Leonardo Leon Vera

Multi-view Attention Mechanism of Representations for Facial Emotions Recognition with Self-Taught Learning

Curitiba - PR, Brasil

2024

Leonardo Leon Vera

Multi-view Attention Mechanism of Representations for Facial Emotions Recognition with Self-Taught Learning

Apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de mestre em Informática.

Pontifícia Universidade Católica do Paraná - PUCPR

Programa de Pós-Graduação em Informática - PPGIa

Supervisor: Alceu de Souza Britto Jr.

Curitiba - PR, Brasil

2024

Leonardo Leon Vera
Multi-view Attention Mechanism of Representations for Facial Emotions Recognition with Self-Taught Learning
Leonardo Leon Vera. – Curitiba - PR, Brasil, 2024-
71 p. : il. (algumas color.) ; 30 cm.

Supervisor: Alceu de Souza Britto Jr.

Dissertação de Tese –
Pontifícia Universidade Católica do Paraná - PUCPR
Programa de Pós-Graduação em Informática - PPGIa, 2024.

1. Palavra-chave1. 2. Palavra-chave2. 2. Palavra-chave3. I. Orientador.

II. Universidade xxx. III. Faculdade de xxx. IV. Título

Leonardo Leon Vera

Multi-view Attention Mechanism of Representations for Facial Emotions Recognition with Self-Taught Learning

Apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de mestre em Informática.

Trabalho aprovado. Curitiba - PR, Brasil, October 24, 2024:

Alceu de Souza Britto Jr.
Orientador(a)

Curitiba - PR, Brasil
2024

Acknowledgements

I extend my heartfelt gratitude to my parents for their unwavering love and steadfast support. Their respect for, and encouragement of, my decisions during my relocation to another country and pursuit of my career has been invaluable.

I am profoundly indebted to Professor Alceu de Souza, whose expert guidance and unwavering support have been instrumental throughout the course of this research. Professor de Souza's dedication and wealth of experience have profoundly influenced the development of this thesis.

My sincere appreciation goes to all those who, in various capacities, contributed to the realization of this project. Their collaboration has been indispensable, and this achievement would not have been possible without their collective efforts.

This work is dedicated to those individuals who believed in me and served as a source of inspiration in reaching my academic goals. Their influence permeates every page of this thesis, and I am grateful for their enduring impact on my scholarly journey.

Abstract

No seguinte trabalho, é apresentado um sistema de reconhecimento de imagens de emoções utilizando aprendizado autodidata. Vários Autoencoders Convolucionais serão treinados para aprender a extrair características de um conjunto de dados de domínio diferente do conjunto de dados alvo. O conjunto de dados alvo será passado pelos codificadores para extrair as representações por imagem. A partir das representações geradas por cada codificador, uma rede neural com várias camadas totalmente conectadas será aplicada sem viés e com a ajuda dos mecanismos de atenção e da técnica de fusão, as multirrepresentações serão combinadas e ponderadas, aumentando o desempenho da rede com o objetivo de realizar a tarefa de classificação no conjunto de dados alvo. Com a validação cruzada Leave-One-Subject-Out (LOSO), a métrica acurácia para os conjuntos de dados JAFFE e CK+ foram de 68,94% e 88,60%, respectivamente, alcançando um desempenho comparável ao trabalho referenciado. **Palavras-chave**: STL, Attention Mechanism, Facial Emotion Recognition, Fusion model, Multi-view Learning.

Abstract

In this ensuing research endeavor, a sophisticated emotion image recognition system utilizing self-taught learning methodology is expounded upon. The approach involves training multiple Convolutional Autoencoders, enabling them to acquire proficiency in extracting features from a domain dataset distinct from the ultimate target dataset. Subsequently, the target dataset undergoes encoding processes through these trained autoencoders to derive distinct representations for each image.

The representations generated by each encoder form the basis for a neural network, incorporating several fully-connected layers. This network operates without bias, integrating attention mechanisms and a fusion technique. Through the judicious combination and weighting of multi-representations, the overall network performance is enhanced, with the primary aim of accomplishing the classification task on the target dataset.

The evaluation of the proposed system involves Leave-One-Subject-Out (LOSO) cross-validation. The attained accuracy metrics for the JAFFE and CK+ datasets stand at 68.94% and 88.60%, respectively. This achievement is notably comparable to the performance reported in the referenced work, validating the efficacy of the developed methodology.

Keywords: Self-Taught Learning, Attention Mechanism, Facial Emotion Recognition, Fusion model, Multi-view Learning.

List of Figures

List of Tables

List of abbreviations and acronyms

STL	Self-Taught Learning
CNN	Convolutional Neural Network
AE-CNN	Autoencoder Convolutional Neural Network
MVL	Multi-view Learning
SVM	Support Vector Machine
CRBM	Convolutional Restricted Boltzmann Machine
LOSO	Leave-One-Subject-Out
PCA	Principal Component Analysis
MSE	Mean Squared Error
CAN	Channel Attention Mechanism
SIFT	Scale-Invariant Feature transform
MDSTFN	Multichannel Deep Spatial–Temporal feature Fusion Neural network
EDLM	Ensemble Deep Learning Model
PNN	Parallel Neural Network
DNN	Deep Neural Network
LBP	Local Binary Patterns
ANN	Artificial Neural Networks
FMPN	Facial Motion Prior Networks
MLP	Multi-Layer Perceptron
RNN	Recurrent Neural Network
ACNN	Attention mechanism CNN
PHRNN	Part-based Hierarchical bidirectional RNN
MSCNN	Multi-Signal Convolutional Neural Network

Contents

1 Introduction

The construction of a model that generalizes well, most supervised learning methods necessitate a substantial amount of labeled training data. However, in numerous real-world scenarios, such data is often scarce, difficult to acquire, and expensive. Consequently, this presents a significant limitation in deep learning tasks. To address the lack of labeled data, knowledge transfer between tasks has emerged as an innovative learning framework. One specific method within this framework is self-taught learning (STL), which leverages data from distributions different from the target problem (RAINA et al., 2007). Unlike conventional transfer learning, STL does not require labeled data from an auxiliary domain. Instead, it learns representations without the need for labeling, effectively handling different data distributions. By doing so, STL provides a versatile approach to improve learning performance in situations where labeled data is insufficient, ultimately enhancing the model's ability to generalize across various tasks and domains.

One of the major advantages of STL has been the use of unlabeled data, as mentioned by the author in their studies (BHANDARI et al., 2018). This is due to the rationale behind STL, which incorporates concepts and principles borrowed from natural human learning processes. It is believed that unlabeled data helps to provide a solid foundation for high-level learning. In other words, STL is becoming a crucial component when (1) there is little labeled data for training, or (2) even when there is a sufficient amount of labeled data, using examples from outside the classes of interest enhances the learning process due to greater generalization power. This approach enhances model performance by leveraging the inherent value of diverse, unlabeled data to strengthen the learning framework.

This paper is mainly concerned with STL and its application in facial emotion recognition (FER). This application is challenging due to the diverse expressions influenced by individual and cultural factors, where it is very common to find datasets with people showing similar expressions for different emotions. This phenomenon is more noticeable in some countries than in others. That being said, we believe that STL has a great opportunity to achieve competitive results.

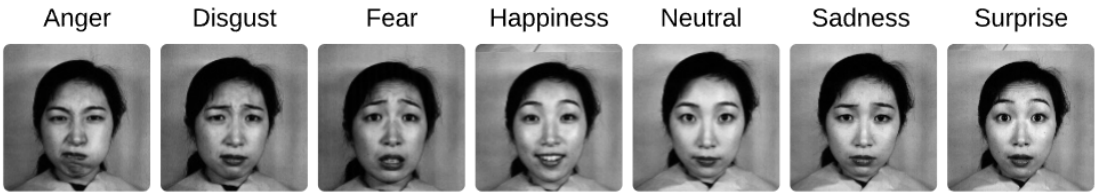
Our research expands on these studies by using Multi-View Learning, which generates several representations from a single image to enrich the information. Our neural network model integrates this attention mechanism to blend these diverse representations effectively, resulting in significant enhancements in FER accuracy metric. This approach not only builds on existing methods but also emphasizes the importance of utilizing multiple views and focused attention to better understand and classify facial expressions.

1.1 Definition of the problem

In the described context, our classification model of facial expressions in images has the main problems:

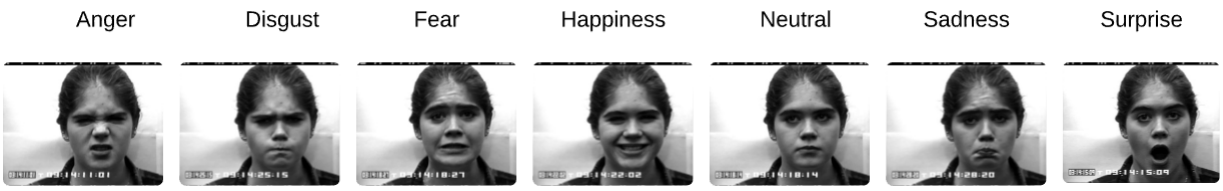
Small Dataset : Both datasets (JAFFE and CK+) used in this work do not contain more than 400 images and 7 classes, so if we applied a deep learning models directly using LOSO validation protocol, it is not as effective due to its small amount of data. Since it is well known in deep learning, the performance on vision tasks increases logarithmically based on volume of training data size (SUN et al., 2017). Some examples of these datasets we can see in Figure 1 and Figure 2. In addition, we have a problem that visually all the emotions are very similar. This makes it an even more difficult problem to solve. More details about the datasets can be found in a future section.

Figure 1 – JAFFE Dataset All Emotions Sample



Source: Author's original work

Figure 2 – CK+ Dataset All Emotions Sample



Source: Author's original work

Feature Extraction : Being a self-taught learning system, an unsupervised CNN Auto-Encoder model was used to extract features from facial expressions. In this context, we will use the latent vector or output of the encoder as the extracted representation of the face. This work employs various strategies to obtain the best representations, making achieving the best representations a challenge.

Classification model : Inspired by emotion recognition works in other modalities (like image, audio, video, text, etc), the attention mechanism was implemented with multi-view learning and fusion strategies from the representations for each image inside the Neural Network with fully-connected layers.

More details about all the datasets we will see in experimental results and discussions section.

1.2 Motivation

Currently, the field of emotion recognition from images is experiencing significant growth due to its wide range of applications in various disciplines. This research aims to explore and contribute to this field of study, providing a Deep Learning model capable of classifying facial expressions from visual images.

There are several solid reasons supporting this research:

Impact on Mental Health and Well-being: Emotion recognition can have a positive impact on people's mental health and well-being by allowing early detection of emotional disorders, providing feedback in emotion-based therapies and improving medical care in general.

Improved Human-Machine Interaction: Advances in emotion recognition can lead to more natural and effective interaction between humans and machines, improving the usability and adaptability of technologies, from virtual assistants to intelligent user interfaces.

Applications in Marketing and Advertising: In the business world, understanding consumers' emotions through images can revolutionize marketing and advertising strategies, personalizing the customer experience and increasing the effectiveness of campaigns.

Education and Personalized Learning: Education can benefit from emotion recognition models to adapt teaching and provide personalized feedback to students, thereby improving knowledge retention and motivation.

Interdisciplinary Research: This research offers opportunities for interdisciplinary collaboration by integrating concepts from psychology, neuroscience, computer science, and other disciplines, thus enriching our understanding of how emotions manifest visually.

Social and Cultural Impact: Advances in this field can influence culture and society, from applications in entertainment and art to understanding emotional expression in various cultures.

In summary, this thesis addresses a crucial topic in the field of artificial intelligence and visual perception: the recognition of emotions from images. The results of this research have a potential to significantly improve our understanding of human emotions and open new doors in a wide range of practical applications and fields of study.

1.3 Objectives

Develop a multi-view system which combine several unsupervised representations for facial emotions recognition problem. To achieve this, we will use attention mechanism with fusion strategy as it is explained in the next steps:

1. Determine the most effective strategy among modifying the latent vector dimension (L), altering the architecture (A), or utilizing both strategies (L/A) in order to generate more diversity.
2. Generate representations from autoencoders trained in an unsupervised manner.
3. Implement attention mechanism with concatenation of representations to improve the classification of facial emotions with neural networks.
4. Evaluate its competitiveness against the current state of the art.
5. Measure the impact of the joint attention mechanism with fusion strategy against the dynamic selection algorithm in the task of classifying unsupervised representations.
6. Assess the impact of the proposed method and determine its significance.
7. Define the best type of Attention Mechanism for the classification stage.

1.4 Research questions

The research questions propose the foundation for the investigation and guide the direction of the study. In this subsection, the specific questions that the research aims to address are formulated and presented.

- Is employing Convolutional Autoencoders trained with unlabeled databases competitive for the Face Expression Recognition task?
- Can the utilization of Attention Mechanism and Multi-View Learning contribute to enhancing the performance of a Face Expression Recognition system?

1.5 Work contributions

The Work Contributions subsection outlines the contributions made within the framework of this thesis.

- A new model based on the combination of autoencoders and attention mechanisms for Face Expression Recognition.

- Competitiveness analysis of the new model when the target dataset has small data.
- Two papers were produced throughout the work:

  - DELAZERI, Bruna; LEON, Leonardo; BARDDAL, J. P.; KOERICH, A. L.; DE, S. B. Alceu. Evaluation of self-taught learning-based representations for facial emotion recognition. 2022 International Joint Conference on Neural Networks (IJCNN), p. 1–8, 2022.

  - LEON, Leonardo; DELAZERI, Bruna; Britto, Alceu. Multiview Attention Model for Facial Emotion Recognition. 2024 European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), 2024.(Submitted)

1.6 Work structure

Chapter 2 is about exploring theoretical foundations and related works that have influenced this research. Chapter 3 explains the proposed method, including the generation of representations, classification model with multi-view and attention mechanism, and methodology. Chapter 4 shows the results obtained from the proposed method, including discussion, comparisons with various strategies via tables, comparison with reference works, and prospects for future work. Finally, chapter 6 named conclusions, summarizes whether the objectives outlined in the introduction were achieved.

2 Theoretical Foundation and Related Works

This chapter presents the works related to the proposed method for Face Expression Recognition task. These were selected for their relationship with feature extraction, combining and improving representations, and basic knowledge in order to understand all this work.

2.1 Self-Taught Learning

Proposed by (RAINA et al., 2007), STL, also described as unsupervised transfer or label-free data transfer, is a machine learning framework that requires little human supervision.

Deep architectures take advantages of the ability of learning hierarchical and high-level features from low-level features. Our STL will include deep architectures, so for simplicity, we will refer to self-taught learning and deep self-taught learning in the same way.

STL catched the attention of researchers, and empirical works already demonstrate its significance in statistical machine learning (BASTIEN et al., 2010). STL simulates the architectural depth of brain, which processes information through multiple stages of transformation and representation, and aims at learning hierarchical and high-level features obtained by the composition of low-level features (GAN et al., 2014).

In general terms, we will outline the stages followed in this framework:

- **Representation Learning (Step 1)**: High-level representation is learned through unlabeled data, which does not necessarily present the same distribution as the labeled data of the target domain. Our way to learn that representations is using AECNN.

- **Feature Building (Step 2)**: Feature vectors are extracted from the labeled data (target domain) using the representation learned in Step 1. In our system, encoders from the AECNN are used to extract those representations for target domain.

- **Training a Classifier (Step 3)**: The feature vectors (representations) extracted in Step 2 are used to train a classifier (Fully-Connected Neural network with Attention mechanism).

