Abstract

Facial Emotion Recognition (FER) is a complex and crucial task in computer vision with widespread applications in fields such as human-computer interaction, psychology, healthcare, and marketing. Its goal is to accurately identify a person's emotional state based on their facial expressions. However, the inherent variability of facial expressions across individuals, cultures, and contexts presents significant challenges. Recent advancements in deep learning, particularly in Convolutional Neural Networks (CNNs), have greatly enhanced the performance of FER systems, enabling near-human accuracy, especially in cases where emotions are clearly expressed.

This thesis addresses FER in the context of multimodal datasets. The proposed method utilizes a CNN to extract salient features from facial images, coupled with a final transformer encoder and a classifier to predict the emotional state. Experimental results demonstrate that this approach achieves competitive performance using the CalD3rMenD3s, BU3DFE and Bosphorus datasets.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

This chapter provides a definition of facial analysis tasks in computer vision, focusing on Facial Expression Recognition (FER), its importance, applications and challenges. Particular attetnion is given to the currently available datasets and their characteristics. The chapter concludes with a brief description of this work contribution.

1.1 Facial Expression Recognition

Facial expression recognition (FER) is the process of automatically identifying and categorizing facial expressions in images or videos. It involves detecting emotions such as happiness, sadness, anger, surprise, disgust, and fear based on the facial features and expressions exhibited by individuals. FER is a fundamental task in human-computer interaction: it enables computers to interpret human emotions and respond accordingly, improving the interaction experience in virtual assistant applications, gaming, and social robotics. While facial expressions are a prominent way to convey human emotions; body movements, voice, and physiological muscle activity can also serve as standalone or additional modalities for recognizing emotions. However, facial expression is one of the most easily recognizable forms of non-verbal communication and is the most widely researched and applied in the field of affective computing. The potential applications of Facial Emotion Recognition (FER) span a wide range of fields, including neuroscience, healthcare, public safety, and anti-fraud systems. Some key examples include:

Healthcare: FER can play a significant role in diagnosing and treating mental health conditions such as depression, anxiety, and autism spectrum disorders. By monitoring and analyzing patients' emotional responses, healthcare providers can gain deeper insights into their mental states, leading to more personalized and effective treatment plans. Additionally, FER can be used to estimate pain levels in patients, providing a non-invasive method for assessing discomfort or distress.

Marketing and Advertising: FER allows for consumer reactions analysis to advertisements, products, and services. By understanding customers' emotional responses,

businesses can refine their marketing strategies, creating more targeted and engaging campaigns. For instance, chatbots and virtual assistants can be designed to respond empathetically to user sentiments, enhancing customer interactions and satisfaction.

Security and Surveillance: FER is increasingly utilized in public safety and surveillance systems to monitor public spaces for suspicious behavior. By identifying unusual emotional patterns or stress indicators, security systems can enhance preventive measures and improve incident response times. One practical application is driver fatigue detection, where monitoring a driver's facial expressions can help prevent accidents by alerting the driver or taking corrective actions.

Entertainment and E-Learning: In the realms of entertainment and education, FER can enhance user experiences by creating interactive environments that adapt to the user's emotional state. For example, in e-learning, FER can assess student engagement and attention levels during educational activities, allowing educators to tailor content and teaching methods to better suit individual learning needs. In gaming and entertainment, FER can create more immersive experiences by adjusting the narrative or gameplay based on the user's emotional reactions.

The growing interest in FER is driven by its potential to bridge the gap between human emotions and technology, leading to more intuitive and effective interactions in various contexts. As a result, there is a rising demand for next-generation software products that can dynamically adapt their behavior to human emotional states. Such applications are likely to be preferred by consumers over traditional, static software products. Moreover, the increasing adoption of wearable technology, such as sensors for tracking physiological signals like heart rate, along with analyzing tone of voice and body postures, is providing more precise contextual information for emotion detection. This integration of multisensory data allows for a more accurate and comprehensive understanding of an individual's emotional state. Consequently, the Internet of Things (IoT) is expected to drive the market for emotion detection systems, enabling more sophisticated and responsive applications across various industries.

According to Allied Market Research[10], the global emotion recognition market is expanding rapidly. It was valued at 21.7 billion in 2021 and is expected to reach 136.2 billion by 2031, with a Compound Annual Growth Rate (CAGR) of 20.5% from 2022 to 2031.

1.2 Tasks definitions

While we have already defined the task of Facial Emotion Recognition (FER), it's important to clarify that there is a broad spectrum of additional tasks requiring facial analysis. To avoid confusion, this section provides brief formal definitions of various computer vision tasks that involve inference over human faces in images, videos, or other representations.

Face Recognition (or Identification) refers to the task of determining the identity of an individual by comparing their facial features with a database of known faces. Given an input face image, the system searches through the database to find the closest match, effectively identifying the person. This technology is widely used in applications such as access control, surveillance, and law enforcement. Common techniques involve the extraction and comparison of facial landmarks, as well as more advanced deep learning approaches that leverage feature embeddings and metric learning.

Face Verification (or Authentication) involves confirming whether a presented face matches a specific identity or reference face. Unlike face recognition, which identifies an unknown individual from a database, face verification involves a binary decision: whether the input face corresponds to the claimed identity or not. This task is critical in applications such as secure login systems, where the system needs to confirm that the user is who they claim to be.

Face Attribute Analysis focuses on identifying and predicting specific attributes or characteristics of a face, such as age, gender, ethnicity, facial hair, presence of glasses, and facial symmetry. This analysis can be useful in demographic studies, targeted marketing, and improving the personalization of services.

Face normalization rrefers to the process of standardizing facial images to a common format, ensuring consistency across all images in terms of size, resolution, and appearance. This process can include various preprocessing steps such as alignment, lighting and color normalization, cropping, and resizing. Face normalization is crucial for reducing the variability in facial datasets, especially those acquired in unconstrained environments, where factors like illumination and pose can differ significantly. The goal is to prepare the dataset for subsequent machine learning tasks by ensuring that all images have similar characteristics.
Common sub-tasks involved in face normalization include:

Face detection is the primary preprocessing step that involves identifying and isolating human faces within an image. The goal is to remove non-facial areas, such as the background, neck, and shoulders, which are irrelevant to the analysis task. Historically, the Viola-Jones [11] algorithm was widely used due to its robustness and low computational cost. However, it struggles with occlusions, such as glasses or hair. Modern deep learning solutions like R-CNN [12] and FaceNet [13] have since become preferred choices due to their improved accuracy and ability to handle more challenging scenarios.

Facial landmark detection is a technique used to locate key points (landmarks) on a human face, such as the corners of the eyes, the tip of the nose, and the corners of the mouth. These landmarks are essential for capturing the deformations and movements associated with different facial expressions[14]. Convolutional Neural Networks (CNNs) are the most commonly used architectures for this task, due to their high accuracy and reliability. Many FER models utilize these landmarks as

part of the data annotation or as a pre-processing step.

Face alignment refers to the process of aligning facial images so that key landmarks are in consistent positions across different images. This process removes variations in pose, scale, and orientation, making subsequent facial analysis tasks more robust and accurate. Face alignment typically involves translation, rotation, and scaling operations to bring detected landmarks into alignment with a predefined template position in a reference frame.
As part of face alignment, **Pose normalization** is often employed to reduce unwanted variations caused by different head poses. Variations in pose can negatively affect the performance of facial analysis models, particularly in tasks like face recognition, where consistency in facial orientation is crucial. However, in tasks like FER, pose can sometimes convey valuable information about a person's emotional state. For instance, a tilted head might suggest curiosity, while a lowered head might indicate sadness. To address these challenges, some FER datasets, like Bosphorus [15], capture multiple views for each expression to help models learn pose-invariant patterns. In unconstrained environments, where head pose cannot be controlled, regularization techniques and pose normalization are often employed to make models more robust.

Although not exclusively facial-related, **Illumination Normalization** is an important preprocessing task, particularly for datasets collected in wild settings. The goal is to mitigate the effects of uneven lighting across images, which can lead to areas being overexposed or underexposed, resulting in loss of detail. A common strategy involves combining histogram equalization with illumination normalization, which has been shown to improve FER performance [16][17]. Histogram equalization adjusts the contrast of an image by redistributing pixel intensity values, enhancing visibility in both dark and bright areas. Illumination normalization standardizes lighting conditions across images, making facial features more detectable. Together, these techniques help standardize both contrast and illumination, ensuring more consistent and reliable feature extraction.

In summary, face normalization acts as a crucial preprocessing step that prepares facial datasets for subsequent feature extraction and analysis. The choice of normalization techniques depends on the specific requirements of the application, the characteristics of the dataset, and the desired trade-offs between model speed, accuracy, and robustness[14].

1.3 FER Datasets

The first datasets characteristic is the type of annotation: Ekman [18] proposed a **categorical model** comprising six basic facial expressions (anger, surprise, happiness, fear, sadness, and disgust) plus the neutral expression, asserting that all humans experience these emotions similarly regardless of the culture. In later studies,s also contempt is considered as an additional distinct emotion. However, recent progress

in psychology [19] asserts that the categorical models are not culture free: some expressions tend to occur with different intensity and frequency in subjects from different cultures. For example, [20] points out the fact that culturally universal emotions are actually only four.
The categorical annotation presents some limitations in describing all the possible emotions humans can experience. To address these limitations, other annotation models have been developed to represent a wider range of emotions. One approach is to switch from hard-labels (categorical) to soft-labels (continuous) which assigns scores to each label, allowing for the representation of mixed emotions in an image.

Another example is the **Facial Action Coding System** (**FACS**) [21] proposed by same authors of categorical model. It breaks down facial expressions into specific muscle movements, called **Action Units** (**AUs**), which correspond to changes in facial appearance (e.g., AU1 Inner Brow Raiser, AU26 Jaw Drop, AU43 Eyes Closed [1]). Unlike the categorical annotation, FACS is concerned with the physical manifestation of emotions through facial expressions, rather than the subjective experience of emotions themselves. It is particularly useful to understand how facial movements are related to emotions across different cultures. In practice, FACS is used along with the categorical model providing a more detailed description of facial expressions: FACS authors produced an Emotion Prediction Table listing facial configurations (in terms of AU combinations) of prototypic and major variants of each emotion (e.g., the occurrence of AUs: 6,12,25 is a sign of happiness, while AUs: 5,15,17 denote anger. Neutral expression is related to no muscle activation (AUs: 0) [3]). Figure 1.1 shows an example of a FACS annotation.

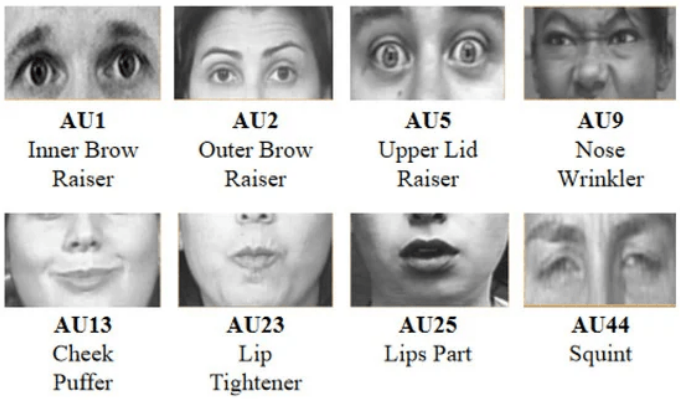


Figure 1.1: FACS annotation example from CK+ dataset [1]

Facial landmarks, mentioned in previous sections, and *AU* are different kinds of annotations: landmarks are points on the face used to describe the facial geometry (e.g., corners of the eyes, tip of the nose) while AUs are muscle movements that change the face's appearance. Note that landmarks contain more identity related information, but can be relevant in FER to let model focus on facial keypoints; instead AUs are identity-free and more expression oriented annotations. Figure 1.2 shows an example of landmarks manually annotated.

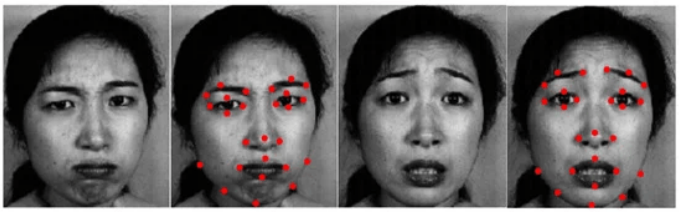


Figure 1.2: Landmarks annotation example from JAFFE dataset [2]

Another annotation model that describes the emotional state of a subject the **Circumplex model** which represents the emotions along two axis: **valence** and **arousal**. The former dimension quantifies whether the perceived emotion has a positive or negative prevalence, while the latter reflects whether an event is exciting/agitating or calm/soothing[5][3]. Figure 1.3 shows the Circumplex annotation model.

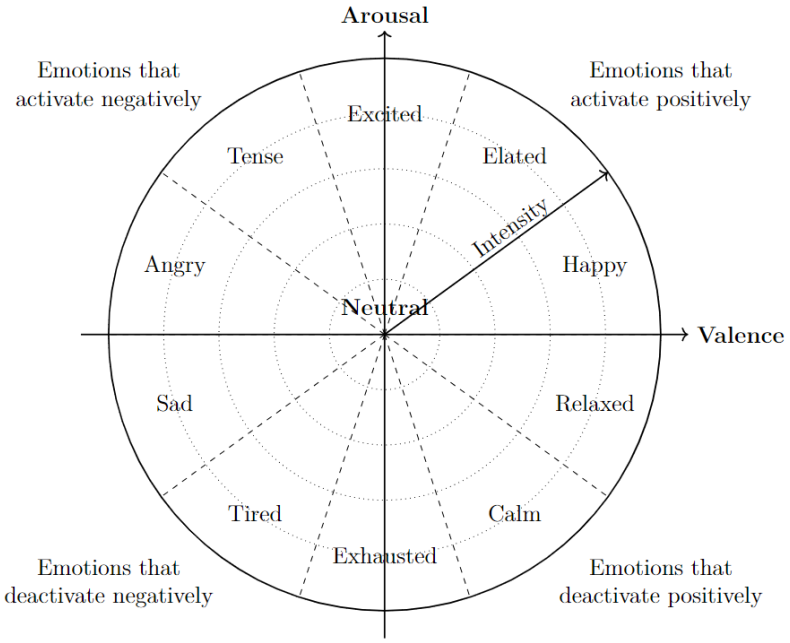


Figure 1.3: Circumplex annotation model

While soft labels, FACS and Circumplex model represent valid research, the simple categorical model remains the strongest baseline and is still widely adopted in recently collected datasets. This is mainly because of its easy representation: soft labels, FACS and Circumplex models introduce more annotation ambiguity. For example, it is not clear how to assign an emotional state to facial muscle activations (some works strictly follow the Emotion Prediction Table, other apply a more loose policy [1]). For the Circumplex model, there is no general method to assign arousal and valence values to a facial image sample. Assigning continuous labels increases the annotation ambiguity, leading to more uninformative variance in the data and making it hard for a model to learn relevant patterns linking such annotations to features.

Another dataset characteristic is the type of data collected: **static** or **dynamic**. Static datasets consist of single image instances, containing only spatial information. In contrast, dynamic datasets comprise sequences of frames that depict the evolution of facial expression over time. This allows for the extraction of features that incorporate both spatial information and temporal relationships among frames, which is very useful for FER. For example, head movements like nodding up and down or lip movements can be captured in dynamic datasets. In dynamic datasets, we can observe different collection methodologies regarding the expression evolution over time: some record the expression from a neutral state to its peak and then stop, while others record the entire evolution from neutral to peak and back to a neutral state. In the context of dynamic datasets, it is worth noting that emotional states are conveyed also through other dynamic signals, which can be collected in multimodal datasets along with facial data (e.g., audio and physiological recordings). These additional modalities provide complementary information and enhance the model's robustness[16].

Wether static or dynamic, datasets can be a collection of **2D** images (RGB or grayscale) or may include additional representations, such as depth maps that associate a distance value from the camera to each pixel (**2.5D**) or other **3D** representations like point clouds or meshes. These additional modalities help to mitigate issues like illumination variations and occlusion because they rely on the face's geometrical structure. 2D representations are cheaper and easier to acquire and process but are susceptible to environmental noise. In contrast, 3D representations offer a more robust description that is invariant to environmental variations irrelevant to the FER task (such as illumination changes), but they come at higher collection cost.

In this context, it's important to distinguish between datasets acquired in **lab settings** and those collected in **wild settings**. Lab settings are controlled environments where images typically have the same background and are shot with controlled angles, head poses, lighting, and occlusions. Subjects are carefully selected to be representative of different genders and ethnicities. Because of the high acquisition costs, laboratory controlled datasets have two issues: they generally contain a lot of repetition and are usually very small. Conversely, wild settings (unconstrained conditions) are much easier to acquire from the internet, ensuring more diversity and much bigger datasets (e.g., AFEW[22] datast contains video clips from movies). However, uncontrolled settings introduce more variance in quality, format, distribution of the subjects, illumination, pose, and occlusion, making inference more challenging and requiring for more elaborated feature extraction methods.

Finally, datasets can be categorized as either **spontaneous** or **posed**. Spontaneous datasets contain facial expressions that occur naturally when subjects are influenced by an external prompt. Spontaneous expressions are more representative of real-world scenarios, providing authentic data for FER tasks[14]. Collecting spontaneous expressions can be expensive due to the need for natural emotional reactions to images or videos, but the resulting data is better for creating models that perform

well in real-life applications. Posed datasets, on the other hand, involve subjects instructed to display specific expression having the same semantic meaning of the target expression [4]. These expressions are often exaggerated and may not reflect real-world spontaneous expressions. Posed datasets are easier to collect and ensure that all desired expressions are represented, which can be useful for initial training and baseline evaluations. Models trained on posed data generally reach better accuracy values but struggle with real-world FER where expressions are less pronounced.

In summary, the choice of dataset, whether static or dynamic, 2D or 3D, collected in lab or wild settings, and spontaneous or posed; impacts the performance and applicability of FER model trained on it. Figure 1.4 shows some examples from common datasets used for FER with respective modality, data type, solicitation and acquisition environment.

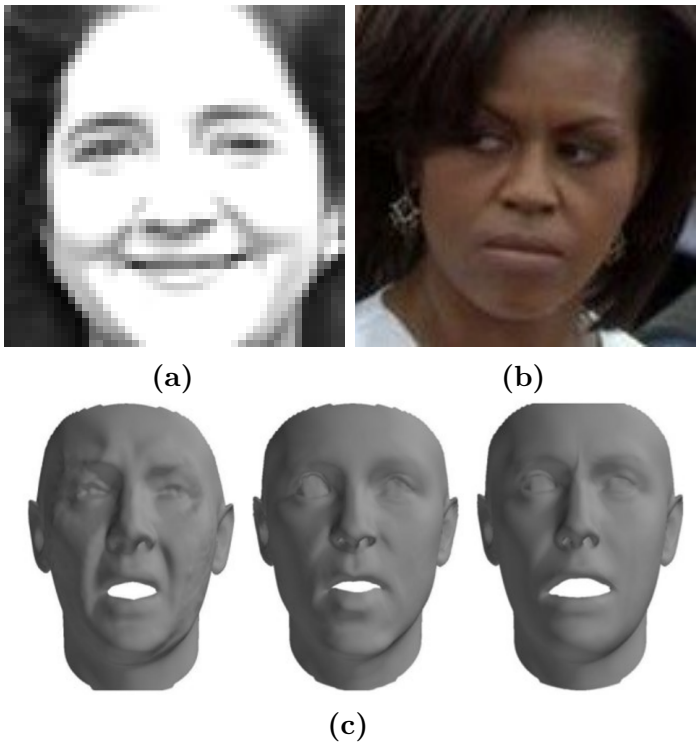


Figure 1.4: FER datasets examples: (a) CK+[1] sample (grayscale, static, posed, lab acquired), (b) bAffectNet[3] sample (RGB, static, spontaneous), (c) 4DFAB[4] sample (3D mesh, dynamic, spontaneous, lab)

1.4 FER's Challenges

Facial Emotion Recognition (FER) faces several challenges that impede its effectiveness and accuracy.

The first challenge is related to the **lack of diverse and exhaustive training**

data. Current state-of-the-art (SOTA) deep networks (especially transformer based ones) are very "data hungry" requiring large, high-quality, and highly diverse datasets to achieve a high level of generalizability[23]. The currently available datasets are neither sufficiently diverse nor large enough for exhaustive FER, leading to overfitting. Overfitting occurs when a model learns the training data too well, including its noise and outliers, and performs poorly on new data. To mitigate this, there is a significant reliance on data augmentation techniques and efforts to merge different datasets. However, merging is not a straightforward task because datasets can differ in many aspects, such as annotation types, collection methodologies, image quality, etc... Moreover, especially in spontaneous datasets, some subject's are not able to satisfactorily represent all the required emotions and therefore we deal with **unbalanced datasets** due to this missing data. Solutions may be to adjust the loss computation to deal with this umbalance or randomly drawn reduced class samples for the overly represented classes.

Note also that FER datasets are very difficult to annotate because of the subjectivity of emotions. This leads to **annotation ambiguity** which can be caused by the inherent ambiguity of emotional expressions or even human error. This noise can negatively impact the model's performance, as it learns from inconsistent labels, leading to overfitting: if annotations are inconsistent, models will learn patterns strictly related to training data rather than learning some underlying generalizable pattern[24]. To address this, some datasets are annotated by multiple annotators and the final label is determined by majority voting.

Another major issue, particularly prevalent in wild settings, is the **intra-class variability**. Intra-class variability refers to the inconsistency within a single emotion category. There can be significant variability in how individuals express the same emotion due to differences in age, ethnicity, gender, and expressivity [14]. Additionally, variations in head pose, illumination, partial occlusion of the subject's face, which are uncontrollable in unconstrained scenarios, contribute to this variability. This makes it difficult for models to detect common patterns across different samples belonging to the same expression class. To address this, normalization techniques are applied to standardize images to a common format, data augmentation is used to increase the diversity of the training data, and regularization techniques are employed to prevent overfitting. Also model architectures are designed to be robust to these variations.

Inter-class similarity is another challenge, where certain expression classes exhibit common characteristics. For instance, frowning eyebrows can be associated with emotions such as disgust, contempt, or anger. Similarly, expressions of surprise and fear may share some common facial configurations, such as wide-open eyes. This overlap can lead to confusion when the model attempts to distinguish between different emotion samples.

Another FER challenge is the models **sensitivity to the scale**: variations in the scale and resolution of the input images can lead to a significant drop in the model's performance. This is particularly true for datasets collected from wild settings (e.g.,

AffectNet is acollection of images from internet) where samples come in a wide range of resolutions and quality. Again, this issue is addressed by normalization techniques.

