Improving Performance and Interpretability in Recognizing Facial Action Units with Deep Neural Networks

Ciprian Adrian Corneanu

Universitat de Barcelona

A dissertation submitted by Ciprian Adrian Corneanu at Universitat de Barcelona to fulfil the degree of Doctor en Matemàtiques. Barcelona, September 30, 2019

Director

Dr. Sergio Escalera
Dept. de Matemàtiques i Informatica
Universitat de Barcelona

Co-director

Dr. Meysam Madadi
Dept. de Matemàtiques i Informatica
Universitat de Barcelona

Părinţilor mei.
To Lenka.

Abstract

Facial expressions are vital ways of communication between humans in social contexts. They are used as conversational markers and they convey information about affective and cognitive state. Many applications would benefit from the advance of automatic facial expression recognition (AFER). Robust AFER would improve human-computer interaction, it would increase driving safety, would help medical personal to better take care of patients with impaired communication ability and could transform online education. In recent years significant advancement has been undertaken in AFER with the use of deep neural networks (DNN). Unfortunately this increase in performance came with increased opacity. The current status of DNNs as "black-box" model hinders the advancement of the field. In this dissertation, we propose a new general framework for analysing deep neural networks based on the systematic study of their topology while they are learning patterns in the data. We use this framework to study a newly proposed DNN, specially built for Action Unit recognition which results in better understanding, control and increased performance. In summary, this dissertation has the following main contributions: a) Definition of comprehensive taxonomy of automatic computer vision approaches to automatic facial expression recognition followed by an extended review of historical and current trends in AFER. b) Proposal of a model that learns representation, patch and output structure of the face end-to-end e) Introduction of a structure inference topology that replicates inference algorithm in probabilistic graphical models by using a recurrent neural network c) Extended ablation study and experimental analysis of the newly proposed architecture d) Analysis and improving performance of the previously proposed architecture for facial expression architecture using the new theoretical framework. e) Formulation of novel general framework for analysis of deep neural networks based on algebraic topology f) Analysis of fundamental topological differences between DNNs that learn and DNNs that memorize g) Demonstrating the use of newly proposed analytical framework on facial action unit recognition using DSIN.

Acknowledgements

All these years, I had the immense pleasure to be part of the Human Pose and Behavior Analysis group (HuPBA) at the University of Barcelona and the Computer Vision Center. I would like to thank to all my colleagues, for their spirit, presence, humor and the long days (and sometimes nights) of hard work. You were true companions. Thank you, Alejandro, Marc, Javi, Carles, Albert, Victor, Gabri, Camilo.

My immense gratitude goes to my advisers. Thank you Sergio for your outstanding commitment, your insight and great leadership. Whenever I'll have to build something where nothing stood before I will think about you and how would you do it. You thought me a lot. Thank you Meysam, for your advice and numerous talks. You were of great help at most important times.

I had the great privilege to visit the group of Prof. Aleix Martinez at the Ohio State University. Aleix is a true inspiration to me. The time I have spent working with him has been challenging intellectually and rewarding personally. My time in OSU has been like going back to my childhood when I would surround myself with books and indulge in the affairs of the spirit.

A special thanks to the great city of Barcelona, it's rebellious spirit, it's people and amazing joy of life. In any other place everything would have been harder.

Last, but not least I would like to thank to my dearest friends and family. You know who you are. Wherever I go, I carry you in my heart.

“The face is the mirror of the mind, and eyes without speaking confess the secrets of the heart.”

St. Jerome

Contents

Chapter 1

Introduction

The face is a fundamental component of human identity. By looking at someone's face humans can almost instantly tell if they know the person, her gender and age, where is she looking and if they are in pain or amused. On whatever airport you would be in the world, in order to verify your identity, a customs officer will look at you and compare your face to a reference photo you have in the passport. The same thing will happen if you get stopped by the police on the street, if you request some service in an office of the local administration, or if you are trying to get a reduction fee when entering into a museum because maybe you are a student. In marketing, when famous artists or athletes endorse some company, product or cause, it will be their face that you will see on a billboard or a TV commercial. The face entered even into politics, many western countries having endless debates and even passing laws regulating if people have the right to cover their faces in public.

Facial shape has great diversity and uniqueness among humans. The specific morphology of the skull and of the soft tissues, muscles, fat and skin that cover it, is unique to every person and changes only very slowly with age. But besides its rigid shape conveying essential information about human identity, faces are also highly morpho-dynamic being a vital conveyor of social signals. Faces are very important in the way we communicate, we picture ourselves and we think about emotion. For example, one of the most used forms of depicting human emotion in art is through facial expressions (see Fig.1.1). By looking at someone's face, we can most times tell if that person is sad or happy, if they are in doubt, if what they say is ironic or serious or if they might be in pain. In general the face is a very rich source of affective and cognitive information and an important channel of communication. In this context it is interesting to note that newborn babies are able to recognize and mimic their mother facial expressions years before they are able to speak or walk. Facial expressions are one of the first motor coordination skills that humans posses. Recognizing facial expressions is essential for being able to tell what others feel and think and to influence their actions which has high importance for survival.

Figure 1.1: It has always been very common in art to portray emotion as facial expression. Sculptures by Franz Messerschmidt, 18th century artist.

This amazing human capability is largely taken for granted. We rarely think how it is possible to be able to recognize so many different facial expressions with such precision and robustness. But as for many other cognitive capacities, their complexity only becomes obvious when we are trying to replicate them into a human engineered device like a computer. *How a computer could replicate these amazing capacities with higher performance and increased transparency is the topic of this dissertation.*

Imagine for a second a lunch with friends. At a certain point one of you grabs her smart phone, takes a photo and shares it on some social media network. Maybe the picture looks something like in Fig. 1.2. What everybody else will see is a nice image with a bunch of dear friends around a table full of great food. A more interested observer will infer an amazing amount of things just by taking a glance at this photo. They will recognize that you are sharing a salad and most of you are having a glass of wine. They will notice what kind of clothes you wear and combined with the strong light coming through the window will probably think that the weather is warm. By the background, they will also think that you are in someone's living room, not in a restaurant and the context is casual. Then by looking at the persons they will easily be able to tell that there are three women and three men around the table and that they are around 30 years old. Closer attention they would pay to everyone's faces more social cues will become obvious. One can tell where people are looking, if they are about to say something and with whom are they interacting to. And by the facial expressions, one could also infer cognitive and affective states and markers of conversation. For example, that some are slightly amused, others slightly impressed and at least one guy tries to falsely act disgusted.

Overall, the amount of information our brains are capable of extracting from a single image is truly impressive especially if one would like to replicate this capacity in an engineered device like a computer. All the things that your friend "sees" are inferences that she is making in her head, starting from the early visual system to the visual cortex

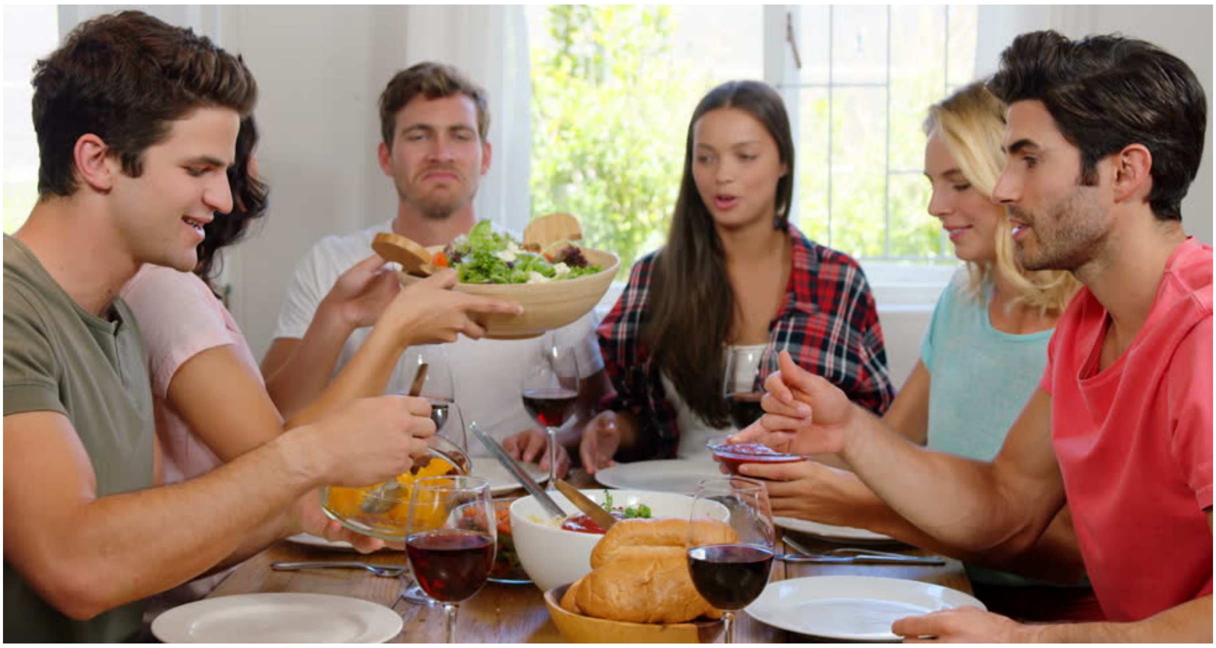


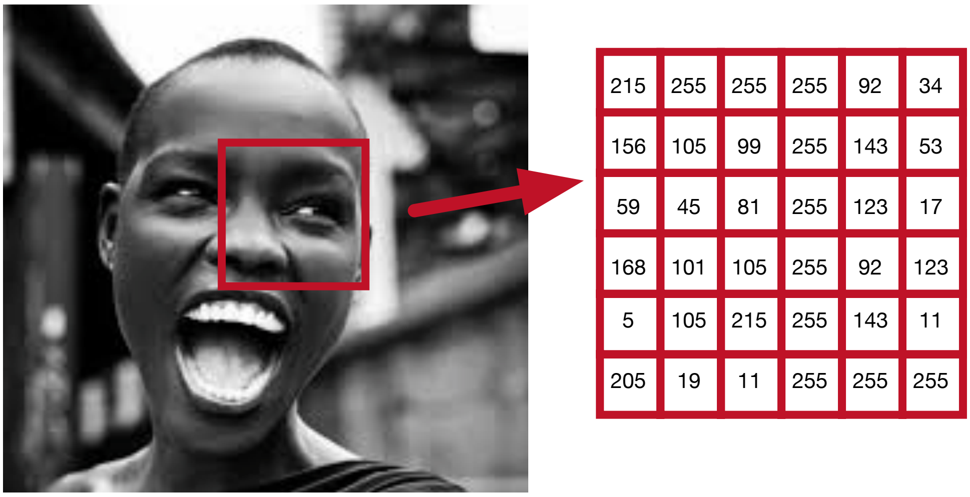
Figure 1.2: A human can infer a lot of information from a static image like object recognition, semantic segmentation, 3D scene reconstruction. In this thesis we focus on automatic facial expression recognition (AFER). Replicating this ability into a machine is challenging.

and beyond. Strictly speaking the image itself of a facial expression is only a collection of numbers with a specific grid-like structure (see Fig. 1.3). The main focus of computer vision is to try to reach a higher level understanding of images. This can be used for object recognition, scene parsing, semantic segmentation or facial expression recognition from this simple collection of numbers represented by an image. *Our focus is on Automatic Facial Expression Recognition (AFER) from images.*

Let's think for a while at the challenges. Look at Fig. 1.2 again. Probably the large majority of humans, if asked, would agree that the man on the left and the blonde lady on the right are slightly amused. But this is the results of a complex pattern recognition algorithm built in our brains. But the collections of numbers contained in the photos would be very different from face to face due to different identity, different poses and different occlusions or hair styles. A robust algorithm would have to learn what relevant information to consider and what to ignore in order to make a decision. These problems are common in any kind of pattern recognition tasks be it from images, audio, or any other type of signal.

In late years, a very powerful model, called Deep Neural Networks (DNN), achieved unprecedented performance in a myriad of tasks, usually when a large amounts of data is available. Unfortunately, at the same time these learning models, *as they became more powerful they also became more complex and opaque.*

Neural networks is not a new idea. The intuition that the designers of a learning



215	255	255	255	92	34
156	105	99	255	143	53
59	45	81	255	123	17
168	101	105	255	92	123
5	105	215	255	143	11
205	19	11	255	255	255

Figure 1.3: An image is a structured collection of numbers. A computer able to gain understanding from an image will need to implement robust pattern recognition algorithms. This is the aim of computer vision. This thesis focuses on improving performance and interpretability of deep learning algorithms that recognize facial expressions from images.

algorithm should look for inspiration to the mammalian brain, was a central way of thinking since the beginning of Artificial Intelligence (AI) in the 1940s. Scientists and innovators focused on greatly simplified mathematical models that describe a biological neuron behaviour. It was theoretically shown that given the right scale, these simplified neurons, when put together are able to approximate any kind of function, no matter how complex. Unfortunately this brings up an important problem. Although the code for specifying the architecture and training of a model can be simple, the results can be very complex, oftentimes effectively resulting in ”black boxes”. When a DNN looks at a picture and decides that it contains the face of a person that looks ”happy” or ”sad” (see Fig. 1.4), the exact functional processes that generate these outputs are hard to interpret even to the scientists who generated the algorithms in the first place, although some progress in interpretability is being made. Finally we end up with a paradoxical situation, in which we have models that perform well but we do not really know how and why. *We are thus interested in improving performance in AFER from images using DNNs and at the same time gain understanding about how these models learn.*

1.1 Motivation and Objectives

We previously outlined the main motivation of this thesis. We restate it here together with the main objectives that consequently derive. The whole manuscript is structured along these lines:

1. **Facial expressions (FE) are vital signaling systems of affect, conveying cues about the emotional and cognitive state of humans**. If we want to

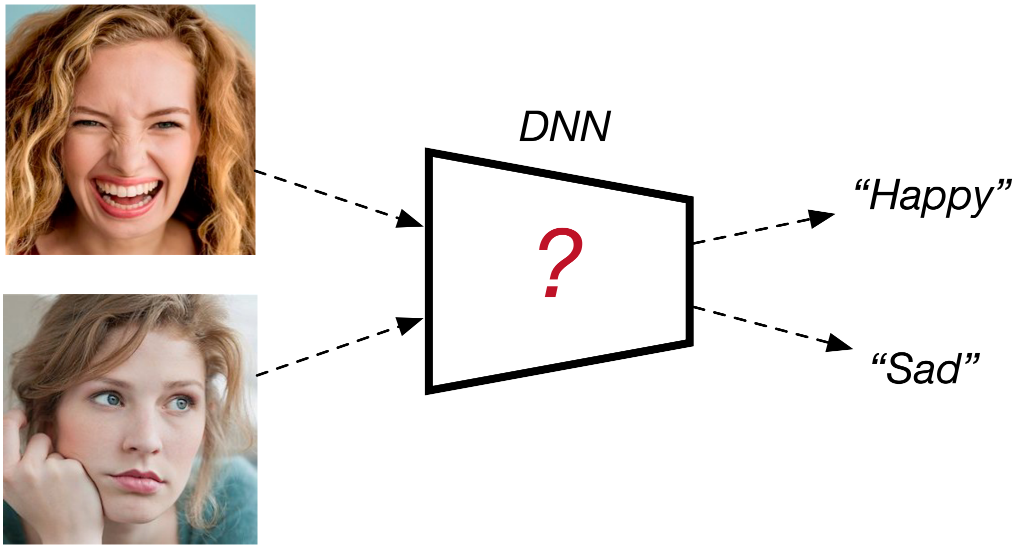


Figure 1.4: Increased performance of deep neural networks in a myriad of computer vision problems also came with greater opaqueness. In this thesis, we propose a novel framework that provides new insight into how deep neural networks learn, with applications into improved facial expressions recognition.

gain better understanding of humans we should **improve performance in facial expression recognition**. This is the main objective of this thesis.

2. **Deep neural networks (DNN) are increasingly opaque as they grow in performance and complexity.** They are regarded as "black-box" models. For better performance, usability and general public acceptance it is important to **gain understanding into what does it mean to learn in deep neural networks**. While this is a general objective and we propose a general framework, applicable to a myriad of problems, we focus our attention to our initial motivation and objective, namely into **what does it mean to learn facial expressions in deep neural networks and how could this be used for increased performance**.

1.2 Contributions and Thesis Outline

We present several contributions to the problem of facial expression recognition. They are grouped in three main categories: contributions related to the taxonomy of facial expression recognition, contributions to improving performance of automatic facial expression recognition and contributions related to interpretability of deep neural networks. There is a direct correspondence between this categorization and the outline of this manuscript. The contributions are the following:

1. **Automatic Facial Expression Recognition. General Framework, Evolutionary Perspective and Trends.** We define a comprehensive taxonomy of

automatic computer vision approaches for automatic facial expression recognition (AFER). The definition and choices of the different components are analyzed and discussed. This is complemented with a section dedicated to the historical evolution of FE approaches and an in-depth analysis of latest trends. Additionally, we provide an introduction into affect inference from the face from an evolutionary perspective. We emphasize research produced since the last major review of AFER in 2009. Our focus on inferring affect, defining a comprehensive taxonomy and treating different modalities is aiming at proposing a more general perspective on AFER and its current trends. The main contributions are:

   (a) An evolutionary perspective of affect inference from the face (Chapter 2.2).
   (b) Definition of comprehensive taxonomy of automatic computer vision approaches to automatic facial expression recognition (Chapter 2.3).
   (c) Extended survey of historical and current trends in AFER (Chapter. 2.4).

2. **Performance in Facial Expression Recognition.** We propose a deep neural architecture that tackles local representation learning and class structure learning by combining learned local and global features in its initial stages and replicating a message passing algorithm between classes similar to a graphical model inference approach in later stages. We show that by training the model end-to-end with increased supervision we improve state-of-the-art facial action unit recognition. The main contributions are:

   (a) Proposal of a model that learns representation, patch and output structure of the face end-to-end (Chapter 3.3).
   (b) Introduction of a structure inference topology that replicates inference algorithm in probabilistic graphical models by using a recurrent neural network (Chapter 3.3).
   (c) Extended ablation study and experimental analysis of the newly proposed architecture (Chapter 3.4).

3. **Interpretability in Facial Expression Recognition.** We derive a novel approach to define what it means to learn in deep networks. We show how this defines the ability of a network to generalize to unseen testing samples and, most importantly, why this is the case. More concretely, there are three main contributions:

   (a) Formulation of novel general framework for analysis of deep neural networks based on algebraic topology (Chapter 4).
   (b) Analysis of fundamental topological differences between DNNs that learn and DNNs that memorize (Chapter 5).

(c) Analyze and improving performance of the previously proposed architecture for facial expression architecture using the new theoretical framework (Chapter 6.)