2.2 CNN Autoencoder

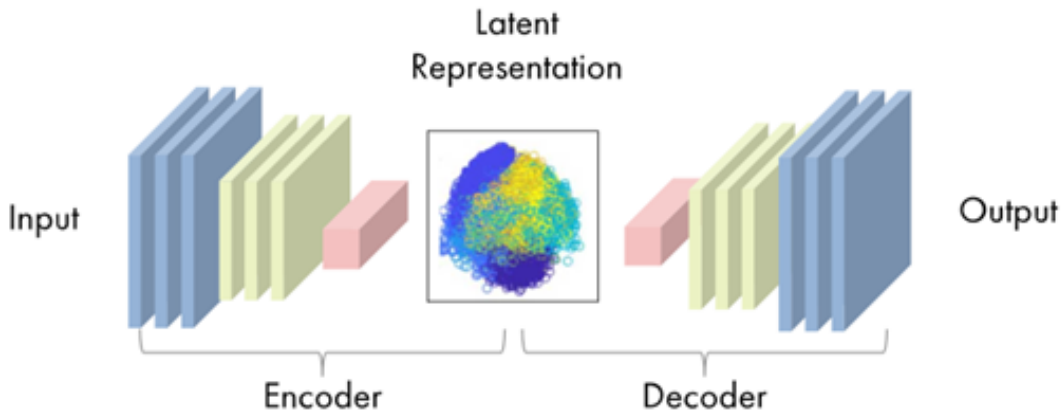
The combination of autoencoders and CNN is used when the input data is images or data with a similar structure. By incorporating convolutional layers in both the encoder and decoder of the autoencoder, the network can learn more robust and meaningful representations for images. The convolutional layers help capture spatial patterns and feature hierarchies, which can be crucial for the effectiveness of the model (LECUN et al., 1998) (GOODFELLOW; BENGIO; COURVILLE, 2016).

In the encoder, convolutional layers are used to extract important features from the input. In our work, it will generate a new target database with new labeled features. Each convolutional layer detects specific patterns in local regions of the image. After passing through the convolutional layers, the data is reduced to a latent space representation. This representation should contain the most important and compact information from the original data (HINTON; SALAKHUTDINOV, 2006).

The network is trained by minimizing the difference between the input and the reconstructed output (RAINA et al., 2007). This process adjusts the weights of the convolutional layers to learn meaningful representations.

In summary, a CNN-based autoencoder is effective for learning efficient representations of visual data, such as images, which can be very useful in various applications like anomaly detection (ZHOU; PAFFENROTH, 2017), text generation (BOWMAN et al., 2016), image generation (KINGMA; WELLING, 2022), image reconstruction, image noise removal, etc. In our work, this efficient representation is the feature extracted from each image.

Figure 3 – Convolutional Autoencoder



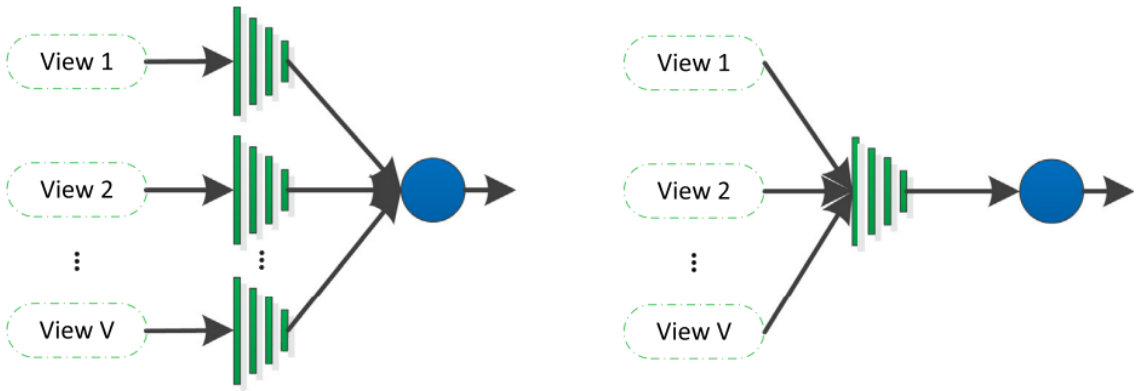
Source: Mathworks

2.3 Multi-view Learning

Multi-view learning (MVL) has attracted increasing attention and achieved great practical success by exploiting complementary information of multiple features or modalities. Recently, due to the remarkable performance of deep models, deep MVL has been adopted in many domains, such as machine learning, artificial intelligence and computer vision (YAN et al., 2021).

In Figure 4, we observe two widely used strategies in the field of MVL (Multi-View Learning): one-view-one-network and multi-view-one-network. The first one (one-view-one) consists of processing a network for each view as input, and then merging the outputs of network into a final representation as illustrated in Figure 4a, which will serve as input for some subsequent processing. The second one (multi-view-one) involves network receiving as input all the representations, previously fused by some "Fusion" technique (for example concatenate or pooling) and then generating a final representation as illustrated in Figure 4b .

Figure 4 – Multi-view Learning



(a) One-view-one-network strategy (b) Multi-view-one-network strategy

Source: (YAN et al., 2021)

For our work, we will employ the initial strategy of One-view-one-network. In this approach, each representation undergoes an attention mechanism, and the output from this mechanism results in a representation of the same dimension as the input. Subsequently, these representations are fused, giving rise to a final representation. This final representation serves as input for the Fully-Connected layers.

2.4 Fusion strategy

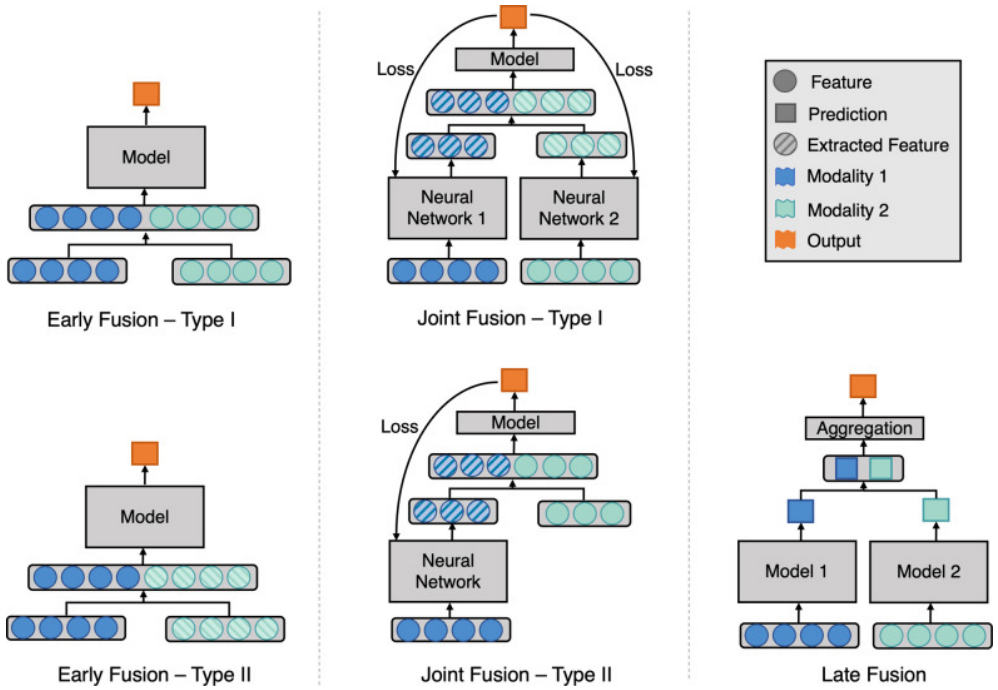
As explained in (SC et al., 2020), there are three approaches to multi-modal learning fusion, but instead of multi-modal we will talk about multi-views:

Early Fusion: It combines different input feature views into a single feature vector before feeding into a machine learning model. This can be achieved through concatenation, pooling, or using a gated unit. It distinguishes between early fusion Figure 5 type I, which merges original features, and early fusion Figure 5 type II, which fuses extracted features, such as predicted probabilities from different views.

Joint Fusion: As seen in Joint fusion Figure 5, It merges learned feature representations from intermediate layers of neural networks with features from other views. Unlike early fusion, the loss is back-propagated to the neural networks extracting features during training, enhancing the representations.

Late Fusion: As seen in late fusion Figure 5, It leverages predictions from separate models to make a final decision, known as decision-level fusion. Models are trained with different views, and the final decision is made using an aggregation function, such as concatenating, averaging, majority voting, weighted voting, or a meta-classifier based on the predictions of each model. The choice of the aggregation function is empirical and depends on the application and input views.

Figure 5 – Fusion strategies



Source: Types of strategies by (SC et al., 2020)

In our application, we use Early and Late Fusion. For example, when we concatenate all the representations before the attention mechanism, we are using Early fusion. And when we concatenate all the attended representations after the attention mechanism, we are using Late Fusion.

2.5 Attention Mechanism

The Attention Mechanism allows a model to focus its processing on specific parts of the input data, rather than treating all the information uniformly. Inspired by human visual attention, this approach is widely used in deep learning applications that require selecting and highlighting certain parts of the data. By prioritizing the most relevant information, the attention mechanism improves the model's accuracy and efficiency.

For example, in **machine translation**, the attention mechanism helps the model focus on key words and phrases in the text (LUONG; PHAM; MANNING, 2015). In **speech recognition**, it highlights the most relevant segments of audio (CHOROWSKI et al., 2015), while in **computer vision**, it identifies the most important regions of an image for recognition tasks (MNIH et al., 2014). Similarly, in **image caption generation**, the attention mechanism helps describe not only the objects in the image but also the relationships between them (XU et al., 2015).

In our work, we employ the attention mechanism for the classification model in such a way that it effectively captures the intra and inter-representations relationships simultaneously through interactions between representations and within itself. In other words, through interactions involving the joint vector and correlation matrices using linear and non-linear operations, the attention mechanism learns which representations to focus on more and the specific value for each representation. As a result, we obtain a better final representation.

2.6 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a dimensionality reduction technique widely used in machine learning and multivariate statistics. The main objective of PCA is to transform a set of correlated variables into a new set of uncorrelated variables known as principal components. These principal components are linear combinations of the original variables and are ordered according to the variance they explain (JOLLIFFE, 2002). This means that the first components capture the largest amount of variability in the data, allowing the primary structure to be represented in fewer dimensions.

Mathematically, the PCA process starts by centering the data around the mean, followed by calculating the covariance matrix between the variables. Then, the eigenvectors and eigenvalues of this matrix are computed to determine the principal directions of maximum variance, i.e., the principal components. The eigenvectors represent the new dimensions, while the eigenvalues indicate how much of the data's variance is explained by each of these vectors.

A key property of PCA is that the principal components are orthogonal to each other,

ensuring that there is no redundancy in the selected dimensions. This helps eliminate multicollinearity among the original variables, which can be critical in models where redundancy must be avoided to improve the interpretability of results (ABDI; WILLIAMS, 2010).

PCA is also effective at filtering noise in the data, as the principal components that explain the least amount of variance often correspond to irrelevant variability or noise. In machine learning applications, dimensionality reduction through PCA can help decrease model complexity, improve computational efficiency, and facilitate faster training of algorithms, especially when dealing with high-dimensional datasets.

In our work, PCA is applied to reduce the dimensionality of the latent vector or image representations. By reducing these representations to their most significant components, we simplify the complexity of the attention model, which not only speeds up training but also helps filter out noisy features that could negatively impact model performance.

2.7 Related works

In this section, we will present some previous works that served as a foundation and inspiration for our research.

In (DELAZERI et al., 2022), we find a work that aims to generate unsupervised representations through the concept of STL for facial emotion recognition. The idea behind it is to use various CNN autoencoders to generate diverse representations, and then train classification models to predict the respective emotion for each face.

In (BHANDARI et al., 2018), the authors perform experiments to compare the performance of STL vs transfer learning for JAFFE in facial emotion recognition task. Unlike us, they use CIFAR-10 as an auxiliary dataset. In the end they come to the conclusion that the performance of transfer learning is superior to STL because this one forces it to remain in a local minimum while transfer learning has a greater probability of finding global minima.

Building upon the aforementioned, in our work, we utilized the STL concept and the leave-one-subject-out (LOSO) strategy to validate our method and see if we could improve upon the state-of-the-art.

Nowadays, NLP models have improved their performance due to the "attention" technique (A. et al., 2017), allowing a model to focus on specific parts of an input during execution. Instead of processing the entire input uniformly, attention assigns variable weights to different parts, enabling the model to pay more attention to the most relevant and contextually important parts. This same idea is applied in the field

of computer vision (MH. et al., 2022), involving the fusion of features, filters, channels, feature maps, etc., or whatever the respective data structure may be. An example of this is (PRAVEEN et al., 2022), where a cross-attention model is proposed that can effectively exploit complementary inter-modal relationships before and after merging image and audio representations, allowing for accurate prediction of emotions.

Facial Expression Recognition (FER) techniques (SAJJAD et al., 2023) have evolved significantly, encompassing both conventional learning-based methods and deep learning-based approaches. Conventional methods, such as SVM, SURF, SIFT, HOG, and Naive Bayes, rely on manual feature engineering techniques and have been extensively employed in FER tasks. In contrast, deep learning has emerged as a powerful paradigm, leveraging hierarchical networks to automatically learn features and perform end-to-end FER. This introduction provides an overview of both conventional and deep learning-based FER techniques, highlighting their distinct approaches and recent advancements.

2.7.1 Conventional Learning-based FER techniques

Conventional learning methods encompass features extractor and descriptors like LBP, SURF, SIFT and HOG, and classifiers like SVM, Naive Bayes and MLP. Traditional practices involve manual feature engineering techniques, such as pre-processing and data augmentation, before feature extraction.

Here are some examples of the aforementioned techniques that will be discussed. A mapped LBP feature was introduced for illumination-invariant FER. SIFT features, resilient against image rotation and scaling, are employed for multi-view FER tasks. The amalgamation of multiple descriptors of texture, orientation, and color, utilized as inputs, aids in enhancing network performance.

In a similar vein, part-based representation involves feature extraction through the exclusion of nonessential elements from the image and the utilization of key components that are pertinent to the task. According to researchers in (CHEN et al., 2018), three regions of interest (ROIs), namely the eyes, mouth, and eyebrows, have been identified as significantly correlated with emotional variations. Table 1 provides an overview of recently published conventional machine learning methods for Facial Expression Recognition (FER).

Table 1 – Contributions of Face Expression Recognition Conventional Techniques for JAFFE and CK+ datasets

Reference	Technique	Contribution
(SAJJAD et al., 2019)	ORB, SVM	ORB features were extracted and fed into an SVM.
(MAKHMUDKHUJAEV et al., 2019)	LPDP	An edge descriptor LPDP was developed which considered statistical details of pixel neighborhoods to collect meaningful and reliable information.
(FERNANDEZ et al., 2019)	FERAtt	An end-to-end architecture which focused on human faces was proposed. The model applied a Gaussian space representation to recognize an expression.
(WANG et al., 2019)	CNN, C4.5 classifier	Features from CNN are combined with C4.5.
(DAMI et al., 2020)	3D CNN	Deep spatiotemporal features were extracted based on deep appearance and neural network.
(WANG et al., 2020)	CNN	An activation function was proposed for CNN models, and a piecewise activation technique was proposed for the procedure of FER tasks.
(JING et al., 2020)	LBP	LBP features extract images textures to catch small faces movements. The network comprised features extraction, attention module, reconstruction module, and classification module components.
(LIANG et al., 2020)	LBP, MSAU-Net	Fine-grained FER in the wild was primarily considered and FG-Emotion was proposed. FG-Emotions provided several features such as LBP and dense trajectories that facilitated the research.