In summary, the challenges faced by FER include the *lack of diverse and exhaustive training data*, *class imbalance*, *annotation ambiguity*, *inter-class similarity*, *intra-class variability*, *scale sensitivity*. Overcoming these challenges is crucial for developing more accurate and reliable FER systems and requires strategies like data augmentation, data normalization, regularization and ad hoc model architecures capable of discerning subtle differences between similar emotional expressions. We will see in the state of the art Section 2.4 the most prominent solutions addressing these challenges.

1.4.1 Higher dimensional data representation

Due to the advantages of higher-dimensional representations described previously, research has made significant efforts to collect high-dimensional descriptions of data (2.5D or 3D). Different representations offer various trade-offs in terms of accuracy and computational complexity. A brief overview of the most common high-dimensional data representations is given below.

Depth Maps provide distance information from the camera for each pixel, offering depth information. They are usually referred as 2.5D representations (or RGB-D) because they are only able to represent the visible portion of the objects in the image: occluded objects are not fully described by depth maps. In facial datasets, depth maps are usually collected using depth sensors like Microsoft Kinect or Intel RealSense. However, they can also be reconstructed from RGB images by feeding a CNN with an RGB image and predict its depth channel. These networks are similar to the ones used in semantic segmentation task, since both tasks require "per-pixel" prediction, with a downsample-upsample architecture and a final convolutional layer with a single filter to be interpreted as depth values per pixel.
Standard loss functions that compare, for each pixel, the predicted depth with the ground truth would not perform well, because of **depth ambiguity** problem: a small object close to the camera looks exactly the same, to a camera sensor, as the same object which is two times bigger and positioned two times more distant. The solution is to use a scale invariant loss function that does not penalize a global multiplicative offset in scale.

Surface Normals provide the normal vector for each pixel, describing the orientation of the object at that pixel. Similar to depth maps, they still cannot fully represent the shape of occluded object portions. Again, to predict surface normals, a convolutional network with a downsample-upsample architecture can be used, employing a per-pixel loss that computes the angle between the predicted vector orientation and the ground truth.

Volumetric Pixels (Voxels) are the 3D equivalent of pixels, representing volume elements in three-dimensional space. Voxel grids represent 3D shapes into a

$V \times V \times V$ grid of occupancies (binary voxels) or arrays containing information such as color, density. The advantage is that voxel grids are discrete 3D representations where each voxel is equidistant from each other voxel, allowing for direct CNN manipulation using 3D convolutions. The drawbacks are the high spatial resolution needed to capture fine structures and the significant storage requirements, which grow cubically with resolution (e.g., storing a small binary voxel grid of $32 \times 32 \times 32$ voxels requires $4GB$ of memory).

3D Point Clouds represent the facial surface as a collection of 3D points characterized by coordinates x, y, z and in some cases also the RGB value. They provide a direct and straightforward representation of 3D data, capturing the geometry of the object in space. PCs can be acquired by scanning the object surface with proper scanners or by reconstructing it from multiple RGB images from different angles. In the latter case, datasets like FACESCAPE [25] shoot multiple high resolution images of the same subject from different angles, allowing for accurate PC reconstruction.
Using standard convolutional neural networks (CNNs) to process point clouds (PCs) is difficult because CNNs are designed for data like images, where the position of each pixel matters. In images, nearby pixels often have similar features, and CNNs take advantage of this local arrangement. However, point clouds are unordered; changing the order of the points in the data structure representing the PC doesn't change the shape they represent. For example, rearranging the points in a point cloud of a sphere still results in a sphere. If you treat a point cloud as an ordered sequence of points and apply a CNN to it, the CNN might give different results just because the point order is different, even though the object itself hasn't changed. This would be incorrect since the point cloud's representation should not depend on the order of the points.
To overcome this, PointNet [26] and PointNet++[27] networks have been developed to process point clouds directly, learning features from the unordered point data. These networks use a symmetric functions (e.g., Max pooling) to aggregate information from the points. These symmetric functions are permutation invariant, meaning they produce the same output regardless of the order of the input points. PointNet++ extends PointNet by introducing a hierarchical structure that captures local and global features at different scales, improving the network's ability to learn complex patterns from point clouds.

3D Meshes are usually generatedd from PCs by connecting points with edges (lines) forming triangular faces offering a continuous object's surface. Exactly as for PCs, meshes cannot be directly processed by CNNs because of their unordered nature. To overcome this, MeshCNN [28] has been developed which is a specialized type of CNN that defines convolution operations directly on the mesh's edges, allowing for feature learning without further processing.

In Figure 1.5 we can see a visual comparison between the different high dimensional data representations. A sample from CalD3rmenD3s [5] dataset is used. The 3D PC representations is extracted from the provided RGB and depth maps by associating the depth value to each pixel and rendering in the 3D space. Mesh is later obtained interpolating the points in the PC with the closest right and bottom points to form

triangular faces. Note these are not state of the art techniques for 3D reconstruction starting from RGB images, which are instead based on deep networks, but they are fast and widely used for representational purposes. Note also that PC in Figure 1.5c fails in giving a continuous representation of the face (especially for nose and cheecks regions) while the mesh is able to capture the face shape with a continuous representation but with poor resolution because we only have a single frontal view RGBD image for reconstruction.

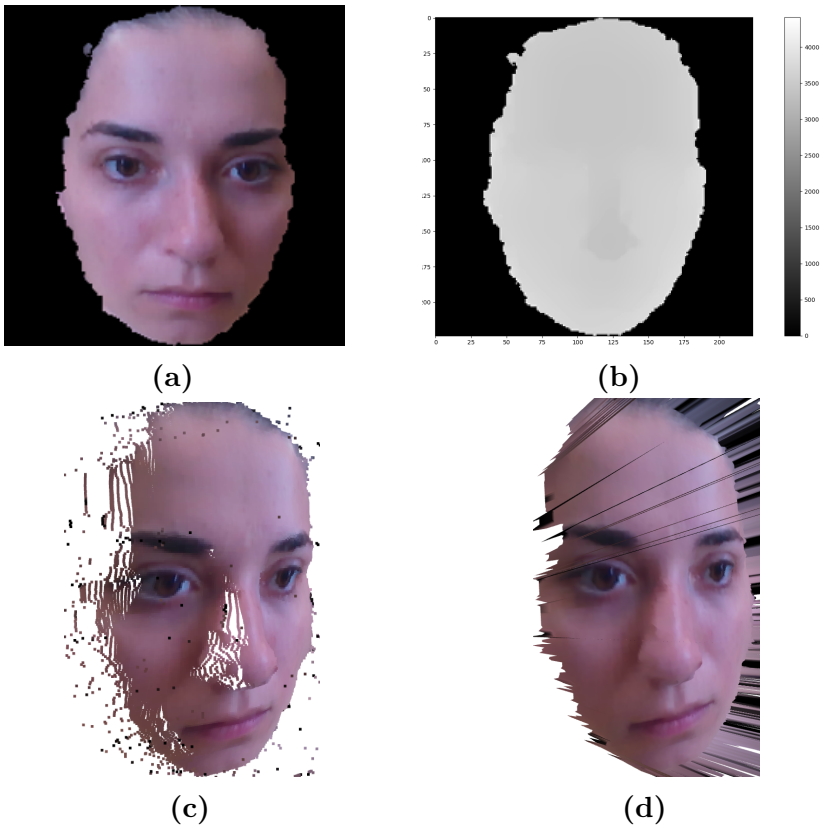


Figure 1.5: (a) RGB, (b) Depth map, (c) 3D Point Cloud, (d) 3D Mesh from CalD3R [5]

1.5 Datasets Review

This Section gives a more in depth description of the most common dataset for FER. Table 1.1 shows a visual comparison.

BU3DFE [9]: 3D meshes are reconstructed by 6 photos shot from different angles, but head pose is always frontal. Six standard expression are portrayed plus neutral expression. Each expression (but the neutral) is presented in 4 intensity levels (low, middle, high and highest). Each mesh is annotated with 83 landmarks. The emotions are mimicked by the subjects making BU3DFE a posed dataset. It is specified that the low intensity level should simulate the spontaneity of the emotional state. It contains 100 subjects, 56 female and 44 male with variety of ethnicities and ranging

age from 18 years to 70 years old. Two views are collected for each expression (+45° and-45°). This leads to 2550 two-views RGB images (1300×900) and geometric 3D meshes.

BU4DFE [29]: This is the dynamic evolution of the static BU3DFE. It contains 101 subjects, 58 female and 43 male with variety of ethnicities and age range of 18-45 years old. For each subject 6 mesh model sequences and 6 RGB video sequences are collected. The initial frame of each video is manually annotated with 83 landmarks. The six emotions are portrayed in the videos from neutral appearance through low intensity, high intensity and back to neutral intensity for a duration of approximately 4 seconds each. This leads to $6emotions \times 101subjects \times 100frames\_per\_video = 60600frames$ posed dataset since subjects, again, are requested to mimic each expression under psychologist supervision.

BU4DFE-S [30]: Here emotions are inducted naturally by submitting subjects to a variety of activities like listening to jokes to elicict happiness, experience unpleasant smell to activate disgust and watch documentaries to elicit sadness. In paticular 8 activities are conducted in order to activate: happiness, sadness, surprise, embarassment, fear, pain, anger and disgust. Moreover, subjects are free to move the head in different poses. There are 41 participants with different ethnicities, 23 woman and 18 men. For each subject and for each of the 8 activities 3D and 2D vidos are recorded. Out of $\sim 1min$ of recorded video, only the most expressive $20s$ are manually annotated with 27 Action Units (FACS), automatically tracked head pose and 2D/3D facial landmarks; for a total of 328 3D+2D sequences. Note that subjects are not explicitly asked to perform with different head poses, since the head pose is considered having a communicative value. The 2D texture videos are high quality ($1040 \times 1392pixels/frame$) and the 3D meshes contain $30000 \sim 50000$ vertices.

Bosphorus [15]: Contains 4652 samples manually annotated into 6 expressions plus neutral and Action Units from Facial Action Coding System (FACS). To be more precise they manually annotate 20 lower face AUs, 5 upper face AUs and 3 AU combinations. Also 24 facial landmarks are manually labelled. 13 poses are collected for each expression with different yaw, pitch and cross rotations. 4 types of realistic occlusions are used (hand, hair, eyeglasses, other accessories). Subjects (60 men and 45 women), with age range 25-35, are mostly Caucasian. Actually dataset is not complete as only 63 of the 105 subject have all the 6 emotions. The acquisition is made using Inspeck Mega Capturor II 3D which is a Stereo 3D digitizer device. It generates coloured texture image of the face (1600×1200 pixels) and point clouds consisting of approximetely 35000 points. Finally, this is a posed database as the subjects are requested to mimic the expressions.

FACESCAPE [25]: contains 18760 3D meshes from 938 different subject with age ranging between 16-70 and mostly Asian. Subjects are instructed to maintain a frontal head pose. No occlusion is possible and lightening is always the same. They extract 3D meshes from RGB images shot, in lab settings, by 68 different cameras from different angles. This is the largest and highest resolution facial dataset available, with 3D meshes containing $2M$ vertices. To make a comparison,

Bosphorus, 4DFAB, BU-3DFE and BU4DFE meshes contained respectively $\sim 35K$, $\sim 100K$, $\sim 10K - 20K$ and $\sim 37K$ vertices. The subjects are asked to mimic the expressions, making it a posed dataset. The expressions are 20 includigng neutral one.

CalD3R & MenD3S [5]: These are two spontaneous datasets. Differently from *BU4DFE-S* [30], emotions are all elicited by submitting images to subjects from the International Affective Picture System (IAPS) [31] and the Geneva Affective Picture Database (GAPED) [32]. In CalD3R there are 104 subjects, 54 woman and 50men mostly from South Europe with age range 19-35 years old. Instead MenD3S comprises 92 subjects, 46 woman and 46 men from Brazil and age range 18-55. They are used together in the presentation paper, bringing the number of subjects to 196. In both cases, light occlusions can be present on some images, such as those caused by hair, beard, and glasses. Different head poses are naturally portrayed. Each sample is described by RGB and Depth map acquired with Intel RealSense SR300 camera.

4DFAB [4]: To record 4D data, a DI4D capturing system has been used, consisting of 6 cameras. Moreover, one frontal grayscale camera to record frontal images and a Kinect to record RGBD has been used. The final database included 1.8M meshes and RGBD images (equivalent of 8.4 hours recording). Over 2-3 frames per subject, 79 landmarks have been annotated. Video clip viewing was the main way to elicit expressions and natural expressions were collected. The database includes 180 subjects (60 female and 120 males) aged from 5 to 75 with different ethnicities. Illumination and pose are constant, while occlusions are present in some cases. The database includes 6 basic expressions plus neutral.

FER2013[33]: A dataset from Kagggle competition of 35,887 grayscale images of faces labeled in 6+neutral classes. The images are collected from the internet and are of low quality, with variations in pose, illumination, and occlusion.

FER+[34]: is the same FER2013 dataset, but re-annotated with 8 classes instead of 7. The new class is "contempt". This has been done using 10 expert annotators and a most-voted strategy to define final label, to reduce annotation ambiguity present in the original FER2013 dataset. FER2013 is no more popular in the literature, since FER+ is a more reliable version of it.

CK+[1]: The Cohn-Kanade dataset is a dynamic dataset of 593 image sequences of 123 subjects with age range 18-50 years. 69% female and 21% male prevalently from Euro-American regions (81%), few from Afro-American (13%) and other regions (6%). The dataset is collected in a lab setting, with controlled lighting and single frontal head pose. Each sequence (with resolution 640×480)is annotated in categorical model (6 plus neutral) and the peak most expressive frame is annotated using FACS. The subjects are asked to mimic the expressions, making it a posed dataset.

AffectNet[3]: Affect from the InterNet is a static dataset of 1M images collected from the internet by querying 1250 emotion related tags. These 1M images are annotated with 66 landmarks automatically extracted with face alignment algorithm. Only 450000/1M are also manually annotated by professional annotators into 8

classes (6, plus neutral and contempt) and with Circumplex model (valence and arousal values).
This means that models for face verification and identification can be trained on the whole dataset, while models for FER can be trained on the subset of 450000 images. Moreover, to reduce annotation ambiguity coming from the fact that images from the internet present high variability, annotators were provided with additional 3 classes: *None* (for additional emotions like sleepy, bored, tired,shame, focused,...), *Uncertain* and *Non-face* (useful if the image is not a photo but, drawing, paint, animation,...). AffectNet is by far the largest database of facial affect in still images which covers both categorical and dimensional models.
Finally, it is worth to note tha papers using Affectnet dataset, make experiments with and without considering the contempt class. In the 7 classes experimentations, the accuracy reached are on average ~4% higher than in the full 8 classes case.

AFEW[22]: The Affect in the Wild dataset is a dynamic dataset of 1426 video clips from movies. The clips last from 300ms up to 5400ms and are annotated into 6 plus neutral categorical model. It can be considered as a Posed dataset since expressions are acted. Exactly as AffectNet[3], it present large variation in occlusions, poses, illuminations and subjects' characteristics, due to the fact it comes from movies.

ExpW[35]: The Expression in the Wild dataset is a static dataset of 91,793 images collected from the internet. The images are annotated into 6 plus neutral categorical model. The dataset is collected in a wild setting, with variations in pose, illumination, and occlusion. As for AffectNet[3], annotators were provided with an additional class for Non-face, later removed, in order to deal with large variance of images on the internet.

RAFDB[36]: The Real-world Affective Faces Database is a static dataset of 29672 images collected from the internet. The images are annotated into 6 plus neutral categorical model by 315 expert annotators. Moreover, also landmarks annotations are given (AU): 5 manually annotated and 37 automatically annotated. Since images are form the internet, different poses and illumination are present, but heavily occluded images have been discarded. So, as additional annotation, bounding boxes around the faces are provided. Subjects age ranges from 0 to 70 and are 52% female, 43% male, and 5% undefined. For racial distribution, there are 77% Caucasian, 8% African-American, and 15% Asian.

1.6 Contribution

In the following, this work's contribution can be summarised into:

- Review of the current state of the art in FER, with a focus on the most common datasets and the challenges faced by FER models.

- A proposal of a deep architecture that is able to deal with multimodal data

Table 1.1: FER Datasets

Database	Year	Data type[a]	Subjects	Samples	Annotation[b]	Acq.[c]	Sol.[d]	Multi- [e]					
								pose	ill.	occ.	age	gender	ethn.
BU3DFE [9]	2006	static RGB, M	100	2500	LM & 6+neut	Lab	P				✓	✓	✓
BU4DFE [29]	2008	dynamic RGB, M	101	~60600frames	LM &6	Lab	P				✓	✓	✓
BU4DFE-S [30]	2013	dynamic RGB, M	41	~328videos	LM & AU & 6+emb+pain	Lab	S	✓			✓	✓	✓
Bosphorus [15]	2008	static RGB, PC	105	4652	LM & AU & 6+neut	Lab	P	✓	✓	✓	✓	✓	
4DFAB [4]	2018	dynamic $RGBD, M$	180	1.8M frames	LM & 6+neut	Lab	S			✓	✓	✓	✓
CalD3r [5]	2023	static $RGBD$	104	4678	6+neut	Lab	S	✓		✓	✓	✓	✓
MenD3s [5]	2023	static $RGBD$	92	4038	6+neut	Lab	S	✓		✓	✓	✓	✓
FER2013 [33]	2013	static RGB	-	35887	6+neut	RW	P	✓	✓	✓	✓	✓	✓
FER+ [34]	2016	static RGB	-	35887	6+neut+cont	RW	P	✓	✓	✓	✓	✓	✓
CK+ [1]	2010	dynamic RGB	123	593videos	AU & 6+cont	Lab	P				✓	✓	
AffectNet [3]	2017	static RGB	-	450000	LM & 6+neut+cont &Circ	RW	P	✓	✓	✓	✓	✓	✓
AFEW [22]	2015	dynamic RGB	330	1426videos clips	6+neut	RW	P	✓	✓	✓	✓	✓	✓
ExpW [35]	2017	static RGB	-	91793	6+neut	RW	S	✓	✓	✓	✓	✓	✓
RAFDB [36]	2017	static RGB	-	29672	LM & 6+neut	RW	P	✓	✓	✓	✓	✓	✓
FACESCAPE [25]	2020	static RGB, M	-	18760	LM & 20	Lab	P				✓	✓	

[a] Data Type: PC= Point Cloud, M= Mesh, $RGBD$= RGB+Depth
[b]Annotation: 6=(anger,disgust,fear,happiness,sadness,surprise), neut=neutral, emb=embarassement, cont=contempt, AU= action units,Circ=circumplex (valence and arousal), LM= landmarks
[c]Acquisition Conditions: L=Laboratory, RW=Real World
[d]Solicitation: P=posed, S=spontaneous
[e]Multi=pose, illumination, occlusion, age,gender, ethnicity

representation (RGBD) and to learn from the unbalanced dataset. The architecture is based on the state of the art in FER and is designed to be robust to the challenges faced by FER models.

- Loss Function analysis and evaluation of the model's performance on the CalD3rMenD3S dataset in different conditions.

- Merge of the CalD3rMenD3S, BU3DFE and Bosphorus datasets to create a larger and more diverse dataset and final evaluation of the model's performance on this new dataset.

Chapter 2

Related Works

This chapter provides an overview of the most relevant works in the FER field. Section 2.1 describes most common baseline architectures and loss functions relevant to the FER task. Next Section 2.4 breaks down the state of the art (SOTA) models adopted to solve FER. Finally, Section 2.5 points out some useful considerations for developing our network proposal in next Chapter 3.

2.1 Foundations

Traditional machine learning methods prior to deep learning, relied on three steps: preprocessing, feature extraction and classification. Noteworthy "shallow" feature extraction techniques include Local Binary Pattern (LBP), Histogram of Oriented Gradient (HOG) and Supervised Descent Method (SDM). These methods, which involve manual feature engineering, will not be described here as they have been gradually replaced by deep learning due to its superior performance and generalization capabilities across various tasks.
In fact, deep networks are more capable, with respect to shallow techniques, to detect very intricate patterns from raw or minimally preprocessed data, eliminating the need for manual feature extraction. Furthermore, the flexibility of deep networks extends to their ability to serve as feature extractors, followed by independent classifiers like Support Vector Machines (SVM) or decision trees.

One of the key advantages of deep networks lies in their scalability. As datasets grow in size and complexity, deep networks can be expanded in depth and width to enhance performance. However, deep networks are more susceptible to overfitting when dealing with smaller datasets, than shallow approaches. **Overfitting** occurs when a model performs well on the training data but fails to generalize to unseen test data. This may happen when the model is too complex and/or trained too much, such that it learns too strict relationships in the training set that are not generalized over unseen data. Overfitting also happens when dataset is too small: if training data is not diverse enough, the model may memorize the training data rather than learning underlying patterns in it. This is a problem in FER because, as noted in

Section 1.4, datasets are usually small and contain lots of repetitions (especially lab acquired ones). To combat overfitting, various regularization techniques such as data augmentation, weight decay, batch normalization, and dropout are commonly employed. These methods help prevent the model from memorizing specific instances in the training data and encourage learning more generalized patterns.

While deep learning has largely overshadowed traditional shallow algorithms in terms of accuracy and performance, shallow methods still hold relevance in scenarios like FER where extensive datasets are not available[14]. This is because, shallow algorithms leverage manually engineered features, like LBP and HOG, designed by domain experts to capture specific data characteristics relevant to the task at hand. In other words, they are designed to be robust and informative features that generalize well to unseen data, even when the dataset is small. This "data efficiency" of shallow algorithms is the reason why they are still alive along with deep approaches in the so called **hybrid models** which are still being developed in FER and other computer vision tasks (e.g., Ma et al., fuse features from RGB and LBP modalities in [37]).

In this work, the focus is exclusively on exploring deep architectures addressing FER challenges and achieving superior performance.

2.1.1 CNNs

A Convolutional Neural Network (CNN) is a class of deep neural networks primarily designed to process visual data such as images and videos. CNNs have gained significant prominence in the field of computer vision due to their ability to automatically extract and learn hierarchical features from raw input data. The architecture of a CNN is characterized by convolutional layers, pooling layers, and fully connected layers, which work together to enable the network to understand spatial hierarchies and patterns within the input data.

CNN are composed by stacking convolutional layers, where filters are convolved with the input data to detect features such as edges, textures, and shapes. Pooling layers are utilized to downsample the feature maps generated by the convolutional layers, reducing the spatial dimensions while retaining essential information. The fully connected layers at the end of the network perform classification based on the learned features, enabling the CNN to make predictions on the input data.

