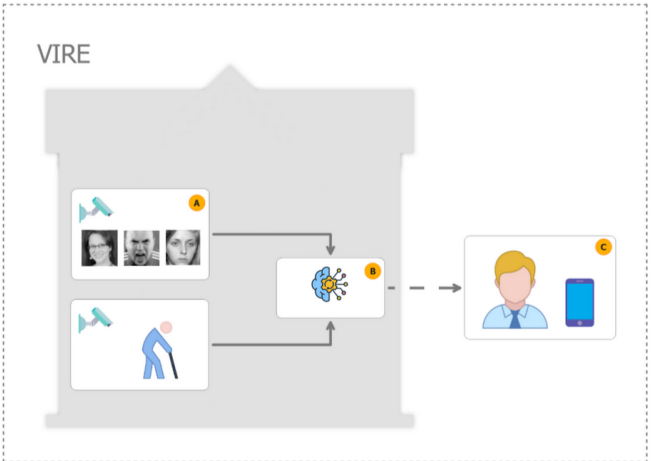

Figure 1. Overview of the VIRE operation

3.1 Data Collection

The detection of emotions by VIRE relies on identifying patterns in the faces of observed individuals. Some emotions, such as happiness, anger, sadness, fear, disgust, and surprise, are universally recognized through the same patterns [Ekman, 2003]. Therefore, this research used the FER2013 (Facial Expression Recognition 2013) dataset [Dumitru *et al*., 2013], which classifies these six basic emotions along with a neutral

one for model training. This dataset was compiled by Pierre-Luc Carrier and Aaron Courville from images found on the internet. It consists of images of faces with dimensions of 48 x 48 pixels, in black and white, separated into the seven mentioned emotions. Although widely used, the FER2013 dataset presents certain limitations that may influence model performance. As the images were collected from the internet, they vary in lighting conditions, angles, and facial occlusion, which may reduce the quality and consistency of the data. Moreover, there is limited information about demographic representation (e.g., age, ethnicity, gender), which can introduce dataset bias and affect generalization, especially in real-world applications.

During the training phase, 28,709 samples were submitted, which corresponds to about 80% of the dataset, distributed across each emotion as shown in Figure 2. For model performance evaluation, the 7,178 images previously separated in FER2013 were used, corresponding to approximately 20% of the dataset.

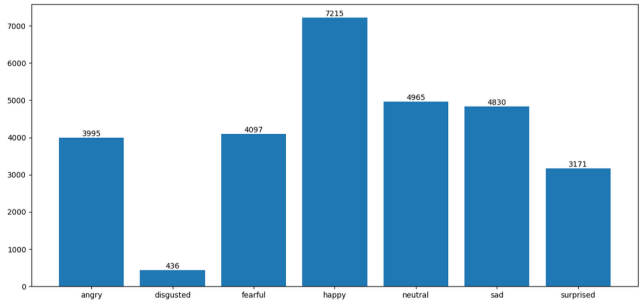

Figure 2. Distribution of images among classes

To optimize the performance of the neural network during training, it is essential that the data is adequately prepared for input into the network. This involves applying various preprocessing techniques to the dataset, one of which is normalization. As illustrated in Figure 3, normalization transforms an image into a floating-point tensor, scaling the values from 0 to 1. In Section A, an image file is decoded into RGB pixel matrices, and subsequently, in Section B, all values are converted to floating-point format. This transformation is crucial for allowing the neural network to operate with smaller values, thereby minimizing the risk of memory overflow and improving training efficiency. Additionally, this preprocessing step helps avoid issues during network processing [Gonzalez and Woods, 2008].

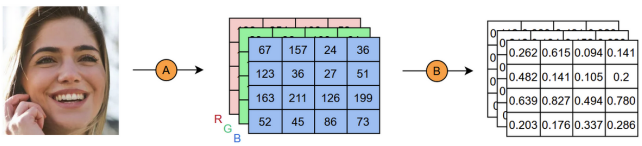

Figure 3. Normalization

To evaluate model performance, the following metrics were used: (i) Accuracy: this metric assesses the percentage of correct predictions relative to the total number of predictions [Mariano, 2021]; (ii) Confusion Matrix: in cases of class imbalance, accuracy may yield biased results [Castro and Braga, 2011]. Accuracy was chosen as a baseline metric to provide a general sense of model performance. However, given that the FER2013 dataset presents class imbalance (e.g., fewer samples for "disgust" compared to "happy" or "neutral"), accuracy alone may be misleading. For this reason, the confusion matrix was also adopted to analyze the classification results per class. This matrix enables the identification of specific patterns of misclassification, such as systematic confusion between similar emotions, and highlights how well the model performs on minority classes, offering a more comprehensive evaluation.

3.2 Recognition Mechanisms

The first mechanism proposed for analysis in this research is the CNN shown in Figure 4. The first six layers consist of two sets of convolutional layers, max pooling, and dropout. These layers are responsible for identifying small local patterns in images, such as edges and textures. This is followed by a Flatten layer, and finally, densely connected layers that will identify global patterns and return a classification for the image submitted to the model.