2.7.2 Deep learning-based FER techniques

In recent times, deep learning has garnered significant attention in the realm of research and has demonstrated cutting-edge performance across various domains (DENG; YU et al., 2014), including computer vision (ULLAH et al., 2022), (ULLAH et al., 2021b) and time-series analysis and prediction (ULLAH et al., 2021a). Unlike traditional methods, deep learning endeavors to capture complex abstractions by employing hierarchical networks that consist of numerous nonlinear transformations and representations. In contrast to conventional approaches for Facial Expression Recognition (FER), where feature extraction

and classification are separate processes, deep networks carry out FER in an integrated manner, thus enabling an end-to-end solution. Deep Neural Network (DNN) models are capable of extracting features, which are subsequently fed into a conventional classifier like Support Vector Machines (SVM) or Random Forest (RF) models for further processing. Additionally, recent studies have introduced a covariance descriptor ((RAZAVIAN et al., 2014), (DONAHUE et al., 2014)) computed from deep Convolutional Neural Network (CNN) features, with classification being executed by Gaussian kernels on a symmetric positive definition. Table 3 and 4 provide an overview of recently published conventional machine learning methodologies.

Table 2 – Table of State-of-the-Art Supervised Models for JAFFE and CK+ datasets

Approach	**Protocol**	**Database**	**Performance**
Transfer Learning DenseNet-161 (AKHAND et al., 2021)	K-Folds	JAFFE	99.51%
Multi-head Self-Attention (WASI et al., 2023)	Holdout	JAFFE	96.67%
ViT + SE (AOUAYEB et al., 2021)	K-Folds	JAFFE	94.83%
Attentional CNN (MINAEE; MINAEI; ABDOLRASHIDI, 2021)	Holdout	JAFFE	92.8%
Patch and Attention MobileNet (NGWE et al., 2023)	K-Folds	CK+	100.0%
ViT + SE (AOUAYEB et al., 2021)	K-Folds	CK+	99.8%
Frame Attention Network (MENG et al., 2019)	K-Folds	CK+	99.7%
Nonlinear Eval on SL + SSL Puzzling (POURMIRZAEI; MONTAZER; ESMAILI, 2022)	K-Folds	CK+	98.23%
Attentional CNN (MINAEE; MINAEI; ABDOLRASHIDI, 2021)	Holdout	CK+	98.0%

Table 3 – Contributions of Face Expression Recognition with Deep Learning Mechanisms for JAFFE and CK+ datasets

Reference	Technique	Contribution
(MOHAMMADPOUR et al., 2017)	CNN	A method based on the LeNet-5 architecture, comprising five trainable parameter layers, two subsampling, and a fully connected layer, was proposed. A SoftMax function was used for the final FER classification.
(JAIN; ZHANG; HUANG, 2020)	PHRNN, MSCNN	A deep evolutional spatial–temporal network (composed of PHRNN and MSCNN) was used to extract the partial-whole, geometry-appearance, and dynamic-still information, thus effectively improving the performance of FER.
(JAIN; ZHANG; HUANG, 2020)	LSTM-CNN	For the facial label prediction, the authors used LSTM-CNN.
(HASANI; MAHOOR, 2017)	3D inception-ResNet-LSTM	A model with layers of an Inception-ResNet model were followed by an LSTM unit was proposed. This method extracted temporal and spatial relations within facial images between different frames in video
(BHANDARI et al., 2018)	STL, Pre-trained CNN	Explores transfer learning and self-taught learning in facial expression classification, emphasizing transfer learning's superiority and deep network layer correlation during weight transfer.
(LI et al., 2018)	CNN, ACNN	A CNN with ACNN was proposed to perceive occlusion regions in the face and emphasize the most discriminative unoccluded regions.
(JAIN et al., 2018)	CNN-RNN	A hybrid CNN and RNN model was used for FER.
(SAJJANHAR; WU; WEN, 2018)	Pre-trained CNN Inception, VGG-Face	Pre-trained state-of-the-art models were used for FER.
(KARTALI et al., 2018)	AlexNet CNN, FER-CNN, SVM, MLP	Five different techniques for real-time basic expression recognition from images were compared.
(RUIZ-GARCIA et al., 2018)	Hybrid CNN-SVM	Humanoid robot for real-time FER was proposed based on convolutional self-learning feature extraction and an SVM classifier.
(CHEN et al., 2019)	FMPN	An FER framework called FMPN was proposed, in which a branch was introduced for facial mask generation to focus on muscle movement regions.

Table 4 – Contributions of Face Expression Recognition with Deep Learning Mechanisms for JAFFE and CK+ datasets

Reference	Technique	Contribution
(SUN; LV, 2019)	SIFT, CNN	Features were extracted from SIFT and CNN.
(REFAT; AZLAN, 2019)	Deep CNN	Different deep learning methods were employed, with a CNN selected as the best algorithm for FER.
(CHEN et al., 2019)	DCNN	A two-staged framework based on a DCNN was proposed that was inspired by the non-stationary nature of facial expressions.
(SUN et al., 2019)	MDSTFN	A multi-channel network was proposed to fuse and learn spatiotemporal features for FER. An optical flow was extracted from the changes between the neutral and peak expression.
(HUA et al., 2019)	CNN, EDLM	Based on ensemble learning model, an algorithm was proposed comprising three sub-networks with different depths. The sub-networks comprised CNN models that were trained separately.
(XI et al., 2020)	PNN, CNN, Residual Network, Capsule Network	A PNN model designed to combine texture features was applied for FER. This network was constructed using CNN, capsule network, and residual network models.
(ZHANG et al., 2016)	Firefly algorithm	An FER technique was proposed based on the firefly algorithm, which was mainly used for feature optimization.
(PENG et al., 2016)	DNN	A DNN was proposed for the classification of facial expression based on a naturalistic dataset.
(TALELE et al., 2016)	LBP, ANN	LBP was implemented for feature extraction from images. GRNN was implemented for the classification of FER based on frame features.
(LEE et al., 2016)	Deep learning methods	A DNN was proposed based on a webcam for a smart TV environment to recognize human facial expressions.

2.7.3 Self-Taught Learning Models techniques

In Table 5, we present three works that represent the state of the art using the self-taught learning framework for three different facial expression recognition datasets:

JAFFE, Cohn-Kanade (CK+), and LFW. In all of these works, the first stage involved a feature extractor based on an unsupervised model such as Sparse Autoencoder, ICA, and CRBM, which were trained on unlabeled datasets. These datasets must belong to a domain significantly different from the labeled dataset due to framework constraints. Then, in the second and final stage, CNN, SVM, and Linear SVM classifiers were trained using the extracted representations from the labeled datasets, employing cross-validation and accuracy as the evaluation metric.

Table 5 – Table of State-of-the-Art Self-Taught Learning Models for JAFFE and CK+ datasets

Paper	Feat. Extractor	Classifier	Unlabeled Dataset	Labeled Dataset	Performance
(BHANDARI et al., 2018)	Sparse Autoencoder	CNN	CIFAR 10	JAFFE	56.45%
(LONG et al., 2012)	ICA	SVM	Hateren's Natural Video	Cohn-Kanade	80.15%
(HUANG; LEE; LEARNED-MILLER, 2012)	CRBM	Linear SVM	Kyoto Nat ural Images	LFW	87.77%

2.8 Final Considerations

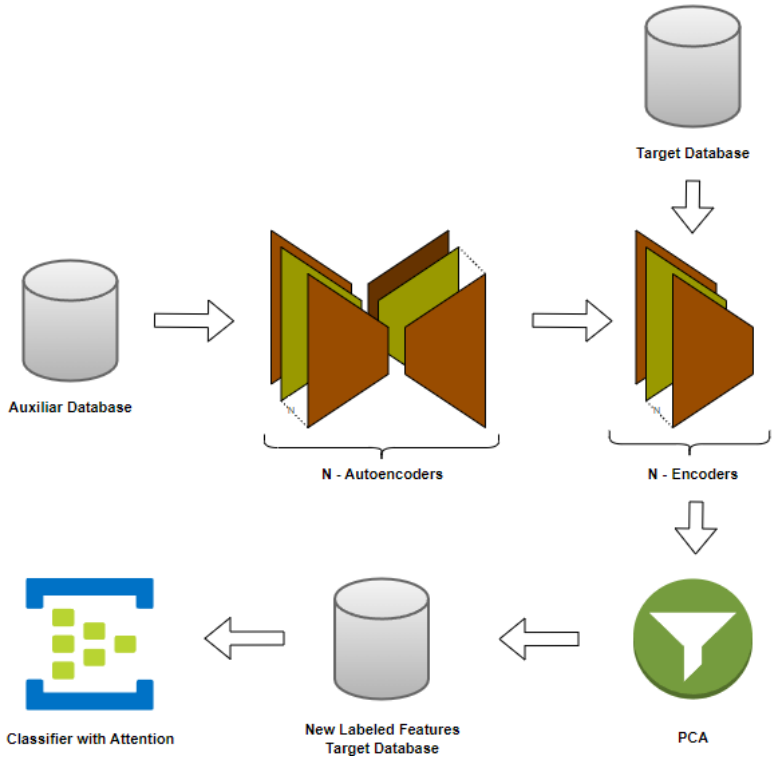
As seen in previous works, the tendency is to continue using CNN for this FER task and since it is a NN, this demands a large amount of data for training, in addition to fine tuning, so for a specific case it will always be a challenge. One way to deal with this large demand for data is to use STL to reuse the learning on a different dataset.

After explaining all the concepts of the work, we will proceed with the explanation of the proposed method in the next section.

3 Proposed Method

In this section we describe the proposed method using Self-Taught learning framework for the Face Expression Recognition problem, which can be found illustrated in Figure 6 where we can see a pool of N autoencoders trained from an auxiliary dataset (unsupervised). From them, we will extract the encoders that, once applied to the target dataset (supervised), will generate representations. These representations will then pass through a dimensionality reduction method to reduce the representation space and then through a classification model with an attention mechanism, which will be trained. Finally, with the trained classification model, we will perform inference on the target dataset to obtain the accuracy metric.

Figure 6 – Self-taught Learning Process



Source: Author's original work

In the Algorithm 1 we have three input variables: images of the auxiliary dataset (X_a), images of the target dataset (X_t) and labels of the target dataset (y_t).

Algorithm 1: Proposed STL Framework Algorithm

Data: X_a, X_t, y_t
Result: Accuracy metric

```
1  pool_hyper_combination = Get_Hyper_Combination(); // Choose a strategy
2  size_pool_hyper_combination = Lenght(pool_hyper_combination);
3  att_model = Get_Attention_Model();          // Initialize attention model
4  for i ← 1 to size_pool_hyper_combination do
5      autoencoder = Get_Autoencoder(pool_hyper_combination[i])
6      W_enc, W_dec = Train_Autoencoder(autoencoder, X_a);
7      X'_t = Get_Representations(W_enc, X_t);
8      X'_t_rd[i] = Reduction_Dimensionality_Method(X'_t);
9  accuracy = Train_Attention_Model(att_model, X'_t_rd, y_t);
10 return accuracy;
```

We start defining which *pool_hyper_combination* (line 1) (pool of combination of hyperparamenters to init the autoencoders) we will use to train the autoencoders, as shown in Table 6. For example, we can initialize an autoencoder using strategy A, which means that we are going to define many autoencoders as possible only modifying the deep of the network using values between 1 and 5 as showed in Table 6.

Immediately, we initialize the attention model in the variable *att_model* using the *Get_Attention_Model* function (line 3).

Table 6 – Table of Hyperparameters strategy

Strategy		Parameters
CAE - Network Arquitecture (A)	Network Depth	N = {5, 4, 3, 2, 1}
CAE - Latent Vector (L)	Latent Vector Size	I = [150, 200, 250, 300, 400, 500, 1000, 1500, 2000, 2500]

Once the initial variables are defined and initialized, we proceed to initialize the autoencoders using the *Get_Autoencoder* function (line 5), passing each one the hyperparameter combination as a parameter. We train the autoencoders with the *Train_Autoencoder* (line 6) function, passing the autoencoder and the auxiliary dataset as parameters. After training each autoencoder, we extract the representation for each image in the target dataset using the *Get_Representation* function (line 7), passing the encoder weights and the images from the target dataset as parameters. Finally, we apply a reduction dimensionality method (line 8) to each representation, reducing its dimension to 150. We can do this last method in 2 ways: applying the PCA method choosing the best subrepresentations of size 150 or using a dense layer with 150 neurons where by training the system it will learn to modify the representations values and transform them into a compressed version.

Once the process of obtaining new features for the target dataset is complete, we

proceed to train the attention model using the *Train_Attention_Model* (line 9) function with parameters: the initialized attention model (att_model), the new representations of the target dataset (X'_t_rd), and the labels of the target dataset (y_t). After training, the accuracy metric is calculated using testing dataset.

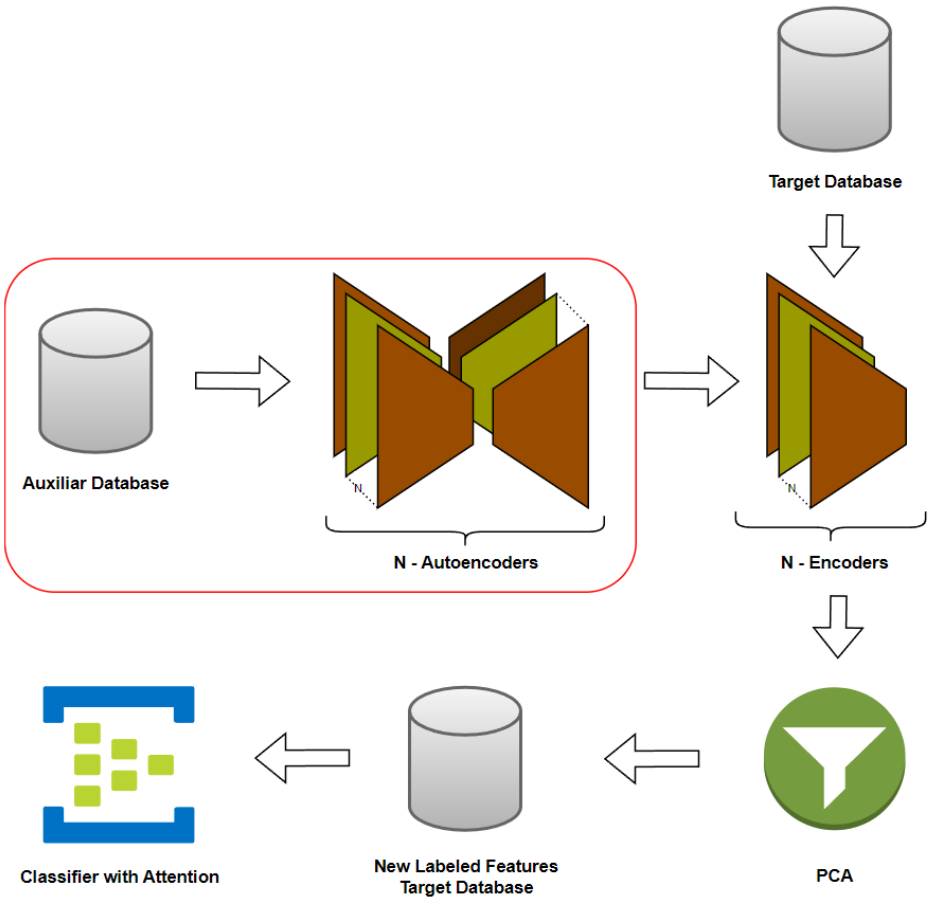
Section 3.1 called Generation of Unsupervised Representations, the strategies employed to construct the autoencoders responsible for extracting features are detailed, along with the specifics of their training.