CNNs leverage parameter sharing and spatial hierarchies to learn increasingly complex features as the network progresses through its layers. This hierarchical feature learning allows CNNs to achieve superior performance in tasks such as image classification, object detection, and facial recognition. The success of CNNs can be attributed to their ability to automatically learn discriminative features from data, making them a powerful tool in various computer vision applications.

Some of the most popular CNN architectures include, VGGNet, GoogLeNet, ResNet, and DenseNet. Figure 2.1 shows a VGG16 architecture used for ImageNet [38] classification: note that the network consists of 13 convolutional layers interleaved by max pooling layers reducing the spatial dimensions while retaining essential information. Finally 3 fully connected layers reduce the spatial dimensions to 1000 classes.

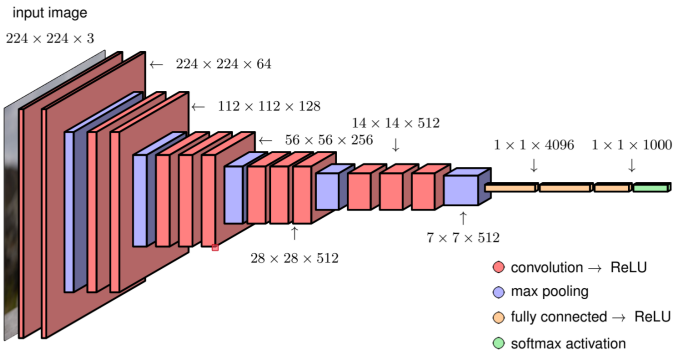


Figure 2.1: VGG16 architecture [6]

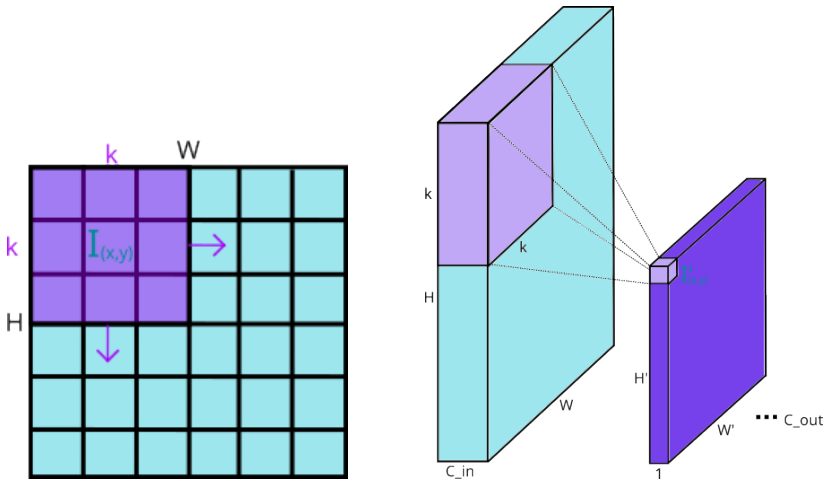
Among the many architectures that have been proposed for image classification, recent development has focused in properly scaling the network depth, width, and resolution to achieve optimal performance. This is known as *compound scaling*, and it is the basis for the EfficientNet[39] architecture presented in 2020. EfficientNet has demonstrated superior performance compared to other architectures like ResNet while requiring significantly fewer parameters and computational resources.

The problem is to find out which is the most general purpose, best performing architecture to scale as needed. We give here a brief overview of the layers which are used to build more complex CNNs.

Convolutional Layer: A convolutional layer applies a set of filters to the input data to extract features. Each filter is convolved with the input data to produce a feature map, which captures specific patterns. The output pixel $I'(x, y)$ of a 2D (descrete) convolution between the input data $I(x, y)$ and a filter K, for a single channel, is computed as follows:

$$\mathbf{I}'(x, y) = \sum_{i=-m//2}^{m//2} \sum_{j=-n//2}^{n//2} \mathbf{I}(\mathbf{x} - \mathbf{i}, \mathbf{y} - \mathbf{j}) \cdot \mathbf{K}(i, j) \tag{2.1}$$

where m and n are the dimensions of the filter K, and $m//2$ is the floor division of m by 2. Usually $m = n$ and we will call it k from now on. An intuitive explanation is given in Figure 2.2 where we can see the filter sliding across the input data with a given stride and computing the dot product between the filter weights and the input data at each position, then summed to produce the pixel in the output. Note that in such standard convolution the depth of the filter is equal to the depth of the input data C_{in}.



(a) **(b)**

Figure 2.2: Standard Convolution: (a) 1 channel frontal view, (b) multi-channel view

The dimension of the output feature map $H' = W'$ is given by:

$$H' = \frac{H - k + 2p}{s} + 1 \tag{2.2}$$

where p is the padding and s is the stride.

The number of parameters in a convolutional layer is:

$$N_{par} = k^2 \cdot C_{in} \cdot C_{out} \tag{2.3}$$

while the number of multiplications is:

$$N_{mult} = k^2 \cdot C_{in} \cdot C_{out} \cdot H \cdot W \tag{2.4}$$

Pooling Layer: Pooling layers are conceptually similar to convolution layers, but are used to reduce the spatial dimensions of the input data without using learnable parameters. They are useful to reduce the amount of computation required for training and inference. There exist different types of pooling layers, such as max pooling and average pooling, respectively reported in equations 2.5 and 2.6:

$$\mathbf{I}'(\mathbf{x}, \mathbf{y}) = \max_{i=0}^{k-1} \max_{j=0}^{k-1} \mathbf{I}(x + i, y + j) \tag{2.5}$$

$$\mathbf{I}'(\mathbf{x}, \mathbf{y}) = \frac{1}{m \cdot n} \sum_{i=0}^{k-1} \sum_{j=0}^{k-1} \mathbf{I}(x - i, y - j) \tag{2.6}$$

Fully Connected Layer: Fully connected layers are applied to 1D features extracted from the convolutional and pooling layers. They multiply each channel in the input 1D feature by a proper learnable weight enabling the network to learn

complex patterns. Fully connected layers are typically used in the final layers of the network, where feature's spatial dimensions $H \times W$ have been collapsed to 1×1, to perform classification. Equation 2.7 shows the mathematical representation of a fully connected layer.

$$\mathbf{y} = \mathbf{W} \cdot \mathbf{x} + \mathbf{b} \tag{2.7}$$

where $\mathbf{y} \in \mathbb{R}^n$ with n being the number of output channels. $\mathbf{x} \in \mathbb{R}^m$ is the input feature with m being the input channels. $\mathbf{W} \in \mathbb{R}^{n \times m}$ is the weight matrix and $\mathbf{b} \in \mathbb{R}^n$ is the bias.

Depthwise Separable Convolution: It replaces the standard convolution with a *depthwise convolution* followed by a *pointwise convolution*. The depthwise convolution applies $k \times k$ filters to each input channel separately. Next, the pointwise convolution applies 1×1 filters. This separation of channel-wise and then spatial-wise convolutions attains the same purpose of a standard convolution (feature extraction) but reduces the number of parameters and computations required, making the network more efficient. Infact, the number of parameters in a depthwise separable convolution is given by:

$$N_{par} = (k^2 \cdot C_{in}) + (1 \cdot C_{in} \cdot C_{out}) \tag{2.8}$$

The number of multiplications is given by:

$$N_{mult} = (k^2 \cdot C_{in} \cdot H \cdot W) + (C_{in} \cdot C_{out} \cdot H \cdot W) \tag{2.9}$$

where in both equations, the first addend is related to depthwise convolution, while second term is related to pointwise convolution.
Figure 2.3 shows a graphical representation of a depthwise separable convolution: note that the depthwise convolution is applied to each channel separately while the pointwise convolution is applied pixel wise for all channels. Depthwise Separable Convolution is now our main block as it is more efficient than a standard convolution.

Inverted Residual Block: MobileNetv2 [40] furtherly developed the depthwise separable convolution, introducing the *inverted residual block*, which consists of:

1. *Pointwise-Conv (Expansion)*: pointwise 1x1 convolution that expands the number of channels to $C' = t \cdot C_{in}$ where t is a fixed expansion factor.

2. *Batchnorm2D* and *SiLu*
3. *Depthwise-Conv*: depthwise convolution (usually 3×3)that reduces spatial dimensions to $H' \times W'$, leaving the number of channels unchanged.

4. *Batchnorm2D* and *SiLu*
5. *Squeeze-Excite*: MobileNetv3 [41] adds an attention mechanism that allows for selection of most informative channels. The squeeze step is a simple global

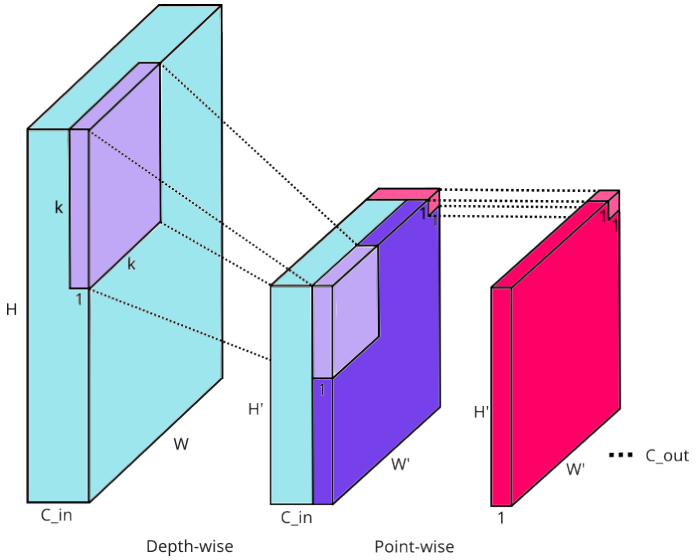


Figure 2.3: Depthwise Separable Convolution

average ($H \times W \times C \rightarrow 1 \times 1 \times C$). The excitation step is performed applying a first FC layer that reduces the channel dimension by a reduction ratio r ($1 \times 1 \times C \rightarrow 1 \times 1 \times C/r$) and a second FC layer that restores the channel dimension C. The output is passed into a sigmoid and interpreted as a weight vector that is multiplied element-wise with the input tensor to select the most informative channels.

6. *Pointwise-Conv (Compression)*: pointwise 1x1 convolution that reduces number of channels to C_{out}.

7. *Batchnorm2D*

A skip connection is present only for those inverted blocks in which $C_{in} = C_{out}$.

Note that the last pointwise convolution is linear (no activation function). The reasons are explained more in detail in [40], we can summarize by saying that the last pointwise convolution (compression) reduces the high-dimensional feature map into a smaller dimensional representation assuming that uninformative features have been filtered out by previous layers. Applying an additional non-linear activation function like ReLU could potentially bring to zero informative negative values and lead to a loss of information.

This is called 'inverted' because the wide part containing more channels is in the middle of the block (*Depthwise-Conv*), instead of at the start/end like in ResNet.

The depthwise convolution is 3×3 for the first inverted residual blocks, then can change to 5×5 or 7×7 for the subsequent blocks in order capture more spatial context.

The most common activation function used in inverted residual block is Swish (also called SiLU):

$$\mathbf{SiLu}(x) = x \cdot \sigma(x) \tag{2.10}$$

where $\sigma(x)$ is the sigmoid function $f(x) = \frac{1}{1+e^{-x}}$.
Its advantage is that it is differentiable everywhere and smoother than ReLU in the neighborhood of 0, but it requires sigmoid computation which is more expensive.

Finally, Figure 2.4 shows a graphical representation of an inverted residual block: note that the squeeze-excitation layer is applied at the point of maximum channel dimension.

Inverted residual block is the basic building block for recent highly performing networks in MobileNet and EfficientNet families which we will use in this work.
It is also worth noting that new generation of MobileNet-v4 [42] presented in 2024, are built as stacks of inverted residual blocks (in which they add an extra initial *Depthwise-Conv* before the *Pointwise-Conv (Expansion)*), and end with Multihead Self Attention mechanism in an "hybrid" setup between CNN and Transformer which generally reaches better performance in terms of accuracy over a variety of datasets. This confirms what we will discover later in related works section that hybrid models; using CNNs to extract feaatures and Transformer to learn long-range dependencies between them, are becoming very popular.

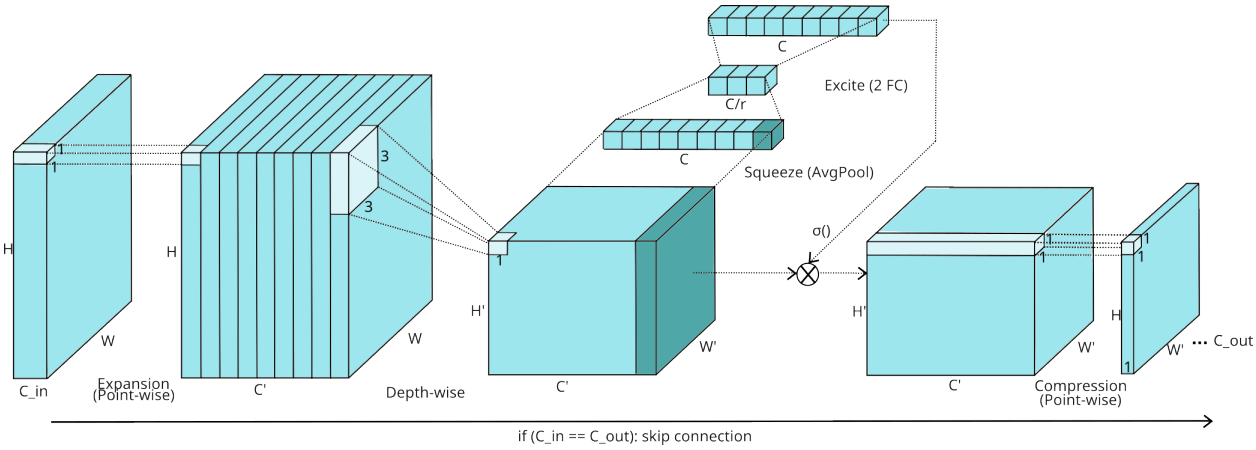


Figure 2.4: Inverted Residual Block

2.1.2 Transformer

A Transformer [43] is a type of deep learning model that has revolutionized natural language processing (NLP) tasks by introducing a novel architecture based on self-attention mechanisms. Unlike traditional recurrent neural networks (RNNs) and convolutional neural networks (CNNs), Transformers do not rely on sequential

processing, allowing for parallelization and more efficient training on long sequences of data.

At the core of a Transformer model are self-attention mechanisms, which enable the model to weigh the importance of different elements in a sequence when making predictions. By attending to all elements simultaneously, Transformers can capture long-range dependencies and contextual information more effectively than traditional models like RNNs or CNNs. In practice, self-attention computes a relevance score for each element of the input data, allowing the model to determine which parts of the input data to focus on. As shown in Figure 2.5, the Transformer architecture employs three learnable matrices: Query, Key, and Value. The Query matrix W_Q computes the attention scores, the Key matrix W_K guides the model's attention, and the Value matrix W_V generates the output of the attention mechanism. The output of the attention mechanism is subsequently processed through a feedforward neural network to yield the final model output.

Typically, the input sequence is divided and processed concurrently by multiple self-attention heads, each specializing in learning distinct patterns within the input data. By stacking numerous self-attention layers, Transformers can discern intricate patterns in the input data and capture extensive dependencies. Transformers have demonstrated unparalleled performance in a myriad of natural language processing tasks, including machine translation, text generation, and text classification. Some of the most prominent transformer architectures encompass BERT, GPT, and T5, which have set new benchmarks in NLP research and applications.

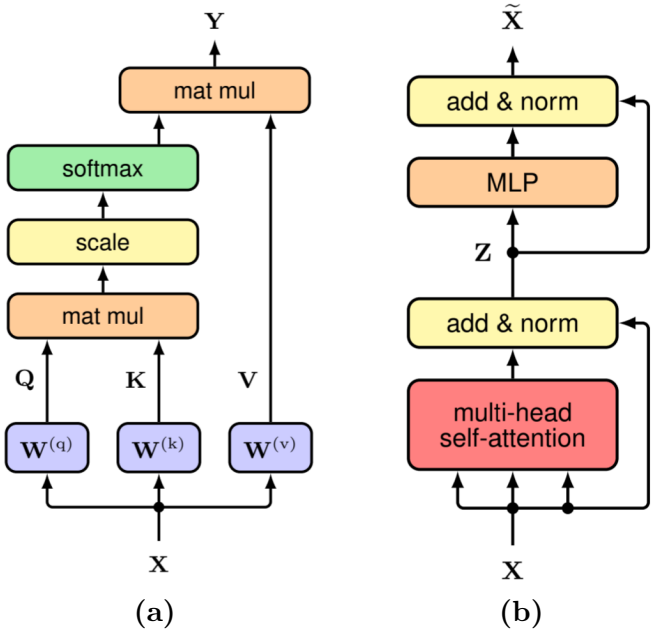


Figure 2.5: (a) Single Self-Attention head ('scale' refers to the normalization of the input to the softmax.), (b) Transformer Encoder layer [6]

Vision Transformer

In 2020, transformers have found application in computer vision tasks, including image classification, object detection, and image segmentation, showcasing promising outcomes. The Vision Transformer (ViT) [44] operates by deviding the input image into a grid of patches, which are subsequently flattened and processed through a sequence of transformer layers to extract meaningful features from the image. This allows the model to capture spatial relationships and contextual information between the image patches, enabling it to make accurate predictions. Vision transformers have demonstrated SOTA performance in image classification tasks, notably excelling in benchmarks like ImageNet. The architecture of a Vision Transformer is depicted in Figure 2.6.

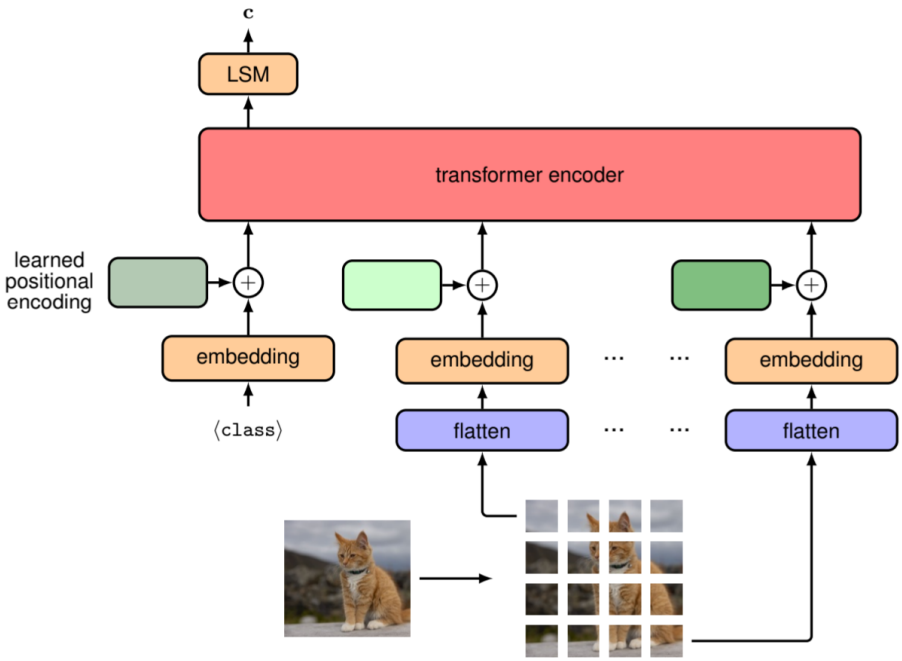


Figure 2.6: Vision Transformer (ViT) architecture [6]

2.2 Loss Layers

The loss layer is a crucial component of the network, as it defines the objective function that the network aims to minimize during training. The choice of the loss function depends on the specific task and the characteristics of the dataset. In FER context, the loss function should take into account the FER's inherent challenges, in particular the high intra-class variance and high inter-class similarity. Furthermore, FER dataset's are usually very unbalanced as some emotions like surprise or contempt are much more infrequent and/or difficult to annotate with respect to others like anger. In order to mitigate these problems, a possible solution is to make use of robust loss layers reported in this Section.

2.2.1 Softmax with Cross Entropy Loss

The Softmax Layer with Cross Entropy Loss is a standard approach used in many classification tasks. The softmax function in Equation 2.11 is typically applied to the output layer of the deep network to convert raw output logits into a probability distribution.

$$\boldsymbol{p}_i = \frac{\exp\left(\boldsymbol{W}_{y_i}^T \boldsymbol{x}_i + b_{y_i}\right)}{\sum_{j=1}^{K} \exp\left(\boldsymbol{W}_j^T \boldsymbol{x}_i + b_j\right)} \tag{2.11}$$

Where $\boldsymbol{p}_i$ is the probability distribution associating to sample i a probability score (p_{ij}) for each class j; $\boldsymbol{x}_i$ is the deep feature of sample i. $\boldsymbol{W}_{y_i}^T$ and $\boldsymbol{W}_j^T$ are, respectively, the column associated with class y_i and the j_{th} column, of the weight matrix of the last fully connected layer. finally, K is the number of classes. In simpler words $\boldsymbol{W}_{y_i}^T \boldsymbol{x}_i + b_{y_i}$ is the score computed by the network for sample i to belong to correct class y_i.
The Cross Entropy Loss (CE loss), reported in Equation 2.12, is then applied to the output of the softmax layer to compute the loss between the predicted probability distribution and the true labels.

$$\mathcal{L}_{CE} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{K} w_j \; y_{ij} \; log(p_{ij}) \tag{2.12}$$

Where N is the number of samples in the training set. Actually, in deep learning N is the number of samples in the mini-batch because we are not able to compute a complete epoch for each update of the network. K is the number of classes, y_{ij} is the binary true label for sample i and class j ($y_{ij} = 1$ if sample i belongs to class j and $y_{ij} = 0$ otherwise).
Or in vectorized formulation:

$$\mathcal{L}_{CE} = -\frac{1}{N} \sum_{i=1}^{N} \boldsymbol{w} \cdot \boldsymbol{y}_i \cdot \log(\boldsymbol{p}_i)^T \tag{2.13}$$

Where $\boldsymbol{y}_i$ is the one-hot encoded label vector for ith sample.

One problem with standard CE loss is that it treats every sample equally, which can be problematic in FER where class imbalance is very common. This can lead to biased models that perform poorly on minority classes. To address this, a weighted version of CE loss can be used, where each class is assigned a weight w_j based on its frequency. This encourages the model to focus more on underrepresented classes, leading to improved performance on imbalanced datasets.