Chapter 2

Automatic Recognition of Facial Expressions and Facial Action Units: Taxonomy, Related Work and Trends

Facial expressions are an important way through which humans interact socially. Building a system capable of automatically recognizing facial expressions from images and video has been an intense field of study in recent years. Interpreting such expressions remains challenging and much research is needed about the way they relate to human affect. This chapter presents a general overview of automatic facial expression analysis. We define a new taxonomy for the field, encompassing all steps from face detection to facial expression recognition, and describe and classify the state of the art methods accordingly. We also present the important datasets and the bench-marking of most influential methods. We conclude with a general discussion about trends, important questions and future lines of research.

2.1 Introduction

Facial expressions (FE) are vital signaling systems of affect, conveying cues about the emotional state of persons. Together with voice, language, hands and posture of the body, they form a fundamental communication system between humans in social contexts. Automatic FE recognition (AFER) is an interdisciplinary domain standing at the crossing of behavioral science, neurology, and artificial intelligence.

Studies of the face were greatly influenced in premodern times by popular theories of physiognomy and creationism. Physiognomy assumed that a person's character or personality could be judged by their outer appearance, especially the face [168]. Leonardo Da Vinci was one of the first to refute such claims stating they were without scientific support [29]. In the 17th century in England, John Buwler studied human communication

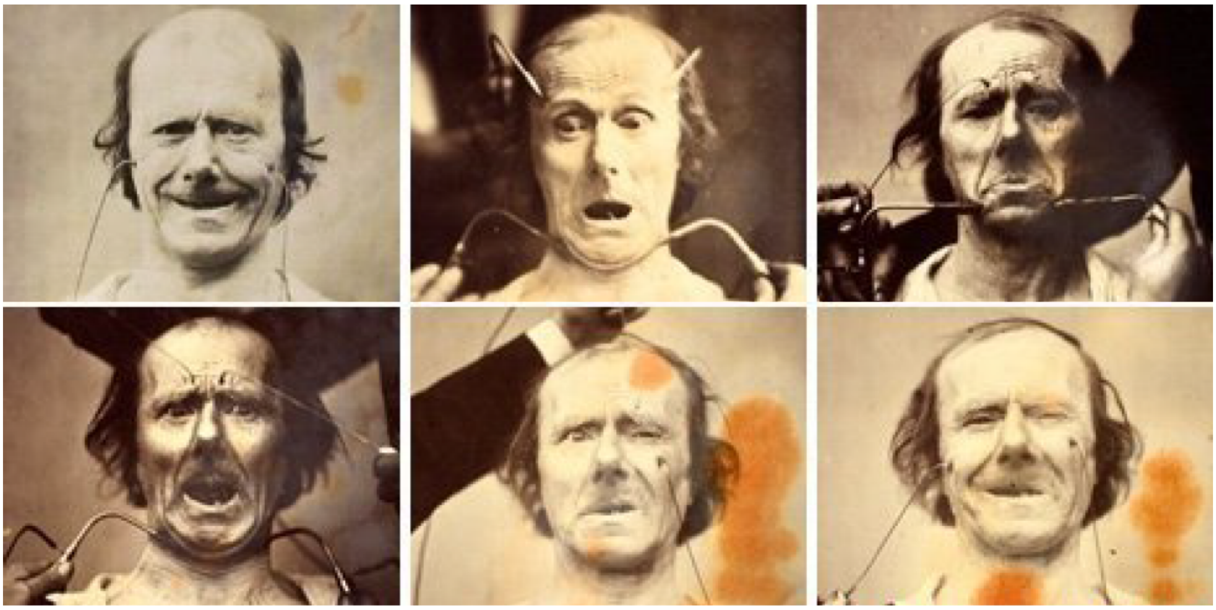


Figure 2.1: In the 19th century, Duchenne de Boulogne conducted experiments on how FEs are produced. From [37].

with particular interest in the sign language of persons with hearing impairment. His book *Pathomyotomia* or *Dissection of the significant Muscles of the Affections of the Mind* was the first consistent work in the English language on the muscular mechanism of FE [78]. About two centuries later, influenced by creationism, Sir Charles Bell investigated FE as part of his research on sensory and motor control. He believed that FE was endowed by the Creator solely for human communication. Subsequently, Duchenne de Boulogne conducted systematic studies on how FEs are produced [37]. He published beautiful pictures of sometimes strange FEs obtained by electrically stimulating facial muscles (see Figure 2.1). Approximately in the same historical period, Charles Darwin firmly placed FE in an evolutionary context [36]. This marked the beginning of modern research of FEs. More recently, important advancements were made through the works of researchers like Carroll Izard and Paul Ekman who inspired by Darwin performed seminal studies of FEs [90, 48, 50].

2.2 Inferring affect from FEs

Depending on context FEs may have varied communicative functions. They can regulate conversations by signaling turn-taking, convey biometric information, express intensity of mental effort, and signal emotion. By far, the latter has been the one most studied.

2.2.1 Describing affect

Attempts to describe human emotion mainly fall into two approaches: categorical and dimensional description.

Categorical description of affect. Classifying emotions into a set of distinct classes that can be recognized and described easily in daily language has been common since at least the time of Darwin. More recently, influenced by the research of Paul Ekman [48, 49] a dominant view upon affect is based on the underlying assumption that humans universally express a set of discrete primary emotions which include happiness, sadness, fear, anger, disgust, and surprise (see Figure 2.2). Mainly because of its simplicity and its universality claim, the universal primary emotions hypothesis has been extensively exploited in affective computing.

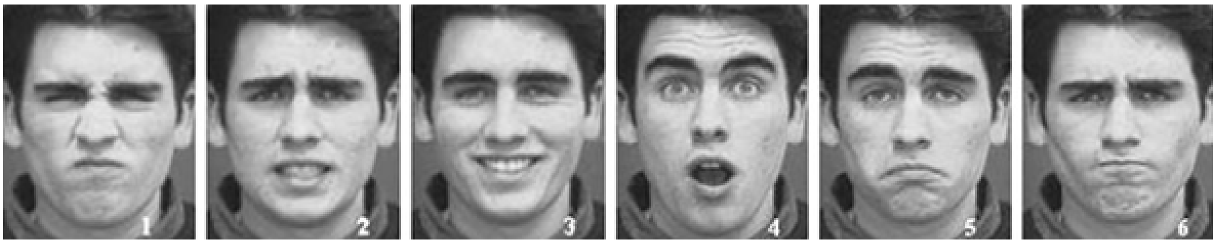


Figure 2.2: Primary emotions expressed on the face. From left to right: disgust, fear, joy, surprise, sadness, anger. From [8].

Dimensional description of affect. Another popular approach is to place a particular emotion into a space having a limited set of dimensions [79, 172, 227]. These dimensions include valence (how pleasent or unpleasent a feeling is) activation[1] (how likely is the person to take action under the emotional state) and control (the sense of control over the emotion). Due to the higher dimensionality of such descriptions they can potentially describe more complex and subtle emotions. Unfortunately, the richness of the space is more difficult to use for automatic recognition systems because it can be challenging to link such described emotion to a FE. Usually automatic systems based on dimensional representation of emotion simplify the problem by dividing the space in a limited set of categories like positive vs negative or quadrants of the 2D space [243].

2.2.2 An evolutionist approach to FE of affect

At the end of the 19th century Charles Darwin wrote *The Expression of the emotion in Man and Animals*, which largely inspired the study of FE of emotion. Darwin proposed that FEs are the residual actions of more complete behavioral responses to environmental challenges. Constricting the nostrils in disgust served to reduce inhalation of noxious or

[1] Also known as arousal.

harmful substances. Widening the eyes in surprise increased the visual field to see an unexpected stimulus. Darwin emphasized the adaptive functions of FEs.

More recent evolutionary models have come to emphasize their communicative functions [65]. [191] proposed a process of exaptation in which adaptations (such as constricting the nostrils in disgust) became recruited to serve communicative functions. Expressions (or displays) were ritualized to communicate information vital to survival. In this way, two abilities were selected for their survival advantages. One was to automatically display exaggerated forms of the original expressions; the other was to automatically interpret the meaning of these expressions. From this perspective, disgust communicates potentially aversive foods or moral violations; sadness communicates request for comfort. While some aspects of evolutionary accounts of FE are controversial [14], strong evidence exists in their support. Evidence includes universality of FEs of emotion, physiological specificity of emotion, and automatic appraisal and unbidden occurrence [47, 105, 143].

Universality. There is a high degree of consistency in the facial musculature among peoples of the world. The muscles necessary to express primary emotions are found universally [186, 77, 25], and homologous muscles have been documented in non-human primates [218, 220, 219]. Similar FEs in response to species-typical signals have been observed in both human and non-human primates [46].

Recognition. Numerous perceptual judgment studies support the hypothesis that FEs are interpreted similarly at levels well above chance in both Western and non-Western societies. Even critics of strong evolutionary accounts [173], [93] find that recognition of FEs of emotion are universally above chance and in many cases quite higher.

Physiological specificity. Physiological specificity appears to exist as well. Using directed facial action tasks to elicit basic emotions, Levenson and colleagues [118] found that HR, GSR, and skin temperature systematically varied with the hypothesized functions of basic emotions. In anger, blood flow to the hands increased to prepare for fight. For the central nervous system, patterns of prefrontal and temporal asymmetry systematically differed between enjoyment and disgust when measured using the *Facial Action Coding System* (FACS) [52]. Left-frontal asymmetry was greater during enjoyment; right frontal asymmetry was greater during disgust. These findings support the view that emotion expressions reliably signal action tendencies [67, 155].

Subjective experience. While not critical to an evolutionary account of emotion, evidence exists as well for concordance between subjective experience and FE of emotion [51, 246]. However, more work is needed in this regard. Until recently, manual annotation of FE or facial EMG were the only means to measure FE of emotion. Because manual annotation is labor intensive, replication of studies is limited.

In summary, the study of FE initially was strongly motivated by evolutionary accounts of emotion. Evidence has broadly supported those accounts. However, FE more broadly figures in cultural bio-psycho-social accounts of emotion. Facial expression signals emotion, communicative intent, individual differences in personality, and psychiatric and medical status, and helps to regulate social interaction. With the advent of automated methods of AFER, we are poised to make major discoveries in these areas.

2.2.3 Applications

The ability to automatically recognize FEs and infer affect has a wide range of applications. AFER, usually combined with speech, gaze and standard interactions like mouse movements and keystrokes can be used to build adaptive environments by detecting the user's affective states [45, 140]. Similarly, one can build socially aware systems [212, 39], or robots with social skills like Sony's AIBO and ATR's Robovie [89]. Detecting students' frustration can help improve e-learning experiences [103]. Gaming experience can also be improved by adapting difficulty, music, characters or mission according to the player's emotional responses [13, 203, 21]. Pain detection is used for monitoring patient progress in clinical settings [136, 101, 88]. Detection of truthfulness or potential deception can be used during police interrogations or job interviews [174]. Monitoring drowsiness or attentive and emotional status of the driver is critical for the safety and comfort of driving [215]. Depression recognition from FEs is a very important application in analysis of psychological distress [75, 185, 98]. Finally, in recent years successful commercial applications like Emotient [3], Affectiva [1], RealEyes [7] and Kairos [5] perform large-scale internet-based assessments of viewer reactions to ads and related material for predicting buying behaviour.

2.3 A taxonomy for recognizing FEs

In Figure 2.3 we propose a taxonomy for AFER, built along two main components: parametrization and recognition of FEs. These are important components of an automatic FE recognition system, regardless of the data modality.

Parametrization deals with defining coding schemes for describing FEs. Coding schemes may be categorized into two main classes. *Descriptive* coding schemes parametrize FE in terms of surface properties. They focus on what the face can do. *Judgmental* coding schemes describe FEs in terms of the latent emotions or affects that are believed to generate them. Please refer to Section 2.3.1 for further details.

An automatic facial analysis system from images or video usually consists of four main parts. First, faces have to be localized in the image. Second, for many methods a face registration has to be performed. During registration, fiducial points (e.g., the corners of the mouth or the eyes) are detected, allowing for a particularization of the face to different poses and deformations. In a third step, features are extracted from the face with techniques dependent on the data modality. The approaches are divided into geometric

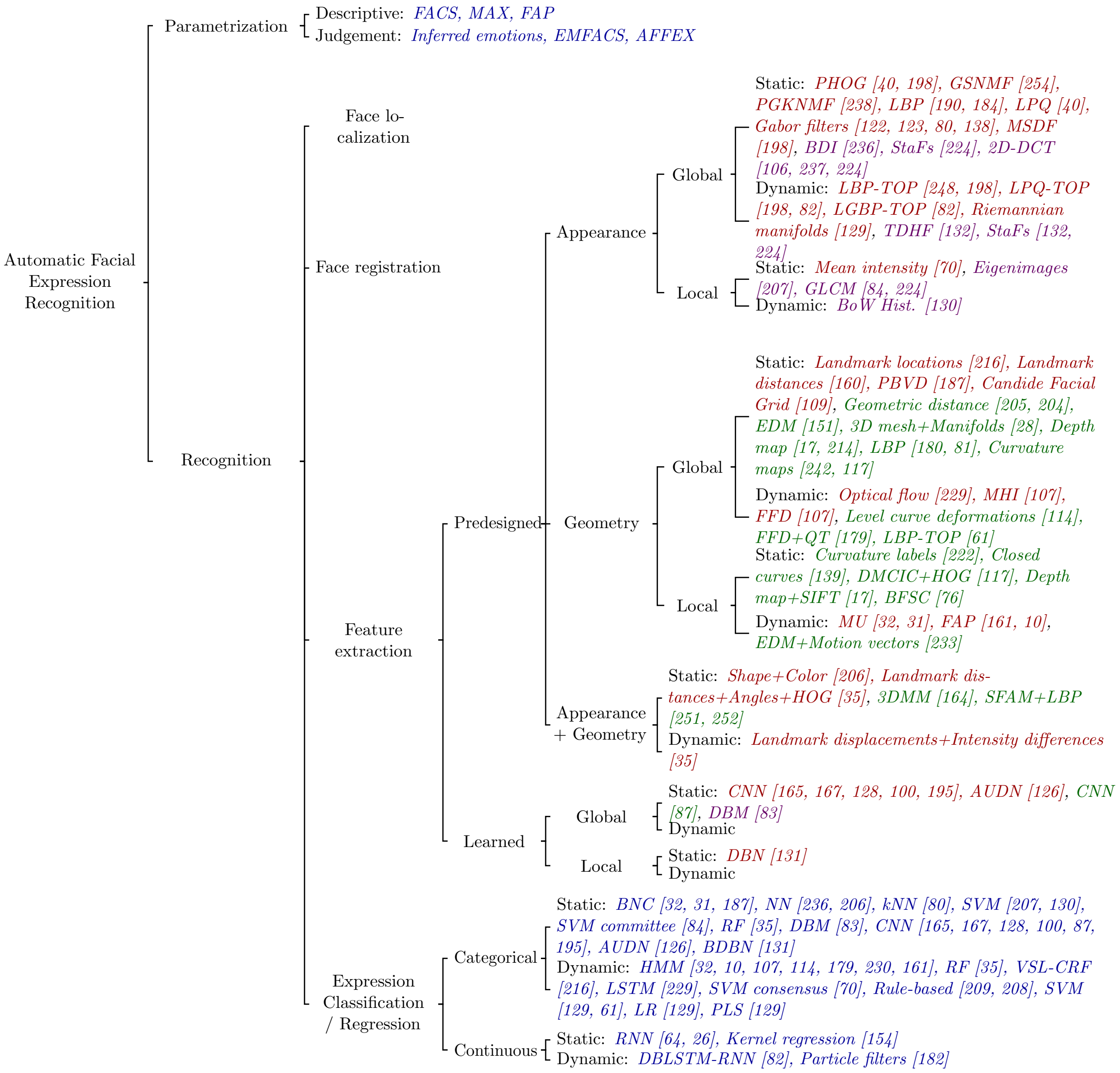

Automatic Facial Expression Recognition
Parametrization
Descriptive: FACS, MAX, FAP
Judgement: Inferred emotions, EMFACS, AFFEX
Recognition
Face localization
Face registration
Feature extraction
Predesigned
Appearance
Global
Static: PHOG [40, 198], GSNMF [254], PGKNMF [238], LBP [190, 184], LPQ [40], Gabor filters [122, 123, 80, 138], MSDF [198], BDI [236], StaFs [224], 2D-DCT [106, 237, 224]
Dynamic: LBP-TOP [248, 198], LPQ-TOP [198, 82], LGBP-TOP [82], Riemannian manifolds [129], TDHF [132], StaFs [132, 224]
Local
Static: Mean intensity [70], Eigenimages [207], GLCM [84, 224]
Dynamic: BoW Hist. [130]
Geometry
Global
Static: Landmark locations [216], Landmark distances [160], PBVD [187], Candide Facial Grid [109], Geometric distance [205, 204], EDM [151], 3D mesh+Manifolds [28], Depth map [17, 214], LBP [180, 81], Curvature maps [242, 117]
Dynamic: Optical flow [229], MHI [107], FFD [107], Level curve deformations [114], FFD+QT [179], LBP-TOP [61]
Local
Static: Curvature labels [222], Closed curves [139], DMCIC+HOG [117], Depth map+SIFT [17], BFSC [76]
Dynamic: MU [32, 31], FAP [161, 10], EDM+Motion vectors [233]
Appearance + Geometry
Static: Shape+Color [206], Landmark distances+Angles+HOG [35], 3DMM [164], SFAM+LBP [251, 252]
Dynamic: Landmark displacements+Intensity differences [35]
Learned
Global
Static: CNN [165, 167, 128, 100, 195], AUDN [126], CNN [87], DBM [83]
Dynamic
Local
Static: DBN [131]
Dynamic
Expression Classification / Regression
Categorical
Static: BNC [32, 31, 187], NN [236, 206], kNN [80], SVM [207, 130], SVM committee [84], RF [35], DBM [83], CNN [165, 167, 128, 100, 87, 195], AUDN [126], BDBN [131]
Dynamic: HMM [32, 10, 107, 114, 179, 230, 161], RF [35], VSL-CRF [216], LSTM [229], SVM consensus [70], Rule-based [209, 208], SVM [129, 61], LR [129], PLS [129]
Continuous
Static: RNN [64, 26], Kernel regression [154]
Dynamic: DBLSTM-RNN [82], Particle filters [182]

Figure 2.3: Taxonomy for AFER in Computer Vision. Red corresponds to RGB, green to 3D, and purple to thermal.

or appearance based, global or local, and static or dynamic (Section 2.3.2.1). Other approaches use a combination of these categories. Finally, machine learning techniques are used to discriminate between FEs. These techniques can predict a categorical expression or represent the expression in a continuous output space, and can model or not temporal information about the dynamics of FEs (Section 2.3.2.2).

Modern FE recognition techniques rely on labeled data to learn discriminative patterns for recognition and, in many cases, feature extraction. For this reason we introduce in Section 2.3.3 the main datasets for all three modalities. These are characterized based on the content of the labeled data, the capture conditions and participants distribution.