Section 3.2 called Classification using Multi-view Attention Mechanism, outlines the procedure for obtaining representations from the previously trained autoencoders.

3.1 Generation of Unsupervised Representations

This the first stage in our proposed method which consist in training autoencoders in order to generate representations. The flow of the proposed method is as shown in Figure 7 in the enclosed red box.

Figure 7 – Train Autoencoder Flow



Source: Author's original work

In the context of autoencoders, to train our model, we feed the same images into both the input and output using the auxiliary unlabeled dataset, with the objective of minimizing the cost function to zero, so that the input, after passing through the network, produces an output that is identical or very similar to the input (POLIC et al., 2019). The concepts explained in the previous section are applied in the context of a convolutional autoencoder, which is divided into two parts: an encoder and a decoder. In the encoder, all the key transformations take place.

Going into more details about the autoencoders, the representation of the image is obtained in the latent vector, which is situated between the encoder and the decoder, as shown in equations 3.1 and 3.2.

$$\mathbf{z} = \text{Encoder}(\mathbf{x}) \tag{3.1}$$

$$\hat{\mathbf{y}} = \text{Decoder}(\mathbf{z}) \tag{3.2}$$

where $\mathbf{x}$ is the autoencoder input and z es the latent vector.

In this work, we add a significant term next to the classic autoencoder cost function, as shown in the formula 3.3:

$$\text{Loss}(y, \hat{y}) = (y - \hat{y})^2 + \frac{\text{c}}{\sum \sqrt{\sum_i (\text{dist\_w}[i] - \text{dist\_w}[-1])^2}} \tag{3.3}$$

The loss function represents a Mean Square Error (MSE) expression in Machine Learning Learning field. The first term compares the output of the autoencoder ($\hat{y}$) with the expected label (y) and the second term contributes to create more diverse representations penalizing representations close to the previous one. It incorporates a regularization variable, $c = 0.001$, controlling the weighting of this term in the equation. Later, dist_w represents a vector with the weights in the output layer of the encoder. The expression in the denominator returns the sum of the distances between each weight and the last one in the batch. In other words, this second expression penalize representations that are very similar to the previous one. This allows the creation of more diverse representations.

The training of the model follows the hyperparameters in Table 6 of the section 3 to obtain the representations:

1. **Modify Latent Vector dimension (L)**: The latent vector is altered with the dimensions specified in Table 6. By choosing this strategy, we create different autoencoders in such a way that the dimension of the latent vector varies between 150 and 2500, but only selecting values from the table mentioned before.

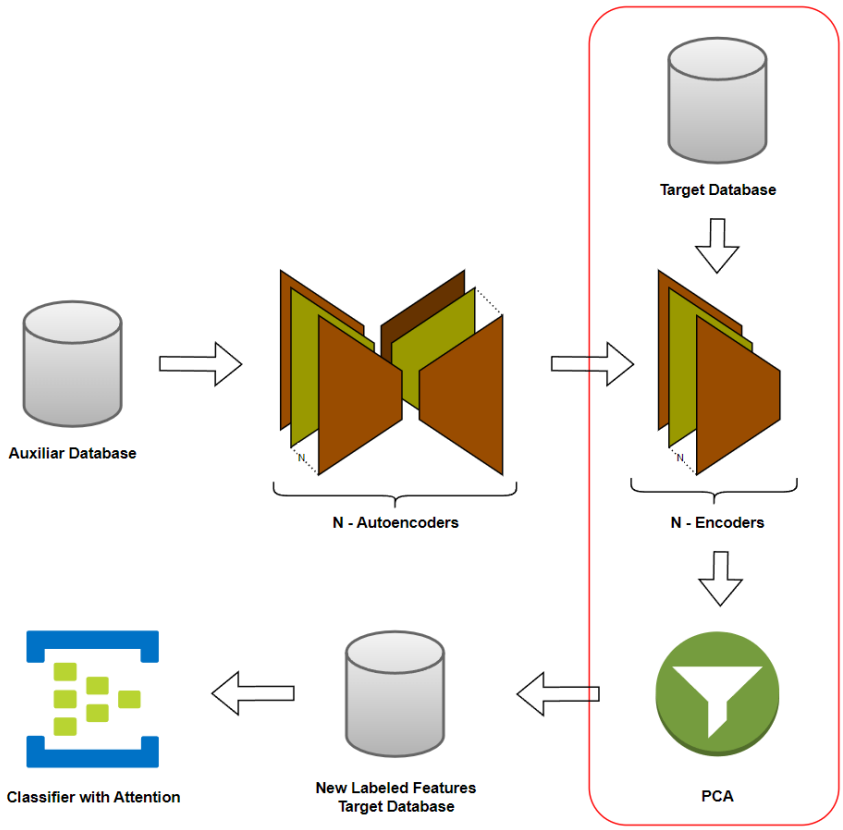
2. **Modify Architecture (A)**: The autoencoder is adjusted by varying the depth or number of convolutional layers, as illustrated in Table 6. By choosing this strategy, we create different autoencoders in such a way that their depth varies from one to five layers.

3. **Modify Latent Vector and Architecture (L/A)**: Both factors are adjusted according to the mentioned items above. By choosing this strategy, a combination of hyperparameters from strategies L and A are randomly selected. With these parameters, the autoencoders are then created.

Once we have some autoencoders trained, we proceed with step two which can be observed in the red box of the Figure 8. That Figure tells us that we have to pass images from the annotated dataset through the encoder, the output of this one returns a vector or representation (in our case). Once this representation is obtained, the dimensionality of the representation is reduced using PCA or other method to 150 components. This provides a fixed dimension for all representations, which is necessary for feeding into the classification model. This one only accepts fixed sized input.

Finally, upon receiving all these representations generated by each encoder and each image, they are combined into a new database referred to as the New Labeled Feature.

Figure 8 – Get Representation Flow



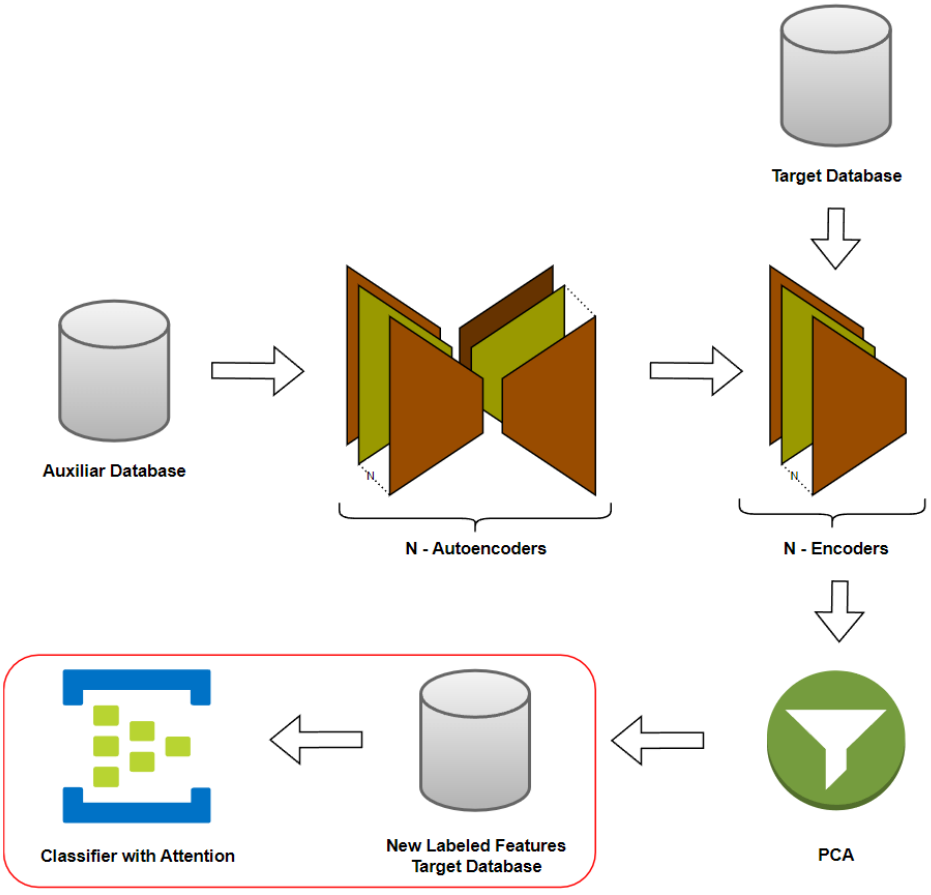
Source: Author's original work

3.2 Classification using Multi-view Attention Mechanism

In the final stage, we proceed to train the attention models. The first one is based on paying attention to the channels (each channel is a representation) of the model's input. It is a simpler model, as shown in Figure 10, where the attention module consists of three layers, and the output will be a vector of the same size as the number of channels, with values between zero and one. Each value indicates the weight assigned to each channel or representation.

The second one is based on a more complex mechanism, as illustrated in Figures 11, 13, 14, 15, and 16, using correlation matrices and representation fusion.

Figure 9 – Train Attention Model Flow



Source: Author's original work

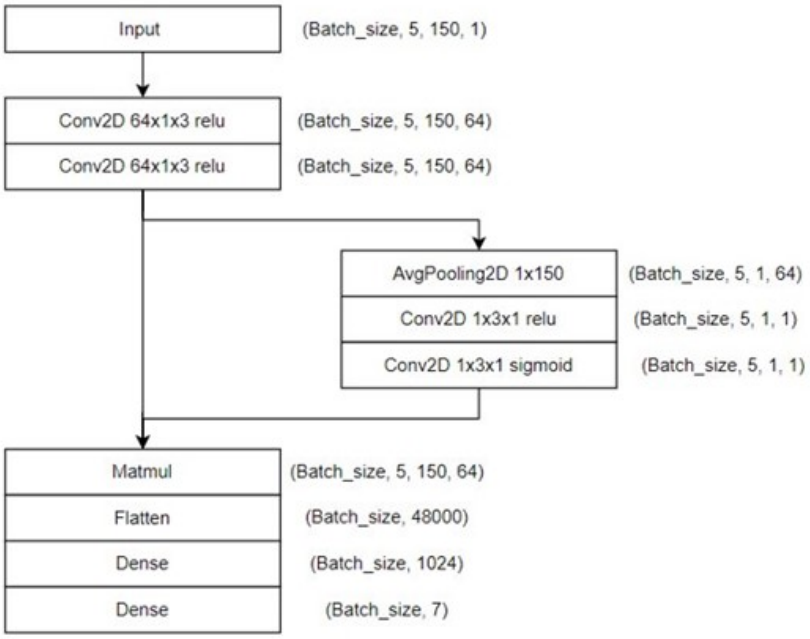
3.2.1 Approach 1: Channel Attention Mechanism (CAN) using CNN

The architecture of this approach was inspired by (BASTIDAS; TANG, 2019), which is introduced as a promising option for multi-spectral images.

A CNN is implemented with two convolutional layers and two dense layers to combine N=5 representations of the JAFFE and CK+ datasets. On the right side of Figure

10, we can see an attention module which will estimate which representation to focus on to obtain the best result.

Figure 10 – Channel Attention Mechanism



Source: Author's original work

In the attention module we have three layers: Average Pooling , Conv2d with ReLU activation function and another Conv2d with sigmoid activation function. The goal of the Average layer Pooling is making an average of all representations and features maps. Being the input dimension 5×150×64 we convert it to 5×1×64. We continue with the Conv2d ReLU layer, which with a single layer we will obtain an output of 5×1×1. Finally, the Conv2d Sigmoid layer outputs an array with a dimension equal to the number of representations, with a range of values between 0 and 1, which will tell us the weight of each representation in the final result. Once the vector of attention weights has been calculated, we are going to multiply it by the initial representations and obtain attention vectors. The latter is going to be squashed to one dimension and then passed through the Fully-Connected layers. In the images we see that 1024 was used but in the experiments it was also used with 128, 256 and 512 since they achieved better results.

The strategy used to validate the model was Leave-One-Subject-Out (LOSO), where each round we leave out images of one emotion to be tested and the other ones for training.

The training of the CNN was conducted with a batch size of 8, 35 epochs, employing early stopping with a patience of 5, cross-entropy cost function, and the Adam optimizer.

3.2.2 Approach 2: Joint Attention Mechanism with Fusion

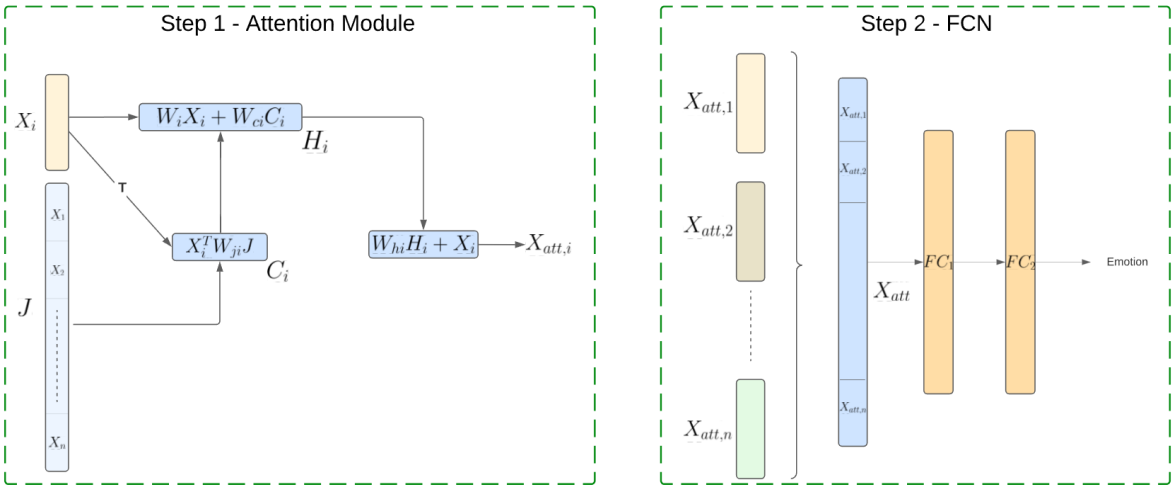
In the search for a more efficient attention mechanism, we found that (PRAVEEN et al., 2022) made significant contributions. As a result, we decided to implement their strategy in our work, which essentially involves a combination of matrix multiplications and fully connected layers, primarily in the initial stage, which we refer to as the attention module (left side Figure 11). This is followed by a feature fusion stage (concatenation) of the features obtained in the first part. Finally, the attended features are passed through an MLP or FCN classification model (right side Figure 11), which predicts the respective emotion. The entire process is detailed in Algorithm 2.