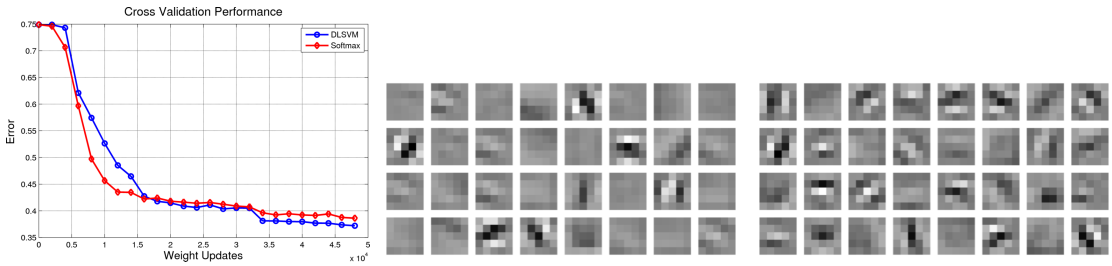
2.2.2 Hinge Loss

Hinge Loss is a margin based loss from Support Vector Machine (SVM) that has been demonstrated to have some benefits with respect to simple CE loss in end-to-end training over two standard datasets such as MNIST and CIFAR-10 [7]. CE loss uses a softmax function to the output layer of the neural network, converting raw scores

(logits) into probabilities that represent the likelihood of each class. In contrast, multiclass SVM aims to find the maximum margin hyperplanes that separates the data points of different classes. So, it is applied directly on the logits. The hinge loss is then used to penalize misclassifications beyond a certain margin to learn robust decision boundaries that generalize well to unseen data. This is usually advantageous in datasets like MNIST and CIFAR-10, where the classes are well-separated. The squared hinge loss formulation for a multi class task is reported in Equation 2.14.

$$\mathcal{L}_{SVM} = \frac{1}{N}\sum_{i=1}^{N}\sum_{j\neq y_i}^{K} max(0, 1 + (p_{ij}) - (p_{iy_i}))^2 \tag{2.14}$$

Intuitevely, the hinge loss penalizes misclassifications beyond a certain margin, which is set to 1 in the standard formulation. If the score of the correct class (p_{iy_i})is greater than the score of the incorrect class (p_{ij}) by more than the margin, the loss is zero. Otherwise, the loss is (quadratically) proportional to the difference between the scores. In particular, CE loss loss penalizes misclassifications exponentially based on the confidence of the prediction, which can be heavily influenced by outliers. In contrast, SVM loss only penalizes misclassifications beyond a certain margin, making it more robust to outliers.
A practical comparison can be found in Image 2.7 where a CNN is trained on the ICML 2013 FER dataset, using CE loss and SVM loss. We can see that, after learning rate is decreased in the last half of the training, SVM loss keeps decreasing more consistently than CE loss which instead has a plateau. Moreover, note that CNN trained with SVM tends to have more textured filters with respect to CNN trained with softmax.



(a) **(b)** **(c)**

Figure 2.7: CE vs SVM losses [7]: (a) Performance of CNN trained with Softmax (red) and L2-SVM (blue), (b) Filters from CNN trained with Softmax, (c) Filters from CNN trained with SVM

2.2.3 Center Loss

To adddress high intra-class variability and high inter-class similarity problems, center loss permits the network to produce features that are both separable (featuers from different classes are far apart in the feature space) and discriminant (features from same class are close to each other in the feature space and so encoding the class

characteristics). The center loss is reported in Equation 2.15.

$$\mathcal{L}_c = \frac{1}{2}\sum_{i=1}^{N}\|\boldsymbol{x}_i - \boldsymbol{c}_{y_i}\|_2 \tag{2.15}$$

Where $\boldsymbol{c}_{y_i}$ is the class center of the class y_i. Center loss increases the discriminative power of the features by explicitly penalizing the distance between the deep features $\boldsymbol{x}_i$ of each face image and their corresponding class centers in the feature space $\boldsymbol{c}_{y_i}$. Ideally, the class centers should be learnt by computing the mean of the deep features produced at each step for all the samples of the same class in the training set. However, this would be inefficient and impractical. So, class centers are actually updated at each iteration by averaging the deep features of the samples in the mini-batch [45]. This may introduce large perturbations in the learninng of the centers (for example, a mini-batch could contain only samples from a single class with a mean very different from the global mean). To avoid this, the learning rate of the centers is controlled by an hyperparameter $\alpha \in [0,1]$.
So an actual center update is computed at each mini-batch through the following SGD update rule:

$$\boldsymbol{c}_j^{t+1} = \boldsymbol{c}_j^t - \alpha\, \mathrm{d}\boldsymbol{c}_j^t \tag{2.16}$$

Where $\mathrm{d}\boldsymbol{c}_j^t$ is computed in the following way:

$$\mathrm{d}\boldsymbol{c}_j^t = \frac{\sum_i^N \sum_j^K y_{ij}\,(\boldsymbol{c}_j^t - \boldsymbol{x}_i)}{1 + \sum_i^N \sum_j^K y_{ij} \cdot 1} \tag{2.17}$$

Note that the numerator is exactly the gradient of the center loss with respect to the class center $\boldsymbol{c}_j$. The denominator is an extra normalization term that ensures that the update is appropriately scaled relative to the number of samples contributing to the gradient. So, if there are many samples in a class, each individual sample should have less influence on the update of the center, ensuring a more stable and consistent update. Moreover, the +1 term in the denominator is used to avoid division by zero when there are no samples for a class in the mini bacth.

The CE loss encourages features separability, reducing the inter-class similarity, but doesn't act on the discriminative power of the features [45]. Therefore, center loss is used along with the standard CE loss in Equation 2.18.

$$\mathcal{L} = \mathcal{L}_{CE} + \lambda\mathcal{L}_C \tag{2.18}$$

where λ is an hyperparameter that balances the two loss functions. Intuitively, the CE loss forces the deep features of different classes staying apart while the center loss efficiently pulls the deep features of the same class to their centers.

Note that other loss functions have been proposed to enhance the discriminative power of the deeply learned features, such as contrastive loss [46], triplet loss [47] which construct the loss function for image pairs and triplets respectively[45]. However, the number of possible pairs and triplets grows quadratically and cubically respectively with the number of samples in the minibatches. A common strategy is

to select meaningful pairs and triplets (those that give significant contribute to the training), but these searching algorithm increase training time. In contrast, center loss is computationally efficient and easy to implement, making it a popular choice for enhancing the discriminative power of the features in FER tasks. This comes at the cost of fine-tuning the hyperparameters λ to balance the two loss functions, and α to control the learning rate of the centers.

2.2.4 Island loss

The limit of center loss is that it only compresses the clusters individually, but does not push clusters apart. This is why it is used in conjunction with CE loss which forces the fetures of different classes to stay apart.

In 2017 Cai et al. [8], further developed the center loss to produce even more discriminative features and trained a CNN with Island Loss for FER. The combined loss will be therefore the sum of Island Loss, center loss and CE loss as presented in Equation 2.19.

$$\mathcal{L} = \mathcal{L}_{CE} + \lambda_1 \mathcal{L}_C + \lambda_2 \mathcal{L}_I \tag{2.19}$$

Where λ_1 and λ_2 are hyperparameters that balance the three loss functions. $\mathcal{L}_I$ is the Island Loss, defined in Equation 2.20.

$$\mathcal{L}_I = \sum_{\boldsymbol{c}_j}^{K} \sum_{\boldsymbol{c}_k \neq \boldsymbol{c}_j}^{K} \left(\frac{\boldsymbol{c}_j \cdot \boldsymbol{c}_k}{\|\boldsymbol{c}_k\|_2 - \|\boldsymbol{c}_k\|_2} + 1 \right) \tag{2.20}$$

$\boldsymbol{c}_j$ and $\boldsymbol{c}_k$ are the class centers of class j and k respectively. Intuitively, we are minimizing the cosine similarity between the class centers, which encourages the features of different classes to be more separable in the feature space. The $+1$ term is necessary to make the loss non-negative, since the cosine exists in $[-1, +1]$ range. The updating strategy is the SGD, equal to the one in Equation 2.16, but the updating value is recomputed in Equation 2.21.

$$\mathrm{d}\boldsymbol{c}_j^t = \frac{\sum_i^N \sum_j^K y_{ij} \left(\boldsymbol{c}_j^t - \boldsymbol{x}_i\right)}{1 + \sum_i^N \sum_j^K y_{ij} \cdot 1} + \frac{1}{1+K} \sum_{\boldsymbol{c}_j}^{K} \sum_{\boldsymbol{c}_k \neq \boldsymbol{c}_j}^{K} \left(\frac{\boldsymbol{c}_k \cdot \|\boldsymbol{c}_j\|_2^2 - \boldsymbol{c}_k \cdot \boldsymbol{c}_j^2}{\|\boldsymbol{c}_k\|_2 \cdot \|\boldsymbol{c}_j\|_2^3} \right) \tag{2.21}$$

where the second term is simply the gradient of the Island Loss with respect to the class center $\boldsymbol{c}_j$, normalized by the number of classes contributing to the gradient plus one to avoid deviding by zero.

Figure 2.8 shows the features learned by a CNN trained with CE loss (a), Center loss (b) and Island loss (c). Note how the center loss aggregates the features of the same expression class towards their centers, thus reducing intra-class variation with

respect to the CE loss (a).
The island loss (c) not only compresses the clusters individually but also pushes clusters apart [8].

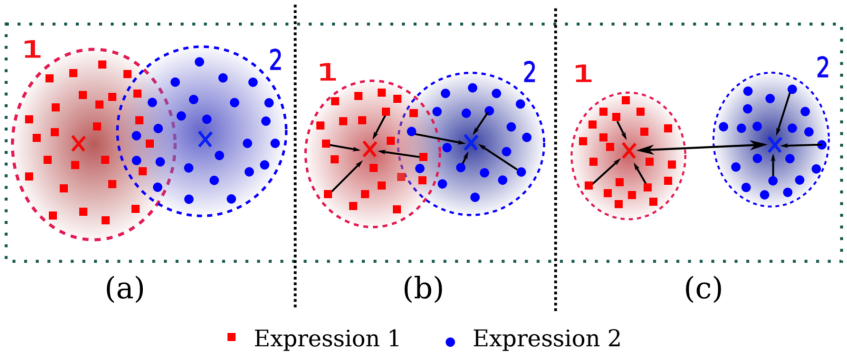


Figure 2.8: Features learned by a CNN trained with CE loss (a), Center loss (b), Island loss (c) [8]

2.2.5 Focal Loss

The Focal loss [48] solves the problem with CE loss which treats all samples equally, without considering the difficulty of recognizing certain samples with respect to others, due to different emotional intensities and visual appearance. Focal loss, defined in Equation 2.22 turns out to be useful to address the issue of class imbalance and hard-to-recognize samples.

$$\mathcal{L}_F = -\frac{1}{N}\sum_{i=1}^{N} \alpha_j \cdot (1 - p_{i,y_i})^{\gamma} \cdot log(p_{i,y_i}) \tag{2.22}$$

Intuitevely, this is the classic CE loss with the addition of the focal term $(1-p_{i,y_i})^{\gamma}$ which dynamically adjusts the loss based on the difficulty of recognizing the sample. It down-weights easy samples and thus focus training on hard ones[48]. For example, when an element of the vector $\boldsymbol{p_i}$ is close to 1, indicating high confidence in the correct class prediction, the modulation factor for that class will be approaching to 0 effectively downweighting the contribution of well-classified samples to the overall loss. $\boldsymbol{\gamma}$ is a tunable parameter used to adjust the balance between easy-to-recognize and hard-to-recognize samples. A higher γ value puts more emphasis on hard-to-recognize samples. α was originally a balancing parameter to address class imbalance by adjusting the weights for each class, but in [48], α is set to 1, indicating equal weighting for all classes.

2.3 Multimodal systems

In FER tasks, the use of multimodal systems, which combine information from multiple modalities (e.g., RGB images, depth maps, audio signals) can lead to improved performance by leveraging complementary information from different sources. Multimodal systems can capture a more comprehensive view of the input data, enhancing the model's ability to recognize and interpret facial expressions. After extracting the features from each modality, the multimodal fusion system combines the features to make a final prediction. In this section, we will discuss the fusion strategies, fusion levels, and training strategies for multimodal fusion systems in FER tasks.

The first simple idea could be **concatenating** the features from the different modalities and then feeding them to a fully connected layer to make the final prediction. However, this approach may not fully leverage the complementary information from the different modalities, as it treats the features independently and does not consider the relationships between them. To address this, more sophisticated fusion strategies based on attention mechanisms can be used to learn the interactions between different modalities. An attention-based fusion network can be more effective than simple concatenation because it allows the model to learn which features from each modality are more important for the task at hand.
We will see in SOTA Section 2.4 that many different attention mechanisms have been developed to fuse features in multimodal FER models. For example, many use a final transformer which applies the multihead self-attention mechanism to learn the inner relationships between the elements of concatenated features from different modalities. Despite we know that such self-attention mechanism is very effective and amenable for GPUs, it is known also that they require very large training dataset in order to capture all meaningful information. Due to the relatively small datasets in FER, some works resort to simpler attention mechanisms tailored for the problem, rather than transformers.

An important aspect to consider when designing a multimodal fusion system is the fusion level, which refers to the stage in the network architecture where the features from different modalities are combined. The fusion level can have a significant impact on the performance of the multimodal system, as it determines how the model integrates information from different sources.
In particular, the earlier layers in a CNN capture low-level features such as edges, textures, and simple patterns, while the later layers capture high-level semantic information such as object categories. The choice of fusion level depends on the task and the characteristics of the data. For example, for tasks requiring spatial context, such as object detection or segmentation, early fusion may be more appropriate, as it retains spatial information from the input images. In contrast, for tasks like classification, where high-level semantic information is more important, late fusion may be more suitable, as it captures the most abstract features from each modality. Note also that early fusions deal with high-dimensional features, which can increase computational complexity and memory requirements. In contrast, late fusions are typically lower-dimensional, reducing computational and memory requirements.

In practice, a tradeoff between early and late fusion is often used, where features from different modalities are fused at multiple levels of the network. This allows the model to leverage both low-level and high-level information from each modality, leading to improved performance.

Another aspect is the training strategy for multimodal fusion systems. There are two main approaches to training multimodal fusion systems: training each modality network separately and then combining them, or training the entire model end-to-end. In the first case, training each modality network separately can simplify the training process and allow for modularity. However, the modality networks doesn't learn to interact with each other during the feature extraction phase, which may limit the potential synergy between the modalities. In contrast, training the entire model end-to-end allows the networks to learn to interact with each other from the beginning, leading to better overall performance. However, this approach is more complex and requires more computational resources.

In practice, training the entire model end-to-end is generally recommended as it allows the model to learn joint representations and leverage complementary information from all modalities more effectively. Many FER works use cross modality attention mechanisms at different stages (early, mid and late) to learn the interactions between the features from different modalities, and train the entire model end-to-end to optimize the fusion process.

2.4 State of the Art

This section, explores some of the SOTA approaches for static FER. This will help us identify the most promising approaches for next Chapter.

Savchenko et al. [49][50] made an effort into building multi-task, CNN based models using architectures like MobileNet [51] and EfficientNet [39] and RexNet [52]. They demonstrated such lightweighted and fast backbones are enough to achieve SOTA performances in a multitude of tasks (face verification, face recognition and FER) providing a strong baseline for many applications. Such multitask networks are usually made of multiple streams (one or more per sub-task) and they compute a final loss is by combining the losses of the individual subtask (usually with a simple sum). In particular, in [50], they pre-train different lightweight CNNs over VGGFace2 [53] (for face recognition) and, in a second experiment, over ImageNet [38] datasets (for object recognition). The pretraining is useful to let the network produce features suitable to discriminate subjects from another. The backbone is appended with 4 stream networks (one per subtask) and finetuned over AffectNet. These 4 stream networks are simple FC layers for gender, age, ethnicity recognition subtasks, while the one for FER subtask also containes some additional convolutional layer. This is because FER is a different tasks than attribute recognitio, so a common approach is to extract earlier features from the backbone CNN and

fine-tune the convolutional layers for the downstream task (FER). They show that pretrained backbone over VGGFace2 leads to better accuracies (~+4%) in FER task over AffectNet with respect to the same backbone pretrained over ImageNet. This is, simply because VGGface2 is a dataset specifically designed for face analysis tasks, so the backbone CNN will learn to extract more relevant features for such tasks.

Another CNN based work is DDAMFN by Zhang et al. [54]. Similarly to Savchenko et al. they decide to use lightweight CNN backbone as a feature extractor which is a MobileFaceNet [55] enhanced for FER task. The common idea raises from the fact that using deeper architectures on small FER tasks could lead to overfitting. The first enhancement consists in applying multiple-size kernels that permit to capture more diverse features from input images, similarly to the Inception Block from GoogleNet [56]. More in depth, the mixed convolution block is composed of 3 parallel convolutional layers with different kernel sizes (3×3, 5×5,7×7) and the output of the block is the concatenation of the outputs of the 3 layers. The second enhancement is the Coordinate Block Attention Module (CBAM)[57] which is used to capture the long-range spatial dependencies between different regions of the feature maps. The CAM is composed of two branches to capture the vertical dependencies and the horizontal dependencies respectively. The two branches are then combined to produce the final attention map which is summed to initial feature map.
They are able to reach better accuracies, directly addressing the high intra-class variance and high inter-class similarity problems by feeding the extracted feature map into an additional attention module which is very similar to the CBAM because it contains 2 separate attention modules acting in vertical and horizontal directions over the input feature map. Finally, the feature maps from the two directions (horizontal and vertical) are Max-pooled to produce the final feature embedding which is fed into a FC layer for classification.

Ning et al. in 2024 [58] produced FMAE wich is a Masked Autoencoder (MAE) from [59] trained over Face9M dataset. Face9M is ad hoc created dataset containing 9M images from the unification of common facial datasets used for face analysis tasks (CelebA [60], FFHQ [61], VGGFace2 [53], CASIA-WebFace [62], UMDFaces [63], MegaFace [64], LAION-5B[65], EDFace-Celeb-1M [66]). As any autoencoder, MAE is composed of an encoder and a decoder: the encoder maps input image into a latent feature with smaller dimension; while the decoder reconstructs the input from the latent feature. Intuitevely, pre-training the encoder in this self-supervised setting allows for learning features that are relevant for reconstruction and so, embedding meaningful information for the classification task. After this pre-training, the decoder is discarded and the encoder is used as a feature extractor.
In Masked AEs, we encourage the encoder to learn useful features by applying random patches to the input image. In other words, by omitting some parts of the input image, the encoder is forced to learn a feature embedding that is invariant to the masked regions. This is particularly useful in computer vision in general, because, differently from language data, image data are not so information-dense and contain a lot of redundancy [59]. Note that this masking idea acts as a regularization (applied at pre-training time) and is very common in FER world to train models more

robust to occlusions.[58] Therefore, [58] pre-trains a Large ViT [44] over Face9M dataset with MAE approach and then use it for FER (and landmark detection) task. It is interesting to see that FMAE is the best performing model on AffectNet and RAF-DB only using this simple pre-training and without any complex network architecture or particular attention mechanism. Of course this comes at the high cost of pre-training a large model over a huge dataset.

Generally speaking, using a CNN backbone for feature extraction and a subsequent attention module to learn relationships from different areas in the extracted feature map is a very common strategy in FER. Recently many of these attention modules are becoming "transformer-based", because they show better ability in evaluating the global context information of the image than CNNs which are limited by the convolutional receptive field. This technique, efficiently overcomes the CNNs limitations in capturing long-range dependencies, but comes at the cost of longer training times and larger datasets since learning complex relationships in sequence-like data with transformers is expensive.

For example, in 2023, Mao et al., proposed POSTERV2 [67], a transformer-based model that achieves SOTA performances on RAF-DB and AffectNet. The model is able to address the intra-class variance and inter-class similarity problems making use of landmarks. It also provides robustness to scale sensitivity which is a problem for landmark-based models. In particular it is composed of two backbones: one extracting features from RGB images (a pre-trained IR-50 [68] that is fine-tuned) and one that extracts features from landmark information (a pre-trained MobileFaceNet [55] used frozen (non fine-tuned)). The two features extracted from the respective backbones are then fused with a transformer used in cross attention setup, which is a simple transformer that formulates queries using the input from the other modality (e.g., $Q_{RGB} = X_{landmark} \cdot W_{Q_{RGB}}$). This is done 3 times in sequence to extract respectively: low, mid and high level features; which are concatenated and fed into a final lightweight ViT [44] (only 2 layers) to learn relationships in the feature embedding and perform classification.

A very similar approach is used by Her et al. in 2024 [24]: they use the same basic architecture with the two stream approach (pre-trained IR-50 for RGB and pre-trained MobileFaceNe for landmarks) with cross attention modules and the final ViT; but they address the annotation ambiguity problem by applying Class Batch Normalization (CBN). According to them, even few "noisy" (badly annotated) samples can significantly degrade FER performance [24] leading to overfitting, so CBN is used. CBN is a variant of Batch Normalization that normalizes the features of each class separately, rather than the entire batch. In this way they assure that the final features are more class-specific, mitigating the noise from some sample in the batch that may be badly annotated, but also, addressing the high intra-class variance problem.

Another similar model is ARBEX [69] from Wasi et al. taht use again the dual stream network, cross attention modules and final ViT.

Xue at al. in TransFER[70] point out the fact that landmark based approaches, like POSTERV2, lack of robustness in wild datasets where landmarks may be not visible because of occlusions, insufficient light or even head pose (for example if the subjects is near profile pose). Another problem is the scaling: image could be too small to precisely detect landmarks. Finally, also subject's characteristics like age and ethnicity could be an obstacle to consistent landmark detection: too large variations in wrinkles, fat distribution, muscle tone can make it more difficult to identify key-points consistently across different age groups.
TransFER uses a stem CNN (pretrained IR-50) to extract feature from the image. This base feature is then processed by many spatial attention networks which apply 2 1×1 convolutional layers reducing the number of channels to 1. These masks are then Maxpooled, multiplied by original feature map and fed to a transformer encoder to learn relationships between them.
The innovation is in addressing overfitting through a dropout-like strategy: one random mask is dropped with probability p1 and also one random head of the transformer is dropped with probability p2, at each forward pass. This strategy is used to prevent the model from overfitting to specific regions of the face and to improve generalization ability.