2.3.1 Parameterization of FEs

Descriptive coding schemes focus on what the face can do. The most well known examples of such systems are *Facial Action Coding System* (FACS) and *Face Animation Paramters* (FAP). Perhaps the most influential, FACS (1978; 2002) seeks to describe nearly all possible FEs in terms of anatomically-based facial actions [53, 56]. The FEs are coded in *Action Units* (AU), which define the contraction of one or more facial muscles (see Figure 2.4). FACS also provides the rules for visual detection of AUs and their temporal segments (onset, apex, offset, ordinal intensity). For relating FEs to emotions, Ekman and Friesen later developed the EMFACS (Emotion FACS), which scores facial actions relevant for particular emotion displays [66]. FAP is now part of the MPEG4 standard and is used for synthesizing FE for animating virtual faces. Is is rarely used to parametrize FEs for recognition purposes [161, 10]. Its coding scheme is based on the position of key feature control points in a mesh model of the face. *Maximally Discriminative Facial Movement Coding System* (MAX) [91], another descriptive system, is less granular and less comprehensive. Brow raise in MAX, for instance, corresponds to two separate actions in FACS. It is a truly sign-based approach as it makes no inferences about underlying emotions.

Judgmental coding schemes, on the other hand, describe FEs in terms of the latent emotions or affects that are believed to generate them. Because a single emotion or affect may result in multiple expressions, there is no 1:1 correspondence between what the face does and its emotion label. A hybrid approach is to define emotion labels in terms of specific signs rather than latent emotions or affects. Examples are EMFACS and AFFEX [92]. In each, expressions related to each emotion are defined descriptively. As an example, enjoyment may be defined by an expression displaying an oblique lip-corner pull co-occurring with cheek raise. Hybrid systems are similar to judgment-based systems in that there is an assumed 1:1 correspondence between emotion labels and signs that describe them. For this reason, we group hybrid approaches with judgment-based systems.

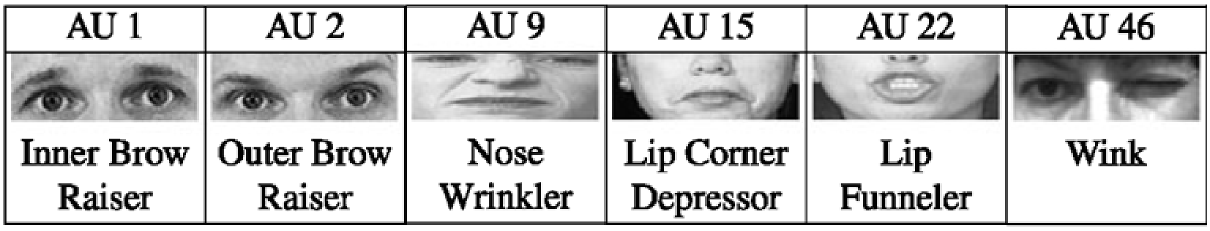


Figure 2.4: Examples of lower and upper face AUs in FACS. Reprinted from [8].

2.3.2 Recognition of FEs

An AFER system consists of four steps: face detection, face registration, feature extraction and expression recognition. As it is out of scope for this thesis, we will not consider the first two steps in detail. We focus instead on feature extraction and expression recognition.

2.3.2.1 Feature extraction

Extracted features can be divided into predesigned and learned. Predesigned features are hand-crafted to extract relevant information. Learned features are automatically learned from the training data. This is the case of deep learning approaches, which jointly learn the feature extraction and classficiation/regression weights. These categories are further divided into global and local, where global features extract information from the whole facial region, and local ones from specific regions of interest, usually corresponding to AUs. Features can also be split into static and dynamic, with static features describing a single frame or image and dynamic ones including temporal information.

Predesigned features can also be divided into appearance and geometrical. Appearance features use the intensity information of the image, while geometrical ones measure distances, deformations, curvatures and other geometric properties. This is not the case of learned features, for which the nature of the extracted information is usually unknown.

Geometric features describe faces through distances and shapes. These cannot be extracted from thermal data, since dull facial features difficult the precise localization of landmarks. Global geometric features, for both RGB and 3D modalities, usually describe the face deformation based on the location of specific fiducial points. For RGB, [160] uses the distance between fiducial points. The deformation parameters of a mesh model are used in [187, 109]. Similarly, for 3D data [205] use the distance between pairs of 3D landmarks, while [151] uses the deformation parameters of an EDM. Manifolds are used in [28] to describe the shape deformation of a fitted 3D mesh separately at each frame of a video sequence through Lipschitz embedding.

The use of 3D data allows generating 2D representations of facial geometry such as depth maps [17, 214]. In [180] *Local Binary Patterns* (LBP) are computed over different 2D representations, extracting histograms from them. Similarly, [81] uses SVD to extract the 4 principal components from LBP histograms. In [242], the geometry is described through the *Conformal Factor Image* (CFI) and *Mean Curvature Image* (MCI). [117] captures the mean curvatures at each location with *Differential Mean Curvature Maps* (DMCM), using HOG histograms to describe the resulting map.

In the dynamic case the goal is to describe how the face geometry changes over time. For RGB data, facial motions are estimated from color or intensity information, usually through *Optical flow* [229]. Other descriptors such as *Motion History Images* (MHI) and *Free-Form Deformations* (FFDs) are also used [107]. In the 3D case, much denser geometric data facilitates a global description of the facial motions. This is done either through deformation descriptors or motion vectors. [114] extracts and segments level curvatures, describing the deformation of each segment. FFDs are used in [179] to register the motion between contiguous frames, extracting features through a quad-tree decomposition. *Flow images* are extracted from contiguous frame pairs in [61], stacking and describing them with LBP-TOP.

In the case of local geometric feature extraction, deformations or motions in localized regions of the face are described. Because these regions are localized, it is difficult to geometrically describe their deformations in the RGB case (being restricted by the precision of the face registration step). As such, most techniques are dynamic for RGB data. In the case of 3D data, where much denser geometric information is available, the opposite happens.

In the static case, some 3D approaches describe the curvature at specific facial regions, either using primitives [222] or closed curves [139]. Others describe local deformations through SIFT descriptors [17] extracted from the depth map or HOG histograms extracted from DMCM feature maps [117]. In [76] the *Basic Facial Shape Components* (BFSC) of the neutral face are estimated from the expressive one, subtracting the expressive and neutral face depth maps at rectangular regions around the eyes and mouth.

Most dynamic descriptors in the geometric, local case have been developed for the RGB modality. These are either based on landmark displacements, coded with *Motion Units* [32, 31], or the deformation of certain facial components such as the mouth, eyebrows and eyes, coded with FAP [161, 10]. One exception is the work in [233] over 3D data, where an EDM locates a set of landmarks and a motion vector is extracted from each landmark and pair of frames.

Although geometrical features are effective for describing FEs, they fail to detect subtler characteristics like wrinkles, furrows or skin texture changes. Appearance features are more stable to noise, allowing for the detection of a more complete set of FEs, being particularly important for detecting microexpressions. These feature extraction techniques are applicable to both RGB and thermal modalities, but not to 3D data, which does not convey appearance information.

Global appearance features are based on standard feature descriptors extracted on the whole facial region. For RGB data, usually these descriptors are applied either over the whole facial patch or at each cell of a grid. Some examples include *Gabor filters* [122, 123], LBP [190, 184], *Pyramids of Histograms of Gradients* (PHOG) [198, 40], *Multi-Scale Dense SIFT* (MSDF) [198] and *Local Phase Quantization* (LPQ) [40]. In [138] a grid is deformed to match the face geometry, afterwards applying *Gabor filters* at each vertex. In [80] the facial region is divided by a grid, applying a bank of *Gabor filters* at each cell and radially encoding the mean intensity of each feature map. An approach called *Graph-Preserving Sparse Non-negative Matrix Factorization* (GSNMF) [254] finds the closest match to a set of base images and assigns its associated primary emotion. This approach is improved in [238], where *Projected Gradient Kernel Non-negative Matrix Factorization* (PGKNMF) is proposed.

In the case of thermal images, the dullness of the image makes it difficult to exploit the facial geometry. This means that, in the global case, the whole facial patch is used. The descriptors exploit the difference of temperature between regions. One of the first works [236] generated a series of *Binary Differential Images* (BDI), extracting the ratio of positive area divided by the mean ratio over the training samples. *2D Discrete Cosine Transform* (2D-DCT) is used in [106, 237] to decompose the frontalized face into cosine waves, from which an heuristic approach extracts features.

Dynamic global appearance descriptors are extensions to 3 dimensions of the already explained static global descriptors. For instance, *Local Binary Pattern histograms from Three Orthogonal Planes* (LBP-TOP) are used for RGB data [248]. LBP-TOP is an extension of LBP computed over three orthogonal planes at each bin of a 3D volume formed by stacking the frames. [198] uses a combination of LBP-TOP and *Local Phase Quantization from Three Orthogonal Planes* (LPQ-TOP), a descriptor similar to LBP-TOP but more robust to blur. LPQ-TOP is also used in [82], along with *Local Gabor Binary Patterns from Three Orthogonal Planes* (LGBP-TOP). In [129], a combination of HOG, SIFT and CNN are extracted at each frame. The first two are extracted from an overlapping grid, while the CNN extracts features from the whole facial patch. These are evaluated independently over time and embedded into Riemannian manifolds. For thermal images, [132] uses a combination of *Temperature Difference Histogram Features* (TDHFs) and *Thermal Statistic features* (StaFs). TDHFs consist of histograms extracted over the difference of thermal images. StaFs are a series of 5 basic statistical measures extracted from the same difference images.

Local appearance features are not used as frequently as global ones, since it requires previous knowledge to determine the regions of interest. In spite of that, some works use them for both RGB and thermal modalities. In the case of static features, [70] describes the appearance of grayscale frames by spreading an array of cells across the mouth and extracting the *mean intensity* from each. For thermal images, [207] generates eigenimages from each region of interest and uses the principal component values as features. In [84] *Gray Level Co-occurrence Maxrices* (GLCMs) are extracted from the interest regions and second-order statistics computed on them. GLCM encode texture information by representing the occurrence frequencies of pairs of pixel intensities at a given distance. As such, these are also applicable to the RGB case. In [224] a combination of StaFs, 2D-DCT and GLCM features is used, extracting both local and global information.

Few works consider dynamic local appearance features. The only one to our knowledge [130] describes thermal sequences by processing them with *SIFT flow* and chunking them into clips. Contiguous clip frames are wrapped and subtracted, spatially dividing the clip with a grid. The resulting cuboids with higher inter-frame variability for either radiance or flow are selected, extracting a *Bag of Words histogram* (BoW Hist.) from each.

Based on the observation that some AU are better detected using geometrical features and others using appearance ones, it was suggested that a combination of both might increase recognition performance [206, 158, 107]. Feature extraction methods combining geometry and appearance are more common for RGB, but it is also possible to combine RGB and 3D. Because 3D data is highly discriminative and robust to problems such as shadows and illumination changes, the benefits of combining it with RGB data are small. Nevertheless, some works have done so [164, 251, 252]. It should also be possible to extract features combining 3D and thermal information, but to the best of our knowledge it has not been attempted.

In the static case, [206] uses a combination of Multi-state models and edge detection to detect 18 different AUs on the upper and lower parts of the face in grayscale images. [35] uses both global geometry features and local appearance features, combining landmark distances and angles with HOG histograms centered at the barycenter of triangles specified by three landmarks. Other approaches use deformable models such as 3DMM [164] to combine 3D and intensity information. In [251, 252] SFAM describes the deformation of a set of distance-based, patch-based and grayscale appearance features encoded using LBP.

When analysing dynamic information, [35] uses RGB data to combine the landmark displacements between two frames with the change in intensity of pixels located at the barycenter defined by three landmarks.

Learned features are usually trained through a joint feature learning and classification pipeline. As such, these methods are explained in Section 2.3.2.2 along with learning. The resulting features usually cannot be classified as local or global. For instance, in the case of CNNs, multiple convolution and pooling layers may lead to higher-level features comprising the whole face, or to a pool of local features. This may happen implicitly, due to the complexity of the problem, or by design, due to the topology of the network. In other cases, this locality may be hand-crafted by restricting the input data. For instance,

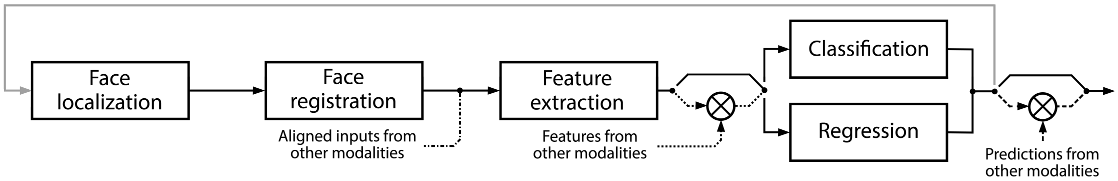

Face localization
Face registration
Aligned inputs from other modalities
Feature extraction
Features from other modalities
Classification
Regression
Predictions from other modalities

Figure 2.5: General execution pipeline for the different modality fusion approaches. The tensor product symbols represent the modality fusion strategy. Approach-specific components of the pipeline are represented with different line types: dotted corresponds to early fusion, dashed to late fusion, dashed-dotted to direct data fusion and gray to sequential fusion.

the method in [131], selects interest regions and describes each one with a *Deep Belief Network* (DBN). Each DBN is jointly trained with a weak classifier in a boosted approach.

2.3.2.2 FE classification and regression

FE recognition techniques are grouped into categorical and continuous depending on the target expressions [142]. In the categorical case there is a predefined set of expressions. Commonly for each one a classifier is trained, although other ensemble strategies could be applied. Some works detect the six primary expressions [122, 109, 187], while others detect expressions of pain, drowsiness and emotional attachment [11, 125, 136], or indices of psychiatric disorder [33, 108].

In the continuous case, FEs are represented as points in a continuous multidimensional space [243]. The advantages of this second approach are the ability to represent subtly different expressions, mixtures of primary expressions, and the ability to unsupervisedly define the expressions through clustering. Many continuous models are based on the activation-evaluation space. In [26], a *Recurrent Neural Network* (RNN) is trained to predict the real-valued position of an expression inside that space. In [154] the feature space is scaled according to the correlation between features and target dimensions, clustering the data and performing *Kernel regression*. In other cases like [64], which uses a RNN for classification, each quadrant is considered as a class, along with a fifth neutral target.

Expression recognition methods can also be grouped into static and dynamic. Static models evaluate each frame independently, using classification techniques such as *Bayesian Network Classifiers* (BNC) [32, 31, 187], *Neural Networks* (NN) [236, 206], *Support Vector Machines* (SVM) [122, 109, 17, 117, 207], SVM committees [84] and *Random Forests* (RF) [35]. In [80] *k-Nearest Neighbors* (kNN) is used to separately classify local patches, performing a dimensionality reduction of the outputs through PCA and LDA and classifying the resulting feature vector.

More recently, deep learning architectures have been used to jointly perform feature extraction and recognition. These approaches often use pre-training [86], an unsupervised layer-wise training step that allows for much larger, unlabeled datasets to be used. CNNs are used in [165, 167, 128, 100, 195]. [126] proposes *AU-aware Deep Networks* (AUDN), where a common convolutional plus pooling step extracts an over-complete representation of expression features, from which receptive fields map the relevant features for each expression. Each receptive field is fed to a DBN to obtain a non-linear feature representation, using an SVM to detect each expression independently. In [131] a two-step iterative process is used to train *Boosted Deep Belief Networks* (BDBN) where eacn DBN learns a non-linear feature from a face patch, jointly performing feature learning, selection and classifier training. [83] uses a *Deep Boltzmann Machine* (DBM) to detect FEs from thermal images. Regarding 3D data, [87] transforms the facial depth map into a gradient orientation map and performs classification using a CNN.

Dynamic models take into account features extracted independently from each frame to model the evolution of the expression over time. Dynamic Bayesian Networks such as *Hidden Markov Models* (HMM) [32, 10, 107, 114, 179, 230, 161] and *Variable-State Latent Conditional Random Fields* (VSL-CRF) [216] are used. Other techniques use RNN architectures such as *Long Short Term Memory* (LSTM) networks [229]. In other cases [209, 208], hand-crafted rules are used to evaluate the current frame expression against a reference frame. In [35] the transition probabilities between FEs given two frames are first evaluated with RF. The average of the transition probabilities from previous frames to the current one, and the probability for each expression given the individual frame are averaged to predict the final expression. Other approaches classify each frame independently (*eg.* with SVM classifiers [70]), using the prediction averages to determine the final FE.

In [187, 61] an intermediate approach is proposed where motion features between contiguous frames are extracted from interest regions, afterwards using static classification techniques. [129] encodes statistical information of frame-level features into Riemannian manifolds, and evaluates three approaches to classify the FEs: SVM, *Logistic regression* (LR) and *Partial Least Squares* (PLS).

More redently, dynamic, continuous models have also been considered. *Deep Bidirectional Long Short-Term Memory Recurrent Neural Networks* (DBLSTM-RNN) are used in [82]. While [182] uses static methods to make the initial affect pedictions at each time step, it uses particle filters to make the final prediction. This both reduces noise and performs modality fusion.

2.3.3 FE datasets

We group datasets' properties in three main categories, focusing on content, capture modality and participants. In the content category we refer to the type of content and labels the datasets provide. We signal intentionality of the FEs (posed or spontaneous), the labels (primary expressions, AUs or others where is the case) and if datasets contain still

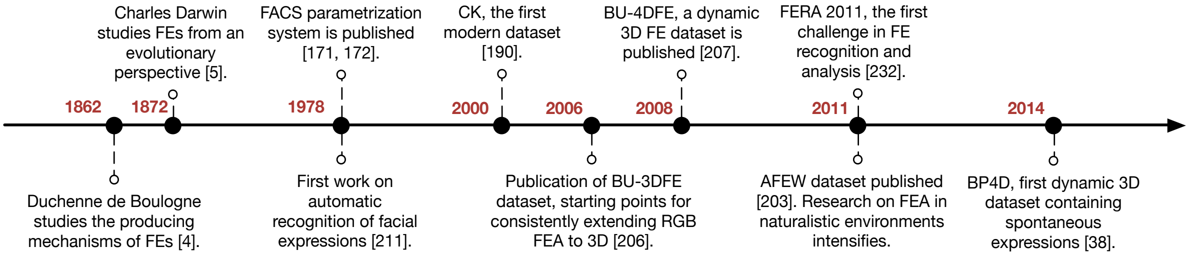

Charles Darwin studies FEs from an evolutionary perspective [5].
FACS parametrization system is published [171, 172].
CK, the first modern dataset [190].
BU-4DFE, a dynamic 3D FE dataset is published [207].
FERA 2011, the first challenge in FE recognition and analysis [232].
1862
1872
1978
2000
2006
2008
2011
2014
Duchenne de Boulogne studies the producing mechanisms of FEs [4].
First work on automatic recognition of facial expressions [211].
Publication of BU-3DFE dataset, starting points for consistently extending RGB FEA to 3D [206].
AFEW dataset published [203]. Research on FEA in naturalistic environments intensifies.
BP4D, first dynamic 3D dataset containing spontaneous expressions [38].

Figure 2.6: Historical evolution of AFER.

images or video sequences (static/dynamic). In the capture category we group datasets by the context in which data was captured (lab or non-lab) and diversity in perspective, illumination and occlusions. The last section compiles statistical data about participants, including age, gender and ethnic diversity. In Tables 2.1 and 2.2 the reader can refer to a complete list of RGB, 3D and Thermal datasets and their characteristics.