The custom CNN used in this research consists of a total of 10 layers. It begins with two convolutional layers using 32 and 64 filters, respectively, both with a kernel size of 3×3 and ReLU activation. Each convolutional layer is followed by a 2×2 max pooling layer and a dropout layer with a rate of 0.25 to reduce overfitting. After the convolutional blocks, the output is flattened and passed to two fully connected (dense) layers with 128 and 64 neurons, both also using ReLU activation. The final output layer uses a softmax activation function with 7 units to predict the emotion class. The network was designed to balance representational capacity with simplicity, and its performance was compared with deeper architectures such as AlexNet and DenseNet.

The other chosen model is the Densenet network. It is a convolutional neural network composed of sequences of convolutional layers, pooling, and ReLU activation functions [Huang *et al.*, 2017]. However, what distinguishes it from other convolutional networks is the concept of dense blocks, Figure 5. A dense block is a set of layers where there is a direct connection between a layer and all subsequent layers, thereby mitigating the gradient vanishing problem common in deep networks.

The DenseNet architecture used in this research, illustrated in Figure 6, begins with an input layer that accepts black and white images of 48 x 48 pixels. The first step involves a convolutional layer that applies a 3 x 3 filter, reducing the image to 24 x 24 pixels and 64 channels. This is followed by batch normalization and a ReLU activation, leading to a pooling layer that decreases the dimensions to 12 x 12 pixels.

After this initial phase, the network employs dense blocks that repeat normalization, activation, and convolution layers with 32 filters, increasing the complexity of the representation to 12 x 12 pixels and 192 channels. A transition layer then reduces the filters to 96 and the dimensions to 6 x 6 pixels. This process continues, with new dense layers increasing the dimensions to 6 x 6 x 224 and subsequently reducing them to 3 x 3 x 112. Finally, the network concludes with a dense layer that generates the model's classification from a vector of 240 units, resulting from the average pooling.

Alexnet is a CNN architecture that emerged as a proposal

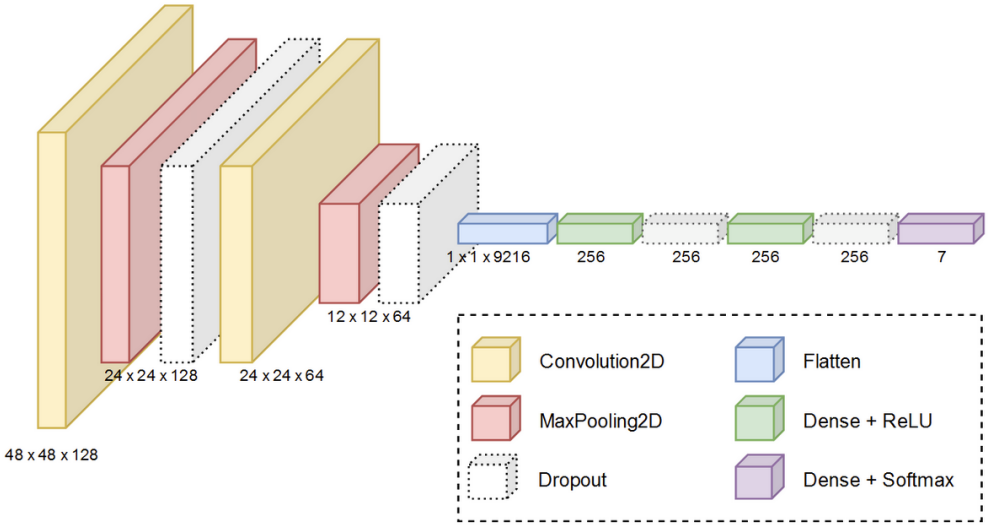

Figure 4. Custom CNN

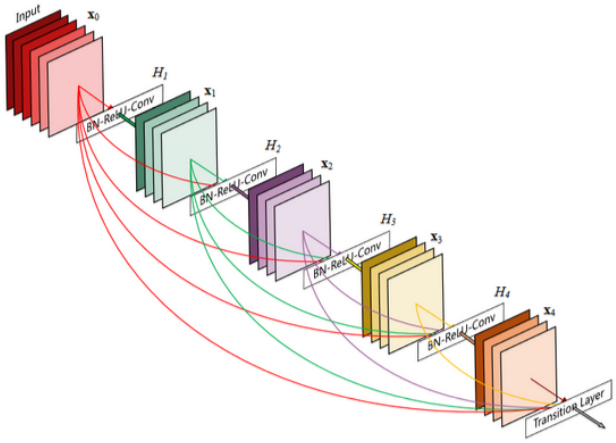

Figure 5. A dense block of 5 layers. [Huang *et al.*, 2017]

for the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [Krizhevsky *et al.*, 2012]. The competition required the model to be trained with a database containing 1,000 categories, each with 1,000 samples. Alexnet achieved significantly superior results compared to its competitors, establishing a new standard for performance in image classification tasks.