Algorithm 2: Attention Model Pseudocode

Data: $X_{1..N}$
Result: class_id
1 J = Concatenation($X_{1..N}$)
2 **for** $i \leftarrow 1$ **to** N; // Step 1 of Fig 11
3 **do**
4 $C_i \leftarrow \tanh\left(\frac{X_i^T W_{ji} J}{\sqrt{d}}\right)$
5 $H_i \leftarrow ReLU(W_i X_i + W_{ci} C_i^T)$
6 $X_{att,i} \leftarrow W_{hi} H_i + X_i$
7 X_{att} = Concatenation($X_{att,1..N}$) ; // Step 2 of Fig 11
8 class_id = FCN(X_{att})
9 **return** *class_id*;

In Algorithm 2, we receive as input all the representations obtained in Section 4.3 from the JAFFE and CK+ target datasets. In line 1, we concatenate all the representations into a vector J, referred to as the joint vector. From lines 2 to 6, we iterate over each representation and apply the formulas outlined in lines 4, 5, and 6. This process forms part of the attention module, as illustrated on the left side of Figure 11. Once the attention process is completed, we obtain the attended representations, denoted as $X_{att,i}$. In line 7, we concatenate all the attended representations into a single vector, and finally, in line 8, we apply the classification model, which returns the class to which the image representations belong. These last two steps are depicted in Figure 11.

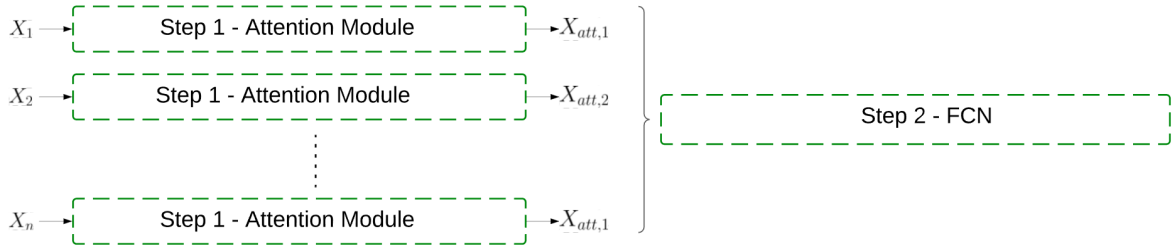
Figure 11 – Attention Mechanism Steps



Source: Author's original work

As a summary of what has been explained above, we have Figure 12, which shows that each representation X_i entering the attention module (step 1) results in an attended representation $X_{att,i}$, which is then concatenated or fused and serves as input to the FCN (step 2)

Figure 12 – Overview Classification model with Attention Modules



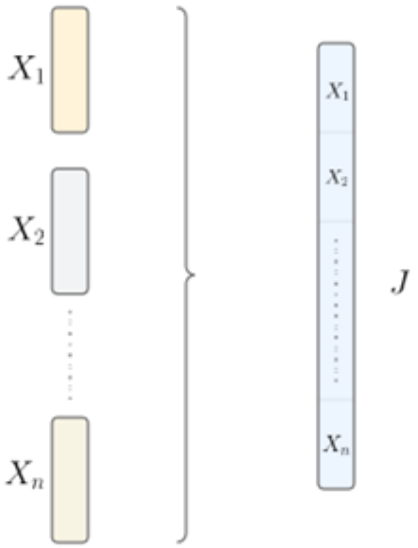
Source: Author's original work

Next, we will explain each step of Algorithm 2, Figure 11 and Figure 12 in greater detail.

3.2.2.1 Fusion of representations

The fusion of the representations, where each representation $X_i \in \mathbb{R}^{1\times d_i}$, $d_i = 150$ (output of PCA with 150 components), is achieved through concatenation. This one is represented by the variable $J \in \mathbb{R}^d$, where $d = \sum_{i=1}^{n} d_i$ and $i \in 1..n$, as seen in Figure 13. The number of representations (also called that each representations belongs to one modality in the context of fusion) can vary between 5 to 50 depending on the experiment being carried out.

Figure 13 – Fusion of Representations

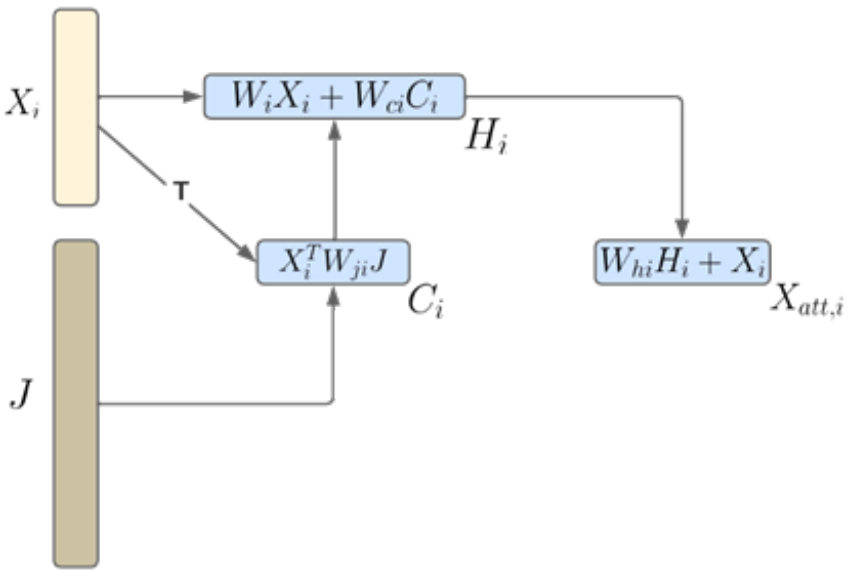


Source: Author's original work

3.2.2.2 Attention mechanism for each representation

In Figure 14, the process of the attention mechanism for each representation is illustrated using a representation X_i as input and the joint vector J as concatenation of all the representations.

Figure 14 – Attention Mechanism for Each representation



Source: Author's original work

First, we compute:

$$C_i = \tanh\left(\frac{X_i^T W_{ji} J}{\sqrt{d}}\right) \tag{3.4}$$

where $C_i \in \mathbb{R}^{d_i \times d}$ is the joint correlation matrix for each feature i. This matrix captures the semantic relevance of feature i across the modality (e.g., image), allowing the model to represent interdependencies between the feature and the shared embedding space. The function tanh introduces non-linearity, which helps normalize the correlation values to a range between -1 and 1, ensuring stable gradients during training.

The matrix $W_{ji} \in \mathbb{R}^{L \times L}$ represents a set of learnable parameters that scale and project the joint features. In this case, $L = 1$, indicating that we treat each frame or sample independently, without considering temporal relationships. This simplification is due to the assumption that individual frames are uncorrelated with each other—each frame is processed as a standalone entity without reference to past or future frames. Thus, no temporal dependencies are modeled.

By scaling the product of the transposed input feature matrix X_i^T, the learned weights W_{ji}, and the joint features J, we normalize the output by the factor $\sqrt{d}$, which accounts for the dimensionality of the feature space d. This scaling prevents the magnitude of the product from growing too large, ensuring that the learning process remains numerically stable, especially in high-dimensional spaces.

Since the dimensions of joint correlation matrices ($C_i \in \mathbb{R}^{d_i \times d}$) and the features of the corresponding modality $X_i \in \mathbb{R}^{L \times d_i}$ differ, we rely on different learnable weight matrices corresponding to features of the individual modalities to compute attention weights of the modalities. Let's see how we combine these mentioned matrices.

Second, we compute:

$$H_i = ReLU(W_i X_i + W_{ci} C_i^T) \tag{3.5}$$

where $W_{ci} \in \mathbb{R}^{k \times d}$ and $W_i \in \mathbb{R}^{k \times L}$. The matrix $H_i \in \mathbb{R}^{k \times d_i}$ represents the attention maps for modality i. These attention maps serve to highlight the most relevant regions or features within the input data for modality i.

To compute the attention weights, the joint correlation matrix C_i and the feature matrix X_i are each multiplied by their respective learnable weight matrices W_{ci} and W_i, and then combined. The result is passed through a ReLU activation function, which introduces non-linearity and ensures that the attention values are non-negative. This step enables the model to focus on the most important features for each modality by dynamically adjusting the attention weights based on both the joint correlations and the input features.

Finally, we will compute the attended representation for each modality i.

$$X_{att,i} = W_{hi}H_i + X_i \tag{3.6}$$

where $W_{hi} \in \mathbb{R}^{k \times L}$ denote the learnable weight matrices.

The term $W_{hi}H_i$ adjusts or modulates the attention map H_i, emphasizing the most relevant elements for modality i.

The term X_i represents the original input that is incorporated into the formula, meaning that the attention mechanism does not exclusively rely on the attention map but also retains information from the original input.

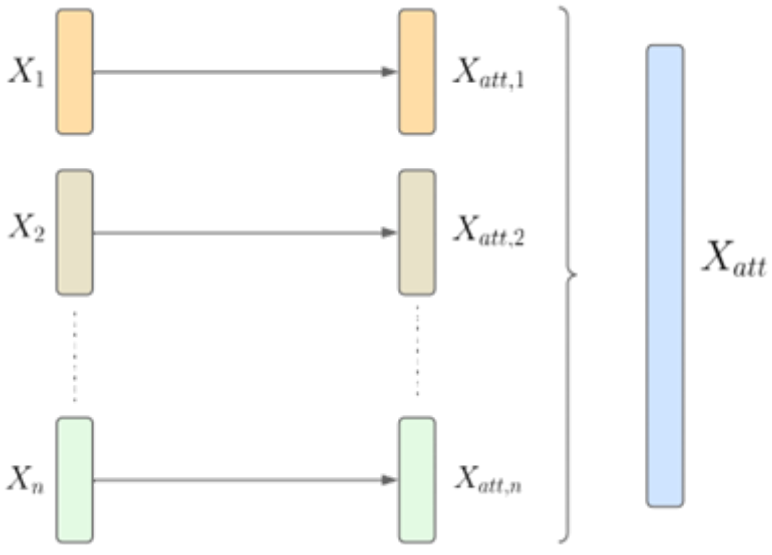
By summing these two terms, $X_{att,i}$ corresponds to the input modified by attention, where the relevance is weighted by H_i and scaled by W_{hi}. In this way, the attention mechanism focuses on the most important parts of the input while still combining them with the original input to create an enhanced representation.

In summary, this formula combines attention information with the original input, allowing the model to prioritize specific parts of the input while keeping a reference to the original data intact.

3.2.2.3 Merging attended representations

In Figure 15 we see that after performing step 3.2.2.2, a representation attended obtained was $X_{att,i}$ for each input representation X_i. Then, we will apply fusion to concatenate the N representations into one.

Figure 15 – Fusion of Attended representations

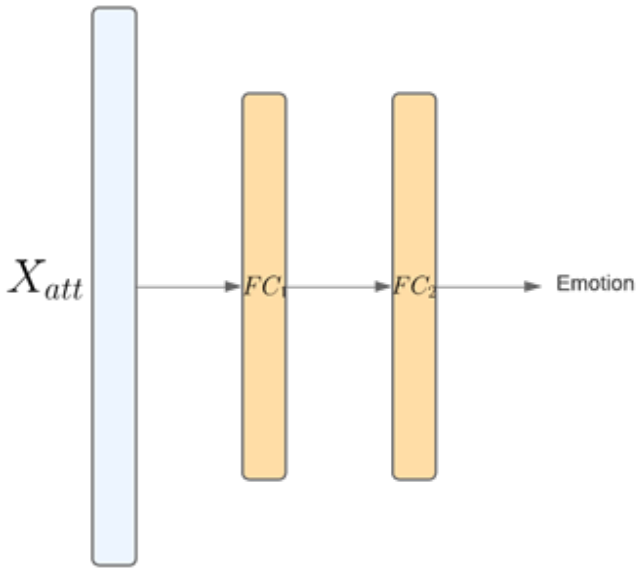


Source: Author's original work

3.2.2.4 Last Fully-Connected layers for classification

After concatenating to a single vector $X_{att} \in R^{1x(150xN)}$, two fully connected of dimensions 128 and 7 with dropout 0.6. The output is a 7-dimensional vector with the probability of each emotion as we see in Figure 16.

Figure 16 – Last Fully-Connected layers for classification



Source: Author's original work

The strategy to validate our method remains LOSO, and it includes using 50 epochs, a batch size of 4, cross-entropy as the cost function, and Adam optimizer with a learning rate of 0.001. After completing all the epochs and with the assistance of the validation dataset, the model with the highest validation accuracy is selected. Once that model is obtained, the accuracy is calculated for the test dataset.

The parameters of our fusion model (W_{ci}, W_i, W_{hi}) are optimized according to cross-entropy loss.

3.3 Final Considerations

In this chapter, the different stages of the proposed method were defined, starting with the representation generation part until the stage of the classification model with attention mechanism.

After describing the proposed method and providing details on the implementation and training of the model, the discussion of the results will follow in the next section.

4 Experimental Results

In this section, we discuss the results obtained from the proposed research methodology in Section 3. We divide in two parts, where the first one presents the results of the first approach, second approach, and in comparison to the state of the art and other works. In the second part, we delve into a discussion regarding the results.

Remember the types of strategies previously mentioned in section 3.1.1:

1. **Modify Latent Vector dimension (L)**: We create different autoencoders in such a way that the dimension of the latent vector varies between 150 and 2500, but only selecting values from the Table 6.

2. **Modify Architecture (A)**: We create different autoencoders in such a way that their depth varies from one to five layers.

3. **Modify Latent Vector and Architecture (L/A)**: A combination of hyperparameters from strategies L and A are randomly selected. With these parameters, the autoencoders are then created.

4.1 Datasets

Accessing and utilizing datasets is a critical aspect of research, yet it is essential to consider the source and permissions required for their use. The following section provides details on the acquisition of four datasets: Kyoto (DOI et al., 2003), LFW (HUANG et al., 2007), JAFFE (LYONS; KAMACHI; GYOBA, 2019), and CK+ (LUCEY et al., 2010). While some datasets are publicly available and easily downloadable, others may require permissions or agreements to ensure compliance with privacy policies and ethical considerations. It is imperative to adhere to these guidelines to prevent any misuse or violation of data usage policies.

4.1.1 Supervised

In the supervised learning paradigm, datasets are labeled, meaning each data point is associated with a corresponding label or category. These labels serve as the ground truth for training machine learning models to make predictions or classifications. In the context of our research, we leverage two widely used supervised datasets, namely the JAFFE dataset and the CK+ dataset. These datasets offer labeled facial expression data, crucial for training our model to accurately recognize and classify facial expressions.

4.1.1.1 JAFFE dataset

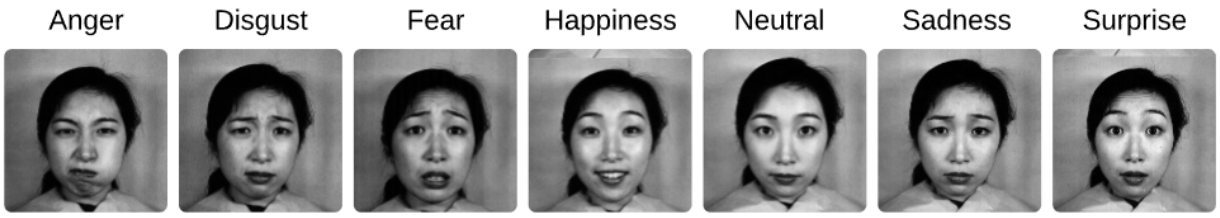
The Jaffe dataset includes 213 grayscale images showcasing various facial expressions from 10 distinct Japanese female participants. Each subject was instructed to display 7 different expressions (6 fundamental expressions plus a neutral one), and the images were labeled with average semantic ratings for each expression, provided by 60 annotators.