Huang et al. uses two attention mechanisms in FERVT [71]: the first is a grid-wise attention mechanism that allows the model to extract features focusing on different areas of the face image. The second mechanism is a vision transfomer fed with these low-level features which captures long-range dependencies .

As another example of transformers on top of CNNs, Ma et al. produced VTFF (Visual Transformers with Feature Fusion) [37] which proposes a fusion strategy for RGB and LBP features (extracted from two backbone ResNet18 [72]) in a global-local attention mechanism that helps the model to be more robust to occlusion and pose variations. More in depth, the features extracted from the two backbones are initially multiplied by two learnable matrixes and then summed. This is done to let the network learn, initially, how much to account for each modality. The overall feature is then processed through a global attention stream (that reduces spatial dimension to 1×1 by Average pooling) and through a local attention stream (which instead retains spatial information). Finally, the attended global and local features are summed and later processed through a large transformer to learn inner relationships between feature components and perform classification. This is similar classification strategy to POSTERV2, TransFER and FERVT. It is interesting to note that, VTFF actually reaches lower accuracies than other state of the art methods, but the global-local attention express its power when, in original paper, they manually add occlusions to the images demostrating that VTFF is more robust to occlusions than other models.

Li et al. introduced FER-former [73] which is a transformer based model for FER in the wild. Again they try to combine features provided by CNNs with transformers in an hybrid model. The innovation in this work is that, inspired by OpenAI CLIP model [74], they address annotation ambiguity by using soft labels instead of the hard labels which are believed to be noisy. This is particularly true for FER datasets where annotation ambiguity is usually addressed with a majority voting system, which is very expensive (many expert annotators must be involved).

FER-former comprises a stem feature extractor (they try both ViT and the same pre-trained IR-50 used in POSTERV2 and transFER). These features are then linearly projected and fed to a transformer encoder. In a parallel stream, the soft labels are fed into the pretrained Text encoder from CLIP (which is a transformer trained on image-text pairs from the internet) to extract context aware features. With "Soft labels" we mean labels that embed the original one-hot label but are also able to give context information to the text encoder (e.g., they try formulations like "This is a face image of expression" or passive sentences like "a/an expression expression is shown in the image").
Later the cosine similarity between the image feature I and the text feature T is computed and used as loss. In this way, the image feature extractor is trained to produce image features that are close to the text features produced from the text encoder, thereby easing the issue of annotation ambiguity.
At test time, the text encoder is discarded and the image feature extractor is used as a standard FER model.

Similarly to VTTF[37], which leverages RGB and LBP modalities, many other works use multiple modalities to improve FER performance. In particular, 2D+3D FER is a common approach that combines RGB images with depth maps or 3D scans (Point Clouds or Meshes) to capture both texture and shape information of the face. This 2D+3D FER task permits to leverage the RGB stream to get discriminant features about the texture of the face and make also use of 3D scans to get features embedding shape information. By fusing the two modalities we can address the limitations of 2D images, such as sensitivity to illumination and pose variations, and capture subtle facial movements and muscle deformations that may not be discernible in 2D images alone.

Note that transformer based models are much more rare for 2D+3D FER because, as already mentioned, transformers generally need more data and there are no large-scale 2D+3D FER datasets due to the high collection costs. This is why most of the works in this field are based on CNNs. To overcome this, some attempts to use lightweighted vision transformers has been made by Li et al. in MFEVIT[75] in 2021. The same authors later abandoned transformers (probably because prone to overfitting considering the very small sized datasets in 2D+3D FER) in AFNET [76], CMANET [77] and FFNET [78]. They opted instead for ad-hoc developed cross-modality attention fusion networks which are less "data-hungry" and require less training than transformer.
A common idea used in these works is the application of Attention Masks: these are images used as prior knowledge to enhance the CNNs feature extraction. Basically, attention masks help the network in learning to extract features by focusing on the salient regions indicated by the masks[76] (e.g., eyes, nose, and mouth). Later, these features are fused with additional attention fusion mechanisms which let the network learn how much to account each modality for. It is interesting to see, in the ablation studies, that they get a very small decrease when they drop the depth modality. This is probably because they use only BU3DFE and Bosphorus as benchmark datasets, which are lab acquired datasets not containing much variation in occlusions, pose and illumination, therefore it is reasonable that the network learns to give much more weight to the RGB modality which is more discriminant in these conditions.

A different approach to solve 2D+3D FER is proposed by Ni et al. in CMFN [79]. Exactly as in above cited works using Bosphorus and BU3DFE datasets, the depth maps are extracted from the 3D meshes, but they convert RGB images into gray scale images and then extract LBP images. Even if not clearified in the paper, this is probably because LBP features are hand-crafted to be representative of texture information and less affected by illumination changes than RGB. So RGB images are grayscaled to avoid redundant information and let the network focus on the LBP stream.
They extract features from the three modalities (RGB, depth maps and LBP) with specifically developed lightweight CNNs and feed them into a cross-modal fusion network that combines them through attention mechanism similar to the AFNET, CMANET and FFNET.

These attention based fusion networks used in 2D+3D FER perform better than simple concatenation or sum fusion because, each modality may inherently emphasize different aspects of the facial expression and it is important to let the network learn how much to account each modality for. As mentioned, 2D images capture surface texture and color variations in the face, while depth maps are better at representing subtle facial geometry. So, it is crucial to consider the complementarity of each modality when combining them, such that the model can compensate for weaknesses in one modality with strengths from another. Moreover it reduces the risk for one modality to dominate the fused feature overshadowing the valuable information provided by the other modality.

Whether multimodal or "full RGB", most of the above cited works address the problem of high intra-class variance and high inter-class similarity trough the use of attention mechanisms (specifically developed or transformer based) and use a final standard classification mechanism with softmax and CE loss. Another approach is to make use of loss functions that enhance the discriminative power of the features. For example, Wen et al. in [45] implement center loss to extract well separated and compact features for a face recognition task over the LFW [80] and YTF [81] datasets. CMFN [79] addresses class unmbalance problem using a focal loss.
Also ARBEX [69] uses complex loss function summing the CE loss, Central Loss and an anchor based loss. Lin et al. in [82] use an orthogonalization loss.

2.5 Summary

Making comparisons between different models is not straightforward as each study, even when using the same dataset, may use slightly different testing protocol (e.g., different training and test split sizes or selection method). Taking this into account, Table 2.1 summarizes all the above cited works ordered by dataset and year for easier comparison. The Overall Accuracy and Average Class Accuracy (if reported in original paper) are both reported in order to give a more comprehensive view of model performance for imbalanced datasets, very common in FER.

$$\text{Ov\_Acc} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.23}$$

$$\text{Avg\_Acc} = \frac{1}{K} \sum_{j=1}^{K} Ov\_Acc_j \tag{2.24}$$

In some cases, some classes are discarded, so Table 2.1 specifies also the results for different number of classes: 8 classes (*neutral*, *sad*, *surprise*, *fear*, *disgust*, *anger*), 7 classes (where *contempt* is discarded) or 6 classes (where also *neutral* is discarded). Finally, models for FER2013 dataset are not reported an its extension FER+ is considered. Also dynamic datasets works are not reported, except for FERVT and VTFF which select frames from CK+ videos.

Some useful observations from Related Works section before going to the development of our model in next Chapter.

- *FER is far from solved*: current SOTA solutions, even if presenting similarities, seem to not converge to a unique general solution for FER, but, rather, to a set of solutions specifically developed for different datasets characteristics. However, we know that more data leads to better generalization, so, despite all the efforts in building robust and efficient architectures, we could say that FER will be solved when enough data will be available. For example, the ViT pretrained in a self-supervised manner over 9 milions images (Face9M) dataset in FMAE [58] is able to reach SOTA performances without any particular architecture.

- *More modalities are better*. For instance, 2D+3D FER models perform better than 2D models on the same dataset. This is because 3D scans are not affected by variations in light and are able to capture shape information and subtle facial movements that are not discernible in 2D images alone. 2D+3D FER is becoming less and less expensive as 3D scanners and depth cameras are now affordable and widely deployed in IoT technologies. So it is reasonable to see multimodal models not as some "niche" research but rather as the future of FER.

- *Deal with high intra-class variance high inter-class similarity*. They can be addressed with attention mechanisms or robust loss functions. Attention mechanisms (ad-hoc developed or transformer based) are usually applied right after the CNN backbone to capture long-range dependencies between different regions of the face. In multimodal systems, *cross-modality* attention mechanisms are used to learn the interactions between the features from different modalities (e.g., POSTERV2 [67]).

- *Deal with class imbalance*: it is usually addressed with weighted CE loss. Also [77] uses a focal loss, which is able to focus training on hard samples by down-weighting easy samples.

- *Deal with Annotation ambiguity*: it is usually addressed by annotating datasets with a majority voting system, which is very expensive (many expert annotators must be involved). In [73] they use text supervision to reduce annotation ambiguity. In [24] they use Class Batch Attention to avoid overfitting to noisy samples.

Table 2.1: FER Models

Dataset	Model	Year	Accuracy (%)		
			#classes	Overall	Avg.
CK+	FERVT [71]	2021	7	100	100
			8	99.46	96.87
	VTFF[a] [37]	2021	7	86.24	-
AffectNet	POSTERV2 [67]	2023	7	67.49	67.45
			8	63.77	63.76
	TransFER [70]	2021	7	66.23	-
	VTFF [37]	2021	7	61.85	64.8
	DDAMFN [54]	2023	7	67.03	-
			8	64.25	-
	FMAE [58]	2024	8	65.00	-
	MobileNet-V1[b] [49]	2022	7	64.71	-
			8	60.20	60.95
	EfficientNet-B0[b] [49]	2022	7	65.74	-
			8	61.32	61.32
	EfficientNet-B2[b] [49]	2022	7	66.34	-
			8	63.03	63.02
	RexNet150 [b] [50]	2021	7	65.54	-
RAF-DB	FERVT [71]	2021	7	88.26	80.63
	POSTERV2 [67]	2023	7	92.21	85.97
	FER-former [73]	2023	7	91.30	85.42
	TransFER [70]	2021	7	90.91	-
	ARBEX [69]	2023	7	92.47	-
	VTFF [37]	2021	7	88.14	81.86
	DDAMFN [54]	2023	7	91.35	-
	FMAE [58]	2024	7	93.09	-
FER+	FER-former [73]	2023	8	90.96	60.40
	FERVT [71]	2021	8	90.04	77.20
	TransFER [70]	2021	7	90.83	-
	ARBEX [69]	2023	7	93.09	-
	VTFF [37]	2021	7	88.81	-
	DDAMFN [54]	2023	7	90.74	-
Bosphorus	AFNET [76]	2023	6	-	88.31
	CMANET [77]	2022	6	-	89.36
	FFNET [78]	2021	6	-	87.65
	CMFN [79]	2023	6	-	85.16
	OGFNET [82]	2021	6	-	89.28
	MFEVIT[75]	2021	6	-	90.28
BU3DFE	AFNET [76]	2023	6	-	90.08
	CMANET [77]	2022	6	-	90.24
	FFNET [78]	2021	6	-	89.82
	CMFN [79]	2023	6	-	88.91
	OGFNET [82]	2021	6	-	89.05
	MFEVIT[75]	2021	6	-	90.83

[a] Pretrained on Affectnet and fine-tuned on CK+

[b] Multitask configuration from Savchenko et al.. Indicated network is the Backbone CNN pretrained on VGGFace2.

Chapter 3

Materials and Methods

This Chapter leverages related works to analyze the CalD3rMenD3s dataset, explaining the preprocessing steps, model architecture and the training process. All the code used in this work is publicly available at `https://github.com/erAI-17/Facial_Expression_Recognition_2024`. One "library-like" repository for Focal, Central and Island loss is available at `https://github.com/erAI-17/Pytorch-Focal-Center-Island-Loss` since no Pytorch official implementations are currently available.

3.1 Dataset

CalD3rMenD3s CalD3r and MenD3s are merged into a single dataset containing a total of 8716 RGBD samples in 224×224 resolution. Since we have only few frontal view images with relatevely low resolution, reconstructing a detailed 3D mesh would lead to poor results as shown in Figure 1.5. This work will therefore focus on a two-stream architecture that leverages the complementary information from RGB and Depth streams. In Figure 3.1 we can see the class distribution is very unbalanced with the majority of samples belonging to the neutral class.

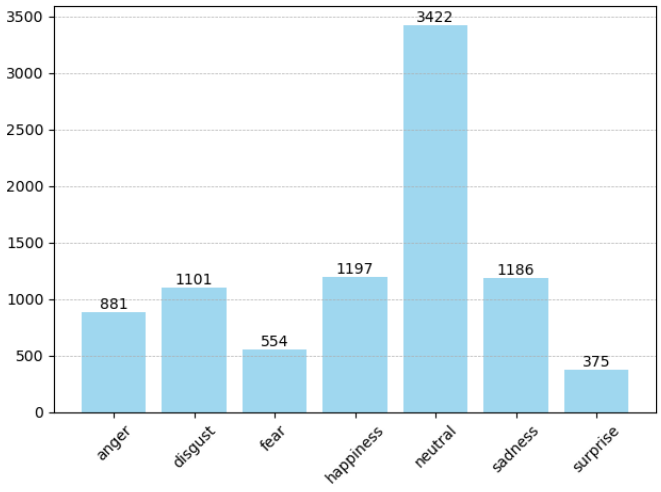


Figure 3.1: Classes distribution of the CalD3r and MenD3s dataset.

Each emotion is elicited using a set of images from psychological research dataset, therefore there is a common framework to effectively annotate images and we expect not to have a large annotation ambiguity. However, none of those images was meant to elicit surprise expression. This is why the dataset contains very few samples annotated as surprise class and, in the model presented along with the dataset in [5], the surprise class has been removed. This introduces some additional difficulty in the classification task because of the few samples, but also because different annotators may have different opinions on what surprise looks like. Figure 3.3 shows a male and a female subject in the seven classes for the two modalities.

BU3DFE Differently from CalD3rMenD3s, BU3DFE comes in RGB and 3D Mesh modalities. Four intensity levels for 7 classes are portrayed by 100 sybjects. Therefore, the total number of samples is 2500 and, as shown in Figure 3.2, it is a balanced dataset but for Neutral class which consists in just 1 intensity level.

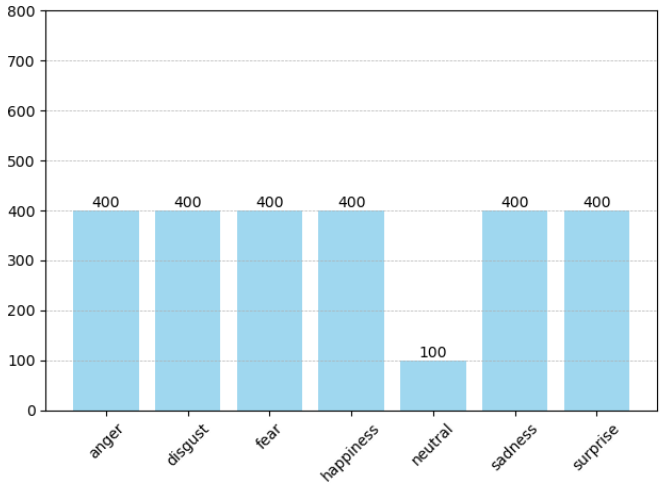


Figure 3.2: Classes distribution of the BU3DFE dataset.

In order to be consistent with the model for CalD3rMenD3s, 3D meshes are converted into depth maps by rendering each mesh in 3D space and taking the depth value. Reconstructiong depth maps that are spatially consistent with corresponding RGB images is challenging because [9] gives no info about the camera device used to capture the RGB image (in particular FOV is not defined) but only the resolution (512×512). Figure 3.4 shows an example of reconstruction where we can see the RGB image (a), the corresponding 3D mesh (b) and the depthmap extracted from the mesh (c).

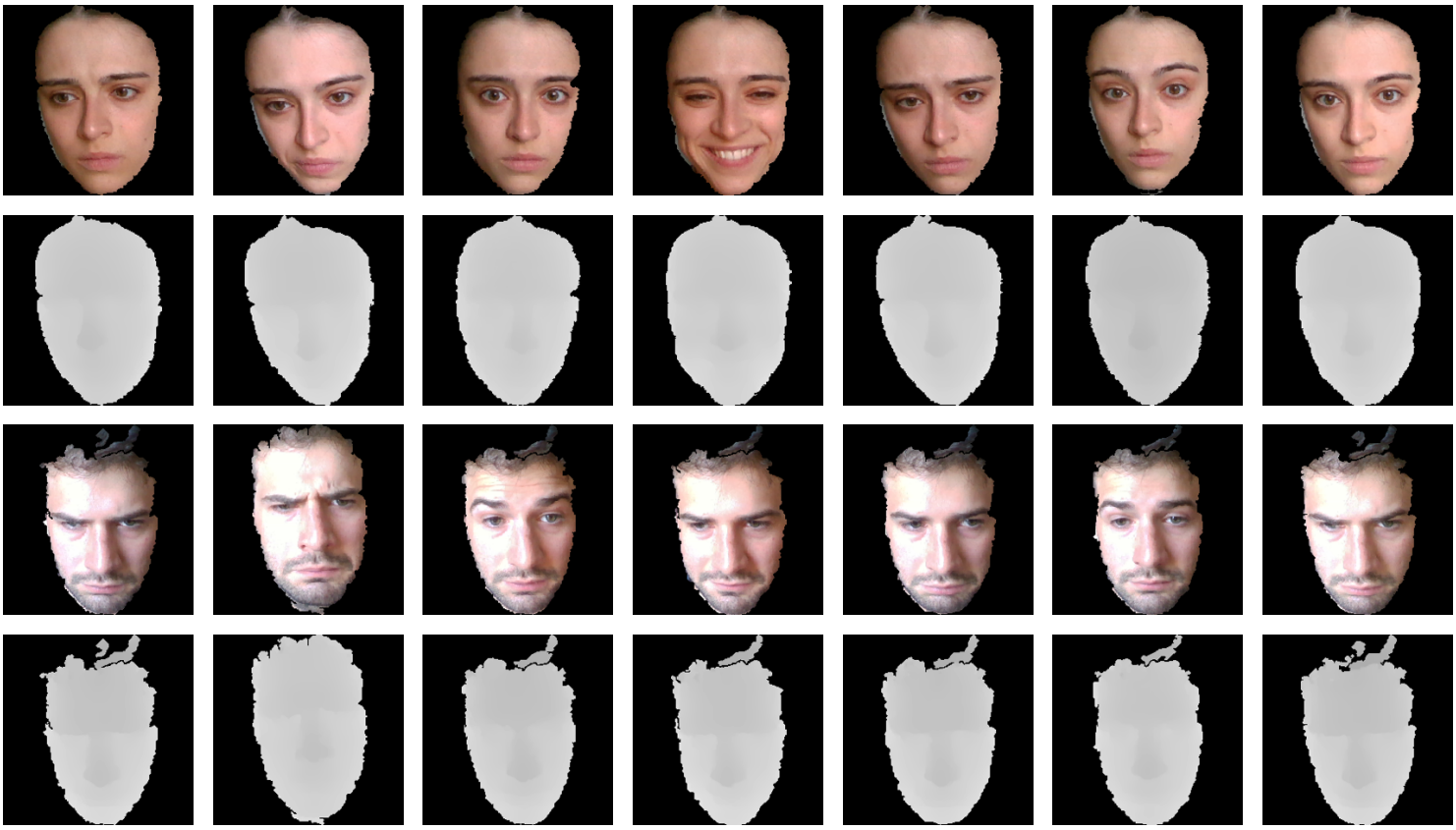
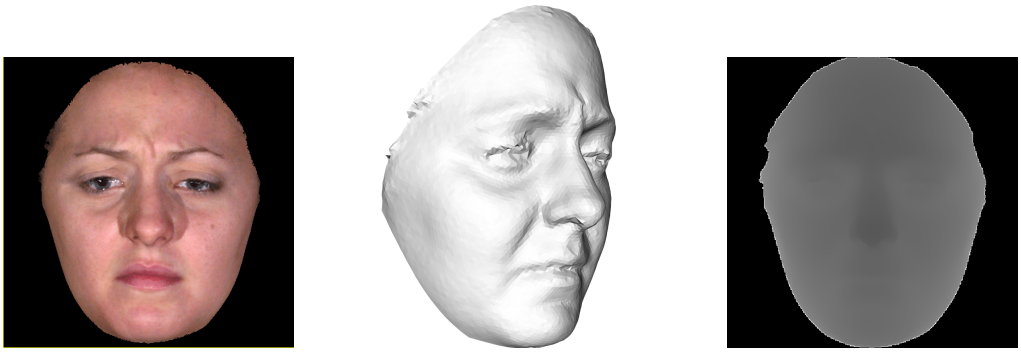


Figure 3.3: Female and Male examples from CalD3rMenD3s. First row RGB, Second row depthmap. Columns order: anger, disgust, fear, happiness, sadness, surprise, neutral [5]



(a) **(b)** **(c)**

Figure 3.4: (a) RGB, (b) 3D Mesh, (c) extracted Depth Map from BU3DFE [9]

3.2 Preprocessing and data augmentation

Both datasets' faces have been already cropped to remove background. Depth images, consisting of 1 channel, are stacked to reach default network input used for RGB stream. To leverage EfficientNetB2 pre-trained model from [49], the input images

are resized to 260×260 with linear interpolation.

We said that head pose might convey expression related information (e.g. head down may indicate sadness), but to detect such patterns we probably need much larger dataset. So, images are aligned using Google's landmark detector MediaPipe [83] which is a CNN trained on faces manually annotated with landmarks. We select the landmarks corresponding to the corners of the eyes, compute the coordinates of the centers of the eyes and the angle between them. Then faces are aligned, by simply rotating images by that angle, as shown in Figure 3.5. These is done only for CalD3rMenD3s dataset since BU3DFE comes already aligned.

(a) **(b)**

Figure 3.5: Face alignment: (a) Original, (b) Aligned

We rescale the pixels values into [0,1] range and z-score normalize as shown in equation 3.1. Common practice would require to use the mean and standard deviation of the pretraining set (VGGFace2), but that would not guarantee zero centered data for our training set which is very different from VGGFace2, especially for the depth modality. Therefore we compute at each validation fold, the mean and standard deviation of the training set, separately for the two modalities, and use them for normalization.

$$\boldsymbol{X}_{norm} = \frac{\boldsymbol{X}/max(\boldsymbol{X})) - \boldsymbol{\mu}}{\boldsymbol{\sigma}} \tag{3.1}$$

Data augmentation will help the model to learn robust patterns to variations in lighting, angle, occlusions, etc.. Furthermore data augmentation could help reducing inter-class similarity as it provides more diverse examples for each class. On the other hand, introducing overly complex data augmentation could introduce unrealistic noise. Online augmentation is chosen as it provides a continuous stream of diverse variations during training process without requiring additional storage. A limited set of realistic augmentations, are selected: 0°-10° rotations, horizontal flips, color jittering (only for RGB images) which simulates variations in lightning and random erasing which simulates occlusion. Data augmentation is not applied on validation samples as they should represent real world conditions to provide an accurate measure of model performance.