The structure, shown in Figure 7, consists of several layers that work together to process images. It begins with five convolutional layers that apply different filters to extract features from the images. The first layer uses 96 filters, followed by a pooling layer that reduces the dimensions of the image. The subsequent layers use smaller filters, with a pooling layer after the last convolutional layer to further simplify the data.

After the convolutional layers, the model includes two dense layers, each with 4096 neurons and ReLU activation, which assist in making final decisions. The model concludes with a classification layer that produces 1000 different outputs, representing the possible categories of the analyzed images. Table 2 presents a structural comparison of the architectures.

3.3 Optimization of Hyperparameter Search

While machine learning models have parameters that are iteratively adjusted during the training process—such as the weights of each neuron—machine learning algorithms rely on hyperparameters [Géron, 2019]. These are defined before training and remain constant throughout the process. Examples of hyperparameters include the learning rate, the number of epochs, and the batch size, each affecting the complexity and efficiency of the model [Bishop, 2016]. The proper selection of hyperparameters is crucial to avoid overfitting, ensuring that the model generalizes well to unseen data [Goodfellow *et al.*, 2016].

This research used Hyperopt, a search framework designed to optimize hyperparameter configurations from a list of possible combinations. The algorithm adopts a Bayesian approach, where, given a model, a score is assigned to each tested hyperparameter configuration [Bergstra and Bengio, 2012]. These configurations and scores are updated iteratively, aiming to maximize the score based on previous results [Bergstra *et al.*, 2015].

Thus, the following sets of values were utilized, as summarized in Table 3. The learning rate, batch size, and number of epochs were adjusted to explore different configurations. Additionally, various weight initialization algorithms were considered, including Glorot Uniform, He Normal, and Lecun Normal. The Tree-structured Parzen Estimator (TPE) algorithm was employed as the search method for hyperparameter optimization, ensuring that the models were refined for enhanced performance.

4 Results and Discussion

This section presents two evaluation scenarios of the VIRE, in which the models undergo hyperparameter tuning and an analysis of the training results for each identified configuration. Through the loss and accuracy metrics obtained from

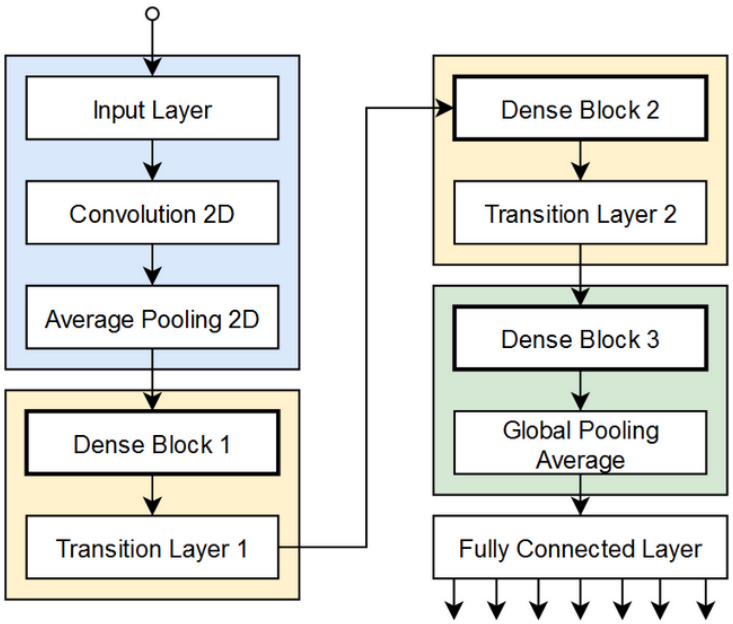

Figure 6. Architecture of the implemented Densenet

Table 2. Structural Comparison of the Architectures

Model	# Conv Layers	# Dense Layers	Activation Functions	Output Layer
CNN (custom)	2	2	ReLU (hidden), Softmax (output)	7 units (Softmax)
AlexNet	5	2	ReLU	1000 units (Softmax)
DenseNet	>20 (via dense blocks)	1	ReLU	7 units (Softmax)

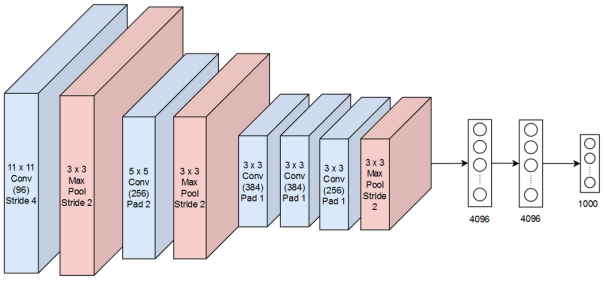

Figure 7. Architecture of Alexnet

Table 3. Hyperparameter Sets Used for Model Optimization

Parameter	Values
Learning Rate	0.0001, 0.001, 0.01
Batch Size	8, 16, 32, 64, 128
Epochs	25, 50, 100, 150
Initializer	Glorot Uniform, He N., Lecun N.
Search Algorithm	TPE

the training and validation sets, the performance of each architecture is compared to identify trends, biases, or issues. In this way, the performance of the VIRE in each evaluated scenario is highlighted.