RGB. One of the first important datasets made public was the Cohn-Kanade (CK) [102], later extended into what was called the CK+ [135]. The first version is relatively small, consisting of posed primary FEs. It has limited gender, age and ethnic diversity and contains only frontal views with homogeneous illumination. In CK+, the number of posed samples was increased by 22% and spontaneous expressions were added. The MMI dataset was a major improvement [160]. It adds profile views of not only the primary expressions but most of the AU of the FACS system. It also introduced temporal labeling of onset, apex and offset. Multi-PIE [163] increases the variability by including a very large number of views at different angles and diverse illumination conditions. GEMEP-FERA is a subset of the emotion portrayal dataset GEMEP, specially annotated using FACS. CASME [232] is an example of a dataset containing microexpressions. A limitation of most RGB datasets is the lack of intensity labels. It is not the case of the DISFA dataset [145]. Participants were recorded while watching a video specially chosen for inducing emotional states and 12 AUs were coded for each video frame on a 0 (not present) to 5 (maximum intensity) scale[145].

While previous RGB datasets record FEs in controlled lab environments, *Acted Facial Expressions In The Wild Database* (AFEW) [41], *Affectiva-MIT Facial Expression Dataset* (AMFED) [146] and SEMAINE [148] contain faces in naturalistic environments. AFEW has 957 videos extracted from movies, labeled with six primary expressions and additional information about pose, age, and gender of multiple persons in a frame. AMFED contains spontaneous FEs recorded in natural settings over the Internet. Metadata consists of frame by frame AU labelling and self reporting of affective states. SEMAINE contains primitive FEs, FACS annotations, labels of cognitive states, laughs, nods and shakes during interactions with artificial agents.

3D. The most well known 3D datasets are BU-3DFE [234], Bosphorus [181] (still

Table 2.1: A non-comprehensive list of RGB FE datasets.

		RGB										
		CK+	**MPIE**	**JAFFE**	**MMI**	**RU_FACS**	**SEMAINE**	**CASME**	**DISFA**	**AFEW**	**SFEW**	**AMFED**
Content	Intention(Posed/Spontaneous)	*P*	*P*	*P*	*P*	*S*	*S*	*S*	*S*	*S*	*S*	*S*
	Label(Primary/AU/DA)	*P/AU*	*P*	*P*	$AU+T$	*P/AU*	*P/AU/DA*[1]	*P/AU*	$AU+I$	*P/*[2]	*P*	*P/AU/Smile*
	Temporality(Static/Dynamic)	*D*	*S*	*S*	*D*	*D*	*D*	*D*	*D*	*D*	*S*	*D*
Capture	Environment(Lab/Non-lab)	*L*	*L*	*L*	*L*	*L*	*L*	*L*	*L*	*N*	*N*	*N*
	Multiple Perspective	○	●	○	●	●	○	○	○	●	●	●
	Multiple Illumination	○	●	○	●	○	○	○	○	●	●	●
	Occlusions	○	●	○	○	●	○	○	○	●	●	○
Subjects	# of subjects	201	337	10	75	100	150	35	27	220	68	5268
	Ethnic Diverse	●	●	○	●	○	○	○	●	●	●	●
	Gender(Male/Female(%))	31/69	70/30	100/0	50/50	-	62/38	37/63	44/56	-	-	58/42
	Age	18-50	$\mu = 27.9$	-	19-62	18-30	22-60	$\mu = 22$	18-50	1-70	-	-

●= Yes, ○ = No, - = Not enough information. DA: Dimensional Affect, I = Intensity labelling, T = Temporal segments. [1] Other labels include Laughs, Nods, Epistemic states(e.g. Certain, Agreeing, Interested etc.) etc. Refer to original paper for details [148]. [2] Pose, Age, Gender. Refer to original paper for details [41].

Table 2.2: A non-comprehensive list of 3D and Thermal FE datasets.

		3D				**RGB+Thermal**			
		BU-3DFE	**BU-4DFE**	**Bosphorus**	**BP4D**	**IRIS**	**NIST**	**NVIE**	**KTFE**
Content	Intention(Posed/Spontaneous)	*P*	*P*	*P*	*S*	*P*	*P*	*S/P*	*S/P*
	Label(Primary/AU)	$P+I$	*P*	*P/AU*	*AU*	*P*	*P*	*P*	*P*
	Temporality(Static/Dynamic)	*S*	*D*	*S*	*D*	*S*	*S*	*D*	*D*
Capture	Environment(Lab/Non-lab)	*L*	*L*	*L*	*L*	*L*	*L*	*L*	*L*
	Multiple Perspective	●	●	-	●	●	●	●	●
	Multiple Illumination	○	○	○	○	●	●	●	●
	Occlusions	●	○	●	○	●	●	●	●
Subjects	# of subjects	100	101	105	41	30	90	215	26
	Ethnic Diverse	●	●	○	●	●	-	○	○
	Gender(Male/Female(%))	56/44	57/43	43/57	56/44	-	-	27/73	38/62
	Age	18-70	18-45	25-35	18-29	-	-	17-31	12-32

●= Yes, ○ = No, - = Not enough information, I = Intensity labelling.

images), BU-4DFE [235] (video) and BP4D [246] (video). In BU-3DFE, 6 expression from 100 different subjects are captured on four different intensity levels. Bosphorus has low ethnic diversity but it contains a much larger number of expressions, different head poses and deliberate occlusions. BU-4DFE is a high-resolution 3D dynamic FE dataset [235]. Video sequences, having 100 frames each, are captured from 101 subjects. It only contains primary expressions. BU-3DFE, BU-4DFE and Bosphorus all contain posed expressions. BP4D tries to address this issue with authentic emotion induction tasks [246]. Games, film clips and a cold pressor test for pain elicitation were used to obtain spontaneous FEs. Experienced FACS coders annotated the videos, which were double-checked by the subject's self-report, FACS analysis and human observer ratings [246].

Thermal. There are few thermal FE datasets, and all of them also include RGB data. The first ones, IRIS [4] and NIST/Equinox [6], consist of image pairs labeled with three posed primary emotions under various illuminations and head poses. Recently the number of labeled FEs has increased, also including image sequences. The *Natural Visible and Infrared facial Expression database* (NVIE) contains 215 subjects, each displaying six expressions, both spontaneous and posed [223]. The spontaneous expressions are triggered through audiovisual media, but not all of them are present for each subject. In the *Kotani Thermal Facial Emotion* (KTFE) dataset subjects display posed and spontaneous motions, also triggered through audiovisual media [153].

2.4 Historical evolution and current trends

2.4.1 Historical evolution

The first work on AFER was published in 1978 [199]. It was tracking the motion of landmarks in an image sequence. Mostly because of poor face detection and face registration algorithms and limited computational power, the subject received little attention throughout the next decade. The work of Mase and Pentland and Paul Ekman marked a revival of this research topic at the beginning of the nineties [99, 54]. The interested reader can refer to some influential surveys of these early works [177, 159, 62].

In 2000, the CK dataset was published marking the beginning of modern AFER [206]. While a large number of approaches aimed at detecting primary FEs or a limited set of FACS AUs [32, 10, 109, 122], others focused on a larger set of AUs [206, 160, 107]. Most of these early works used geometric representations, like vectors for describing the motion of the face [32], active contours for describing the shape of the mouth and eyebrows [10], or deformable 2D mesh models [109]. Others focused on appearance representations like Gabor filters [122], optical flow and LBPs [190] or combinations between the two [206]. The publication of the BU-3DFE dataset [234] was a starting point for consistently extending RGB FE recognition to 3D. While some of the methods require manual labelling of fiducial vertices during training and testing [222, 196, 204], others are fully automatic [117, 76, 214, 242]. Most use geometric representations of the 3D faces, like principal directions of surface curvatures to obtain robustness to head rotations [222], and normalized Euclidean distances between fiducial points in the 3D space [204]. Some encode global deformations of facial surface (depth differences between a basic facial shape component and an expressional shape component) [76] or local shape representations [180]. Most of them target primary expressions [222] but studies about AUs were published as well [183, 180].

In the first part of the decade static representations were the primary choice in both RGB [206, 122], 3D [222, 221, 204, 76, 17, 117] and thermal [84]. In later years various ways of dynamic representation were also explored like tracking geometrical deformations across frames in RGB [109, 160] and 3D [28, 114] or directly extracting features from RGB [107] and thermal frame sequences [153, 223].

Besides extended work on improving recognition of posed FEs and AUs, studies on expressions in ever more complex contexts were published. Works on spontaneous facial expression detection [137, 187, 211, 244], analysis of complex mental states [57], detection of fatigue [96], frustration [103], pain [124, 11, 125], severity of depression [75] and psychological distress [133] opened new territory in AFER research.

In summary, research in automatic AFER started at the end of the 1970's, but for more than a decade progress was slow mainly because of limitations of face detection and face registration algorithms and lack of sufficient computational power. From RGB static representations of posed FEs, approaches advanced towards dynamic representations and spontaneous expressions. In order to deal with challenges raised by large pose variations, diversity in illumination conditions and detection of subtle facial behaviour, alternative modalities like 3D and Thermal have been proposed. While most of the research focused on primary FEs and AUs, analysis of pain, fatigue, frustration or cognitive states paved the way to new applications in AFER.

In Figure 2.6 we present a timeline of the historical evolution of AFER. Next we will focus on two important recent trends.

2.4.2 AFER for detecting non-primary affective states

Most of AFER was used for predicting primary affective states of basic emotions, such as anger or happiness but FEs were also used for predicting non-primary affective states such as complex mental states [57], fatigue [96], frustration [103], pain [124, 11, 125], depression [75, 134], mood and personality traits [19, 178].

Approaches related to mood prediction from facial cues have pursued both descriptive (e.g., FACS) and judgmental approaches to affect. In a paper from 2009, Cohn et al. studied the difference between directly predicting depression from video by using a global geometrical representation (AAM), indirectly predicting depression from video by analyzing previously detected facial AUs and prediction depression from audio cues [33]. They concluded that specific AUs have higher predictive power for depression than others suggesting the advantage of using indirect representations for depression prediction. The AVEC, a challenge, is dedicated to dimensional prediction of affect (valance, arousal, dominance) and depression level prediction. The approaches dedicated to depression prediction are mainly using direct representations from video without detecting primitive FEs or AUs [228, 192, 189, 94]. They are based on local, dynamic representations of appearance (LBP-TOP or variants) for modelling continuous classification problems. Multimodality is central in such approaches either by applying early fusion [189] or late fusion [94] with audio representations.

As humans rely heavily on facial cues to make judgments about others, it was assumed that personality could be inferred from FEs as well. Usually studies about personality are based on the BigFive personality trait model which is organized along five factors: openness, conscientiousness, extraversion, agreeableness, and neuroticism. While there are works on detecting personality and mood from FEs only [19, 178] the dominant approach is to use multimodality either by combining acoustic with visual cues [19, 20] or physiological with visual cues [9]. Visual cues can refer to eye gaze [15, 18], frowning, head orientation, mouth fidgeting [15], primary FEs [19, 178] or characteristics of primary FEs like presence, frequency or duration [19]. In [19], Biel et al. use the detection of 6 primary FEs and of smile to build various measures of expression duration or frequency. They show that using FEs is achieving better results than more basic visual activity measures like gaze activity and overall motion of the head and body; however performance is considerably worse than when estimating personality from audio and especially from prosodic cues.

In summary, in recent years, the analysis of non-primary affective states mainly focused on predicting depression. For predicting levels of depression, local, dynamic representations of appearance were usually combined with acoustic representations [228, 192, 189, 94]. Studies of FEs for predicting personality traits had mixed conclusions until now. First, FEs were proven to correlate better than visual activity with personality traits [33]. Practically though, while many studies have showed improvements of prediction when combined with physiological or acoustic cues, FEs remain marginal in the study of personality trait prediction [19, 15, 20, 18].

2.4.3 AFER in naturalistic environments

Until recently AFER was mostly performed in controlled environments. The publication of two important naturalistic datasets, AMFED and AFEW marked an increasing interest in naturalistic environment analysis. AFEW, *Acted Facial Expressions in the Wild* dataset contains a collection of sequences from movies labelled for primitive FEs, pose, age and gender among others [41]. Additional data about context is extracted from subtitles for persons with hearing impairment. AMFED on the other hand, contains videos recording reactions to media content over the Internet. It mostly focuses on boosting research about how attitude to online media consumption can be predicted from facial reactions. Labels of AUs, primitive FEs, smiles, head movements and self reports about familiarity, liking and disposal to rewatch the content are provided.

FEs in naturalistic environments are unposed and typically of low to moderate intensity and may have multiple apexes (peaks in intensity). Large head pose and illumination diversity are common. Face detection and alignment is highly challenging in this context, but vital for eliminating rigid motion and head pose from facial expressions. Not surprisingly, in an analysis of errors in AU detection in three-person social interactions, [74] found that head yaw greater than 20 degrees was a prime source of error. Pixel intensity and skin color, by contrast, were relatively benign.

While approaches to FE detection in naturalistic environments using static representations exist [71, 42], dynamic representations are dominant [129, 193, 128, 216, 127, 100]. This follows the tendency in spontaneous FE recognition in controlled environments where dynamic representations improve the ability to distinguish between subtle expressions. In [128], spatio-temporal manifolds of low level features are modelled, [193] uses a maximum of a BoW (Bag of Words) pyramid over the whole sequence, [100] captures spatio-temporal information through autoencoders and [216] uses CRFs to model expression dynamics.

Some of the approaches use predesigned representations [71, 193, 42, 127, 43] while recent successful approaches learn the best representation [131, 100, 128] or combine predesigned and learned features [129]. Because of the need to detect subtle changes in the facial configuration, predesigned representations use appearance features extracted either globally or locally. Gehrig et al. in their analysis of the challenges of naturalistic environments use DCT, LBP and Gabor Filters [71], Sikka et al. use dense multi-scale SIFT BoWs, LPQ-TOP, HOG, PHOG and GIST to get additional information about context [193], Dhall et al. use LBP, HOG and PHOG in their baseline for the SFEW dataset (static images extracted from AFEW) [42] and LBP-TOP in their baseline for the EmotiW 2014 challenge [43], and Liu et al. use convolution filters for producing mid-level features [128].

Some representative approaches using learned representation were recently proposed [131, 100, 128, 129]. In [131], a BDBN framework for learning and selecting features is proposed. It is best suited for characterizing expression-related facial changes. [100] proposes a configuration obtained by late fusing spatio-temporal activity recognition with audio cues, a dictionary of features extracted from the mouth region and a deep neural network for FEs recognition. In [129], predesigned (HOG, SIFT) and learned (deep CNN features) representations are combined and different image set models are used to represent the video sequences on a Riemannian manifold. In the end, late fusion of classifiers based on different kernel methods (SVM, Logistic Regression, Partial Least Squares) and different modalities (audio and video) is conducted for final recognition results. Finally, [216] encodes dynamics with a *Variable-State Latent Conditional Random Fields* (VSL-CRF) model that automatically selects the optimal latent states and their intensity for each sequence and target class.

Most approaches presented target primitive FEs. Methods for recognizing other affective states have also been proposed, namely cognitive states like boredom, confusion, delight, concentration and frustration [22], positive and negative affect from groups of people [44] or liking/not-linking of online media for predicting buying behaviour for marketing purposes [147].

In summary, large head pose rotations and illumination changes make FE recognition in naturalistic environments particularly challenging. FEs are by definition spontaneous, usually have low intensity, can have multiple apexes and can be difficult to distinguish from facial displays of speech. Even more, multiple persons can express FEs simultaneously. Because of the subtleness of facial configurations most predesigned representations are dynamically extracting the appearance [71, 193, 127, 43]. Recently successful methods learn representations [131, 100, 128, 129] from sequences of frames. Most approaches target primitive FEs of affect, but others recognize cognitive states [22], postive and negative affect from groups of people [44] and liking/not-linking of online media for predicting buying behaviour for marketing purposes [147].

2.5 Conclusion

By looking at faces humans extract information about each other, such as age, gender, race, and how others feel and think. Building automatic AFER systems would have tremendous benefits. Despite significant advances, automatic AFER still faces many challenges like large head pose variations, changing illumination contexts and the distinction between facial display of affect and facial display caused by speech. Finally, even when one manages to build systems that can robustly recognize FEs in naturalistic environments, it still remains difficult to interpret their meaning. In this chapter we have focused in providing a general introduction into the broad field of AFER. We have started by discussing how affect can be inferred from FEs and its applications. An in-depth discussion about each step in a AFER pipeline followed, including a comprehensive taxonomy and many examples of techniques used on data captured with different video sensors (RGB, 3D, Thermal). Then, we have presented important recent evolutions in recognition of non-primary affective states and analysis of FEs in naturalistic environments. In the next chapter we will focus on improving facial action unit recognition by proposing a novel deep neural network architecture.

Chapter 3

Learning Facial Action Units with the Deep Structure Inference Network

Facial expressions are combinations of basic components called Action Units (AU). Recognizing AUs is key for general facial expression analysis. Recently, efforts in automatic AU recognition have been dedicated to learning combinations of local features and to exploiting correlations between AUs. We propose a deep neural architecture that tackles both problems by combining learned local and global features in its initial stages and replicating a message passing algorithm between classes similar to a graphical model inference approach in later stages. We show that by training the model end-to-end with increased supervision we improve state-of-the-art by 5.3% and 8.2% performance on BP4D and DISFA datasets, respectively.

3.1 Introduction

Facial expressions (FE) are important cues for recognizing non-verbal behaviour. The ability to automatically mine human intentions, attitudes or experiences has many applications like building socially aware systems [212, 39], improving e-learning [103], adapting game status according to player's emotions [13], and detecting deception during police interrogations [112].

The Facial Action Unit System (FACS) [55] is a descriptive coding scheme of FEs that focuses on what the face can do without assuming any cognitive or emotional value. Its basic components are called Action Units (AU) and they combine to form a complete representation of FEs. The reader can refer to Sec. 2.3.1 for a detailed introduction to these concepts.