It is worth noting that preprocessing comprising face landmark extraction, alignment and data augmentations, are very fast and done online during training.

3.3 Model Architecture

Let I_{rgb} and I_{depth} be the RGB and depth images respectively. Resizing and channel stacking for depth images are applied such that $I_{rgb}, I_{depth} \in \mathbb{R}^{H=260 \times W=260 \times C=3}$.

I_{rgb} and I_{depth} are processed in a two stream network in Figure 3.8, composed of two identical feature extractors (EfficientNetB2) pretrained for face verification on VGGFace2 [50]. The architecture of the EfficientNetB2 backbone is shown in Table 3.1, where *Reps* is the number of repetitions for a block. Note that kernel size k and stride s, in Inverted Residual Block (IRB) are referred to the *Depthwise-Conv*, since the *Pointwise-Conv* is always 1×1 with stride 1. Features are extracted after the last IRB leading to $X_{rgb}, X_{depth} \in \mathbb{R}^{H=9 \times W=9 \times C=352}$.

Each Inverted residual block (IRB) contains a Squeeze-Excite module which permits to select informative channels and suppress less discriminant ones. The same concept can be applied on the spatial information using spatial attention mechanisms. Coordinate Block Attention Module [57] creates the spatial attention map $C \times 1 \times 1$ by concatenationg the MaxPooling and AvgPooling of the feature and then applying convolution and sigmoid activation, as shown in Figure 3.6(a). Instead we use the LANet [84] approach that uses additional learnable convolution rather than pooling, as shown in Figure 3.6(b). We apply $S = 4$ of these spatial Attention Modules and MaxPool their results, like in TransFER [70], but instead of using drop-out of random attention masks, we only use batchnorm, believing it provides similar regularization effect but without hyperparametrs (no dropout probability).

Moreover, we use this spatial attention in a cross modality setup: the masks M_{rgb} and M_{depth} are computed separately for the two modalities, averaged and multiplied by X_{rgb}. The idea is to select the most informative spatial regions from the RGB and Depth stream and average them such that high activations in the RGB image, which could be due to high local illumination, can be suppressed by low activations in the depth image. Viceversa, high activations in the depth image, which could be caused by occlusions, can be suppressed by low activations in the RGB image. In this way the Depth modality is used to guide the learning of the RGB stream.

The feature X_{out} is passed through a 1×1 convolutional layer and reshaped to create a sequence like input for a small Vision Transformer Encoder (only 4 layers). This setup is the same from POSTERV2 [67]. Finally, two fully connected layers (MLP), followed by a softmax layer are used to classify the output.

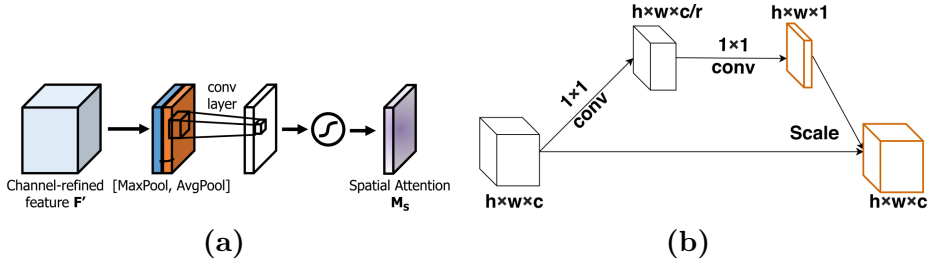


Figure 3.6: Spatial Attention mechanisms: (a) CBAM [57], (b) LANet [84]

Table 3.1: EfficientNetB2 Architecture

Layer	C_{in}	C_{out}	k	stride	Reps
Conv2D (stem) [a]	3	32	3×3	2	1
Conv2D (stem)[a]	32	16	3×3	2	1
IRB [b]	16	24	3×3	2	1
IRB [b]	24	24	3×3	1	2
IRB [b]	24	48	5×5	2	1
IRB [b]	48	48	5×5	1	2
IRB [b]	48	88	3×3	2	1
IRB [b]	88	88	3×3	1	2
IRB [b]	88	120	5×5	1	1
IRB [b]	120	120	5×5	1	3
IRB [b]	120	208	5×5	2	1
IRB [b]	208	208	5×5	1	4
IRB [b]	208	352	3×3	1	2
IRB [b]	352	352	3×3	1	1
Conv2D[a]	352	1408	1×1	1	1
MaxPool	1408	1408	-	-	1
FC	1408	7	-	-	1

[a] Stem Depthwise Separable block, with batchnorm2D and SiLU

[b] Inverted Residual Block. Residual connection is present only if $C_{in} = C_{out}$

3.4 Training

Because we use pretrained backbones, the initial weights are biased towards a very different dataset from ours, especially for depth modality. For this reason, common practice suggests to adopt a "warm-up" strategy where the learning rate is initially set to a very low value and then increased to the desired value. This allows the model to slowly adapt to the new dataset without abrupt oscillations caused by large gradients.

A popular learning rate schedule that adopts this strategy is the One-Cycle schedule [85] where the learning rate starts very low, increases up to a maximum value and then decreases following a cosine annealing schedule. In Table 3.2 we show the learning rate setup for the two backbones and the "fusion network" (comprising the Cross Modality Spatial Attention, the transformer encoder and the MLP) all using One-Cycle schedule. Note that the maximum value is higher for the fusion network because it is trained from scratch, while the backbones are pretrained and only need to be fine-tuned. The learning rate schedule for the fusion network is shown in Figure 3.7.

Table 3.2: Learning rate setup

	Max LR	Min LR	Schedule
Backbones	1^{-4}	10^{-6}	One-cycle
Fusion Network	1^{-3}	10^{-6}	One-cycle

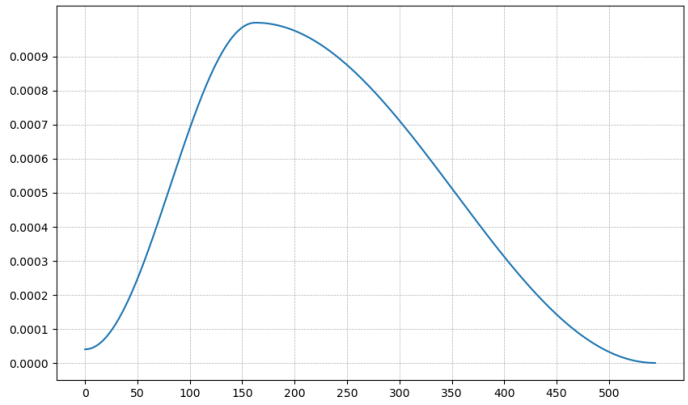


Figure 3.7: Onecycle learning rate schedule for the fusion network

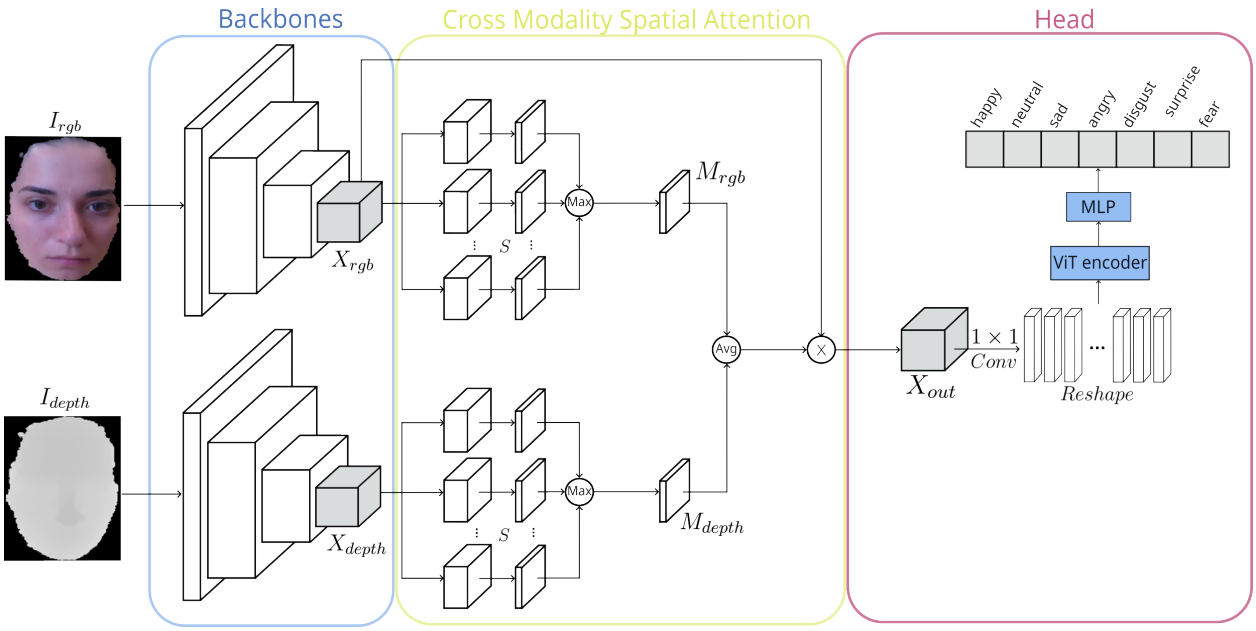


Figure 3.8: Model architecture

AdamW optimizer is used. Island Loss is set up with $\lambda_1 = 1^{-2}$ and $\lambda_2 = 10$, with centers learnt using an SGD optimizer with fixed learning rate $\alpha = 0.5$.

To make the network faster, Automatic Mixed Precision (AMP) is used to make the forward pass in half precision (16bits) while keeping the weights in single precision (32bits). This allows to reduce the memory usage and the computational time.

Batch size is set to 128 with gradient accumulation to make up for memory constraints. Using relatively small batches, allows to introduce a regularization effect

in the training due to the noise in the gradients computed on smaller batches with respect to larger ones used in other works (e.g. 1024 in [74] or bigger). These values of batch size, learning rate, weight decay, lambdas and S are hyperparameters tuned through a random search strategy (with Optuna framework [86]) to find the best combination of hyperparameters.

K-fold cross validation is used to evaluate the model over CalD3RMenD3s with 5 folds. In this way, at each fold, the model is trained over the 80% of dataset and validated on the remaining 20% to have a more reliable accuracy estimate of the model considering low sized FER datasets. For BU3DFE, the same strategy from [77] [76] (and others) is used: randomly select 60% of dataset for training and 40% for validating and then averaging the accuracies. The model is trained for 10 epochs on each fold and validated every 10% of an epoch. We use Tesla T4 GPU with 12GB memory taking approximately 1,15 hours per fold on CalD3rMenD3s.

Figure 3.9 show the evolution of the loss, training and validation accuracy over the 5 different folds on CalD3rMenD3s in the 6 classes setup. We can see consistent results between different folds ensuring robustness of this validation strategy. Note that, since we use gradient accumulation, the number of iterations is shown on the x-axis. The translation into epochs is:

$$\text{Epochs} = \frac{\text{Iterations}}{\text{training set size/Batch size}} \tag{3.2}$$

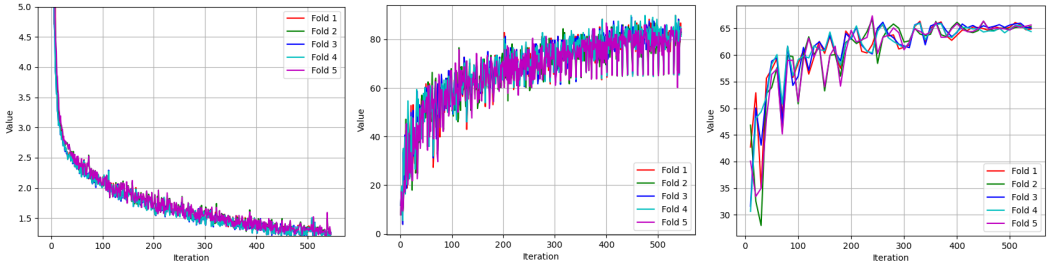
As an example, for CalD3rMenD3s, at each fold, the training set size is $(8716/5)\times 4$ and batch size is 128 which means that 54,4 iterations correspond to 1 epoch.

We can see that the loss and the validation accuracy are respectively decreasing and increasing consistently over the training, which is a good sign that the model is learning the task without overfitting. However, an increasing oscillating behavior on the training accuracy starts after 5-6 epochs setting a lower bound to $\sim 62\%$. This is a sign of overfitting on the training set: the model is not able to generalize well from a batch to another. This oscillation behaviour translates into a a plateau in the validation accuracy probably because the validation set is much bigger than the batch size and this oscillation is washed-out. This is a proof that the model is learning very fast, probably due to the pre-trained backbones, and we could actually apply an early stop after 6 or 7 epochs.

3.5 Results and Comparisons

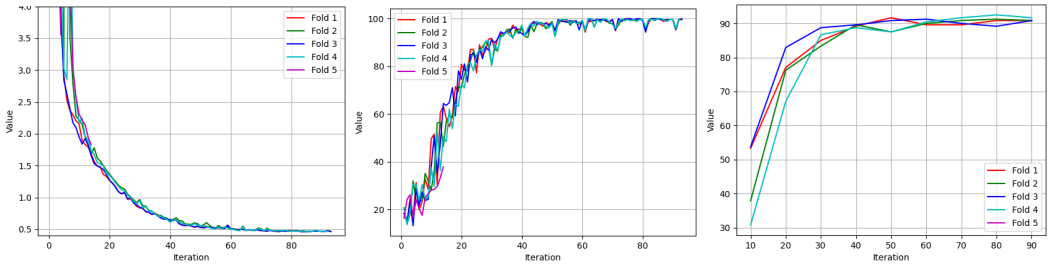
Accuracies and confusion matrices are reported in Tables 3.3 and 3.4 and Figure 3.10. They consist in the averaged values over all the folds.

The only known model available for comparison on CalD3rMenD3s is the one presented along with the dataset in [5] which is a multimodal network comprising a MobilNetV3 [41] for the RGB stream and a lightweight 3D CNN over a 3D voxel representation reconstructed from the depth map. However, the authors validated the model over 6 classes, excluding *surprise* class because it is not represented enough. Unfortunately, also every work on the BU3DFE dataset considers only 6 classes discarding the neutral one probably because less represented. Therefore, in this



(a) Training Loss **(b)** Training Accuracy **(c)** Validation Accuracy

(d) Training on CalD3RMenD3s 6 classes



(e) Training Loss **(f)** Training Accuracy **(g)** Validation Accuracy

(h) Training on BU3DFE 6 classes(only 2 highest intensity levels)

Figure 3.9: Training and Validation Statistics for CalD3RMenD3s, BU3DFE 6 classes

work, for both datasets, the setups with 6 and 7 classes are both evaluated for easier comparison with other works.

Note that all the known models validated on BU3DFE are based on a two stream RGB (or grayscale) and Depth architecture with different fusion strategies (none of them is using the mesh directly).

Table 3.3: Results on CalD3RMenD3s

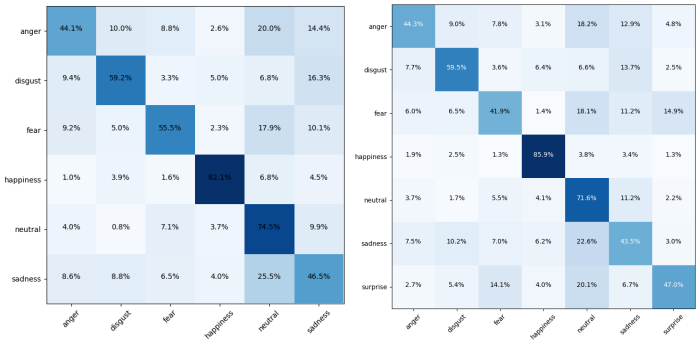
Model	Year	Accuracy (%)	Classes
CalD3RMenD3s [5]	2023	58,30	6[a]
This	2024	65,11	6[a]
		62,50	7

[a] w/o Surprise class

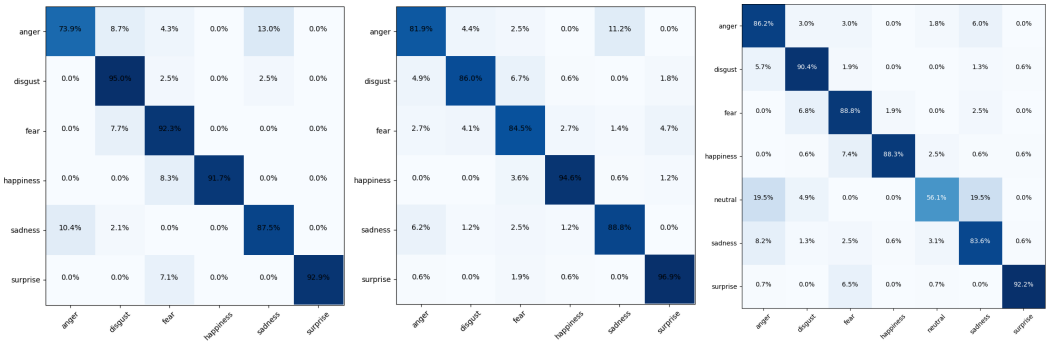
Table 3.4: Results on BU3DFE

Model	Year	Accuracy (%)	Classes
MFEVIT [75]	2021	90,83	6^{ab}
CMANET [77]	2022	90,24	6^{ab}
CMFN [79]	2023	88,91	6^{ab}
AFNET [76]	2023	90,08	6^{ab}
This	2024	90,83	6^{ab}
		88,78	6^{a}
		87,47	7

[a] w/o Neutral class
[b] only 2 highest intensity levels



(a) CalD3RMenD3s 6 classes **(b)** CalD3RMenD3s 7 classes



(c) BU3DFE 6 classes only 2 highest intensity **(d)** BU3DFE 6 classes **(e)** BU3DFE 7 classes

Figure 3.10: Confusion Matrices. X-axis are the predicted class, Y-axis is the true class.

Considering the very fast network and that we resize images to half resolution (260×260) with respect to other works, we can say that the accuracy is satisfying. By comparing results between the two datasets we can confirm what we said in datasets section: FER over posed datasets is much simpler task than over spontaneous dataset. In fact, in Figure 3.11, we can see that the features extracted by the model on the BU3DFE dataset are very well separated since the posed expressions are more different from each other with respect to the spontaneous ones in CalD3rMenD3s

analysed in Figure **??**. In general we can see from confusion matrix that there are many neutral samples confused with other classes. For BU3DFE it may be because of the low number of neutral samples (only 100 samples), while for CalD3RMenD3s it may be because spontaneous expression may be slightly activated and the model is not able to distinguish between them. Also the couple Surprise-Fear and Sadness-Anger are difficult to distinguish, probably because they share similar facial features like open mouth and wide eyes for the first couple and closed mouth and frowning eyes for the second couple.

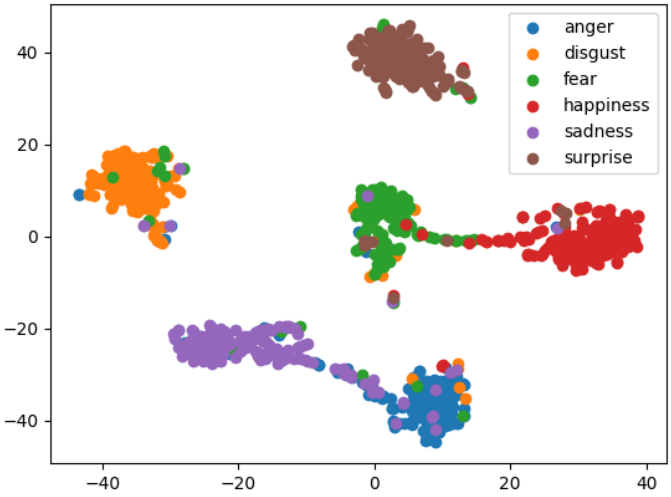
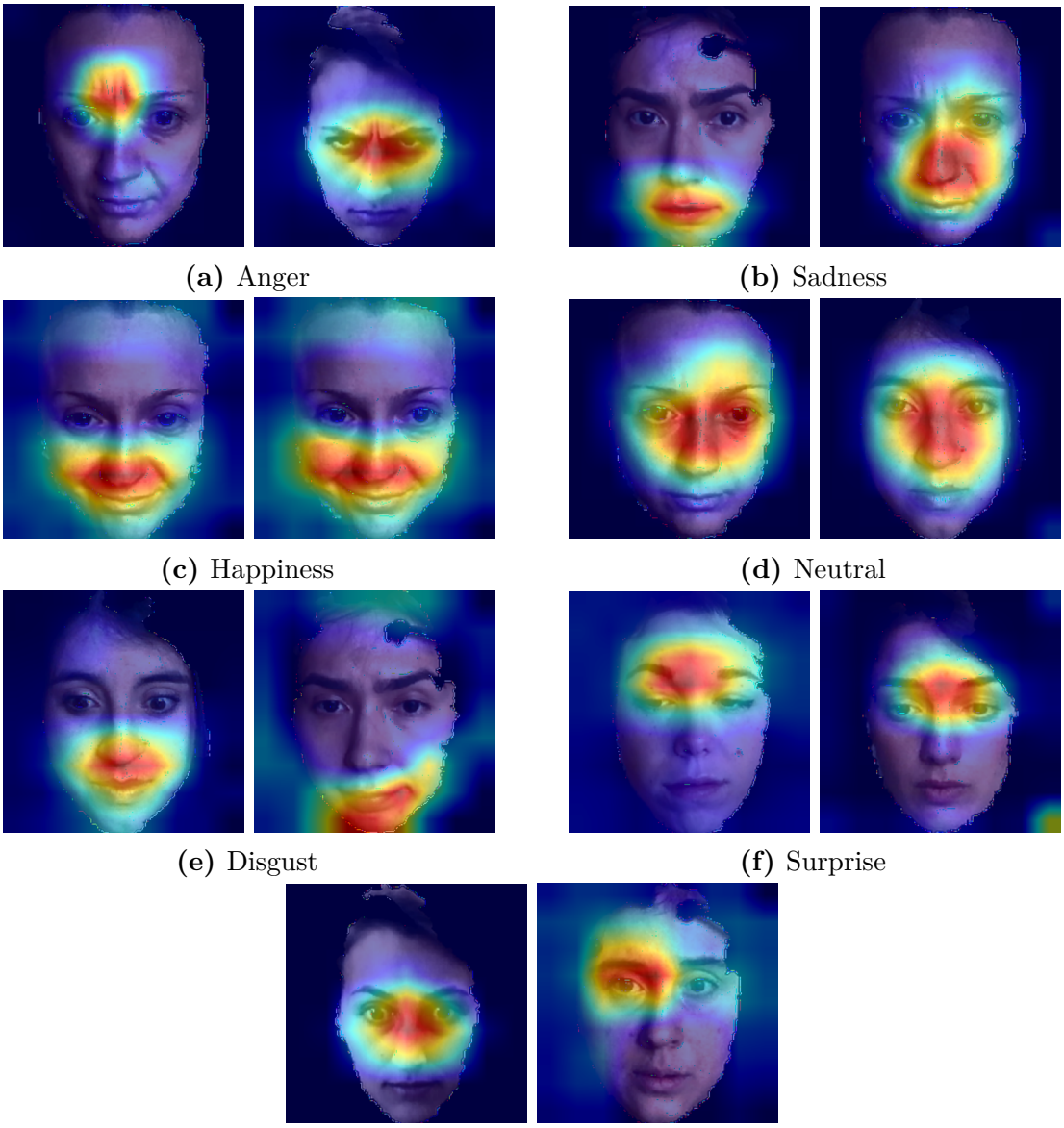


Figure 3.11: BU3DFE 6 classes features

3.6 Attention analysis

GRADCAM [87] is used to visualize where the model is focusing on the input image to make the prediction. Gradcam computes the score of the target class at the output of a convolutional layer and then performs backpropagation using that score as a loss value. We show the attention maps for 2 correctly classified validation samples per each class in Figure 3.12 from the CalD3RMenD3s 7 classes experiment. Wether no real pattern can be detected for the Neutral class, it is clear that regions of the face that are important for the classification are the eyebrows for Anger and Surprise, the mouth for Happiness and the eyes for Fear.