4.1 Impact of Results in Scenario I

Hyperopt was run over ten trials, with each iteration involving the selection of a specific hyperparameter configuration, training the respective neural network, and evaluating its performance. This iterative process was repeated until the predefined number of trials was reached. Figure 8 shows the loss values observed for each configuration tested, comparing the three proposed architectures: AlexNet, DenseNet, and CNN. It is noticeable that DenseNet exhibited higher variability and increasing loss values in the later trials, while AlexNet and CNN showed more consistent and lower results, with AlexNet displaying the lowest loss values throughout the trials. This behavior can be attributed to the greater complexity of DenseNet, which may be more sensitive to the selection of inappropriate hyperparameters. DenseNet's architecture includes densely connected layers, where each layer receives inputs from all previous layers. While this enhances feature propagation and mitigates vanishing gradients, it also increases the number of paths through which information flows. Consequently, small variations in learning rate, batch size, or weight initialization can lead to unstable training dynamics. Furthermore, the increased depth makes the network more prone to overfitting on limited or noisy data, especially when hyperparameters are not optimally tuned. These factors combined contribute to the observed variability across trials.

It is important to highlight that the AlexNet model reached its lowest loss value in the seventh iteration, while DenseNet achieved its minimum in the fourth iteration, and the custom CNN reached the lowest value in the eighth iteration. These results, listed in Table 4, reflect important differences in the ideal configurations for each architecture. For instance, while AlexNet and DenseNet used the Lecun Normal initializer with quite similar learning rates, the custom CNN used the He

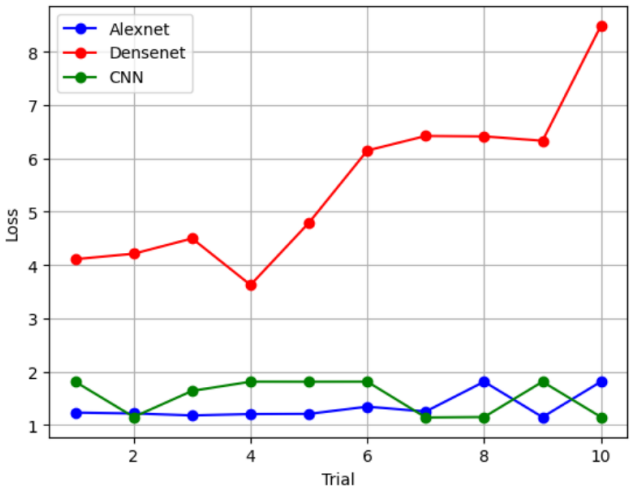

Figure 8. Performance of each configuration in the hyperparameter search phase

Normal initializer and had a significantly higher learning rate. These differences indicate that more complex architectures, such as DenseNet, may converge more quickly with fewer epochs but are more sensitive to fine-tuning, whereas models like the custom CNN may require more training to reach their best performance due to its initialization configuration and higher learning rate.

Table 4. Neural Network Model Configurations

Model	Learning Rate	Batch	Epochs	Initializer
AlexNet	0.000141436	64	50	Lecun N.
DenseNet	0.000139557	8	25	Lecun N.
CNN	0.000828395	8	50	He N.

4.1.1 Alexnet

The performance of the Alexnet network throughout the training can be observed in Figure 9. In the loss graph, it is noted that the value for the training set decreases progressively, indicating that the model fits well to this dataset. In contrast, for the validation data, it increases, showing that the model was unable to generalize to new data. This behavior is a strong indication of overfitting. This hypothesis is reinforced by observing that, in the accuracy graph, the result for each dataset maintains a significant discrepancy between them.

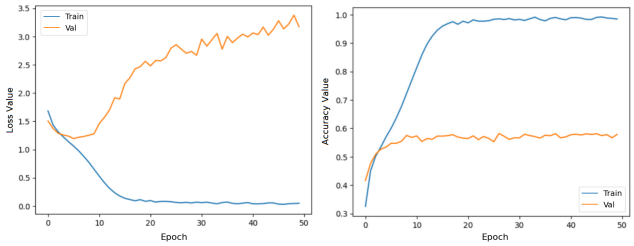

Figure 9. Performance of Alexnet in Training I

4.1.2 Densenet

In the DenseNet network, the loss graph presented in Figure 10 shows that the model adapts well to the training data, with a continuous reduction in loss until values close to zero. However, in the validation set, the loss exhibits the opposite behavior, gradually increasing over the epochs. When analyzing accuracy, it is observed that, while the model achieves growing and consistent performance in training, there is a significant difference compared to validation, with accuracy stabilizing at a lower level and with greater fluctuation. These patterns suggest that the network suffers from overfitting, excessively adapting to the training data and losing its ability to generalize to unseen data.