AUs are patterns of muscular activation and the way they modify facial morphology

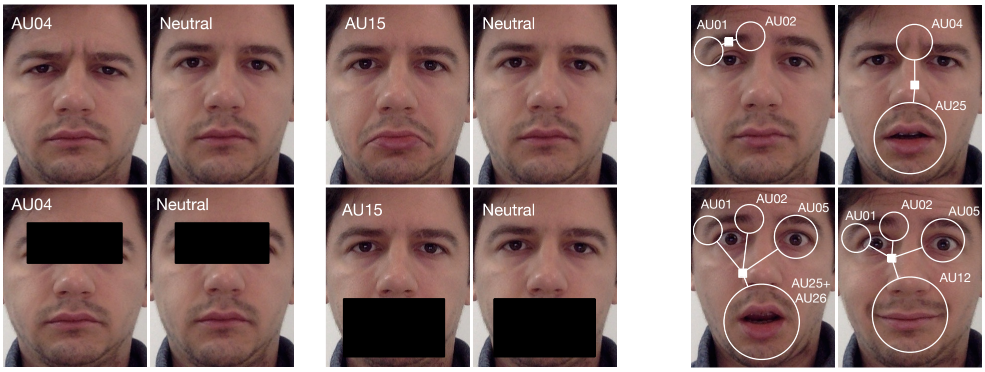


Figure 3.1: Patch and structure learning are key problems in AU recognition. (a) By masking a region an expressive face becomes indistinguishable from neutral. (b) Multiple, correlated AUs can be active at the same time.

is localized (Fig. 3.1). While initial AU recognition methods (like JPML [249] and APL [255]) were using shallow predefined representations, recent methods (like DRML [250], ROI [119] and GL [60]) applied deep learning to learn richer local features that capture facial morphology. Therefore one could predict specific AUs from informative face regions selected depending on the facial geometry. For instance, contrary to non-adaptive methods like DRML [250] and APL[255], ROI [119] and JPML [249] extract features around facial landmarks which are more robust with respect to non-rigid shape changes. Patch learning is challenging as the human face is highly articulated and different patches can contribute to either specific or groups of AUs. Learning the best patch combination together with learning specific features from each patch could be beneficial for AU recognition.

AU recognition is also multi-label. Several AUs can be active at the same time and certain AU combinations are more probable than others (Fig. 3.1). AU prediction performance could be improved by considering probabilistic dependencies. In deep learning approaches, correlations can be addressed implicitly in the fully connected layers (e.g. DRML [250], GL [60] and ROI [119]). However, structure is not learned explicitly and inference and sparsity are implicit by design. JPML [249] treats the problem by including pre-learned priors about AU correlations into their learning. Learning structured outputs has also been studied by using Graphical Models [249, 217, 58]. However, these models are not end-to-end trainable.

In this chapter, we claim that patch and the structure learning are key problems in dealing with AU recognition. We propose a deep neural network that tackles those problems in an integrated way through an incremental and end-to-end trainable approach. First, the model learns local and holistic representations exhaustively from facial patches. Then it captures structure between patches by predicting specific AUs. Finally, AU correlations are captured by a structure inference network that replicates message passing inference algorithms in a connectionist fashion. Tab. 3.1 compares some of the most important features of the proposed method to the state-of-the-art (specifically JPML

method	LRL	AP	PL	SL	EE	method	LRL	AP	PL	SL	EE
APL [255]	×	×	✓	×	×	GL [60]	×	×	✓	×	✓
JPML [249]	×	✓	✓	×	×	ROI [119]	✓	✓	✓	×	✓
DRML [250]	✓	✓	×	×	✓	DSIN (ours)	✓	✓	✓	✓	✓

Table 3.1: Features of our model and related work. LRL: local representation learning, AP: adaptive patch, PL: patch learning, SL: structured learning, EE: end-to-end.

[249], APL [255], DRML[250], GL[60] and ROI[119]). We show that by separately treating problems in different parts of the network and being able to optimize them jointly, we improve state-of-the-art by 5.3% and 8.2% performance on BP4D and DISFA datasets, respectively. Summarizing, our 2 main contributions are: 1) we propose a model that learns representation, patch and output structure end-to-end, and 2) we introduce a structure inference topology that replicates inference algorithm in probabilistic graphical models by using a recurrent neural network.

This chapter is organized as follows. Sec. 3.2 presents related work. Sec. 3.3 details the proposed model and Sec. 3.4 the results. Sec. 3.5 concludes the chapter.

3.2 Related Work

Related work is discussed in relation to patch learning or structure learning.

Patch Learning. Inspired by locally connected convolutional layers [202], Zhao et al. [250] proposed a regional connected convolutional layer that learns specific convolutional filters from sub-areas of the input. In [119], different CNNs are trained on different parts of the face merging features in an early fusion fashion with fully connected layers. Zhao et al. [249] performed patch selection and structure learning with shallow representations where patches for each AU were selected by group sparsity learning. Jaiswal et al. [95] used domain knowledge and facial geometry to pre-select a relevant image region for a particular AU, passing it to a convolutional and bi-directional Long Short-Term Memory (LSTM) neural network. Zhong et al. [255] proposed a multi-task sparse learning framework for learning common and specific discriminative patches for different expressions. Patch location was predefined and did not take into account facial geometry.

Structure Learning. Zhang et al. [245] proposed a multi-task approach to learn a common kernel representation that describes AU correlations. Elefteriadis et al. [58] adopted a latent variable Conditional Random Field (CRF) to jointly detect multiple AUs from predesigned features. While existing methods capture local pairwise AU dependencies, Wang et al. [226] proposed a restricted Boltzmann machine that captures higher-order AU interactions. Together with patch-learning, Zhao et al. [249] used positive and negative competitions among AUs to model a discriminative multi-label classifier. Walecki et al. [217] placed a CRF on top of deep representations learned by a CNN. Both components are trained iteratively to estimate AU intensity. Wu et al. [231] used a Restricted Boltzman Machine that captures joint probabilities between facial landmark locations and AUs. More recently, Benitez et al. [60] proposed a loss combining the recognition of isolated and groups of AUs.

3.3 Method

Let $\mathcal{D} = \{\mathbf{X}, \mathbf{Y}\}$ be a set of pairs of input images $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_M\}$ and output AU labels $\mathbf{Y} = \{\mathbf{y}_1, ..., \mathbf{y}_M\}$ with M number of instances. Each image $\mathbf{x}_i$ is composed of P patches $\{I_1, ..., I_P\}$ and output label $\mathbf{y}_i$ is a set of N AUs $\{y_1, ..., y_N\}$ taking a binary value $\{0, 1\}$. Several AU classes can be active for an observation as a multi-label problem. Predicting such output is challenging as a softmax function can not be applied on the set of outputs contrary to the standard mono-label/multi-class problems. In addition, using independent AU activation functions in losses like cross-entropy, ignores AU correlations. Including the ability to learn structure in the model design is thus relevant.

Two main ways of solving multi-label learning in AU recognition are either capturing correlations through fully-connected layers [250, 60, 119] or inferring structure through probabilistic graphical models (PGM) [249, 217, 58]. While the former can capture correlations between classes, this is not done explicitly. On the other hand, PGMs offer an explicit solution and their optimization is well studied. Unfortunately, placing classical PGMs on top of neural network predictions considerably lowers the capacity of the model to learn high order relationships since it is not end-to-end trainable. One solution is to replicate graphical model inference in a conectionist fashion which would make possible joint optimization. Jointly training CNNs and CRFs has been previously studied in different problems [253, 30, 38]. Following this trend, in this work we formulate AU recognition by a graphical model and implement it by neural networks, more specifically CNNs and recurrent neural network (RNN). This way, AU predictions from local regions along AU correlations are learned end-to-end.

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote a graph with vertices $\mathcal{V} = \mathbf{y}$ specifying AUs and edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ indicating the relationships between AUs. Given the Gibbs distribution we compute conditional probability $P(\mathbf{y}|\mathbf{x}, \Theta)$ as:

$$P(\mathbf{y}|\mathbf{x}, \Theta) = \frac{1}{Z(\mathbf{y}, \mathbf{x}, \Theta)} e^{-E(\mathbf{y}|\mathbf{x}, \Theta)}, \tag{3.1}$$

where Θ are model parameters, Z is a normalization function and E is an energy function. The model can be updated by introducing latent variables $\mathbf{p}$ as:

$$P(\mathbf{y}|\mathbf{x}, \Theta) = \sum_{\mathbf{p}} P(\mathbf{y}, \mathbf{p}|\mathbf{x}, \Theta), \tag{3.2}$$

where $\mathbf{p}$ is given as the output of CNN. The vertices and edges in the graph $\mathcal{G}$ can be updated as $\mathcal{V} = \mathbf{y} \cup \mathbf{p}$ and $\mathcal{E} = \mathcal{E}_y \cup \mathcal{E}_{py} \cup \mathcal{E}_p$. Although edges $\mathcal{E}_y$ can be defined by a prior knowledge taken from a given dataset, we use a fully connected graph independent to the dataset and assign a mutual gating strategy to control information passing through edges (more details in Sec. 3.3.3). We define $\mathcal{E}_{py}$ as edges between $\mathbf{p}$ and $\mathbf{y}$, and use a selective strategy to define edges in this set. Finally, edges $\mathcal{E}_p$ is an empty set, since in our model an independent CNN is trained on each image patch I_j and we do not assign any edge among $\mathbf{p}$. Given this assumption, probability distribution $P(\mathbf{y}, \mathbf{p}|\mathbf{x}, \Theta)$ is given by:

$$P(\mathbf{y}, \mathbf{p}|\mathbf{x}, \Theta) = P(\mathbf{y}|\mathbf{p}, \mathbf{x}, \Theta) \prod_k P(p_k|\mathbf{x}, \Theta). \tag{3.3}$$

As in CRF, energy function $E(.)$ is computed by unary and pairwise terms as:

$$E(\mathbf{y}, \mathbf{p}, \mathbf{x}, \Theta) = \sum_k \varphi_p(p_k, \mathbf{x}, \pi) + \sum_{(i,k)\in\mathcal{E}_{py}} \psi_{py}(y_i, p_k, \phi) + \sum_{(i,j)\in\mathcal{E}_y} \psi_y(y_i, y_j, \omega), \tag{3.4}$$

where $\varphi(.)$ is a unary term, $\psi_*(.)$ are pairwise terms and $\Theta = \pi \cup \phi \cup \omega$. Fig. 3.2 presents our Deep Structure Inference Network (DSIN). It consists of three components each designed to solve a term in Eq. 3.4. We refer to the initial part as *Patch Prediction* (PP), whose purpose is to exhaustively learn deep local representations from facial patches and produce local predictions. Then, the *Fusion* (F) module performs patch learning per AU. The final stage, *Structure Inference* (SI), refines AU prediction by capturing relationships between AUs. The DSIN is end-to-end trainable and CNN features can be trained based on gradients back-propagated from structure inference in a multi-task learning fashion.

3.3.1 Patch Prediction

Given image patches $\mathbf{x}$, unary terms $\varphi_p(\mathbf{p}, \mathbf{x}, \pi)$ provide AUs confidences for each patch which are defined as the log probability:

$$\varphi_p(\mathbf{p}, \mathbf{x}, \pi) = \log P(\mathbf{p}|\mathbf{x}, \pi). \tag{3.5}$$

Probability $P(\mathbf{p}|\mathbf{x}, \pi)$ is modeled by independent patch prediction functions $\{\Pi_i(I_i; \pi_i)\}_{i=1}^{P}$, where I_i is input image patch and π_i are function parameters. Each Π_i is a CNN computing N AUs probabilities through sigmoid function at last layer. P independent predictions are provided at this stage, each being a vector of AU predictions. Although image patches may overlap, we assume independence to let each network be expert at predicting AUs on local regions. By learning independent global representations and local representations, we can better capture facial morphology and address AU locality.

In Fig. 3.3(a) we detail the topology of the CNNs used for learning the patch prediction functions. Many complex topologies have been proposed in recent years and searching for

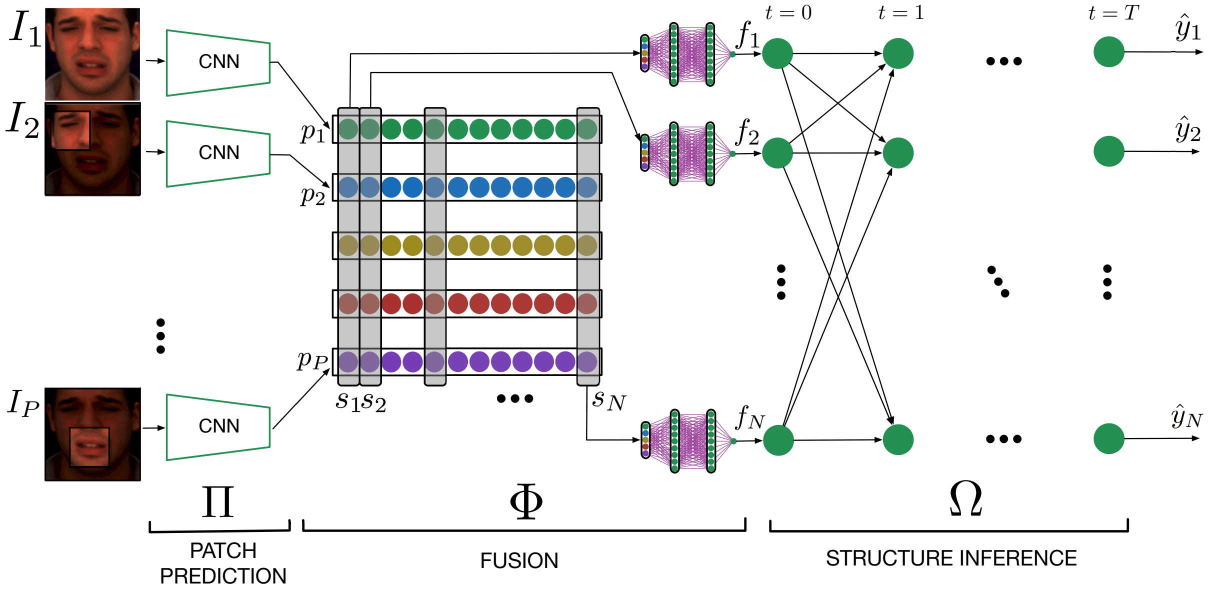


Figure 3.2: Deep Structure Inference Network (DSIN) learns independent AU predictions from global and local learned features. It refines each AU prediction by taking into account correlation to the other AUs. Each input image is cropped into a set of patches $\{I_i\}_{i=1}^{P}$ which is used for training an independent CNN for producing a probability vector p_i for N AUs (φ_p in Eq. 3.4). From s_j (the patch predictions for a specific AU) we learn a combination for producing a single AU prediction f_j (simplified ψ_{py} in Eq. 3.4). Final predictions y_j are computed by inferring structure among AUs through iterative message passing similar to inference in a probabilistic graph model (ψ_y in Eq. 3.4).

the best is out of the scope of this work. The chosen topology, a shallow network, follows the intuition behind well known models like VGG [194].

3.3.2 Fusion

Computational complexity to marginalize pairwise relationships in $\mathcal{E}_{py}$ is high. In our formulation, we simplify edges such that $\mathcal{E}_{py}$ becomes directed from nodes in $\mathbf{p}$ to nodes in $\mathbf{y}$. It means we omit mutual relationships among $\mathbf{p}$ and $\mathbf{y}$. Therefore, nodes in $\mathbf{y}$ are conditioned on the nodes in $\mathbf{p}$. However, we want each AU node in $\mathbf{y}$ to be conditioned on the same AU nodes in $\mathbf{p}$ from different patches. It means different patches can provide complementary information to predict target AU independent to other AUs. Finally, $\psi_{py}(\mathbf{y}, \mathbf{p}, \phi)$ is defined as the log probability of $P(\mathbf{y}|\mathbf{p}, \phi)$ which is modeled by a set of independent functions, so called fusion functions $\{\Phi_j(s_j; \phi_j)\}_{j=1}^{N}$, where $s_j \subset \mathbf{p}$ corresponds to the set of j-th AU predictions from all patches and ϕ_j is function parameters. We simply model each function Φ_j with 2 fully connected layers with 64 hidden units, each followed by a sigmoid layer, as shown in Fig. 3.3(b). We found 64 hidden units works well in practice while higher dimensionality does not bring any additional performance and quickly starts over-fitting. The output of each Φ_j is the predicted probability f_j for j-th

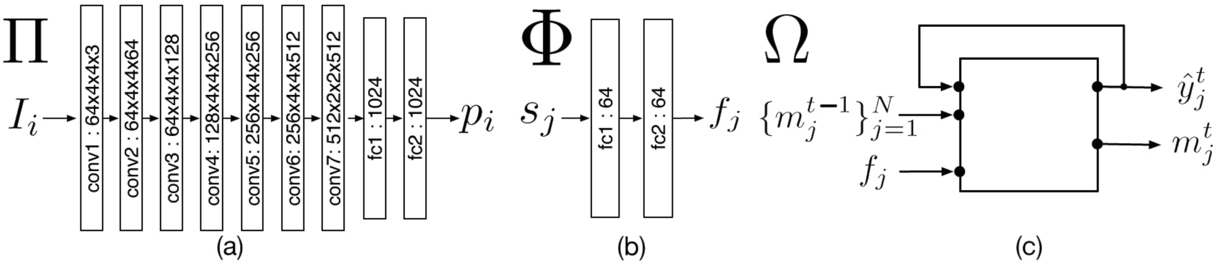


Figure 3.3: (a) Topology of patch prediction CNNs. Each convolutional block has stride 2 and batch normalization. Number of filters followed by the size of the kernel are marked. The last layers are fully-connected (FC) layers marked with the number of neurons. All neurons use ReLU activations. (b) Each fusion unit is a stack of 2 FC layers. (c) A structure inference unit. For better visualization, we just show the interface of the unit without the inner topology. See details in Sec. 3.3.3

.

AU.

3.3.3 Structure Inference

Up to now, we computed individual AU probabilities in a feed-forward neural network without taking AU relationships explicitly into account. The goal is to model pairwise terms ψ_y such that the whole process is end-to-end trainable in a compact way. Belief propagation and message passing between nodes is one of the well known algorithms for PGM inference. Inspired by [38], which proposes a connectionist implementation for action recognition, we build a *Structure Inference* (SI) module in the final part of DSIN.

The SI updates each AU prediction in an iterative manner by taking into account information from other AUs. The intuition behind this is that by passing information between predictions in an explicit way, we can capture AU correlations and improve predictions. The structure inference module is a collection of interconnected recurrent structure interference units (SIU) (see Fig. 3.3(c)). For each AU there is a dedicated SIU. We denote the computations done by SIU by a function Ω. Let $\{\Omega_j\}_{j=1}^N$ be the set of SIU functions $\Omega_j : \mathbb{R}^{N+2} \to \mathbb{R}^2$ where:

$$\hat{y}_j^t, m_j^t = \Omega_j(f_j, m_1^{t-1}, m_2^{t-1}, ..., m_N^{t-1}, \hat{y}_j^{t-1}; \omega_j). \tag{3.6}$$

At each iteration t, Ω_j takes as input the initial prediction f_j for its class, a set of incoming messages $\{m_j^{t-1}\}_{j=1}^N$ from the SIUs corresponding to the other classes and its own previous prediction $\hat{y}_j^{t-1}$. Each function Ω_j has two inline units: producing j-th AU prediction $\hat{y}_j^t$ and message m_j^t for next time step. In this way, predictions are improved iteratively by receiving information from other nodes. Computationally, we replicate this iterative message passing mechanism in the collection of SIUs with a recurrent neural network that shares function parameters Ω_j across all time steps. We show a SIU unit in Fig. 3.3(c).