(a) Anger **(b)** Sadness

(c) Happiness **(d)** Neutral

(e) Disgust **(f)** Surprise

(g) Fear

Figure 3.12: Heatmaps for each emotion class

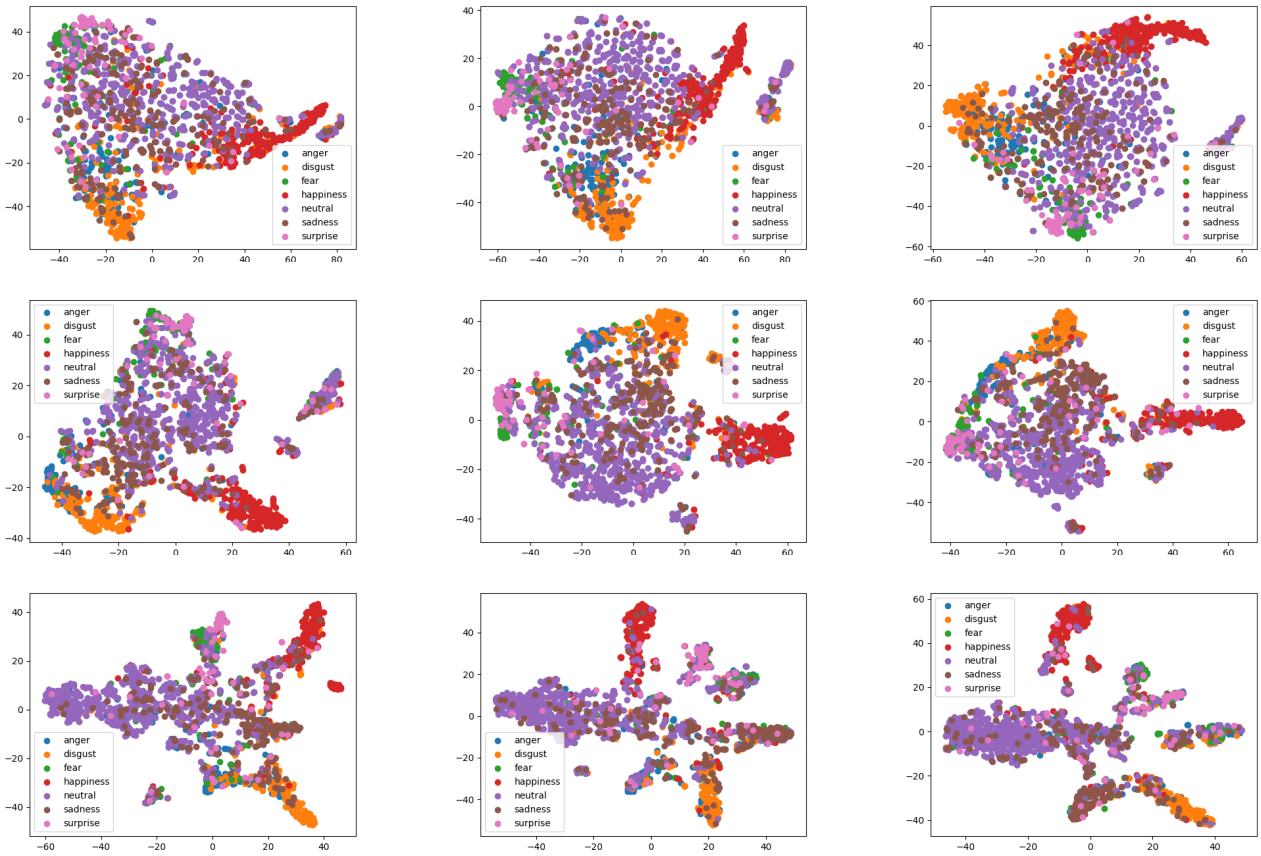
3.7 Loss function analysis

This section shows the discriminative power of the features at the exit of the fusion network, when trained using different loss functions. The network is trained for 10 epochs using CE Loss, Central Loss and Island Loss over CalD3rMenD3s in the 7 classes setup. Figure 3.13 shows the features of the validation set, plot in 2D space using the t-SNE dimensionality reduction. We can see that the features produced by the network trained with Center Loss are more clustered around each class center compared to the CE Loss, especially for the Neutral class which is very spread and more difficult to distinguish. Note that the results from confusion matrices are confirmed by the overlapping between classes Surprise-Fear. The Island Loss seems

to provide a good clustering of the classes with better separation, even if it still struggles in the overlapping couples. The final accuracy of the model reported in Table 3.5 for the three cases, confirms this observation, with the Island Loss providing the best performance, followed by the Center Loss and the CE Loss.

Table 3.5: Loss experiments on CalD3rMenD3s 7 classes

Loss	Accuracy (%)
CE	59,81
Central	61,73
Island	62,50



2 epochs 6 epochs 10 epochs

Figure 3.13: t-SNE plot features of validation samples for Cross Entropy (first row), Central (second row) and Island (third row) loss functions after 2, 6 and 10 epochs training.

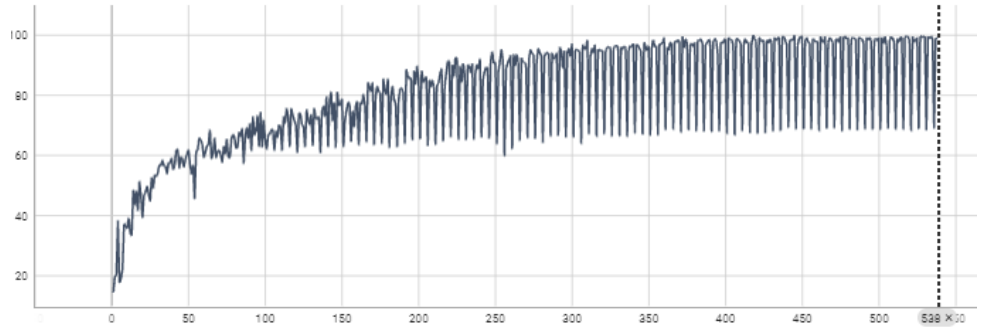
3.8 Modality Ablation

The contribution of depth modality is now evaluated by training the model only using the RGB stream. The results shown in Table 3.6 are referred to the same cross validation used for the previous experiments.

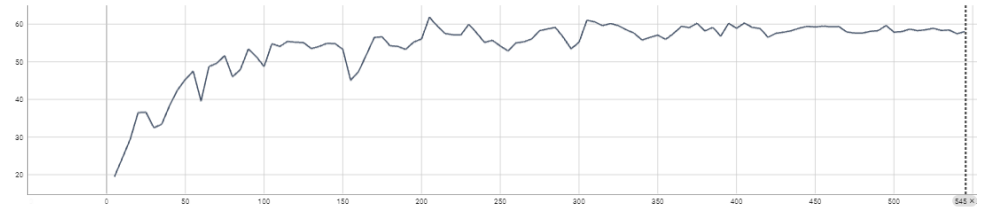
Table 3.6: Modality Ablation on CalD3rMenD3s 7 classes

Modality	Accuracy (%)
RGB	59.82
RGB+Depth	62,50

We can see that the depth modality provides a significant improvement in the accuracy, confirming the importance of the depth information. Moreover, in Figure 3.14(a) we see a 10 epoch training (only 1 fold is shown for better representation): note how the oscillation in the training accuracy is much higher and occurs earlier. This is a sign that the depth modality is actually helping the model to learn more robust features and generalize better.



(a) Training Accuracy



(b) Validation Accuracy

Figure 3.14: CalD3rMenD3s 7 classes 10 epochs training only RGB modality, (1 fold shown)

3.9 Modules Ablation

Table 3.7 compares the full model with and without final ViT encoder (which is substituted by a final AvgPooling), and without the Cross Modality Spatial Attention which is replaced with simpler sum fusion between the features from the two modalities:

$$\boldsymbol{p} = Softmax(MLP(AvgPool((\boldsymbol{X_{rgb}}) + (\boldsymbol{X_{depth}})))) \tag{3.3}$$

Table 3.7: Modules Ablation on CalD3rMenD3s 7 classes

Model	Accuracy(%)
w/o ViT encoder[a]	62,32
w/o Cross Mod. Spatial Att. [b]	61,73
Full	62,50

[a] Average pooling

[b] Sum Fusion

We can see a consistent drop in performance when not using the Cross Modality Spatial Attention, which is a good sign that the model is learning to combine the information from the two modalities in a more effective way than a simple sum. However, the improvement using the ViT encoder is not so relevant, which is a sign that the ViT encoder probably needs a larger dataset to be fully exploited.

3.10 Dataset Merge

As an additional experiment, CalD3rMenD3s, BU3DFE and Bosphorus datasets are merged into a single dataset for a final testing of the model. Whether it's true that CalD3rMenD3s is spontaneous while BU3DFE and Bosphorus are posed, the low intensity expressions from BU3DFE can be assimilated as spontaneous expression, while the overly exaggerated expressions both from BU3DFE and Bosphorus can be useful for the model to capture the most important features of the expressions and transfer that knowledge over less intense expressions.

For the Bosphorus dataset, all of the images already annotated into the 7 classes are selected. Also, some of the images annotated with Action Units are converted into the categorical annotation by following the indications in CK+ [1] in a more conservative way such that only images undoubtedly belonging to a class are converted. Table 3.8 shows the conversion from AUs to categorical annotation. Depth maps are extracted from the Point Clouds similarly to BU3DFE and the RGB images are cropped accordingly to the depth maps. In this way the Bosphorus dataset is aligned with the other two datasets.

Table 3.8: Bosphorus AUs to Categorical conversion

AUs	Class
Lip Presser (AU24)	Anger
Nose Wrinkler (AU9)	Disgust
Lip Corner Puller (AU12)	Happiness
Lip Corner Depressor (AU15)	Sadness
Outer Brow Raiser (AU2)	Surprise
Inner Brow Raiser (AU1)	Fear

The final dataset is composed of 12098 images which distribution is shown in Figure 3.15 which is then split into a training set and a test set using 80%-20% policy.

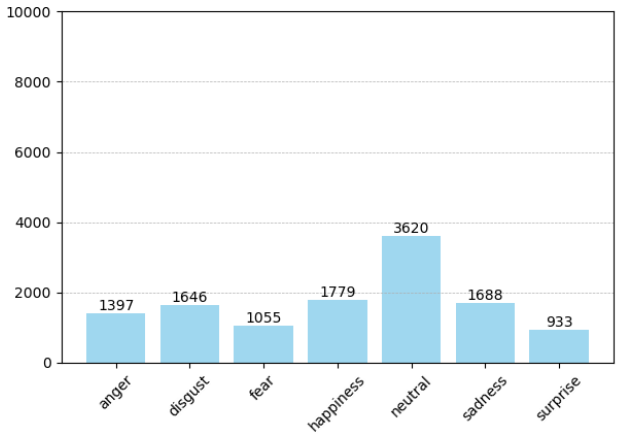
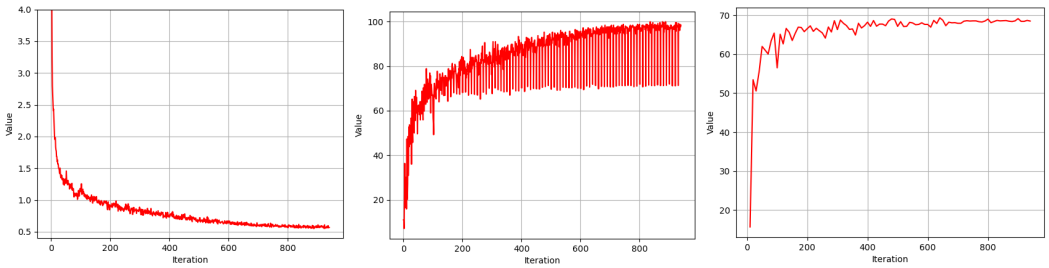


Figure 3.15: Classes distribution of the Global dataset

Final results are shown in Table 3.9 with confusion matrix and training statistics in Figures 3.17 and 3.16

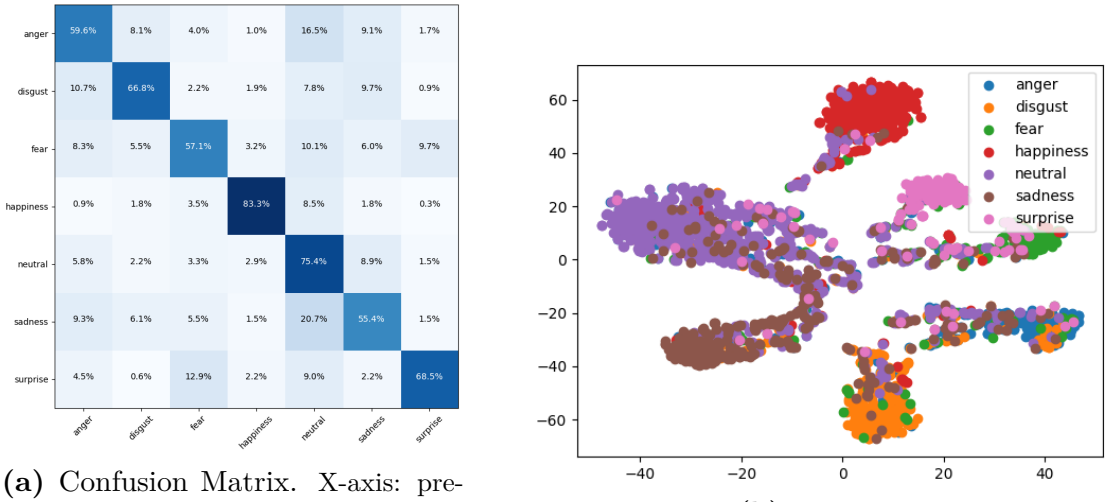
Table 3.9: Results on CalD3RMenD3s

Model	Year	Accuracy (%)	Classes
This	2024	68,48	7



(a) Training Loss **(b)** Training Accuracy **(c)** Validation Accuracy

Figure 3.16: Training and Validation Statistics for the Global dataset 7 classes



(a) Confusion Matrix. X-axis: predicted class, Y-axis: true class

(b) Features

Figure 3.17: Training results on Global dataset 7 classes

Chapter 4

Conclusions

4.1 Summary

This work addressed 2D+3D FER with an hybrid deep network which uses features extracted from a CNN to train a Vision Transformer Encoder. After a detailed analysis of currently available datasets and related works, a network proposal has been evaluated in different experimental setups over CalD3rMenD3s, BU3DFE and a new multimodal dataset born from union of the two, plus the Bosphorus dataset.

4.2 Future Work

The proposed model stregth relies in actively selecting most meaningful spatial regions in the features extracted from the CNN, and then using a transformer to learn a more global representation of the image. Spatial Attention could be introduced directly in the backbones, rather than only at the end, creating an "overloaded" Squeeze-Excite module that doesn't only apply attention on the channel dimension but also over spatial dimension. This can be done independently in each modality backbone or with the cross-modality setup. Moreover it could be implemented in different flavours, for example rotating the input features like in CBAM [57].

It could be interesting to expand furtherly the dataset, for example by including 4DFAB [4], to improve the generalization ability of the model. Moreover, leveraging larger dataset, deeper and larger backbones could improve the performance of the model.

The model could be extended to address annotation ambiguity by using a text encoder like in OpenAI's [74] or by using a variant of a batch normalization where the mean and standard deviation used for normalization are computed differently for each class, allowing the network to learn to ignore the noise in the labels.

It could be interesting to perform a more thorough cross dataset analysis, for example, by testing the model trained on the Global dataset on other datasets like the 4DFAB to see if it generalizes well.

Finally, given the simple nature of the model, it could be embedded in a real-time application that can be used in real-world scenarios. This could be done using a device recording in real-time the RGBD images and feeding them to the model to get a robust prediction even in cases of occlusions or low lighting conditions.

Bibliography

[1] Patrick Lucey, Jeffrey F. Cohn, Takeo Kanade, Jason M. Saragih, Zara Ambadar, and I. Matthews. «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* (2010), pp. 94–101. URL: https://api.semanticscholar.org/CorpusID:3329621 (cit. on pp. iv, 5, 6, 8, 14, 16, 55).

[2] Michael Lyons, Shigeru Akamatsu, Miyuki Kamachi, and Jiro Gyoba. «Coding facial expressions with gabor wavelets». In: *Proceedings Third IEEE international conference on automatic face and gesture recognition*. IEEE. 1998, pp. 200–205 (cit. on pp. iv, 6).

[3] Ali Mollahosseini, Behzad Hassani, and Mohammad H. Mahoor. «AffectNet: A Database for Facial Expression, Valence, and Arousal Computing in the Wild». In: *CoRR* abs/1708.03985 (2017). arXiv: 1708.03985. URL: http://arxiv.org/abs/1708.03985 (cit. on pp. iv, 5, 6, 8, 14–16).

[4] Shiyang Cheng, Irene Kotsia, Maja Pantic, and Stefanos Zafeiriou. «4DFAB: A Large Scale 4D Facial Expression Database for Biometric Applications». In: *CoRR* abs/1712.01443 (2017). arXiv: 1712.01443. URL: http://arxiv.org/abs/1712.01443 (cit. on pp. iv, 8, 14, 16, 58).

[5] Luca Ulrich, Federica Marcolin, Enrico Vezzetti, Francesca Nonis, Daniel C. Mograbi, Giulia Wally Scurati, Nicolò Dozio, and Francesco Ferrise. «CalD3r and MenD3s: Spontaneous 3D facial expression databases». In: *Journal of Visual Communication and Image Representation* 98 (2024), p. 104033. ISSN: 1047-3203. DOI: https://doi.org/10.1016/j.jvcir.2023.104033. URL: https://www.sciencedirect.com/science/article/pii/S1047320323002833 (cit. on pp. iv, 6, 11, 12, 14, 16, 42, 43, 48, 49).

[6] Christopher Michael Bishop and Hugh Bishop. *Deep Learning - Foundations and Concepts*. Ed. by Springer Cham. 1st ed. 2023. ISBN: 978-3-031-45468-4. DOI: https://doi.org/10.1007/978-3-031-45468-4 (cit. on pp. iv, 19, 24, 25).

[7] Yichuan Tang. «Deep Learning using Support Vector Machines». In: *CoRR* abs/1306.0239 (2013). arXiv: 1306.0239. URL: http://arxiv.org/abs/1306.0239 (cit. on pp. iv, 26, 27).

[8] Jie Cai, Zibo Meng, Ahmed-Shehab Khan, Zhiyuan Li, James O'Reilly, and Yan Tong. «Island Loss for Learning Discriminative Features in Facial Expression Recognition». In: *CoRR* abs/1710.03144 (2017). arXiv: 1710.03144. URL: http://arxiv.org/abs/1710.03144 (cit. on pp. iv, 29, 30).

[9] Lijun Yin, Xiaozhou Wei, Yi Sun, Jun Wang, and M.J. Rosato. «A 3D facial expression database for facial behavior research». In: *7th International Conference on Automatic Face and Gesture Recognition (FGR06)*. 2006, pp. 211–216. DOI: 10.1109/FGR.2006.6 (cit. on pp. iv, 12, 16, 42, 43).

[10] Onkar Sumant Tanmay Sharma. In: *Emotion Detection and Recognition Market Size, Share, Competitive Landscape and Trend Analysis Report by Software Tool, by Application, by Technology, by End User : Global Opportunity Analysis and Industry Forecast, 2021-2031*. 2023, p. 232. URL: https://www.alliedmarketresearch.com/emotion-detection-and-recognition-market (cit. on p. 2).

[11] P. Viola and M. Jones. «Rapid object detection using a boosted cascade of simple features». In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. Vol. 1. 2001, pp. I–I. DOI: 10.1109/CVPR.2001.990517 (cit. on p. 3).

[12] Wenqi Wu, Yingjie Yin, Xingang Wang, and De Xu. «Face Detection With Different Scales Based on Faster R-CNN». In: *IEEE Transactions on Cybernetics* 49.11 (2019), pp. 4017–4028. DOI: 10.1109/TCYB.2018.2859482 (cit. on p. 3).

[13] Florian Schroff, Dmitry Kalenichenko, and James Philbin. «FaceNet: A unified embedding for face recognition and clustering». In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 815–823. DOI: 10.1109/CVPR.2015.7298682 (cit. on p. 3).

[14] Thomas Kopalidis, Vassilios Solachidis, Nicholas Vretos, and Petros Daras. «Advances in Facial Expression Recognition: A Survey of Methods, Benchmarks, Models, and Datasets». In: *Information* 15.3 (2024). ISSN: 2078-2489. DOI: 10.3390/info15030135. URL: https://www.mdpi.com/2078-2489/15/3/135 (cit. on pp. 3, 4, 7, 9, 18).