Since all the images come from Japanese women, the dataset does not capture diversity in gender, race, or ethnicity, which may limit the generalization of models trained on this dataset.

All the images were taken under controlled conditions, with uniform lighting and aligned faces. This may not reflect the complexity of real-world situations, where lighting and facial positions vary significantly.

The dataset can be downloaded from the creators' website (https://zenodo.org/records/3451524), but it is necesary to create an account and ask for permission to download their dataset. It is a small dataset in grayscale color, it weighs 14.1 MB.

Figure 17 – JAFFE dataset



Source: Author's original work

4.1.1.2 CK+ dataset

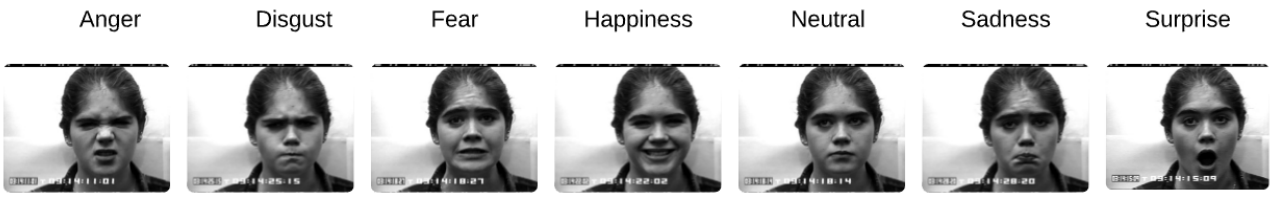
The CK+ dataset contains 593 video sequences from 123 distinct subjects, aged between 18 and 50, representing a range of genders and ethnic backgrounds. Each video captures a facial transition from a neutral expression to a specific peak expression. The recordings were made at 30 frames per second (FPS) with a resolution of either 640x490 or 640x480 pixels.

Since our model does not require video sequences, we will extract the last frame from each sequence and assign it the corresponding label based on the sequence. In the end, we will have 593 images divided among 123 subjects, so each subject will have approximately 5 facial images.

Same way like JAFFE dataset, it can be downloaded from the creators' website (https://www.jeffcohn.net/Resources/), but it is necesary to fill a couple of documents

and send them for permission to download their dataset. It is not a small dataset when you download it (its size is only 957MB), but we are only using a small part of it.

Figure 18 – CK+ dataset



Source: Author's original work

4.1.2 Unsupervised

In contrast to supervised learning, unsupervised learning involves datasets with unlabeled data points, where the objective is to discover patterns or structures inherent in the data without explicit guidance from labeled examples. Our research also utilizes an unsupervised dataset, namely the Kyoto dataset. This dataset provides unlabeled facial images, allowing us to explore novel approaches for representation learning and feature extraction without relying on pre-existing labels. Additionally, we briefly mention the Labeled Faces in the Wild (LFW) dataset, which although labeled, is often used in unsupervised or semi-supervised settings due to its large size and diverse range of facial images.

4.1.2.1 Kyoto dataset

The dataset includes images of natural scenes, such as forests, mountainous areas, bodies of water, and rural landscapes. These images focus on natural environments without large human-made structures.

The dataset consist of 62 natural images of 256×200 pixels in RGB color space. It is used by several works that apply the concepts of STL, thanks to its large variability for example in lighting conditions, seasons of the year, and different types of vegetation and natural landscapes.

It can be downloaded from its creator website (https://eizaburo-doi.github.io/kyoto_natim/). This dataset is very small, it only weighs 1MB, because of the 62 small images in RGB.

4.1.2.2 Labeled Faces in the Wild (LFW) dataset

The images are of famous individuals, primarily taken in public situations such as press conferences, interviews, and other events. These images were not captured under controlled conditions, resulting in variability in lighting, posture, and facial expressions.

Figure 19 – Kyoto dataset

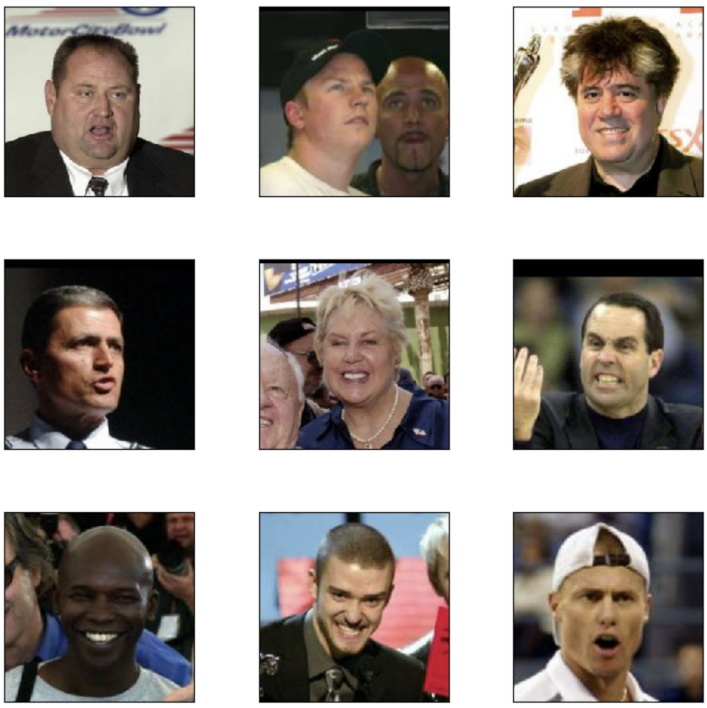
Source: Author's original work

It contains 13233 RGB faces images collected on the web from 5,749 people. Unlike Kyoto, LFW was selected for our protocol because its images belong to a domain related to the FER problem.

In this way, we aim to determine the impact of an auxiliary dataset that belongs to the same or a different domain from the target dataset. It can be downloaded from CS UMASS webiste (https://vis-www.cs.umass.edu/lfw/#download).

The Figure 20 presents a sample of LFW dataset.

Figure 20 – LFW dataset



Source: Author's original work

4.2 Experimental Results and Discussions

In this section, we present the results of our experiments and discuss the insights derived from them. We begin by analyzing the results obtained when defining the unsupervised dataset and explore how it impacts the model's performance. Then, we examine the outcomes of different approaches, like the channel attention mechanism and the multi-view attention mechanism of representations . Additionally, we discuss the choice of dimensionality reduction methods, such as Principal Component Analysis (PCA) and a Simple Dense Layer, also explore the influence of the number of components on model performance. Finally, we summarize and discuss the results in comparison with reference works and offer conclusions on our proposed approach.

It's important to mention that the way these accuracy values (percentages) were chosen was performing the same experiment six times, or in other words, run the same model with the same hyper-parameters in six different docker containers at same time. Each docker container runs a model and returns a value, then we choose the biggest value among those six and show it in the table.

4.2.1 Experiment to define Unsupervised dataset

One of the questions we raised at the beginning of this work was the variability of auxiliary datasets. Variability largely depends on the similarity between the domains of the datasets. We will present two datasets that are very different from each other and measure the impact through the accuracy metric to determine whether the closeness between domains matters or not.

Table 7 presents some comparisons of metric accuracy for the auxiliars datasets LFW and Kyoto. The first one contains images close to the target domain; however, the domain of the second dataset is very different from the target dataset as it does not contain facial images. The comparisons are made for the three different strategies (L, A and L/A). For this experiment, we limited to a single target dataset (JAFFE) for simplicity, since we noticed similar behaviour for CK+ dataset in different experiments.

Table 7 – Table of Accuracy for the table datasets LFW and Kyoto vs Strategies vs Number of Representations of JAFFE dataset as target dataset using Joint Attention Mechanism

Rep/Strategy	**LFW**			**Kyoto**		
	L	**A**	**L/A**	**L**	**A**	**L/A**
10	13.19	**19.26**	13.62	65.10	**68.94**	67.49
20	15.15	16.41	17.77	65.34	63.85	67.53
30	15.53	18.38	14.51	63.45	64.32	66.20
50	18.34	18.79	16.89	65.14	68.52	66.68

We see that the results are not competitive when using an auxiliary dataset that has a similar domain to the target dataset.

While the accuracy of testing dataset for Kyoto dataset is 68.94%, the accuracy for LFW dataset is 19.26%. So we can infer that in our Self-Taught learning framework, an auxiliary dataset which belongs to a completely different domain of target dataset reach much better results than an auxiliary dataset that belongs to a similar domain of the target dataset.

We observe that the difference is significant, and we consider this to be due to the fact that the LFW dataset is large, and the learned representations have been highly biased towards this domain. On the other hand, with the smaller and more varied dataset, no bias was achieved that could significantly affect the representations in a negative way.

4.2.2 Channel Attention Mechanism results (First approach)

This is our first implementation of an attention mechanism. We used the Channel Attention Mechanism, which focuses on identifying which channel of the input data structure has a greater impact on the final outcome.

Tables 8 and 9 present comparisons of accuracy for the different models: with and without channel attention mechanism and the reference works of (DELAZERI et al., 2022) SVM and Stacking SVM for the datasets JAFFE and CK+ using the Latent Vector strategy (L).

Table 8 – Table of Strategy L vs Models for JAFFE dataset using first approach

Model evaluated in JAFFE	**L**
With Attention	44.27
Without Attention	54.78
SVM (DELAZERI et al., 2022)	59.96
Stacking SVM (DELAZERI et al., 2022)	62.26

Table 9 – Table of Strategy L vs Models for CK+ dataset using first approach

Model evaluated in CK+	**L**
With Attention	78.83
Without Attention	81.50
SVM ((DELAZERI et al., 2022))	87.00
Product SVM ((DELAZERI et al., 2022))	86.99

Those comparisons of accuracy are presented using the L strategy where the latent vector size is varied between 150 and 2500 choosing specific values as shown in the table 6. The reason we use specific size values is because we want to make the most fair comparison against state-of-the-art works.

The two types of models that were evaluated with the attention mechanism and without it. Where we hypothesized that the attention mechanism could better choose which channel (another name for representation) to focus more on or give greater priority and thus give us better results, but we see that it was not what we expected. Without the attention mechanism we obtained better results for either the JAFFE or CK+ dataset of about 10.51 and 2.67 points respectively.

Furthermore, comparing with other reference works where SVM and Stacking SVM are used, up to a difference below 17.99 and 8.16 was obtained than expected for the JAFFE and CK+ datasets respectively.

Then, having such a wide difference in the results between both models (with Attention and Stacking SVM), it was decided that it was not worth continuing with other experiments, for example by increasing the number of representations greater than five (only five were used for the experiments) representations and other strategies in addition to L using said channel attention mechanism since it was most likely to obtain similar results, so it was decided to approach another solution.

4.2.3 Joint Attention Mechanism with Fusion results (Second approach)

This is our second implementation of an attention mechanism. We used Joint Attention Mechanism, which focuses on using an joint vector (concatenation of all the representations) performing calculations against each representation to identify how the representations behave both against each other and internally within each representation. Based on this, the influence of each representation on the final result is determined.

The tables 10 and 11 show the results of the accuracy metric for approach two (Joint Attention Mechanism with Fusion) where on the one hand we have the three different strategies (L, A and L/A) versus the number of representations per image for the JAFFE and CK+ datasets.

Table 10 – Table of Strategies vs Number of Representation for JAFFE dataset using second approach

Rep/JAFFE	**L**	**A**	**L/A**
10	65.10	**68.94**	67.49
20	65.34	63.85	67.53
30	63.45	64.32	66.20
50	65.14	68.52	66.68

Since the experiments of the first approach were not successful at all, it was decided to change to a very different method that explores a different type of attention mechanism using fusion of representations and correlation matrices.

Table 11 – Table of Strategies vs Number of Representation for CK+ dataset using second approach

Rep/CK+	L	A	L/A
10	86.68	85.52	85.59
20	85.67	83.02	84.65
30	85.73	83.07	86.34
50	86.57	**88.60**	85.74

Unlike the first experiments, positive results were obtained according what it was expected and it was also evaluated in different representations from 10 to 50 for both datasets. As mentioned before in the discussions of the approach, it was decided to keep the hyper-parameters of the strategies from related works, to make the fair comparisons respect to the state-of-the-art and previous works.

As we see in the results of both datasets, there is a slight tendency to improve accuracy as the number of representations increases, but at the same time, greater RAM and GPU resources are needed to carry it out. On our machine, we were able to use 32GB of RAM and 11GB of GPU memory, but in order to run 70 representations and four models at the same time (strategy to reduce testing and training time), we got Out-of-memory (OOM) in every container docker. Even training for 50 representations it lasted approximately four hours for the CK+ dataset that has more data. So we were limited to work up to 50 representations for hardware and time limitations, but we believe that we got good results.

4.2.4 Presence of Attention Mechanism

After conducting experiments with both attention mechanisms, we observed that the second approach yielded better results. Therefore, we will continue using it for more specific experiments moving forward. Next, we will determine whether the use of an attention mechanism actually improves the accuracy of our classification model.

In table 12 and 13 we can see the results of the accuracy metric for approach two in the JAFFE and CK+ datasets respectively, comparing three different strategies (L, A and L/A) against the model with and without attention. For the sake of simplicity in our experiments, we will fix the number of representations to 10 and copy the results from experiment 4.2.1 for "w/ attention 10" for both datasets, as the same procedure is used.

Table 12 – Table of Strategies vs With and without attention for 10 representations in JAFFE dataset using second approach

Rep/JAFFE	L	A	L/A
w/ attention 10	**65.10**	**68.94**	**67.49**
wo/ attention 10	26.68	28.54	24.86

Table 13 – Table of Strategies vs With and without attention for 10 representations in CK+ dataset using second approach

Rep/CK+	**L**	**A**	**L/A**
w/ attention 10	**86.68**	**88.60**	**86.34**
wo/ attention 10	45.40	42.66	35.35

We see that there is an improvement in the accuracy of the models that present the attention mechanism compared to those that don't have it.

As mentioned before for approach 1, the channel attention mechanism was not as effective for our problem unlike Joint Attention Mechanism with Fusion representations. This is reflected in the notable improvement of almost 100% compared to the model without attention mechanism for the two datasets.

4.2.5 Choice of dimensionality reduction method

We shows two types of dimensionality reduction methods: PCA and Fully Connected Layer. As their names suggest, these methods aim to reduce the high dimensionality of feature representations, which can reach up to 2500, down to a smaller value, such as 150. This reduction not only allows for faster system processing but also helps mitigate overfitting. Additionally, these methods enable us to set a fixed input dimension for the classification model.

We show that using an approach where we put a dense layer with 150 neurons instead of PCA where at the moment of training the system, it will learn to modify the representations values and transform them into a compressed version one. This is not a very commom strategy but it worths in order to enrich our work with different ideas.

In Tables 14 and 15, the results are shown for the different amounts of representations of the accuracy metric for the second approach using Kyoto and LFW respectively as auxiliary datasets and only the JAFFE dataset as the target dataset for reasons of simplicity explained above. These mentioned results are compared with the three different strategies (L, A and L/A) for a dimensionality reduction method based on Fully-Connected Layer.