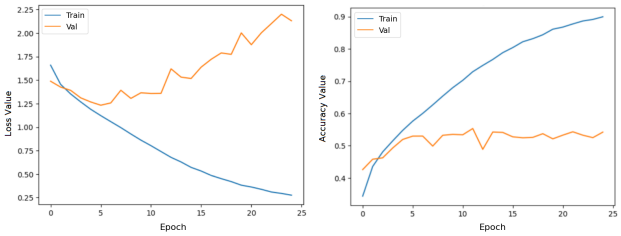

Figure 10. Performance of Densenet in Training I.

4.1.3 CNN

The initial observation to be made in the case of the CNN is that the loss and accuracy values in Figure 11 follow similar trajectories for both the training and validation sets, which suggests satisfactory generalization and partially excludes the possibility of overfitting. However, the low accuracy levels suggest that the model could be trained for more epochs to achieve better results.

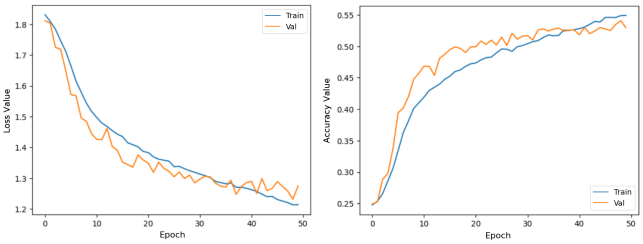

Figure 11. Performance of CNN in Training I.

Given this possibility, the model was subjected to training again, this time with 300 epochs. In Figure 12, it is observed that, between the 40th and the 60th epochs, the loss and accuracy values for the training and validation sets start to diverge. While the model continues to fit the training data, the loss values for the validation set remain relatively stable, showing no significant improvement. This increasing divergence indicates that the model is suffering from overfitting before achieving satisfactory results.

4.1.4 Results

Table 5 presents a summary of the results obtained for each training, and it is possible to conclude that, even using the parameters proposed by Hyperopt, all models faced generalization issues, resulting in overfitting at some point.

The Alexnet and Densenet networks achieved a loss close to zero and a training accuracy close to one, indicating a good

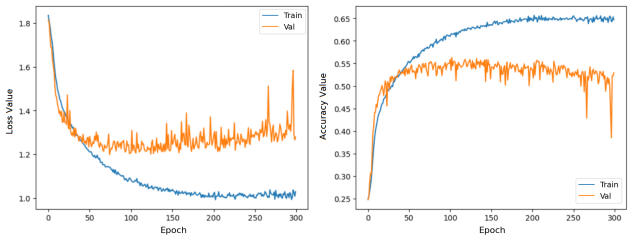

Figure 12. Performance of CNN with 300 epochs in Training I.

Table 5. Training Results in Phase I

Model	Train Loss	Val Loss	Train Acc	Val Acc
AlexNet	0.0466	3.1725	0.9846	0.5786
DenseNet	0.2761	2.1306	0.8996	0.5421
CNN	1.2139	1.2749	0.5493	0.5297
CNN 300e	1.0284	1.2782	0.6467	0.5295

fitting capacity over the training data. However, both presented a very high loss on the validation data, resulting in low accuracy and consequently unsatisfactory generalization capability. The CNN, despite initially showing promising results, when retrained for more epochs, showed little variation in the indicators and did not present an improvement over the results obtained with 50 epochs, besides evidencing the overfitting issue.

4.2 Impact of Results in Scenario II

In this phase, the training was conducted again, but with a dataset whose images underwent data augmentation transformations aimed at mitigating overfitting. From the test dataset, this technique generates new images by varying the existing ones through rotation, translation, distortion, zoom, and cropping. The applied transformations are listed in Table 6, along with their respective intensity levels.

Table 6. Hyperparameter Configuration for Data Augmentation

Attribute	Value
rotation_range	10%
width_shift_range	10%
height_shift_range	10%
shear_range	10%
zoom_range	10%
horizontal_flip	True
fill_mode	nearest

In Figure 13, it can be seen that, in this execution, the worst loss value is significantly lower than that of the previous execution. The AlexNet and custom CNN models exhibited considerable variation throughout the trials, achieving their best loss values in the second and third iterations, respectively. In contrast, the DenseNet showed much less variance, attaining the best loss value among the three in the third iteration. The configurations identified in the re-execution of Hyperopt are listed in Table 7.