[15] Arman Savran, Nese Alyuz, Hamdi Dibeklioglu, Oya Çeliktutan, Berk Gokberk, Bulent Sankur, and Lale Akarun. «Bosphorus Database for 3D Face Analysis». In: Jan. 2008, pp. 47–56. ISBN: 978-3-540-89990-7. DOI: 10.1007/978-3-540-89991-4_6 (cit. on pp. 4, 13, 16).

[16] Shan Li and Weihong Deng. «Deep Facial Expression Recognition: A Survey». In: *CoRR* abs/1804.08348 (2018). arXiv: 1804.08348. URL: http://arxiv.org/abs/1804.08348 (cit. on pp. 4, 7).

[17] Zhiding Yu and Cha Zhang. «Image based Static Facial Expression Recognition with Multiple Deep Network Learning». In: *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*. ICMI '15. Seattle, Washington, USA: Association for Computing Machinery, 2015, pp. 435–442. ISBN: 9781450339124. DOI: 10.1145/2818346.2830595. URL: https://doi.org/10.1145/2818346.2830595 (cit. on p. 4).

[18] Paul Ekman and W V Friesen. «Constants across cultures in the face and emotion.» In: *Journal of personality and social psychology* 17 2 (1971), pp. 124–9. URL: https://api.semanticscholar.org/CorpusID:14013552 (cit. on p. 4).

[19] Rachael E. Jack, Oliver G. B. Garrod, Hui Yu, Roberto Caldara, and Philippe G. Schyns. «Facial expressions of emotion are not culturally universal». In: *Proceedings of the National Academy of Sciences* 109 (2012), pp. 7241–7244. URL: https://api.semanticscholar.org/CorpusID:2661203 (cit. on p. 5).

[20] Rachael E. Jack, Wei Sun, Ioannis Delis, Oliver G. B. Garrod, and Philippe G. Schyns. «Four not six: Revealing culturally common facial expressions of emotion.» In: *Journal of experimental psychology. General* 145 6 (2016), pp. 708–30. URL: https://api.semanticscholar.org/CorpusID:19069039 (cit. on p. 5).

[21] Paul Ekman and Wallace V. Friesen. «Facial Action Coding System: Manual». In: 1978. URL: https://api.semanticscholar.org/CorpusID:140895661 (cit. on p. 5).

[22] Abhinav Dhall, Roland Goecke, Simon Lucey, and Tom Gedeon. «Collecting Large, Richly Annotated Facial-Expression Databases from Movies». In: *IEEE MultiMedia* 19.3 (2012), pp. 34–41. DOI: 10.1109/MMUL.2012.26 (cit. on pp. 7, 15, 16).

[23] Muhammad Sajjad, Fath U. Min Ullah, Mohib Ullah, Georgia Christodoulou, Faouzi Alaya Cheikh, Mohammad Hijji, Khan Muhammad, and Joel J.P.C. Rodrigues. «A comprehensive survey on deep facial expression recognition: challenges, applications, and future guidelines». In: *Alexandria Engineering Journal* (2023). URL: https://api.semanticscholar.org/CorpusID:256783294 (cit. on p. 9).

[24] Myung Beom Her, Jisu Jeong, Hojoon Song, and Ji-Hyeong Han. *Batch Transformer: Look for Attention in Batch*. 2024. arXiv: 2407.04218 [cs.CV]. URL: https://arxiv.org/abs/2407.04218 (cit. on pp. 9, 34, 39).

[25] Haotian Yang, Hao Zhu, Yanru Wang, Mingkai Huang, Qiu Shen, Ruigang Yang, and Xun Cao. «FaceScape: a Large-scale High Quality 3D Face Dataset and Detailed Riggable 3D Face Prediction». In: *CoRR* abs/2003.13989 (2020). arXiv: 2003.13989. URL: https://arxiv.org/abs/2003.13989 (cit. on pp. 11, 13, 16).

[26] Charles Ruizhongtai Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. «PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation». In: *CoRR* abs/1612.00593 (2016). arXiv: 1612.00593. URL: http://arxiv.org/abs/1612.00593 (cit. on p. 11).

[27] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. «PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space». In: *CoRR* abs/1706.02413 (2017). arXiv: 1706.02413. URL: http://arxiv.org/abs/1706.02413 (cit. on p. 11).

[28] Rana Hanocka, Amir Hertz, Noa Fish, Raja Giryes, Shachar Fleishman, and Daniel Cohen-Or. «MeshCNN: A Network with an Edge». In: *CoRR* abs/1809.05910 (2018). arXiv: 1809.05910. URL: http://arxiv.org/abs/1809.05910 (cit. on p. 11).

[29] Lijun Yin, Xiaochen Chen, Yi Sun, Tony Worm, and Michael Reale. «A high-resolution 3D dynamic facial expression database». In: *2008 8th IEEE International Conference on Automatic Face & Gesture Recognition*. 2008, pp. 1–6. DOI: 10.1109/AFGR.2008.4813324 (cit. on pp. 13, 16).

[30] Xing Zhang, Lijun Yin, Jeffrey F. Cohn, Shaun Canavan, Michael Reale, Andy Horowitz, and Peng Liu. «A high-resolution spontaneous 3D dynamic facial expression database». In: *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*. 2013, pp. 1–6. DOI: 10.1109/FG.2013.6553788 (cit. on pp. 13, 14, 16).

[31] Margaret M. Bradley and Peter J. Lang. «International Affective Picture System». In: *Encyclopedia of Personality and Individual Differences*. Ed. by Virgil Zeigler-Hill and Todd K. Shackelford. Cham: Springer International Publishing, 2017, pp. 1–4. ISBN: 978-3-319-28099-8. DOI: 10.1007/978-3-319-28099-8_42-1. URL: https://doi.org/10.1007/978-3-319-28099-8_42-1 (cit. on p. 14).

[32] Elise S. Dan-Glauser and Klaus R. Scherer. «The Geneva affective picture database (GAPED): a new 730-picture database focusing on valence and normative significance». In: *Behavior Research Methods* 43 (2011), pp. 468–477. URL: https://api.semanticscholar.org/CorpusID:207655542 (cit. on p. 14).

[33] Kaggle. *Facial Expression Recognition Challenge 2013*. 2013. URL: https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/ (cit. on pp. 14, 16).

[34] Emad Barsoum, Cha Zhang, Cristian Canton-Ferrer, and Zhengyou Zhang. «Training Deep Networks for Facial Expression Recognition with Crowd-Sourced Label Distribution». In: *CoRR* abs/1608.01041 (2016). arXiv: 1608.01041. URL: http://arxiv.org/abs/1608.01041 (cit. on pp. 14, 16).

[35] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. «From Facial Expression Recognition to Interpersonal Relation Prediction». In: *CoRR* abs/1609.06426 (2016). arXiv: 1609.06426. URL: http://arxiv.org/abs/1609.06426 (cit. on pp. 15, 16).

[36] Shan Li, Weihong Deng, and JunPing Du. «Reliable Crowdsourcing and Deep Locality-Preserving Learning for Expression Recognition in the Wild». In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 2584–2593. DOI: 10.1109/CVPR.2017.277 (cit. on pp. 15, 16).

[37] Fuyan Ma, Bin Sun, and Shutao Li. «Facial Expression Recognition With Visual Transformers and Attentional Selective Fusion». In: *IEEE Transactions on Affective Computing* 14.2 (2023), pp. 1236–1248. DOI: 10.1109/TAFFC.2021.3122146 (cit. on pp. 18, 35, 36, 40).

[38] Olga Russakovsky et al. «ImageNet Large Scale Visual Recognition Challenge». In: *CoRR* abs/1409.0575 (2014). arXiv: 1409.0575. URL: http://arxiv.org/abs/1409.0575 (cit. on pp. 19, 32).

[39] Mingxing Tan and Quoc V. Le. «EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks». In: *CoRR* abs/1905.11946 (2019). arXiv: 1905.11946. URL: http://arxiv.org/abs/1905.11946 (cit. on pp. 19, 32).

[40] Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. «Inverted Residuals and Linear Bottlenecks: Mobile Networks for Classification, Detection and Segmentation». In: *CoRR* abs/1801.04381 (2018). arXiv: 1801.04381. URL: http://arxiv.org/abs/1801.04381 (cit. on pp. 21, 22).

[41] Andrew Howard et al. «Searching for MobileNetV3». In: *CoRR* abs/1905.02244 (2019). arXiv: 1905.02244. URL: http://arxiv.org/abs/1905.02244 (cit. on pp. 21, 48).

[42] Danfeng Qin et al. *MobileNetV4 – Universal Models for the Mobile Ecosystem*. 2024. arXiv: 2404.10518 [cs.CV]. URL: https://arxiv.org/abs/2404.10518 (cit. on p. 23).

[43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. «Attention Is All You Need». In: *CoRR* abs/1706.03762 (2017). arXiv: 1706.03762. URL: http://arxiv.org/abs/1706.03762 (cit. on p. 23).

[44] Alexey Dosovitskiy et al. «An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale». In: *CoRR* abs/2010.11929 (2020). arXiv: 2010.11929. URL: https://arxiv.org/abs/2010.11929 (cit. on pp. 25, 34).

[45] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. «A Discriminative Feature Learning Approach for Deep Face Recognition». In: *European Conference on Computer Vision*. 2016. URL: https://api.semanticscholar.org/CorpusID:4711865 (cit. on pp. 28, 37).

[46] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. «Supervised Contrastive Learning». In: *CoRR* abs/2004.11362 (2020). arXiv: 2004.11362. URL: https://arxiv.org/abs/2004.11362 (cit. on p. 28).

[47] Florian Schroff, Dmitry Kalenichenko, and James Philbin. «FaceNet: A Unified Embedding for Face Recognition and Clustering». In: *CoRR* abs/1503.03832 (2015). arXiv: 1503.03832. URL: http://arxiv.org/abs/1503.03832 (cit. on p. 28).

[48] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. «Focal Loss for Dense Object Detection». In: *CoRR* abs/1708.02002 (2017). arXiv: 1708.02002. URL: http://arxiv.org/abs/1708.02002 (cit. on p. 30).

[49] A. Savchenko, Lyudmila V. Savchenko, and Ilya Makarov. «Classifying Emotions and Engagement in Online Learning Based on a Single Facial Expression Recognition Neural Network». In: *IEEE Transactions on Affective Computing* 13 (2022), pp. 2132–2143. URL: https://api.semanticscholar.org/CorpusID:250298227 (cit. on pp. 32, 40, 43).

[50] Andrey V. Savchenko. «Facial expression and attributes recognition based on multi-task learning of lightweight neural networks». In: *2021 IEEE 19th International Symposium on Intelligent Systems and Informatics (SISY)*. 2021, pp. 119–124. DOI: 10.1109/SISY52375.2021.9582508 (cit. on pp. 32, 40, 45).

[51] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. «MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications». In: *ArXiv* abs/1704.04861 (2017). URL: https://api.semanticscholar.org/CorpusID:12670695 (cit. on p. 32).

[52] Dongyoon Han, Sangdoo Yun, Byeongho Heo, and Young Joon Yoo. «ReXNet: Diminishing Representational Bottleneck on Convolutional Neural Network». In: *CoRR* abs/2007.00992 (2020). arXiv: 2007.00992. URL: https://arxiv.org/abs/2007.00992 (cit. on p. 32).

[53] Qiong Cao, Li Shen, Weidi Xie, Omkar M. Parkhi, and Andrew Zisserman. «VGGFace2: A dataset for recognising faces across pose and age». In: *CoRR* abs/1710.08092 (2017). arXiv: 1710.08092. URL: http://arxiv.org/abs/1710.08092 (cit. on pp. 32, 33).

[54] Saining Zhang, Yuhang Zhang, Ye Zhang, Yufei Wang, and Zhigang Song. «A Dual-Direction Attention Mixed Feature Network for Facial Expression Recognition». In: *Electronics* 12.17 (2023). ISSN: 2079-9292. DOI: 10.3390/electronics12173595. URL: https://www.mdpi.com/2079-9292/12/17/3595 (cit. on pp. 33, 40).

[55] Sheng Chen, Yang Liu, Xiang Gao, and Zhen Han. «MobileFaceNets: Efficient CNNs for Accurate Real-time Face Verification on Mobile Devices». In: *CoRR* abs/1804.07573 (2018). arXiv: 1804.07573. URL: http://arxiv.org/abs/1804.07573 (cit. on pp. 33, 34).

[56] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. «Going Deeper with Convolutions». In: *CoRR* abs/1409.4842 (2014). arXiv: 1409.4842. URL: http://arxiv.org/abs/1409.4842 (cit. on p. 33).

[57] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. «CBAM: Convolutional Block Attention Module». In: *CoRR* abs/1807.06521 (2018). arXiv: 1807.06521. URL: http://arxiv.org/abs/1807.06521 (cit. on pp. 33, 45, 58).

[58] Mang Ning, Albert Ali Salah, and Itir Onal Ertugrul. *Representation Learning and Identity Adversarial Training for Facial Behavior Understanding*. 2024. arXiv: 2407.11243 [cs.CV] (cit. on pp. 33, 34, 38, 40).

[59] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross B. Girshick. «Masked Autoencoders Are Scalable Vision Learners». In: *CoRR* abs/2111.06377 (2021). arXiv: 2111.06377. URL: https://arxiv.org/abs/2111.06377 (cit. on p. 33).

[60] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. «Deep Learning Face Attributes in the Wild». In: *Proceedings of International Conference on Computer Vision (ICCV)*. 2015 (cit. on p. 33).

[61] Haoran Bai, Di Kang, Haoxian Zhang, Jin-shan Pan, and Linchao Bao. «FFHQ-UV: Normalized Facial UV-Texture Dataset for 3D Face Reconstruction». In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), pp. 362–371. URL: https://api.semanticscholar.org/CorpusID:254018271 (cit. on p. 33).

[62] Dong Yi, Zhen Lei, Shengcai Liao, and S. Li. «Learning Face Representation from Scratch». In: *ArXiv* abs/1411.7923 (2014). URL: https://api.semanticscholar.org/CorpusID:17188384 (cit. on p. 33).

[63] Ankan Bansal, Anirudh Nanduri, Carlos Domingo Castillo, Rajeev Ranjan, and Rama Chellappa. «UMDFaces: An annotated face dataset for training deep networks». In: *2017 IEEE International Joint Conference on Biometrics (IJCB)* (2016), pp. 464–473. URL: https://api.semanticscholar.org/CorpusID:66176 (cit. on p. 33).

[64] Daniel Miller, Ira Kemelmacher-Shlizerman, and Steven M. Seitz. «MegaFace: A Million Faces for Recognition at Scale». In: *ArXiv* abs/1505.02108 (2015). URL: https://api.semanticscholar.org/CorpusID:15074951 (cit. on p. 33).

[65] Christoph Schuhmann et al. «LAION-5B: An open large-scale dataset for training next generation image-text models». In: *ArXiv* abs/2210.08402 (2022). URL: https://api.semanticscholar.org/CorpusID:252917726 (cit. on p. 33).

[66] Kaihao Zhang, Dongxu Li, Wenhan Luo, Jingyun Liu, Jiankang Deng, Wei Liu, and Stefanos Zafeiriou. «EDFace-Celeb-1M: Benchmarking Face Hallucination With a Million-Scale Dataset». In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45 (2021), pp. 3968–3978. URL: https://api.semanticscholar.org/CorpusID:238583342 (cit. on p. 33).

[67] Jiawei Mao, Rui Xu, Xuesong Yin, Yuanqi Chang, Binling Nie, and Aibin Huang. *POSTER++: A simpler and stronger facial expression recognition network*. 2023. arXiv: 2301.12149 [cs.CV]. URL: https://arxiv.org/abs/2301.12149 (cit. on pp. 34, 38, 40, 45).

[68] Jiankang Deng, Jia Guo, and Stefanos Zafeiriou. «ArcFace: Additive Angular Margin Loss for Deep Face Recognition». In: *CoRR* abs/1801.07698 (2018). arXiv: 1801.07698. URL: http://arxiv.org/abs/1801.07698 (cit. on p. 34).

[69] Azmine Toushik Wasi, Karlo Šerbetar, Raima Islam, Taki Hasan Rafi, and Dong-Kyu Chae. *ARBEx: Attentive Feature Extraction with Reliability Balancing for Robust Facial Expression Learning*. 2023. arXiv: 2305.01486 [cs.CV]. URL: https://arxiv.org/abs/2305.01486 (cit. on pp. 34, 37, 40).

[70] Fanglei Xue, Qiangchang Wang, and Guodong Guo. «TransFER: Learning Relation-aware Facial Expression Representations with Transformers». In: *CoRR* abs/2108.11116 (2021). arXiv: 2108.11116. URL: https://arxiv.org/abs/2108.11116 (cit. on pp. 35, 40, 45).

[71] Qionghao Huang, Changqin Huang, Xizhe Wang, and Fan Jiang. «Facial expression recognition with grid-wise attention and visual transformer». In: *Information Sciences* 580 (2021), pp. 35–54. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2021.08.043. URL: https://www.sciencedirect.com/science/article/pii/S0020025521008495 (cit. on pp. 35, 40).

[72] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. «Deep Residual Learning for Image Recognition». In: *CoRR* abs/1512.03385 (2015). arXiv: 1512.03385. URL: http://arxiv.org/abs/1512.03385 (cit. on p. 35).

[73] Yande Li, Mingjie Wang, Minglun Gong, Yonggang Lu, and Li Liu. *FERformer: Multi-modal Transformer for Facial Expression Recognition*. 2023. arXiv: 2303.12997 [cs.CV]. URL: https://arxiv.org/abs/2303.12997 (cit. on pp. 35, 39, 40).

[74] Alec Radford et al. «Learning Transferable Visual Models From Natural Language Supervision». In: *CoRR* abs/2103.00020 (2021). arXiv: 2103.00020. URL: https://arxiv.org/abs/2103.00020 (cit. on pp. 35, 48, 58).

[75] Hanting Li, Mingzhe Sui, Zhaoqing Zhu, and Feng Zhao. «MFEViT: A Robust Lightweight Transformer-based Network for Multimodal 2D+3D Facial Expression Recognition». In: *CoRR* abs/2109.13086 (2021). arXiv: 2109.13086. URL: https://arxiv.org/abs/2109.13086 (cit. on pp. 36, 40, 50).

[76] Mingzhe Sui, Hanting Li, Zhaoqing Zhu, and Feng Zhao. «AFNet-M: Adaptive Fusion Network with Masks for 2D+3D Facial Expression Recognition». In: *2023 IEEE International Conference on Image Processing (ICIP)*. 2023, pp. 116–120. DOI: 10.1109/ICIP49359.2023.10222441 (cit. on pp. 36, 40, 48, 50).

[77] Zhaoqing Zhu, Ming-Fa Sui, Hanting Li, and Feng Zhao. «CMANET: Curvature-Aware Soft Mask Guided Attention Fusion Network for 2D+3D Facial Expression Recognition». In: *2022 IEEE International Conference on Multimedia and Expo (ICME)* (2022), pp. 1–6. URL: https://api.semanticscholar.org/CorpusID:251847634 (cit. on pp. 36, 38, 40, 48, 50).

[78] Ming-Fa Sui, Zhaoqing Zhu, Feng Zhao, and Feng Wu. «FFNet-M: Feature Fusion Network with Masks for Multimodal Facial Expression Recognition». In: *2021 IEEE International Conference on Multimedia and Expo (ICME)* (2021), pp. 1–6. URL: https://api.semanticscholar.org/CorpusID:236227917 (cit. on pp. 36, 40).

[79] Rongrong Ni, Biao Yang, Xu Zhou, Angelo Cangelosi, and Xiaofeng Liu. «Facial Expression Recognition Through Cross-Modality Attention Fusion». In: *IEEE Transactions on Cognitive and Developmental Systems* 15.1 (2023), pp. 175–185. DOI: 10.1109/TCDS.2022.3150019 (cit. on pp. 37, 40, 50).

[80] Gary B. Huang, Marwan A. Mattar, Tamara L. Berg, and Eric Learned-Miller. «Labeled Faces in the Wild: A Database forStudying Face Recognition in Unconstrained Environments». In: 2008. URL: https://api.semanticscholar.org/CorpusID:88166 (cit. on p. 37).

[81] Lior Wolf, Tal Hassner, and Itay Maoz. «Face recognition in unconstrained videos with matched background similarity». In: *CVPR 2011*. 2011, pp. 529–534. DOI: 10.1109/CVPR.2011.5995566 (cit. on p. 37).

[82] Shisong Lin, Mengchao Bai, Feng Liu, Linlin Shen, and Yicong Zhou. «Orthogonalization-Guided Feature Fusion Network for Multimodal 2D+3D Facial Expression Recognition». In: *IEEE Transactions on Multimedia* 23 (2020), pp. 1581–1591. URL: https://api.semanticscholar.org/CorpusID:225718070 (cit. on pp. 37, 40).

[83] Ivan Grishchenko, Artsiom Ablavatski, Yury Kartynnik, Karthik Raveendran, and Matthias Grundmann. «Attention Mesh: High-fidelity Face Mesh Prediction in Real-time». In: *CoRR* abs/2006.10962 (2020). arXiv: 2006.10962. URL: https://arxiv.org/abs/2006.10962 (cit. on p. 44).

[84] Qiangchang Wang and Guodong Guo. «LS-CNN: Characterizing Local Patches at Multiple Scales for Face Recognition». In: *IEEE Transactions on Information Forensics and Security* 15 (2020), pp. 1640–1653. DOI: 10.1109/TIFS.2019.2946938 (cit. on p. 45).

[85] Leslie N. Smith and Nicholay Topin. «Super-Convergence: Very Fast Training of Residual Networks Using Large Learning Rates». In: *CoRR* abs/1708.07120 (2017). arXiv: 1708.07120. URL: http://arxiv.org/abs/1708.07120 (cit. on p. 46).

[86] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. «Optuna: A Next-generation Hyperparameter Optimization Framework». In: *CoRR* abs/1907.10902 (2019). arXiv: 1907.10902. URL: http://arxiv.org/abs/1907.10902 (cit. on p. 48).

[87] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. «Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization». In: *International Journal of Computer Vision* 128.2 (Oct. 2019), pp. 336–359. ISSN: 1573-1405. DOI: 10.1007/s11263-019-01228-7. URL: http://dx.doi.org/10.1007/s11263-019-01228-7 (cit. on p. 51).