Table 14 – Table of Strategies vs Number of Representation using JAFFE as target dataset and Kyoto as auxiliar dataset under the second approach not using PCA

Rep/JAFFE	**L**	**A**	**L/A**
10	61.31	62.88	58.75
20	57.38	62.32	**64.75**
30	59.63	61.59	61.60
50	57.70	61.01	62.0

Table 15 – Table of Strategies vs Number of Representation using JAFFE as target dataset and LFW as auxiliar dataset under the second approach not using PCA

Rep/JAFFE	L	A	L/A
10	16.31	14.57	13.07
20	16.43	17.75	17.19
30	17.85	15.97	16.40
50	16.79	16.50	**18.13**

As we see in the results of the tables above it was not possible to surpass the best results obtained using PCA 68.94% for JAFFE 12.

4.2.6 Choice of the Number of PCA Components

Let's find out how much information can be preserved by reducing the dimensionality of the data. By calculating the cumulative variance, we determine how much total variance from the original data is preserved when retaining a specific number of principal components.

Cumulative variance is useful for decision-making on how many principal components to retain. When performing dimensionality reduction with PCA, we typically decide how many principal components (or dimensions) to keep. Cumulative variance helps you understand how much information you are retaining in relation to the total number of components.

In Figure 21, we have randomly selected a pool of hyper-parameters that uses 10 representations. We will observe how the cumulative variance behaved against the number of components for each representation of the target dataset.

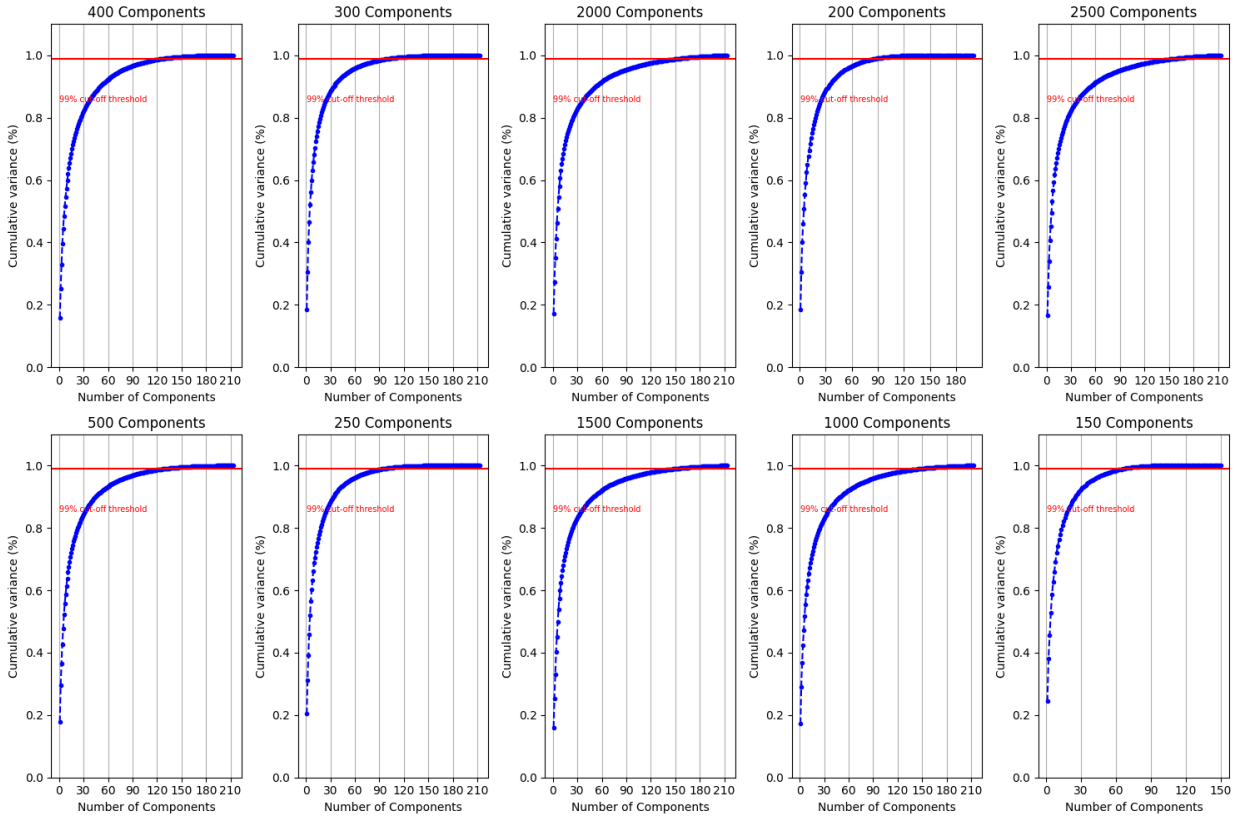
For instance, deciding to retain 99% of the cumulative variance means that we are willing to accept a loss of 1% of the original information. This approach can be useful for eliminating noise and reducing model complexity, especially when working with high-dimensional datasets, as is the case when dealing with data of up to 2500 components or dimensions.

In practice, we plot the cumulative variance against the number of components and select the number of components that allows us to retain the desired amount of variance. This is often known as the "elbow" in the graph, and the point where the curve flattens indicates the number of components to retain.

Judging by the "elbow" in the graphs, the number of components that occurs most frequently is 150, making it a suitable candidate for use in our work.

Then, an important question to answer is whether it is worth applying PCA to the representations, and the answer is 'yes'. Because, for the attention model, we need a fixed

Figure 21 – Commulative Variance vs Number of Components



Source: Author's original work

input, and historically, PCA has proven to be one of the best methods for dimensionality reduction. In this way, we can reduce the dimension of all arrays with different sizes to a fixed dimension, which is smaller or equal to all of them.

4.2.7 Final results

In Table 16 we see a comparison of the best results of the accuracy metric in the JAFFE and CK+ datasets, for each model described in our work against some works used as a reference that use a methodology similar to ours.

Table 16 – Table of Best results of the proposed method compared to the reference works

Rep/Dataset	**JAFFE**	**CK+**
(DELAZERI et al., 2022)	62.26	87.21
(BHANDARI et al., 2018) CNN	56.45	-
(LONG et al., 2012) SVM	-	80.15
Channel Attention	44.26	78.83
Multi-view Attention (10 Reps limit)	68.94	86.68
Multi-view Attention	**68.94**	**88.60**

4.2.8 General discussions of the results

Since the results obtained in (DELAZERI et al., 2022) were achieved with 10 representations, we will only consider our results in our experiments with the same amount of representations, thus ensuring a fairer comparison. With this in mind, we observe that for the JAFFE dataset, we outperform it by 6.68 points, in contrast to the CK+ dataset where it surpasses us by 0.53 points. However, if we consider the results without restrictions talking about number of representations, we surpass it in both datasets.

The way (BHANDARI et al., 2018) worked was using CNN as a classification model in STL where the differential of our work was to use different initializations in the weights for each layer and the best result obtained was 56.45 for the JAFFE dataset. With this said, all our experiments for the JAFFE dataset manage to be better in the accuracy metric except for approach one of Channel Attention. Unfortunately, that paper does not mention any results for CK+.

After discussing the results, we move on to presenting the conclusions in the next section.

5 Conclusion

A multi-view of representations system with attention and fusion mechanisms was developed, capable of surpassing the state of the art on both target datasets using unsupervised representations.

We observe that the impact of an attention mechanism in a multi-view of representations was positive compared to the dynamic selection algorithm. In JAFFE and CK+, it had a positive and negative impact of 6.68 and 0.53 points, respectively. However, without the restrictions of the 10-repetition limit, the positive impact on both datasets increases significantly. Finally, the best strategy was "Modifying Depth of Autoencoder" (A) for JAFFE and CK+ datasets.

A strength of the method is that it is not reliant on large volumes of annotated data to achieve competitive results. A weakness is accurately parameterizing the algorithm, as it depends on several parameters to be adjusted.

As future work, we aim to address the challenge of generating representations with greater diversity. For instance, modifying the cost function during autoencoder training, determining new parameterization strategies for autoencoder initialization, and testing on diverse datasets with greater variation.

Bibliography

A., V. et al. Attention is all you need. In: *In Advances in neural information processing systems*. [S.l.]: NIPS, 2017. p. 5998–6008. Cited in pag 30.

ABDI, Hervé; WILLIAMS, Lynne J. Principal component analysis. *WIREs Computational Statistics*, v. 2, n. 4, p. 433–459, 2010. Disponível em: <https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wics.101>. Cited in pag 30.

AKHAND, M. A. H.; ROY, Shuvendu; SIDDIQUE, Nazmul; KAMAL, Md Abdus Samad; SHIMAMURA, Tetsuya. Facial emotion recognition using transfer learning in the deep cnn. *Electronics*, v. 10, n. 9, 2021. ISSN 2079-9292. Disponível em: <https://www.mdpi.com/2079-9292/10/9/1036>. Cited in pag 33.

AOUAYEB, Mouath; HAMIDOUCHE, Wassim; SOLADIE, Catherine; KPALMA, Kidiyo; SEGUIER, Renaud. *Learning Vision Transformer with Squeeze and Excitation for Facial Expression Recognition*. 2021. Disponível em: <https://arxiv.org/abs/2107.03107>. Cited in pag 33.

BASTIDAS, Alexei A.; TANG, Hanlin. Channel attention networks. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. [S.l.: s.n.], 2019. Cited in pag 42.

BASTIEN, Frédéric; BENGIO, Yoshua; BERGERON, Arnaud; BOULANGER-LEWANDOWSKI, Nicolas; BREUEL, Thomas; CHHERAWALA, Youssouf; CISSE, Moustapha; CôTé, Myriam; ERHAN, Dumitru; EUSTACHE, Jeremy; GLOROT, Xavier; MULLER, Xavier; LEBEUF, Sylvain Pannetier; PASCANU, Razvan; RIFAI, Salah; SAVARD, Francois; SICARD, Guillaume. *Deep Self-Taught Learning for Handwritten Character Recognition*. 2010. Cited in pag 25.

BHANDARI, Piyush; BIJARNIYA, Rakesh Kumar; CHATTERJEE, Subhamoy; KOLEKAR, Maheshkumar. Analysis for self-taught and transfer learning based approaches for emotion recognition. In: *2018 5th International Conference on Signal Processing and Integrated Networks (SPIN)*. [S.l.: s.n.], 2018. p. 509–512. Cited 6 times in pages 19, 30, 34, 36, 61, and 62.

BOWMAN, Samuel R.; VILNIS, Luke; VINYALS, Oriol; DAI, Andrew; JOZEFOWICZ, Rafal; BENGIO, Samy. Generating sentences from a continuous space. In: RIEZLER, Stefan; GOLDBERG, Yoav (Ed.). *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*. Berlin, Germany: Association for Computational Linguistics, 2016. p. 10–21. Disponível em: <https://aclanthology.org/K16-1002>. Cited in pag 26.

CHEN, Jingying; LV, Yongqiang; XU, Ruyi; XU, Can. Automatic social signal analysis: Facial expression recognition using difference convolution neural network. *Journal of Parallel and Distributed Computing*, Elsevier, v. 131, p. 97–102, 2019. Cited in pag 35.

CHEN, Luefeng; ZHOU, Mengtian; SU, Wanjuan; WU, Min; SHE, Jinhua; HIROTA, Kaoru. Softmax regression based deep sparse autoencoder network for facial emotion recognition in human-robot interaction. *Information Sciences*, v. 428, p. 49–61, 2018. ISSN 0020-0255.

Disponível em: <https://www.sciencedirect.com/science/article/pii/S0020025517310496>. Cited in pag 31.

CHEN, Yuedong; WANG, Jianfeng; CHEN, Shikai; SHI, Zhongchao; CAI, Jianfei. Facial motion prior networks for facial expression recognition. In: IEEE. *2019 IEEE Visual Communications and Image Processing (VCIP)*. [S.l.], 2019. p. 1–4. Cited in pag 34.

CHOROWSKI, Jan K; BAHDANAU, Dzmitry; SERDYUK, Dmitriy; CHO, Kyunghyun; BENGIO, Yoshua. Attention-based models for speech recognition. In: CORTES, C.; LAWRENCE, N.; LEE, D.; SUGIYAMA, M.; GARNETT, R. (Ed.). *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2015. v. 28. Disponível em: <https://proceedings.neurips.cc/paper_files/paper/2015/file/1068c6e4c8051cfd4e9ea8072e3189e2-Paper.pdf>. Cited in pag 29.

DAMI, Jeong; BYUNG-GYU, Kim; SUH-YEON, Dong; ASASD asdfasdfasfd. Deep joint spatiotemporal network (djstn) for efficient facial expression recognition. *Sensors*, v. 20, n. 7, 2020. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/20/7/1936>. Cited in pag 32.

DELAZERI, Bruna; LEON, Leonardo; BARDDAL, J. P.; KOERICH, A. L.; DE, S. B. Alceu. Evaluation of self-taught learning-based representations for facial emotion recognition. *2022 International Joint Conference on Neural Networks (IJCNN)*, p. 1–8, 2022. Cited 4 times in pages 30, 56, 61, and 62.

DENG, Li; YU, Dong et al. Deep learning: methods and applications. *Foundations and trends® in signal processing*, Now Publishers, Inc., v. 7, n. 3–4, p. 197–387, 2014. Cited in pag 32.

DOI, E.; INUI, T.; LEE, T.-W.; WACHTLER, T.; SEJNOWSKI, T. J. Spatiochromatic receptive field properties derived from information-theoretic analyses of cone mosaic responses to natural scenes. In: . [S.l.: s.n.], 2003. p. 397–417. Cited in pag 51.

DONAHUE, Jeff; JIA, Yangqing; VINYALS, Oriol; HOFFMAN, Judy; ZHANG, Ning; TZENG, Eric; DARRELL, Trevor. Decaf: A deep convolutional activation feature for generic visual recognition. In: PMLR. *International conference on machine learning*. [S.l.], 2014. p. 647–655. Cited in pag 33.

FERNANDEZ, Pedro D. Marrero; PEñA, Fidel A. Guerrero; REN, Tsang Ing; CUNHA, Alexandre. *FERAtt: Facial Expression Recognition with Attention Net*. 2019. Cited in pag 32.

GAN, Junying; LI, Lichen; ZHAI, Yikui; LIU, Yinhua. Deep self-taught learning for facial beauty prediction. *Neurocomputing*, v. 144, p. 295–303, 2014. ISSN 0925-2312. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0925231214006468>. Cited in pag 25.

GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. *Deep Learning*. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>. Cited in pag 26.

HASANI, Behzad; MAHOOR, Mohammad H. Facial expression recognition using enhanced deep 3d convolutional neural networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. [S.l.: s.n.], 2017. p. 30–40. Cited in pag 34.

HINTON, G. E.; SALAKHUTDINOV, R. R. Reducing the dimensionality of data with neural networks. *Science*, v. 313, n. 5786, p. 504–507, 2006. Disponível em: <https://www.science.org/doi/abs/10.1126/science.1127647>. Cited in pag 26.

HUA, Wentao; DAI, Fei; HUANG, Liya; XIONG, Jian; GUI, Guan. Hero: Human emotions recognition for realizing intelligent internet of things. *IEEE Access*, IEEE, v. 7, p. 24321–24332, 2019. Cited in pag 35.

HUANG, Gary B.; LEE, Honglak; LEARNED-MILLER, Erik. Learning hierarchical representations for face verification with convolutional deep belief networks. In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2012. p. 2518–2525. Cited in pag 36.

HUANG, Gary B.; RAMESH, Manu; BERG, Tamara; LEARNED-MILLER, Erik. *Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments*. [S.l.], 2007. Cited in pag 51.