4.2.1 Alexnet

Despite training with the transformed data, Alexnet displayed poor performance right after the twentieth epoch, similar to

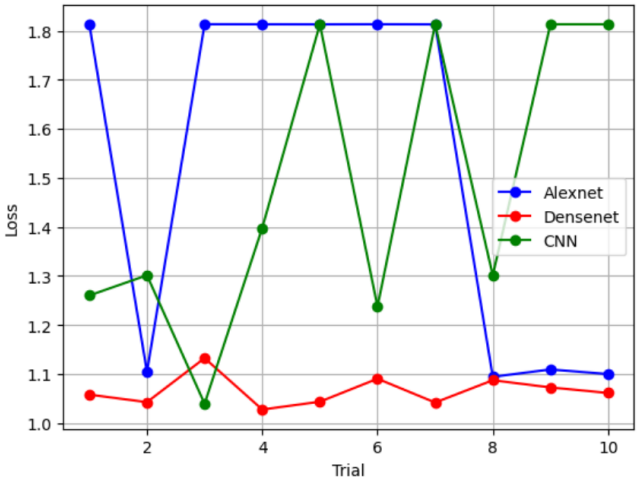

Figure 13. Performance of each configuration in the second hyperparameter search phase

Table 7. Hyperparameters suggested by Hyperopt in the second training.

Model	Learning Rate	Batch	Epochs	Initializer
AlexNet	0.000248422	8	150	Lecun N.
DenseNet	0.000433757	8	150	He N.
CNN	0.000167163	16	150	Glorot Uni.

the behavior in the previous training. It is noted, in Figure 14, that the network manages to fit the training data but fails to generalize, reflecting high loss values in the validation dataset. The accuracy remained stable and did not substantially exceed the values from the previous training.

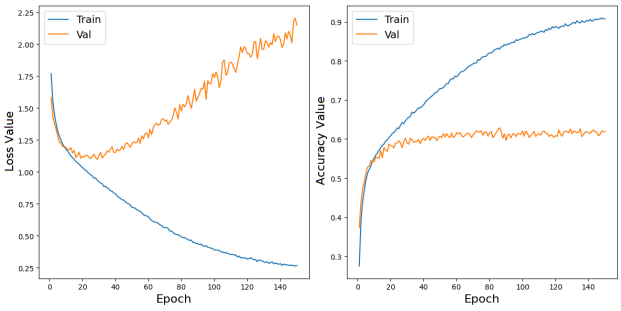

Figure 14. Performance of Alexnet in Training II.

4.2.2 Densenet

Densenet, on the other hand, achieved better results than the previous ones as observed in Figure 15. The loss remained relatively stable throughout the epochs, but despite the improvement, the validation value did not drop significantly, highlighting difficulties in generalization. This behavior reflected in the validation accuracy, which, although showed superior performance, was still not much better than Alexnet.

4.2.3 CNN

Observing the graphs in Figure 16, it is noted that the accuracy and loss maintained a certain regularity until the end. This stability, along with the small difference between the training and validation metrics, reveals that the model man-

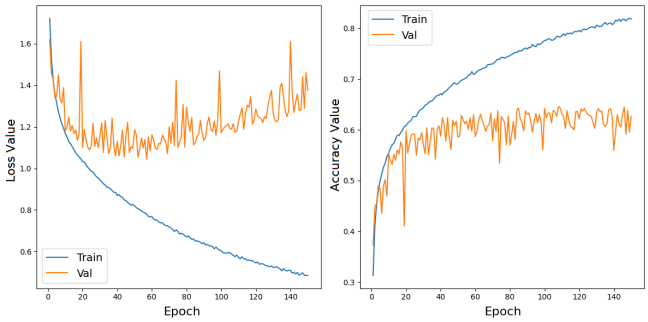

Figure 15. Performance of Densenet in Training II.

aged to handle overfitting, unlike the training without data augmentation.

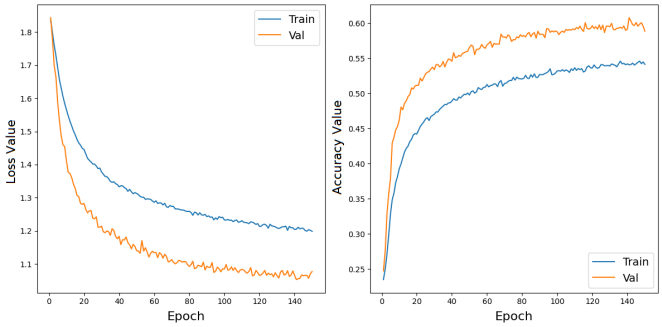

Figure 16. Performance of CNN in Training II.

4.2.4 Results

Table 8 provides a summary of the results achieved by each model during this new training phase. It is observed that, although the validation accuracy did not grow significantly compared to the previous training, all models benefited from the use of data augmentation, especially as it helped reducing the validation loss values. Among all the models, the one that benefited the most from this method was CNN, which showed a substantial improvement compared to the initial training. This improvement indicates that data augmentation was effective in enhancing CNN's ability to generalize to new data, making it more robust and efficient.