A message unit basically corresponds to the distribution of the AU node. A message unit from a SIU is a parametrized function of the previous messages, the initial fused prediction and the previous prediction of the same SIU:

$$m_j^t = \sigma\left(\omega_j^m \left[\mu(m_1^{t-1}, ..., m_N^{t-1}), f_j, \hat{y}_j^{t-1}\right] + \beta_j^m\right), \tag{3.7}$$

where $\sigma(.)$ is the sigmoid function, $\mu(.)$ is the mean function, $\omega_j^m \in \mathbb{R}^3$ and $\beta_j^m \in \mathbb{R}$ are message function parameters. Messages between two nodes at each time step have a mutual relationship which can be controlled by a gating strategy. Therefore, a set of correction factors are computed as:

$$\chi_j^t = \sigma\left(\omega_j^g \left[\mu(m_1^t, ..., m_N^t), f_j, \hat{y}_j^{t-1}\right] + \beta_j^g\right), \tag{3.8}$$

where $\omega_j^g \in \mathbb{R}^3$ and $\beta_j^g \in \mathbb{R}$ are gating function parameters. Then, a message $m_{i \to j}^t$ that is passed from AU node i to j will be updated by the mutual factors of the gate between nodes i and j as:

$$\overline{m}_j^t = \mu(\chi_i^t, \chi_j^t) m_{i \to j}^t. \tag{3.9}$$

Finally, updated messages coming to the j-th node along with initial estimation f_j are used to produce output prediction $\hat{y}_j^t$ as:

$$\hat{y}_j^t = \sigma\left(\omega_j^y \left[\mu(\overline{m}_1^t, ..., \overline{m}_N^t), f_j\right] + \beta_j^y\right), \tag{3.10}$$

where $\omega_j^y \in \mathbb{R}^2$ and $\beta_j^y \in \mathbb{R}$ are prediction function parameters. By doing this, we are able to combine representation learning in function Π, patch learning in function Φ and structure inference in the Ω in a single end-to-end trainable model. We introduce our training strategy in Sec. 3.4.1.

3.4 Experimental Analysis

In the following, we describe experimental settings and results.

3.4.1 Experimental Setting

Data. We used BP4D [247] and DISFA [144] datasets. BP4D contains 2D and 3D videos of 41 young adults. It has 328 videos (8 videos for 41 participants) with 12 coded AUs, resulting in about 140k valid face images [247]. DISFA contains 27 adults (12 women and 15 men) with ages between 18 to 50 years and relative ethnic diversity. The data corpus consists of approximately 130k frames in total. AU intensity is coded for each video frame on a 0 (not present) to 5 (maximum intensity) ordinal scale. For our purpose we consider

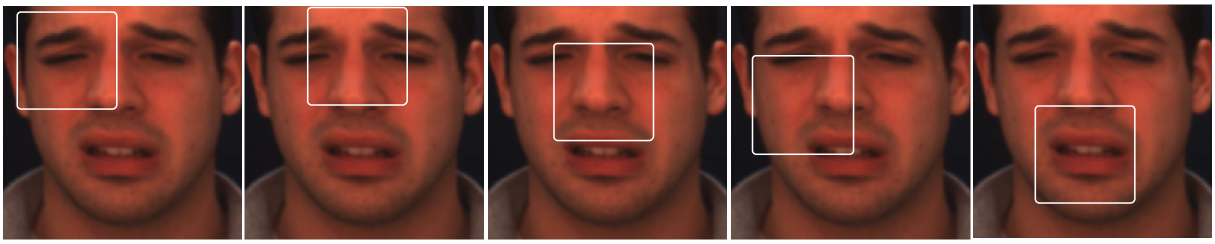


Figure 3.4: Each input image is aligned and cropped into 5 patches.

all labels with intensity greater than 3 as active and the rest as non-active. Both datasets are widely used in most recent AU recognition works.

Preprocessing. For each image, facial geometry is estimated using [104]. From all neutral faces we compute 3 reference anchors as the mean of the eyes and the mouth landmarks. Faces are resized to $224 \times 224 \times 3$ and a rigid transformation is applied for registering to the anchors, reducing variance to scale and rotation. We crop 5 patches of size $56 \times 56 \times 3$ around points defined by the detected landmarks (see Fig. 3.4). For reducing redundancy we ignore corresponding, symmetrical patches like the left eye and cheek.

Training. We incrementally train each part of DSIN before end-to-end model training. During training we use supervision on the patch prediction p, the fusion f and the structure inference outputs $\hat{y}$. On p we use a weighted L_2 loss denoted by $L_\Pi(p, y)$. The weights are inversely proportional to the ratio of positives in the total number of observations for each AU class in training. The weighting gives more importance to the minority classes in each training batch which ensures a more equal gradient update across classes and overall better performance. On the fusion and structure inference outputs we apply a binary cross-entropy loss (denoted by $L_\Phi(f, y)$ and $L_\Omega(\hat{y}, y)$). For the structure inference we include a regularization on the correction factors (denoted by χ in Eq. 3.8 and Eq. 3.9) to force sparsity in the message passing. Details of the training procedure are shown in Alg. 1. We use an Adam optimizer with learning rate of 0.001 and mini-batch size 64 with early stopping. Experimentally, we found the individual loss contributions $w_1 = 0.25$, $w_2 = 0.25$ and $w_3 = 0.5$ to work well in training. For both datasets we perform a subject exclusive 3-fold cross-validation. Similarly to [119], on DISFA we take the best CNNs trained for patch prediction on the BP4D and retrained fully connected layers for the new set of outputs. We fix the convolutional filters throughout the rest of the training.

Methods and metrics. We compare against CPM [241], APL [255], JPML [249], DRML [250], and ROI [119] state-of-the-art alternatives. We evaluate $F1$-frame score as $F1 = 2\frac{PR}{P+R}$, where $P = \frac{tp}{tp+fp}$, $R = \frac{tp}{tp+fn}$, tp being true positives, fn false negatives and fp false positives. All metrics are computed per AU and then averaged. Targeted AUs shown in Fig. 6.1.

Algorithm 1 Training procedure of DSIN.

```
Training data: $\{\{I\}_{i=1}^{P}, y\}$
Model parameters: patch prediction: $\{\pi_i\}_{i=1}^{P}$, fusion $\{\phi_i\}_{i=1}^{N}$, structure inference $\{\omega_i\}_{i=1}^{N}$

Step 0: random initialization around 0: $\pi, \phi, \omega \leftarrow \mathcal{N}(0, \sigma^2)$

Step 1: train patch prediction: $\pi_i \leftarrow \min_{\pi}(L_{\Pi}(\Pi_i(I_i; \pi_i)), y), \forall i \in \{1, ..., P\}$
Step 2: freeze patch prediction; train fusion: $\phi \leftarrow \min_{\phi} L_{\Phi}(\Phi(\Pi; \phi), y)$
Step 3: train patch prediction and fusion jointly:

    $\pi, \phi \leftarrow \min_{\pi,\phi}(L_{\Pi}(\Pi(I; \pi)), y) + L_{\Phi}(\Phi(\Pi; \phi), y))$
Step 4: freeze patch prediction and fusion; train structure inference:

    $\omega \leftarrow \min_{\omega} L_{\Omega}(\Omega(\Phi; \omega), y)$
Step 5. train all:
$\pi, \phi, \omega \leftarrow \min_{\pi,\phi,\omega}(w_1 L_{\Pi}(\Pi(I; \pi)), y) + w_2 L_{\Phi}(\Phi(\Pi; \phi), y) + w_3 L_{\Omega}(\Omega(\Phi; \omega), y))$
Output: optimized parameter: $\pi^{opt}$, $\phi^{opt}$, $\omega^{opt}$
```

3.4.2 Results

In the following, we explore the effect design decisions included in the DSIN followed by comparison against state-of-the-art alternatives in Sec. 3.4.2.2 and qualitative examples in Sec. 3.4.2.3.

3.4.2.1 Ablation Study.

We analyze DSIN design decisions in the following.

Class balancing. In both datasets, classes are strongly imbalanced. This can be harmful during training. To alleviate this, we use a weighted loss on patch prediction CNNs. Tab. 3.2 shows results with and without class balancing. This overall improves performance, especially on poorly represented classes. On BP4D the classes with ratios of positives in the total of samples lower than 30% are AU01, AU02, AU04, AU17, AU24. These are the classes that are improved the most. AUs like AU07 or AU12 have positives to total rations higher than 50%. Balancing can reduce performance on these classes.

Choice of prediction topology. In Tab. 3.2 we compare the proposed CNN for patch prediction (PP(face)) against VGG-16. The VGG-16 model used was trained for face recognition [162] and fine-tuned on our data for AU recognition. Our model shows superior performance.

Targeting subsets of AUs. We explore the effect of the considered target set on the overall prediction performance. In Tab. 3.2 we show prediction results from the right eye and from the mouth patches when training either on the full set of targets ([*method*]) or on individual targets ([*method*]^{ind}). When training on individual AUs the decision for the classifier is simpler. On the other hand any correlation information between classes that

	method	AU01	AU02	AU04	AU06	AU07	AU10	AU12	AU14	AU15	AU17	AU23	AU24	avg
	VGG(face)^{ft}	**35.2**	31.2	25.4	73.1	**72.1**	80.1	59.2	35.1	32.1	52.3	26.1	**36.2**	46.5
	PP(face)^{ncb}	35.1	**38.1**	**53.9**	**77.2**	70.7	**83.1**	**86.2**	**56.1**	**39.8**	**54.5**	**37.2**	31.4	**55.3**
PP	PP(right eye)^{ind}	**46.8**	**40.4**	45.3	68.3	69.2	-	-	-	-	-	-	-	-
	PP(mouth)^{ind}	-	-	-	-	-	78.6	82.0	54.2	38.6	54.7	[39.3]	**43.3**	-
	PP(right eye)	38.0	[37.7]	48.3	69.5	71.0	72.4	77.4	50.7	15.0	38.9	13.8	15.3	45.7
	PP(between eye)	41.7	34.8	45.9	64.9	65.5	72.1	73.9	54.9	19.7	33.9	13.9	7.0	44.0
	PP(mouth)	12.4	7.3	22.4	75.5	70.5	78.9	81.3	**66.2**	35.8	59.6	37.6	[42.8]	49.3
	PP(right cheek)	30.5	18.4	41.8	75.2	73.2	79.1	81.9	[61.9]	35.7	55.1	35.5	35.7	52.0
	PP(nose)	41.6	28.4	46.4	71.1	70.5	78.8	78.0	57.1	21.3	43.7	34.0	20.3	49.3
	PP(face)	43.8	37.5	[54.9]	**77.4**	[71.2]	[79.2]	**84.0**	56.6	[39.7]	[59.7]	39.2	39.5	[56.9]
	PP+F	[44.8]	35.8	**57.1**	[76.7]	**74.3**	**79.6**	[83.7]	56.6	**41.1**	**61.8**	**42.2**	40.1	**57.8**
DSIN	DSIN$_2^{ncf}$	46.7	34.1	**62.0**	76.5	**74.1**	[83.1]	84.9	60.9	36.0	57.1	**43.3**	36.1	57.9
	DSIN$_2$	47.7	36.5	55.6	76.3	[73.7]	80.1	85.0	64.0	[39.2]	60.6	[43.1]	39.9	58.2
	DSIN$_5$	[49.7]	36.3	57.3	**76.8**	73.4	81.6	84.5	[64.7]	38.5	[63.0]	39.0	37.3	58.5
	DSIN$_{10}$	**51.7**	[40.4]	56.0	76.1	73.5	79.9	[85.4]	62.7	37.3	62.9	38.6	[41.6]	[58.9]
	DSIN$_{10}^{tt}$	**51.7**	**41.6**	[58.1]	[76.6]	**74.1**	**85.5**	**87.4**	**72.6**	**40.4**	**66.5**	38.6	**46.9**	**61.7**

Table 3.2: Recognition results on BP4D. PP([patch]) stands for patch prediction on the indicated patch. F stands for the fusion and DSIN is the final model. We indicate the results when training on individual AUs with [method]^{ind}, fine tuning on the validation dataset of the decision threshold by DSIN^{tt}, number of iterations of the structure inference by DSIN$_T$ and training without correction factors as DSIN^{ncf}. VGG(face)^{ft} is a pre-trained VGG-16 [162] fine-tuned on BP4D. PP(face)^{ncb} is a patch prediction without class balancing. All results are obtained by 3-fold cross-validation on BP4D.

could be captured by the FC layers is ignored. In certain cases the individual prediction is superior to the exhaustive prediction. In the case of the right eye patch this is particularly true for AU01. But this is rather the exception. On average and across patches training on groups of AUs or on all AUs is beneficial as correlation information between classes is employed by the network in the fully connected layers. Additionally, predicting AU individually with independent nets would quickly increase the number of parameters with considerable effects on the training speed and final model performance.

Tab. 3.2 and 3.3 show AU recognition results on both datasets trained on patches. That proves the locality assumption. When training on the mouth the performance on the upper face AUs is greatly affected. Similarly, training on the eye affects the performance on the lower face AUs. This is expected as the patch prediction can only infer the other AUs from the ones visible in the patch.

Learning Local Representations. On average, face prediction compared to patch prediction performs better on the entire output set. However, when individual AUs are considered, this is no longer the case. For BP4D, the performance on AU15 and AU24 are considerably higher when predicting from the mouth patch than from the face (see Tab. 3.2). On DISFA the prediction from the whole face is the best on just 3 AUs (see Tab. 3.3). The nose patch is better for predicting AU06 and AU09, the mouth patch is better for AU12, AU25 and AU26, and the between eye patch for AU01.

Patch Learning. Tab. 3.2 and 3.3 show results of AU-wise fusion for BP4D and DISFA (PP+F). On both, patch learning through fusion is beneficial, but on DISFA

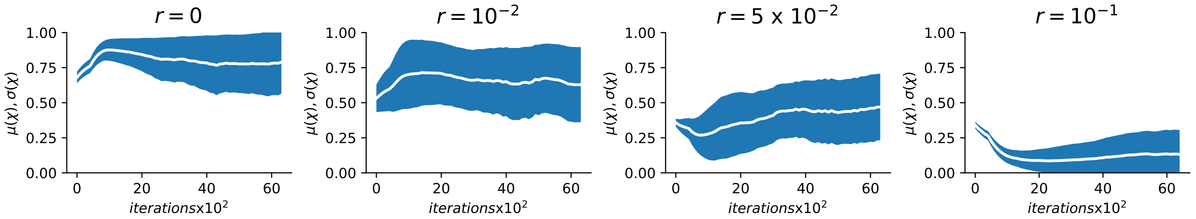


Figure 3.5: Different levels of regularization on the mean $\mu(\chi)$ (white line) and standard deviation $\sigma(\chi)$ (envelope) of the correction factors during training. Small regularization values force the correction factors to diverge faster. Increasing regularization collapses the correction factors hurting the message passing.

method	AU01	AU02	AU04	AU06	AU09	AU12	AU25	AU26	avg
PP(right eye)	27.2	15.4	58.8	8.0	18.2	53.6	73.3	9.1	33.0
PP(between eye)	34.6	13.2	59.7	15.4	21.1	50.9	72.9	8.5	34.5
PP(mouth)	7.5	6.4	44.6	28.5	23.9	**72.1**	87.5	[27.3]	37.2
PP(right cheek)	24.6	12.2	46.1	31.2	45.2	71.5	84.5	22.4	33.8
PP(nose)	21.9	19.1	52.0	**32.0**	**50.9**	66.5	76.6	8.9	41.0
PP(face)	29.8	[31.4]	64.6	26.8	21.3	70.1	87.0	20.3	43.9
PP+F	[40.1]	18.6	**70.8**	25.4	42.1	[71.8]	[88.8]	26.4	[48.0]
DSIN	**42.4**	**39.0**	[68.4]	[28.6]	[46.8]	70.8	**90.4**	**42.2**	**53.6**

Table 3.3: Results of DSIN on DISFA. PP([patch]) stands for patch prediction on the indicated patch. F stands for the fusion. DSIN is the final model. For DISFA we only show the DSIN with $T = 10$, the best performing on BP4D.

benefits are higher. This might be due to the fact that prediction results on DISFA are considerably more balanced across patches. Overall on BP4D the fusion improves results on almost all AUs compared to face prediction. This shows that even though the other patches perform worse on certain classes, there is structure to learn from their prediction that helps to improve performance. However, the fusion is not capable to replicate the result of the mouth prediction on AU14. On DISFA, in almost every case fusion gets close or higher to the best patch prediction. In both cases, fusion has greater problems in improving individual patches in cases where input predictions are already very noisy.

Structure Learning. Tab. 3.2 and 3.3 show results of the final DSIN model. For BP4D, we also perform a study of the number of iterations T considered for structure inference. Since parameters ω_j are shared across iterations, more iterations are beneficial to capture AU relationships in a fully connected graph with a large number of nodes (12 in our case). We also trained DSIN without correction factors (Eq. 3.9 is not applied in this case). Results are inferior compared with the same model with correction factors. In the case of DISFA, we only applied the structure inference with the best previously found $T = 10$ steps. Structure inference is beneficial in both cases. On BP4D, it considerably improves AU2 and AU14. For DISFA, the results are even more conclusive. Adding the structure inference brings more than 5% improvement over the fusion.

Correction factor regularization. Fig. 3.5 shows the effect of increasing regularization applied on the correction factors χ. Overall, regularizing χ does not bring significant

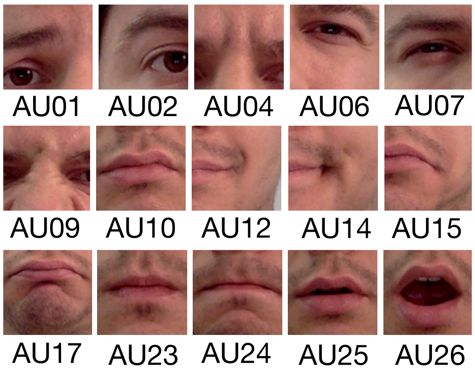


Figure 3.6: Facial Action Units targeted in this work.

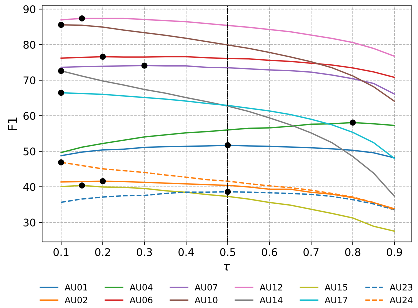


Figure 3.7: τ vs AU performance on BP4D validation set. Black circles denote best score.

method	AU01	AU02	AU04	AU06	AU07	AU10	AU12	AU14	AU15	AU17	AU23	AU24	AVG
JPML [249]	32.6	25.6	37.4	42.3	50.5	72.2	74.1	[65.7]	38.1	40.0	30.4	[42.3]	45.9
DRML [250]	36.4	**41.8**	43.0	55.0	67.0	66.3	65.8	54.1	33.2	48.0	31.7	30.0	48.3
CPM [241]	[43.4]	40.7	43.3	59.2	61.3	62.1	68.5	52.5	36.7	54.3	**39.5**	37.8	50.0
ROI [119]	36.2	31.6	43.4	**77.1**	[73.7]	[85.0]	[87.0]	62.6	**45.7**	58.0	38.3	37.4	56.4
DSIN	**51.7**	40.4	[56.0]	76.1	73.5	79.9	85.4	62.7	37.3	[62.9]	[38.8]	41.6	[58.9]
DSIN^{tt}	**51.7**	[41.6]	**58.1**	[76.6]	**74.1**	**85.5**	**87.4**	**72.6**	[40.4]	**66.5**	38.6	**46.9**	**61.7**

Table 3.4: AU recognition results on BP4D. Best results are shown in bold. Second best results are shown in brackets. For the proposed model we show an additional set of results (DSIN$_{tt}$) obtained when the decision threshold is tuned per AU.

benefits. When comparing $r = 10^{-2}$ with no regularization the differences are minimal. The network has the ability to learn sparse message passing by itself without regularization. Still, small values of r lead to faster divergence of χ and faster convergence of the network. The difference in performance is not significant. On the other hand values of $r > 5 \times 10^{-2}$ negatively affect performance as most of χ get closer to 0 and no messages are passed anymore. For these reasons, we keep $r = 5 \times 10^{-3}$.