JAIN, Deepak Kumar; ZHANG, Zhang; HUANG, Kaiqi. Multi angle optimal pattern-based deep learning for automatic facial expression recognition. *Pattern Recognition Letters*, Elsevier, v. 139, p. 157–165, 2020. Cited in pag 34.

JAIN, Neha; KUMAR, Shishir; KUMAR, Amit; SHAMSOLMOALI, Pourya; ZAREAPOOR, Masoumeh. Hybrid deep neural networks for face emotion recognition. *Pattern Recognition Letters*, Elsevier, v. 115, p. 101–106, 2018. Cited in pag 34.

JING, Li; KAN, Jin; DALIN, Zhou; NAOYUKI, Kubota; ZHAOJIE, Ju. Attention mechanism-based cnn for facial expression recognition. *Neurocomputing*, Elsevier, v. 411, p. 340–350, 2020. Cited in pag 32.

JOLLIFFE, I. T. *Principal Component Analysis*. New York: Springer-Verlag, 2002. (Springer Series in Statistics). ISBN 0-387-95442-2. Disponível em: <http://www.springer.com/statistics/statistical+theory+and+methods/book/978-0-387-95442-4>. Cited in pag 29.

KARTALI, Aneta; ROGLIĆ, Miloš; BARJAKTAROVIĆ, Marko; ĐURIĆ-JOVIČIĆ, Milica; JANKOVIĆ, Milica M. Real-time algorithms for facial emotion recognition: a comparison of different approaches. In: IEEE. *2018 14th Symposium on Neural Networks and Applications (NEUREL)*. [S.l.], 2018. p. 1–4. Cited in pag 34.

KINGMA, Diederik P; WELLING, Max. *Auto-Encoding Variational Bayes*. 2022. Disponível em: <https://arxiv.org/abs/1312.6114>. Cited in pag 26.

LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, v. 86, n. 11, p. 2278–2324, 1998. Cited in pag 26.

LEE, Injae; JUNG, Heechul; AHN, Chung Hyun; SEO, Jeongil; KIM, Junmo; KWON, Ohseok. Real-time personalized facial expression recognition system based on deep learning. In: IEEE. *2016 IEEE International Conference on Consumer Electronics (ICCE)*. [S.l.], 2016. p. 267–268. Cited in pag 35.

LI, Yong; ZENG, Jiabei; SHAN, Shiguang; CHEN, Xilin. Occlusion aware facial expression recognition using cnn with attention mechanism. *IEEE Transactions on Image Processing*, IEEE, v. 28, n. 5, p. 2439–2450, 2018. Cited in pag 34.

LIANG, Liqian; LANG, Congyan; LI, Yidong; FENG, Songhe; ZHAO, Jian. Fine-grained facial expression recognition in the wild. *IEEE Transactions on Information Forensics and Security*, IEEE, v. 16, p. 482–494, 2020. Cited in pag 32.

LONG, Fei; WU, Tingfan; MOVELLAN, Javier R.; BARTLETT, Marian S.; LITTLEWORT, Gwen. Learning spatiotemporal features by using independent component analysis with application to facial expression recognition. *Neurocomputing*, v. 93, p. 126–132, 2012. ISSN 0925-2312. Disponível em: <https://www.sciencedirect.com/science/article/pii/S092523121200361X>. Cited 2 times in pages 36 and 61.

LUCEY, Patrick; COHN, Jeffrey F.; KANADE, Takeo; SARAGIH, Jason; AMBADAR, Zara; MATTHEWS, Iain. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*. [S.l.: s.n.], 2010. p. 94–101. Cited in pag 51.

LUONG, Thang; PHAM, Hieu; MANNING, Christopher D. Effective approaches to attention-based neural machine translation. In: MÀRQUEZ, Lluís; CALLISON-BURCH, Chris; SU, Jian (Ed.). *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, 2015. p. 1412–1421. Disponível em: <https://aclanthology.org/D15-1166>. Cited in pag 29.

LYONS, Michael; KAMACHI, Miyuki; GYOBA, Jiro. The Japanese Female Facial Expression (JAFFE) Dataset. In: . Zenodo, 2019. Disponível em: <https://doi.org/10.5281/zenodo.3451524>. Cited in pag 51.

MAKHMUDKHUJAEV, Farkhod; ABDULLAH-AL-WADUD, M.; IQBAL, Md Tauhid Bin; RYU, Byungyong; CHAE, Oksam. Facial expression recognition with local prominent directional pattern. *Signal Processing: Image Communication*, v. 74, p. 1–12, 2019. ISSN 0923-5965. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0923596518306556>. Cited in pag 32.

MENG, Debin; PENG, Xiaojiang; WANG, Kai; QIAO, Yu. Frame attention networks for facial expression recognition in videos. In: *2019 IEEE International Conference on Image Processing (ICIP)*. [S.l.: s.n.], 2019. p. 3866–3870. Cited in pag 33.

MH., Guo; TX., Xu; JJ., Liu; AL. et. Attention mechanisms in computer vision: A survey. *Computational Visual Media 8*, p. 331–368, 2022. Cited in pag 31.

MINAEE, Shervin; MINAEI, Mehdi; ABDOLRASHIDI, Amirali. Deep-emotion: Facial expression recognition using attentional convolutional network. *Sensors*, v. 21, n. 9, 2021. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/21/9/3046>. Cited in pag 33.

MNIH, Volodymyr; HEESS, Nicolas; GRAVES, Alex; KAVUKCUOGLU, koray. Recurrent models of visual attention. In: GHAHRAMANI, Z.; WELLING, M.; CORTES, C.; LAWRENCE, N.; WEINBERGER, K.Q. (Ed.). *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2014. v. 27. Disponível em: <https://proceedings.neurips.cc/paper_files/paper/2014/file/09c6c3783b4a70054da74f2538ed47c6-Paper.pdf>. Cited in pag 29.

MOHAMMADPOUR, Mostafa; KHALILIARDALI, Hossein; HASHEMI, Seyyed Mohammad R; ALYANNEZHADI, Mohammad M. Facial emotion recognition using deep convolutional networks. In: IEEE. *2017 IEEE 4th international conference on knowledge-based engineering and innovation (KBEI)*. [S.l.], 2017. p. 0017–0021. Cited in pag 34.

NGWE, Jia Le; LIM, Kian Ming; LEE, Chin Poo; ONG, Thian Song. *PAtt-Lite: Lightweight Patch and Attention MobileNet for Challenging Facial Expression Recognition*. 2023. Disponível em: <https://arxiv.org/abs/2306.09626>. Cited in pag 33.

PENG, Xianlin; XIA, Zhaoqiang; LI, Lei; FENG, Xiaoyi. Towards facial expression recognition in the wild: A new database and deep recognition system. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. [S.l.: s.n.], 2016. p. 93–99. Cited in pag 35.

POLIC, Marsela; KRAJACIC, Ivona; LEPORA, Nathan; ORSAG, Matko. Convolutional autoencoder for feature extraction in tactile sensing. *IEEE Robotics and Automation Letters*, v. 4, n. 4, p. 3671–3678, 2019. Cited in pag 40.

POURMIRZAEI, Mahdi; MONTAZER, Gholam Ali; ESMAILI, Farzaneh. *Using Self-Supervised Auxiliary Tasks to Improve Fine-Grained Facial Representation*. 2022. Disponível em: <https://arxiv.org/abs/2105.06421>. Cited in pag 33.

PRAVEEN, R Gnana; MELO, Wheidima Carneiro de; ULLAH, Nasib; ASLAM, Haseeb; ZEESHAN, Osama; DENORME, Théo; PEDERSOLI, Marco; KOERICH, Alessandro L.; BACON, Simon; CARDINAL, Patrick; GRANGER, Eric. A joint cross-attention model for audio-visual fusion in dimensional emotion recognition. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. [S.l.: s.n.], 2022. p. 2485–2494. Cited 2 times in pages 31 and 44.

RAINA, Rajat; BATTLE, Alexis; LEE, Honglak; PACKER, Benjamin; NG, Andrew Y. Self-taught learning: transfer learning from unlabeled data. In: *Proceedings of the 24th International Conference on Machine Learning*. New York, NY, USA: Association for Computing Machinery, 2007. (ICML 2007), p. 759–766. ISBN 9781595937933. Disponível em: <https://doi.org/10.1145/1273496.1273592>. Cited 3 times in pages 19, 25, and 26.

RAZAVIAN, Ali Sharif; AZIZPOUR, Hossein; SULLIVAN, Josephine; CARLSSON, Stefan. Cnn features off-the-shelf: an astounding baseline for recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. [S.l.: s.n.], 2014. p. 806–813. Cited in pag 33.

REFAT, Chowdhury Mohammad Masum; AZLAN, Norsinnira Zainul. Deep learning methods for facial expression recognition. In: IEEE. *2019 7th International Conference on Mechatronics Engineering (ICOM)*. [S.l.], 2019. p. 1–6. Cited in pag 35.

RUIZ-GARCIA, Ariel; ELSHAW, Mark; ALTAHHAN, Abdulrahman; PALADE, Vasile. A hybrid deep learning neural approach for emotion recognition from facial expressions for socially assistive robots. *Neural Computing and Applications*, Springer, v. 29, p. 359–373, 2018. Cited in pag 34.

SAJJAD, Muhammad; NASIR, Mansoor; ULLAH, Fath U Min; MUHAMMAD, Khan; SANGAIAH, Arun Kumar; BAIK, Sung Wook. Raspberry pi assisted facial

expression recognition framework for smart security in law-enforcement services. *Information Sciences*, v. 479, p. 416–431, 2019. ISSN 0020-0255. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0020025518305425>. Cited in pag 32.

SAJJAD, Muhammad; ULLAH, Fath U Min; ULLAH, Mohib; CHRISTODOULOU, Georgia; Alaya Cheikh, Faouzi; HIJJI, Mohammad; MUHAMMAD, Khan; RODRIGUES, Joel J.P.C. A comprehensive survey on deep facial expression recognition: challenges, applications, and future guidelines. *Alexandria Engineering Journal*, v. 68, p. 817–840, 2023. ISSN 1110-0168. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1110016823000327>. Cited in pag 31.

SAJJANHAR, Atul; WU, ZhaoQi; WEN, Quan. Deep learning models for facial expression recognition. In: IEEE. *2018 digital image computing: Techniques and applications (dicta)*. [S.l.], 2018. p. 1–6. Cited in pag 34.

SC, Huang; A, Pareek; S, Seyyedi; I, Banerjee; MP, Lungren. Fusion of medical imaging and electronic health records using deep learning: a systematic review and implementation guidelines. *NPJ Digit Med*, v. 3, 2020. Disponível em: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7567861/>. Cited 2 times in pages 27 and 28.

SUN, Chen; SHRIVASTAVA, Abhinav; SINGH, Saurabh; GUPTA, Abhinav. Revisiting unreasonable effectiveness of data in deep learning era. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. [S.l.: s.n.], 2017. Cited in pag 20.

SUN, Ning; LI, Qi; HUAN, Ruizhi; LIU, Jixin; HAN, Guang. Deep spatial-temporal feature fusion for facial expression recognition in static images. *Pattern Recognition Letters*, Elsevier, v. 119, p. 49–61, 2019. Cited in pag 35.

SUN, Xiao; LV, Man. Facial expression recognition based on a hybrid model combining deep and shallow features. *Cognitive Computation*, Springer, v. 11, n. 4, p. 587–597, 2019. Cited in pag 35.

TALELE, Kiran; SHIRSAT, Archana; UPLENCHWAR, Tejal; TUCKLEY, Kushal. Facial expression recognition using general regression neural network. In: IEEE. *2016 IEEE Bombay Section Symposium (IBSS)*. [S.l.], 2016. p. 1–6. Cited in pag 35.

ULLAH, Fath U Min; KHAN, Noman; HUSSAIN, Tanveer; LEE, Mi Young; BAIK, Sung Wook. Diving deep into short-term electricity load forecasting: comparative analysis and a novel framework. *Mathematics*, MDPI, v. 9, n. 6, p. 611, 2021. Cited in pag 32.

ULLAH, Fath U Min; MUHAMMAD, Khan; HAQ, Ijaz Ul; KHAN, Noman; HEIDARI, Ali Asghar; BAIK, Sung Wook; ALBUQUERQUE, Victor Hugo C de. Ai-assisted edge vision for violence detection in iot-based industrial surveillance networks. *IEEE Transactions on Industrial Informatics*, IEEE, v. 18, n. 8, p. 5359–5370, 2021. Cited in pag 32.

ULLAH, Fath U Min; OBAIDAT, Mohammad S; MUHAMMAD, Khan; ULLAH, Amin; BAIK, Sung Wook; CUZZOLIN, Fabio; RODRIGUES, Joel JPC; ALBUQUERQUE, Victor Hugo C de. An intelligent system for complex violence pattern analysis and detection. *International Journal of Intelligent Systems*, Wiley Online Library, v. 37, n. 12, p. 10400–10422, 2022. Cited in pag 32.

WANG, Yingying; LI, Yibin; SONG, Yong; RONG, Xuewen. The influence of the activation function in a convolution neural network model of facial expression recognition. *Applied Sciences*, MDPI, v. 10, n. 5, p. 1897, 2020. Cited in pag 32.

WANG, Yao-Chin; QU, Hailin; YANG, Jing; ASDFASDF asdfkjafjkasfasdf. The formation of sub-brand love and corporate brand love in hotel brand portfolios. *International Journal of Hospitality Management*, v. 77, p. 375–384, 2019. ISSN 0278-4319. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0278431917303602>. Cited in pag 32.

WASI, Azmine Toushik; ŠERBETAR, Karlo; ISLAM, Raima; RAFI, Taki Hasan; CHAE, Dong-Kyu. *ARBEx: Attentive Feature Extraction with Reliability Balancing for Robust Facial Expression Learning*. 2023. Disponível em: <https://arxiv.org/abs/2305.01486>. Cited in pag 33.

XI, Zhenghao; NIU, Yuhui; CHEN, Jieyu; KAN, Xiu; LIU, Huaping. Facial expression recognition of industrial internet of things by parallel neural networks combining texture features. *IEEE Transactions on Industrial Informatics*, IEEE, v. 17, n. 4, p. 2784–2793, 2020. Cited in pag 35.

XU, Kelvin; BA, Jimmy Lei; KIROS, Ryan; CHO, Kyunghyun; COURVILLE, Aaron; SALAKHUTDINOV, Ruslan; ZEMEL, Richard S.; BENGIO, Yoshua. Show, attend and tell: Neural image caption generation with visual attention. In: *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*. [S.l.]: JMLR.org, 2015. (ICML'15), p. 2048–2057. Cited in pag 29.

YAN, Xiaoqiang; HU, Shizhe; MAO, Yiqiao; YE, Yangdong; YU, Hui. Deep multi-view learning methods: A review. *Neurocomputing*, v. 448, p. 106–129, 2021. ISSN 0925-2312. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0925231221004768>. Cited in pag 27.

ZHANG, Tao; JIA, Wenjing; HE, Xiangjian; YANG, Jie. Discriminative dictionary learning with motion weber local descriptor for violence detection. *IEEE transactions on circuits and systems for video technology*, IEEE, v. 27, n. 3, p. 696–709, 2016. Cited in pag 35.

ZHOU, Chong; PAFFENROTH, Randy C. Anomaly detection with robust deep autoencoders. In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: Association for Computing Machinery, 2017. (KDD '17), p. 665–674. ISBN 9781450348874. Disponível em: <https://doi.org/10.1145/3097983.3098052>. Cited in pag 26.