Table 8. Training Results in Phase II

Model	Train Loss	Val Loss	Train Acc	Val Acc
AlexNet	0.2570	2.1214	0.9115	0.6092
DenseNet	0.4944	1.2913	0.8163	0.6443
CNN	1.2052	1.0547	0.5399	0.6013

When analyzing the performance of the CNN, Figure 17 presents the confusion matrices for the three models. In the CNN matrix, it is highlighted that no instance of the "disgust" class was correctly classified, likely due to the limited number of samples for this emotion in the dataset. The "happiness" class also shows high confusion with other emotions. For instance, 439 instances of "happiness" were incorrectly classified as "sadness", which is comparable to the number of correct classifications for "happiness". Quantitatively, AlexNet also exhibited a similar confusion pattern, with approximately 41.3% of "happiness" instances mislabeled as "sadness", while DenseNet misclassified about 36.8% in the same way. These high rates suggest that the models struggle to distinguish between these two emotions, which may share overlapping visual features such as subtle smiles or neutral expressions. Furthermore, variations in facial expressiveness among individuals, cultural differences in how emotions are displayed, and the relatively low resolution of the FER2013 images may contribute to this ambiguity.

Overall, for the model to perform better, the number of predictions in the main diagonal should be greater compared to the other cells in the graph. But as the confusion matrix reveals, even with CNN having the best performance compared to the others, it still struggles to distinguish between some specific classes.

5 Conclusion

This research presented VIRE, an image analysis system designed for recognizing human emotions and assisting in mental health. The study explored the main concepts in emotion classification and utilized machine learning, computer vision, and deep learning techniques to implement the recognition module, which is based on a convolutional neural network. To ensure that the core of the system could classify images satisfactorily, the research proposed training and analyzing three different architectures: Alexnet, Densenet, and a CNN.

The training was conducted in two complementary phases, using the FER2013 dataset. Each phase aimed to mitigate issues identified in the previous phase, following the same sequence of steps: seeking the best hyperparameters and training each model with the found values; analyzing the models based on loss and accuracy metrics for the training and validation datasets; and finally, concluding the results of the stage.

Thus, it was possible for all models to recognize emotions, although with variations in performance among them. All achieved an accuracy of about 60% and demonstrated a reasonable capacity for generalization to new data. However, the results indicate that the best of the evaluated models still faces challenges with false positives and misclassifications, which may compromise the viability of the proposed system. Nevertheless, at the end of the training, the CNN showed a tendency to overcome overfitting over the epochs, maintaining good performance indicators and standing out as the best candidate to compose the recognition core of VIRE.

In future work, we intend to expand the evaluation to include newer, more widely adopted architectures, such as ResNet variants and transformer-based models. Furthermore, exploring zero-shot classification techniques using models like CLIP could enable VIRE to recognize emotional expressions even with limited annotated data, increasing the system's flexibility and generalizability. Furthermore, we plan to explore transfer learning techniques by fine-tuning pre-trained models on large-scale datasets like ImageNet. This approach is known to improve performance on facial emotion recognition tasks and could substantially increase VIRE's accuracy and robustness, especially when using deeper and more complex architectures.

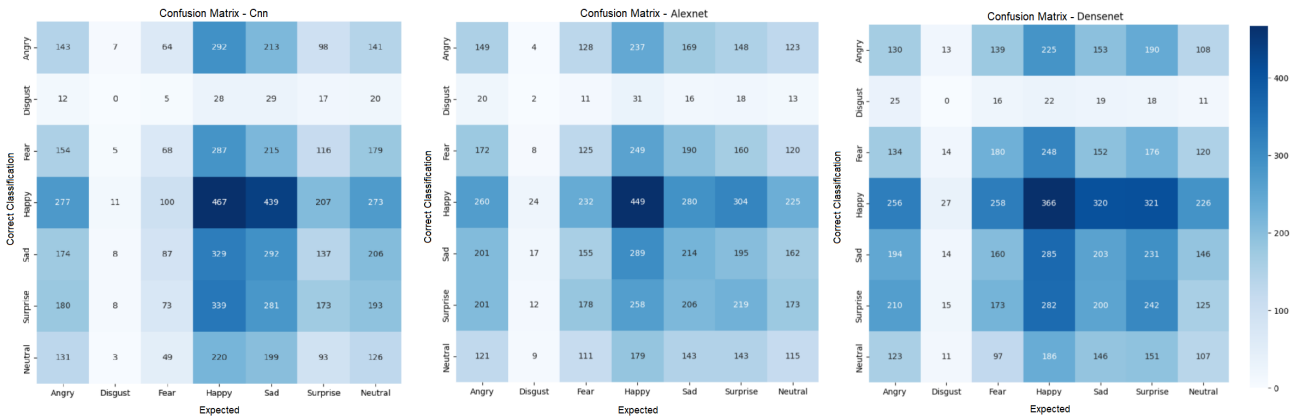

Figure 17. Confusion Matrix for each model

Declarations

Acknowledgements

The authors would like to thank the Coordination for the Improvement of Higher Education Personnel CAPES (CAPES) and the National Council for Scientific and Technological Development (CNPq).

Funding

No Funding.

Authors' Contributions

All authors contributed to the writing of this article, read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Availability of data and materials

Data can be made available upon request.