Threshold Tuning. Prediction value per AU takes values between 0 and 1. In all results, we compute the performance by binarizing the output with respect to threshold $\tau = 0.5$. Although class balancing as a weighted loss is beneficiary, it does not totally solve data imbalance. Fig. 3.7 shows performance in terms of τ for validation set of BP4D. As shown, a threshold $\tau = 0.5$ is not an ideal value. For most classes $\tau \in [0.1, 0.3]$ is preferable. Exception is AU04. Tables 3.2 and 3.3 show the performance of the proposed model after tuning τ per class (DSIN^{tt}). This way 2.8% and 3.1% of performance is gained on BP4D and DISFA, respectively.

3.4.2.2 Comparison with state-of-the-art.

Tables 6.2 and 6.3 show how our model compares against the state-of-the-art related methods on BP4D and DISFA, respectively. DSIN and ROI are the best performing in both datasets. Both methods learn deep local representations and patch combinations

method	AU01	AU02	AU04	AU06	AU09	AU12	AU25	AU26	avg
APL[255]	11.4	12.0	30.1	12.4	10.1	65.9	21.4	26.0	23.8
DRML [250]	17.3	17.7	37.4	29.0	10.7	37.7	38.5	20.1	26.7
ROI [119]	41.5	26.4	66.4	**50.7**	8.5	**89.3**	88.9	15.6	48.5
DSIN	[42.4]	[39.0]	[68.4]	28.6	[46.8]	70.8	[90.4]	[42.2]	[53.6]
DSIN^{tt}	**46.9**	**42.5**	**68.8**	[32.0]	**51.8**	[73.1]	**91.9**	**46.6**	**56.7**

Table 3.5: AU recognition results on DISFA. Best results are shown in bold. Second best results are shown in brackets.

end-to-end. The worst performing methods, JPML on BP4D and APL on DISFA, use predefined features and are not end-to-end trained. Comparing DSIN and ROI with DRML one can observe the advantage in learning independent local representation. Both ROI and our model learn independent local representations, while DRML disentangles the representation learning in just one layer of their network. Interestingly though, there is also an exception. On BP4D, CPM performs slightly better than DRML even though it is not a deep learning method. When comparing our proposed model with ROI on BP4D our CNN trained just on face without class balancing has inferior results. When we include class balancing and patch learning our topology improves performance, further enhanced by structure inference and end-to-end final training. In the case of DISFA, single CNN trained on the whole face with class balancing has a performance of 43.9, being 4.6% lower than ROI. When we add patch prediction fusion (PP+F) we get just 0.5% lower than ROI while the addition of the structure inference and threshold tuning improves ROI performance. Finally, DSIN shows the best results on both datasets. For BP4D, from the 12 AUs target it performs best on 5 and second best on additional 5. In the case of DISFA the improvement over ROI is greater, DSIN performing best in all but one AU. Overall, we obtain 5.3% absolute and 9.4% relative performance improvement on BP4D and 8.2% absolute and 16.9% relative performance improvement on DISFA, respectively.

3.4.2.3 Qualitative results.

Fig. 3.8(a) shows examples of how structure inference tends to correct predictions following AU correlations. We show the magnitude of AU correlations on BP4D in Fig. 3.8(b). In the first 3 column examples, AU06 and AU07 are not correctly classified by the fusion model (middle row). Both these AUs are highly correlated with already detected AUs like AU10, AU12 and AU14. Such correlation could be captured by SI (bottom row). The rightmost example shows how AU17, a false positive, is corrected. As shown in Fig. 3.8(b), AU17 is negatively correlated with AU4, which was already detected. In Fig. 3.8(c) we show a class activation map [188] for AU24 of the patch prediction (left) vs. the DSIN (right). Contrary to very localized patch prediction, the attention on right expands to a larger area of the face where possible correlated AUs might exist.

In Fig. 3.9 and 3.10 we show more examples of predictions produced by DSIN. For

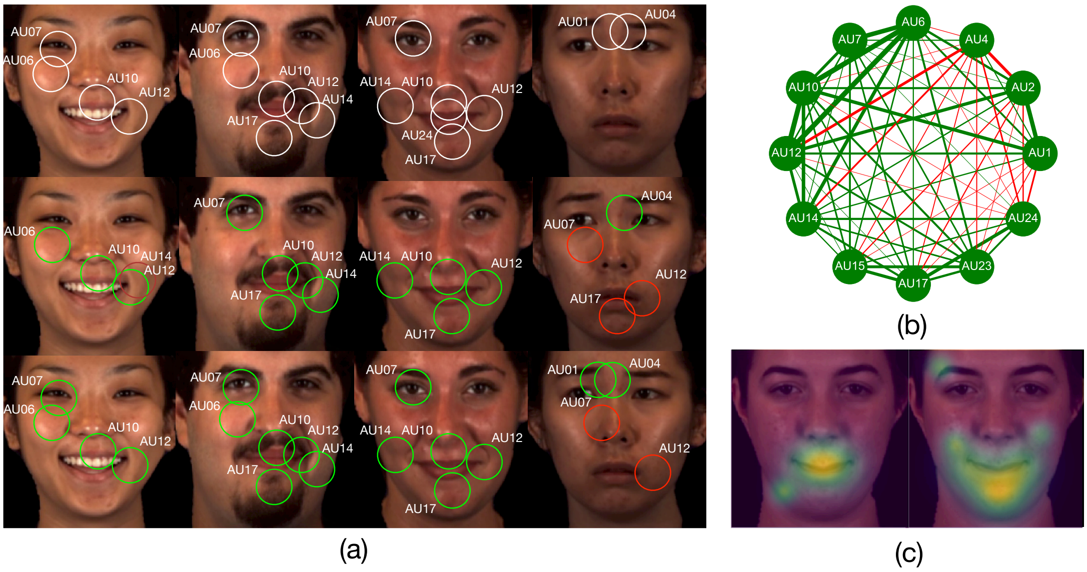


Figure 3.8: (a) Examples of AU predictions: ground-truth (top), fusion module (middle) and structure inference (bottom) prediction (●: true positive, ●: false positive). (b) AUs correlation in BP4D (●: positive, ●: negative). Line thickness is proportional with correlation magnitude. (c) Class activation map for AU24 that shows the discriminative regions of simple patch prediction (left) and DSIN (right). Best seen in color.

each example, we first show the real AU values (GT), the values predicted by each of the patch prediction CNNs (PP([patch])), the predictions of the fusion (F) and the final prediction that includes the Structure Inference (DSIN). We discuss each of them in the following.

Patch Prediction In general the patch predictions have a tendency to be more accurate for predicting the AUs visible in the patch and less accurate on the other AUs. For example, in the case of DISFA, the right eye patch prediction usually produces false positives on the lower face AUs like AU12 (see Fig. 3.9(c,d,h)) and AU25 (see Fig. 3.9(c,g,h)). The same tendency can be observed on the between eye patch prediction (false positives for AU25 on Fig. 3.9(b) and AU12 on Fig. 3.9(e,g)) and nose patch prediction (false positives for AU25 Fig. 3.9(c,g) and AU12 Fig. 3.9(g)).

In the case of BP4D, the right eye patch prediction produces false positives for AU10 (Fig. 3.10(c,e,h)), AU12 (Fig. 3.10(d,f)) and AU17 (Fig. 3.10(e)). Similarly the between eye patch prediction is falsely predicts lower-face AUs like AU10 (Fig. 3.10(f)), AU12 (Fig. 3.10(f)) and AU17 (Fig. 3.10(h)). In the case of the nose or mouth two clear examples of false positives are for AU7 in Fig. 3.10(b). It might be interesting to the reader to compare these mouth and nose patch predictions of AU7 in Fig. 3.10(b) with false positives on the same AU, this time predicted on the right eye (Fig. 3.10(e)) or the between eye patch (Fig. 3.10(d)). In the first case the eye is wide opened but not visible to the mouth or nose patch. In the second case, AU7 is visible to the patch but the eyes are more closed which might have confused the model.

Some additional comments can be made about the limitations of patch prediction. In the case of DISFA, an interesting example can be seen in Fig. 3.9(e) where AU25 (Lips Apart) is not predicted by any of the patches except the between eye where it is not even visible (AU25 is highly correlated with AU12). This is a confusing example, because the subject has his tongue between the lips. In the case of the face prediction, some other examples of false negatives can be seen in Fig. 3.9(c,g,h,i). A particularly interesting example is shown in Fig. 3.9(g) where none of the patches is able to predict AU4, despite being quite visible. Also on DISFA, in Fig. 3.9(j), the mouth prediction is not capable to detect AU12, the Lip Corner Puller.

On BP4D as well, we can refer to several examples where the patch prediction fails to detect visible AUs. For example, in Fig. 3.10(b) the face prediction, fails to detect AU2 or in Fig. 3.10(c), the mouth prediction fails to detect AU12, AU15, AU23 and AU24. There are also cases like Fig.3.10(e) where even-though marked as active in the ground-truth, AU1 and AU4 have low intensity and are not really visible. Some of the patches (like the face) have problems detecting them.

On DISFA, the ground-truth is an intensity label from 0 (neutral) to 5 (most intense). In order to keep results comparable with the related work, we have considered the low intensity levels (below 3) as neutral samples and all the other as active. This can be misleading to the model. Some examples can be found in Fig. 3.9(g,i). In both these cases, AU12 (Lip Corner Puller) has low intensity and not marked in our binarised ground-truth. Nevertheless the mouth patch prediction detects it. The same is true for AU25 (Lips Apart) for Fig. 3.9(i). Another interesting example is Fig. 3.9(h). The AU23 (Lip Tightener) is not targeted in the case of DISFA but nevertheless it is present in this example.

Fusion In general in the fusion predictions, two general tendencies can be observed. The fusion gives more weight to AUs that have been independently predicted by most patches and also to patch predictions that have higher probability of being correct.

In the case of DISFA, the majority vote can be observed on Fig. 3.9(a) for AU12 and AU25, on Fig. 3.9(b) for AU4, in Fig. 3.9(c) for AU1, AU2, AU12, AU25, just to name a few. In the case of BP4D, some examples can be found in Fig. 3.10(a) where AU7 is predicted by the fusion but AU2 not, or in Fig. 3.10(b) where AU10 is predicted but AU12 and AU14 not. Counter-examples can be found as well, for example in Fig. 3.10(e) AU10 is falsely predicted by several patches, some of them expert in this AU (e.g mouth) but nevertheless the fusion manages to correct it.

The fusion might also take into account if the probability of a patch to produce accurate predictions for a specific AU is high. For example, in the case of DISFA the between eye patch prediction of AU4 in in Fig. 3.9(a) is propagated to the fusion but the prediction of AU25 in Fig. 3.9(b) is not. This might happen because the between eye patch prediction has more confidence when predicting AU4 than AU25. A similar example can be found in

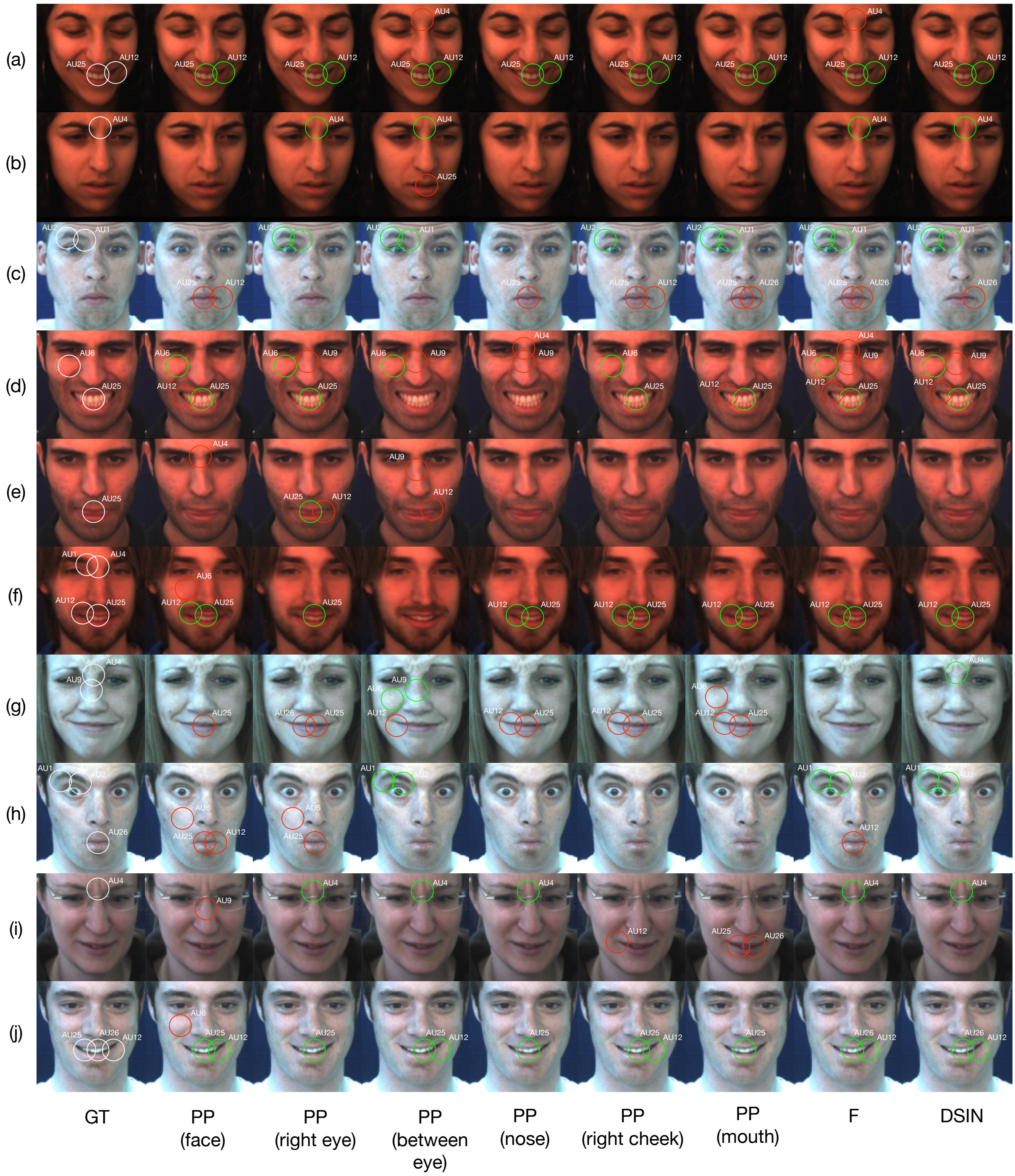


Figure 3.9: Examples of predictions of DSIN on DISFA. GT: ground truth (y), PP: patch prediction ($\{p_i\}_{i=1}^{N}$), F: fusion (f), DSIN: final predictions ($\hat{y}$), ●: true positive, ●: false positive.

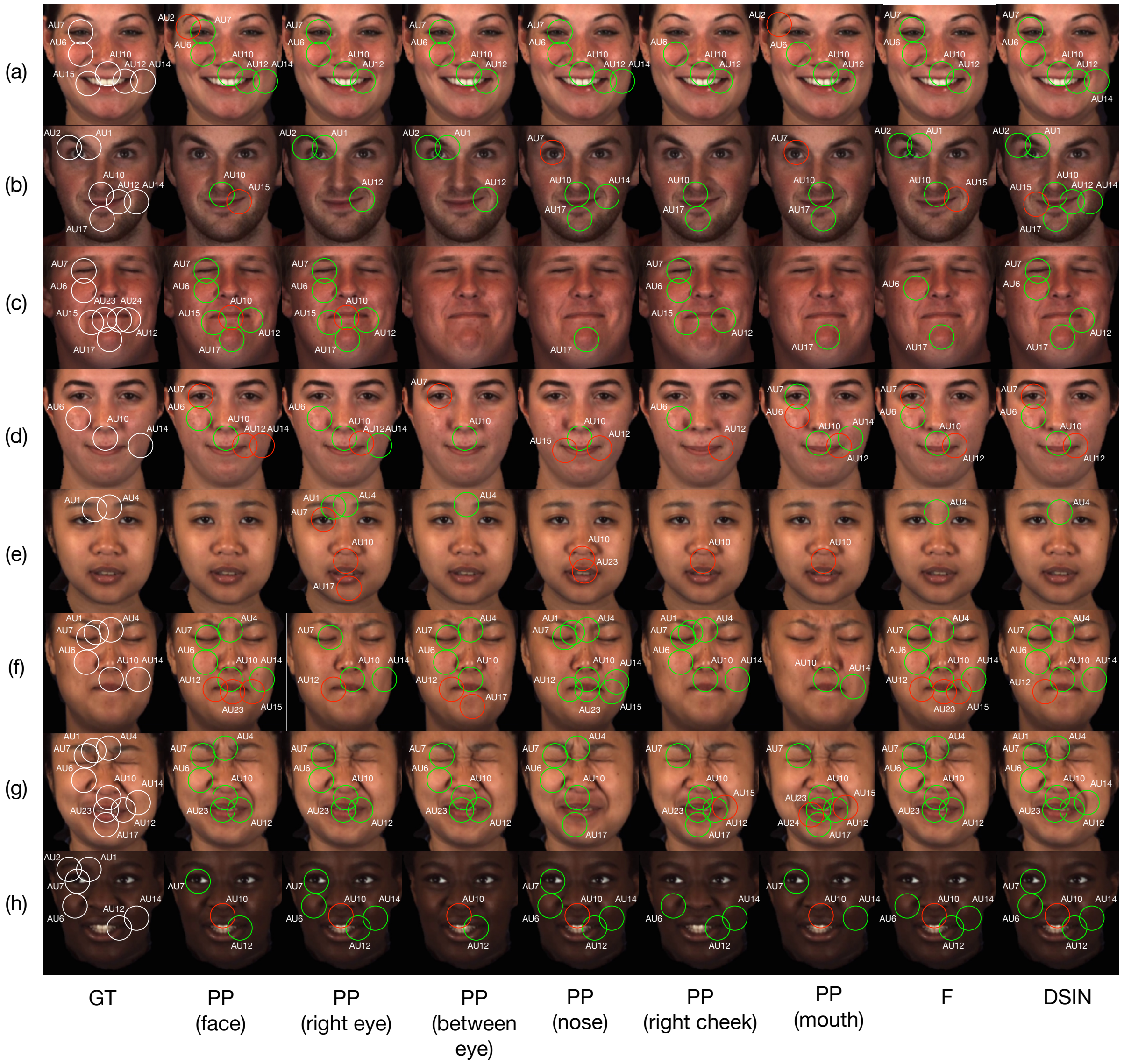
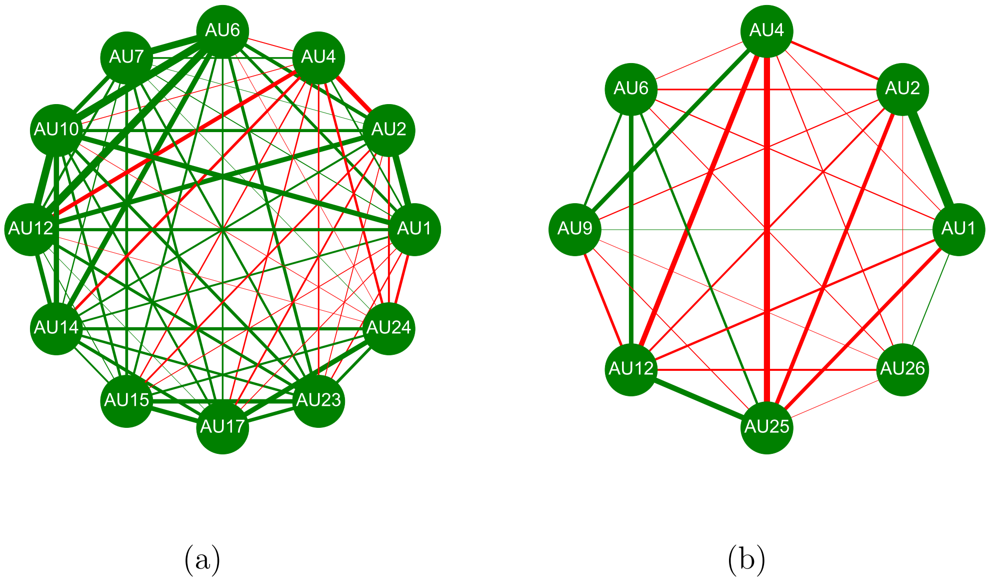


Figure 3.10: Examples of predictions of DSIN on BP4D. GT: ground truth (y), PP: patch prediction ($\{p_i\}_{i=1}^{N}$), F: fusion (f), DSIN: final predictions ($\hat{y}$), 🟢: true positive, 🔴: false positive.