References

Albraikan, A. A., Alzahrani, J. S., Alshahrani, R., Yafoz, A., Alsini, R., Hilal, A. M., Alkhayyat, A., and Gupta, D. (2022). Intelligent facial expression recognition and classification using optimal deep transfer learning model. *Image and Vision Computing*, 128:104583. DOI: 10.1016/j.imavis.2022.104583.

American Psychiatric Association (2014). *DSM-5: Manual diagnóstico e estatístico de transtornos mentais*. Artmed Editora. Book.

Attrah, S. (2025). Emotion estimation from video footage with lstm.

Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *J. Mach. Learn. Res.*, 13(10):281–305. Available at:`https://jmlr.org/papers/v13/bergstra12a.html`.

Bergstra, J., Komer, B., Eliasmith, C., Yamins, D., and Cox, D. D. (2015). Hyperopt: a python library for model selection and hyperparameter optimization. *Computational Science amp; Discovery*, 8(1):014008. DOI: 10.1088/1749-4699/8/1/014008.

Bishop, C. (2016). *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer New York. DOI: 10.1117/1.2819119.

Castro, C. L. d. and Braga, A. P. (2011). Aprendizado supervisionado com conjuntos de dados desbalanceados. *Sba: Controle amp; Automação Sociedade Brasileira de Automatica*, 22(5):441–466. DOI: 10.1590/s0103-17592011000500002.

Chollet, F. (2018). *Deep Learning with Python*. Manning Publications, Shelter Island. Book.

Dalgalarrondo, P. (2019). *Psicopatologia e semiologia dos transtornos mentais*. Artmed, Porto Alegre, 3 edition. Book.

Dumitru, I., Goodfellow, W., Cukierski, Y., and Bengio (2013). Challenges in representation learning: facial expression recognition challenge. Available at:`https://kaggle.com/competitions/challenges-in-representation-learning-facial-expression-recognition-challenge`.

Ekman, P. (2003). *Emotions Revealed*. Times Books, New York. DOI: 10.1136/sbmj.0405184.

Ekman, P. and Friesen, W. V. (1971). Constants across cultures in the face and emotion. *Journal of Personality and Social Psychology*, 17(2):124–129. DOI: 10.1037/h0030377.

Filho, G. P. R., Meneguette, R. I., Mendonça, F. L. L. d., Enamoto, L., Pessin, G., and Gonçalves, V. P. (2024). Toward an emotion efficient architecture based on the sound spectrum from the voice of portuguese speakers. *Neural Computing and Applications*, 36(32):19939–19950. DOI: 10.1007/s00521-024-10249-4.

Frenay, B. and Verleysen, M. (2014). Classification in the presence of label noise: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 25(5):845–869. DOI: 10.1109/tnnls.2013.2292894.

Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly

Media. Available at:https://books.google.com.br/books?id=HHetDwAAQBAJ.

Gonzalez, R. and Woods, R. (2008). *Digital Image Processing*. Prentice Hall. DOI: 10.2307/1574313.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. Adaptive Computation and Machine Learning series. MIT Press. DOI: 10.1038/nature14539.

Huang, G., Liu, Z., van der Maaten, L., and Weinberger, K. Q. (2017). Densely Connected Convolutional Networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, Honolulu, HI, USA. IEEE Computer Society. DOI: 10.1109/cvpr.2017.243.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, page 1097–1105, Red Hook, NY, USA. Curran Associates Inc.. DOI: 10.1145/3065386.

Lonkar, S. (2021). Facial expressions recognition with convolutional neural networks. DOI: 10.48550/ARXIV.2107.08640.

Mariano, D. (2021). *Métricas de avaliação em machine learning: acurácia, sensibilidade, precisão, especificidade e F-score*. Alfahelix. DOI: 10.51780/978-6-599-275326-15.

Minaee, S., Minaei, M., and Abdolrashidi, A. (2021). Deep-emotion: Facial expression recognition using attentional convolutional network. *Sensors*, 21(9):3046. DOI: 10.3390/s21093046.

Oguine, O. C., Oguine, K. J., Bisallah, H. I., and Ofuani, D. (2022). Hybrid facial expression recognition (fer2013) model for real-time emotion classification and prediction. DOI: 10.48550/ARXIV.2206.09509.

Qu, D., Dhakal, S., and Carrillo, D. (2023). Facial emotion recognition using cnn in pytorch. DOI: 10.48550/ARXIV.2312.10818.

Solomon, A. (2014). *O demônio do meio-dia: uma anatomia da depressão*. Companhia das Letras, São Paulo, 2 edition. Tradução de Myriam Campello.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2015). Rethinking the inception architecture for computer vision. DOI: 10.48550/ARXIV.1512.00567.

Zhang, Z., Lin, W., Liu, M., and Mahmoud, M. (2020). Multimodal deep learning framework for mental disorder recognition. In *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)*, pages 344–350. DOI: 10.1109/FG47880.2020.00033.