Fig. 3.9(d). There the nose patch prediction falsely detects AU4 and AU9. This patch usually has high confidence on these AUs and thus the fusion takes it into account. In the case of BP4D a suggestive example can be seen in Fig.3.10(b) where AU1 and AU2 are predicted by two different highly confident patches (right eye and between eye) and AU7 by two different low confidence patches (mouth and nose). The AU1 and AU2 get propagated by the fusion while AU7 not.

Structure Inference Finally, we present DSIN predictions. For helping the reader we include the AU correlations calculated on the BP4D and DISFA train dataset (Fig.3.11). Due to the stimulus used to elicit affect during the data collection process, these correlations might slightly vary between datasets. Nevertheless, on these datasets there is structure



(a)

(b)

Figure 3.11: AU correlations on BP4D (a) and DISFA (b). 🟢: positive correlation, 🔴: negative correlation. Thicker the line, higher the correlation.

consistency on the shared AU subset (AU1, AU2, AU4, AU6 and AU12). AU1 and AU2 or AU6 and AU12 are two examples of AU pairs that strongly correlate on both datasets and AU4 and AU12 is a good example of negatively correlated AU pair.

In general the structure inference manages to correct some false positives or negatives following correlations between classes shown in Fig. 3.11. In the case of DISFA some suggestive examples can be found in Fig. 3.9 (a) and Fig. 3.9(c) where AU4 and AU25 respectively are correctly predicted as non-active. Both these AUs are negatively correlated to already detected AUs. In the case of BP4D, in Fig. 3.10(a) AU14 is added to the final prediction, in the case of Fig. 3.10(c) AU7 and AU12. In general, if one looks at the real predicted confidence number per AU (not displayed here), when the prediction is ambiguous (around 0.5) the structure inference uses the other classes to take a better decision. This is less the case for samples were the DSIN has high confidence (close to 0 or 1).

3.5 Conclusion

We proposed the Deep Structured Inference Network, designed to deal with both patch and structure learning for AU recognition. DSIN first learns independent local and global representations and corresponding predictions. Then, it learns relationships between predictions per AU through stacked fully connected layers. Finally, inspired by inference algorithms in graphical models, DSIN replicates a message passing mechanism in a connectionist fashion. This adds the ability to capture correlations in the output space.

The model is end-to-end trainable and improves state-of-the-art results by 5.3% and 8.2% performance on BP4D and DISFA datasets, respectively.

In the next two chapters (Chapter 4 and Chapter 5) we will make a detour from the problem of facial action unit recognition. We will first define a novel generic theoretical framework for analysing deep neural network behaviour during optimization. This new tool opens the “black-box” of neural networks, giving important cues about how they learn and most importantly when do they start memorizing. We will use this new insights in Chapter 6 for training DSIN with improved control, performance and without the need of validation data.

Chapter 4

A Novel Framework for Analysing Deep Neural Networks

In this chapter we introduce a novel generic theoretical framework for analysing deep neural networks. The basic idea is to project a deep network into a topological space and then use established methods in Algebraic Topology to count non-trivial patterns in this space. We will show in next chapters that this new perspective sheds light on fundamental aspects in learning theory, namely the overfitting problem (or learning vs memorization) and the generalization gap.

4.1 Preliminaries

Let $G = (V, E)$ be an undirected graph that consists of a pair of finite sets (V, E), where the elements v_i of V are called the vertices, and the elements e_{ij} of E are called the edges.

A *n-clique*, $\sigma \subset G$ is a subset of $n + 1$ vertices of G that are connected to each other, and n is the degree of the clique. Any subgraph $\phi \subset \sigma$ of a n-clique is itself a clique of lower degree and is called a *face* of that clique. A clique that is not a face of a clique of larger degree, is called a *maximal clique*.

The *clique complex* of a graph G is the collection of all its cliques, $S(G) = \{S_0(G), S_1(G), \ldots S_n(G)\}$, where $S_k(G)$ is the set of all $(k + 1)$-cliques in G [197]. A clique complex defines a topological space.

The *geometric realization* of this topological space is an object. A few examples of such objects are shown in Figure 4.1(a). As seen in this figure, a 0-clique is realized by a single node, a 1-clique by a line segment (i.e. edge) connecting two nodes, a 2-clique is the filled triangle that connects three nodes (i.e. the three edges of the three nodes define a filled triangle, that is, a 2D face), etc.

The geometric realizations of cliques in a clique complex intersect on common faces forming a well-defined object as shown in Fig. 4.1(b).

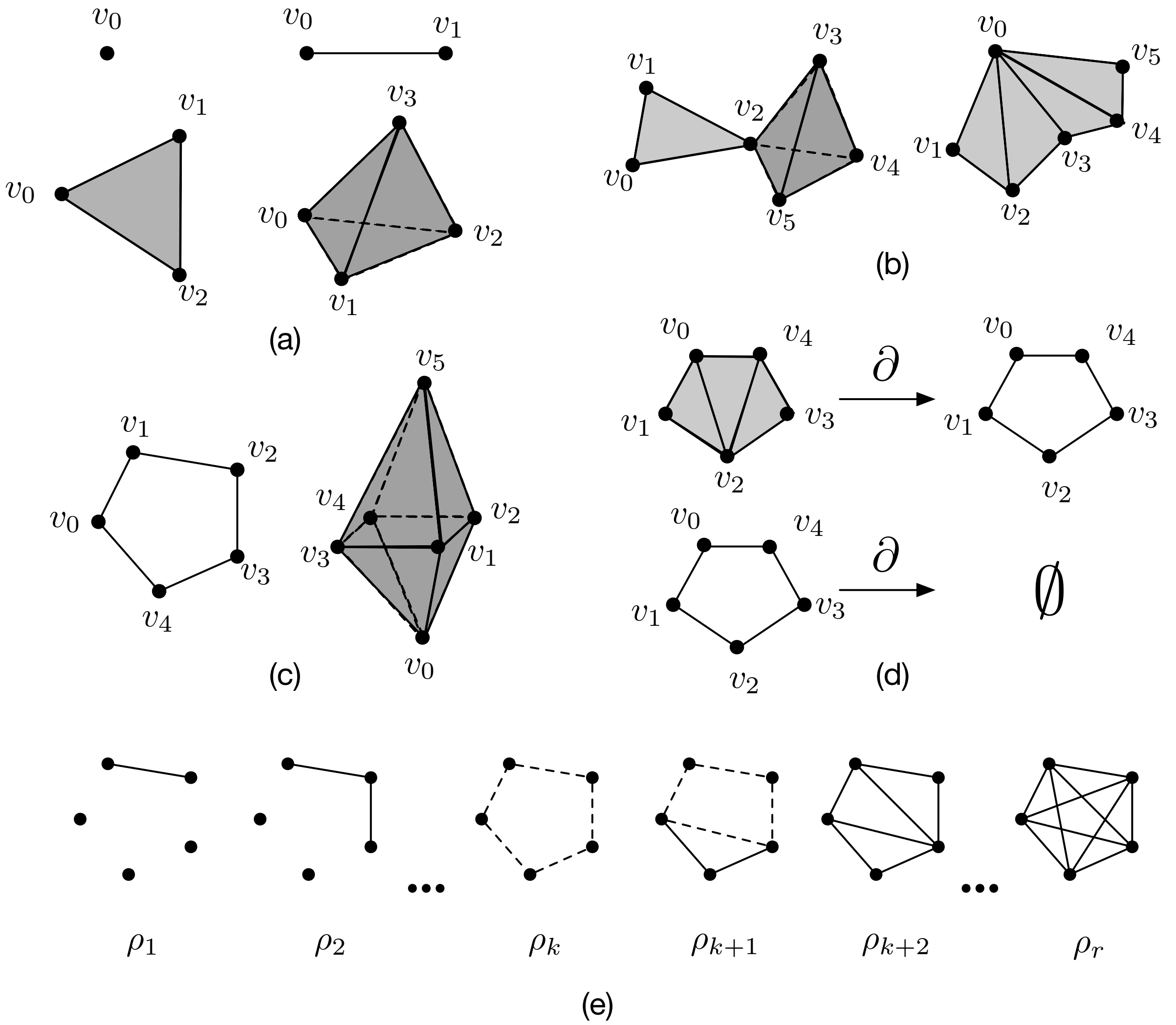


Figure 4.1: (a) Shown here are examples of a 0-clique (a point or node of a graph), 1-clique (two nodes connected by a 1D undirected edge), 2-clique (three nodes connected by the 1D edges and 2D faces filling in the space between the edges), and 3-clique (four nodes connected by their 1D edges, 2D faces, and a 3D solid filling in the space between the faces). These simplicies correspond to topological realization of a 0-clique, 1-clique, 2-clique, and 3-clique, respectively. Note that a n-clique defines a nD space. (b) Examples of clique complexes: cliques glue on common faces to form topological objects. (c) Example cavities in the topological space. (d) The boundary operator of a chain surrounding a clique complex (top) and a cavity (bottom). (e) Persistent homology detects birth and destruction of a cavity (shown with dashed line) during the mapping of a weighted graph onto binary graphs (Algorithm 2).

We define the correlation activation of nodes in a DNN as topological objects, as those illustrated in Fig. 4.1(a-c).

We study the topological properties of these objects. For this, we turn to several concepts necessary to compute homology, a method in algebraic topology capable of counting *cavities* in topological objects.

4.2 Homology and cavities

We define a *chain complex* $C(S)$ of a clique complex, to be the sequence $\{C_n(S, \mathbb{F}_2)\}_{n \geq 0}$ (abbreviated C_n), where C_n is the $\mathbb{F}_2$-vector space whose bases are the (n+1)-cliques $\sigma \in S_n, \forall n \geq 0$, and $\mathbb{F}_2 = \{0, 1\}$. In other words the elements of C_n are interconnections of (n+1)-cliques in S. For example, elements of C_1 are linear combinations of edges (2-cliques) and elements of C_2 are linear combinations of *filled* triangles (3-cliques, i.e. 2D faces).

For each $n \geq 1$, there exists a linear transformation called a *boundary operator* that maps C_n onto a lower dimensional chain complex:

$$\partial_n : C_n \to C_{n-1}, \tag{4.1}$$

with

$$\partial_n(\sigma_{0,1,\ldots,n}) = \sum_{k=0}^{n} \sigma_{0,1,\ldots,k-1,k+1,\ldots,n}. \tag{4.2}$$

Geometrically, the boundary of a n-clique is the set of $(n-1)$-cliques bordering it. Fig. 4.1(d) shows an example. Hence, the boundary operator takes a collection of n-cliques and maps them to their boundary, i.e., a collection of $(n-1)$-cliques.

Similarly, a n-cycle is a path in our graph that forms closed structures, , $v_1 \to v_2 \to v_3 \to v_1$. Since the boundary of any clique is a cycle, the boundary operator provides a mechanism to compute cycles. Formally, a n-cycle is given by any element $l \in C_n$ that satisfies $\partial_n(l) = 0$, Fig. 4.1(d).

Let us call the subset of n-cycles, $k(\partial_n)$. And, let the n-cycle that defines the boundary of C_n be $b(\partial_n)$.

Note that two n-cycles, $l_1, l_2 \in C_n$, are equivalent if their sum in $\mathbb{F}_2$ defines the boundary of a path of $n+1$ vertices, i.e., a $(n+1)$-chain.

Formally, we define this homology of dimension n as,

$$H_n(S, \mathbb{F}_2) = k(\partial_n)/b(\partial_n + 1), \tag{4.3}$$

for $n \geq 1$, and

$$H_0(S) = C_0/b(\partial_1). \tag{4.4}$$

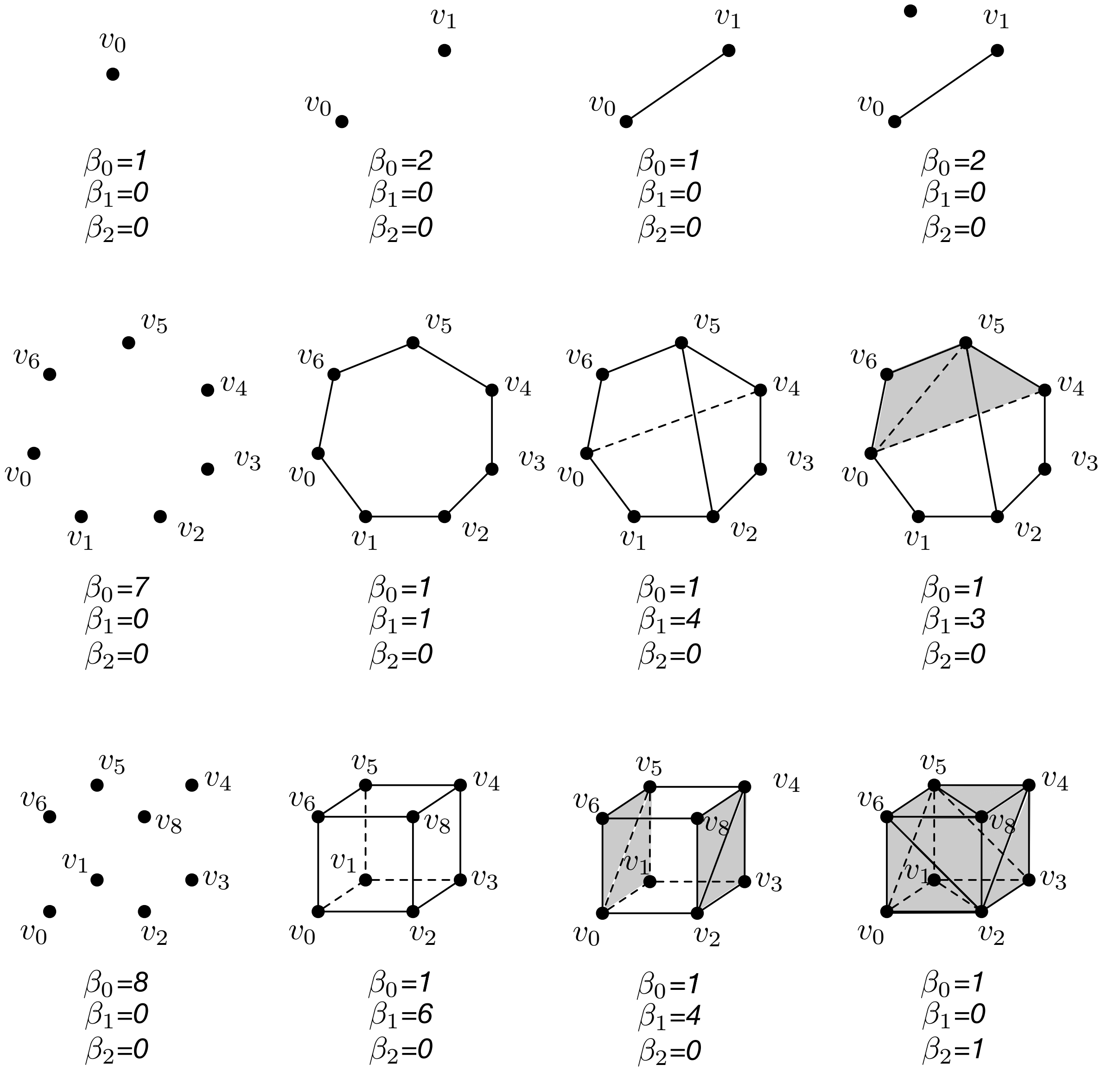


Figure 4.2: The Betti numbers are sequences of natural numbers counting the cavities of a topological object in a corresponding dimension. Low dimensional Betti numbers have intuitive interpretation: β_0 counts connected components, β_1 2D cavities, β_2 3D cavities. We show several examples of clique simplices and corresponding Betti numbers. By adding edges, lower dimensional cavities are destroyed while higher dimensional are formed. For example, note how a first cavity $l_0 = (v_0, v_1, ..., v_6)$ is formed (middle row, second column), then by adding two more edges (v_2, v_4) and (v_0, v_4) four new cavities appear $l_1 = (v_0, v_4, v_5, v_6)$, $l_2 = (v_0, v_1, v_2, v_3, v_4)$, $l_3 = (v_2, v_3, v_4, v_5)$ and $l_4 = (v_0, v_1, v_2, v_5, v_6)$ (middle row, third column). By definition, a geometrical realization of a clique always includes all the enclosed space between the nodes (simplices). Adding another edge (v_0, v_5) fills l_1, thus β_1 drops to 3. Some edges are dashed for facilitating visualization.

Therefore, H_n defines the vector space spanned by the class of n-cycles. Its dimensionality is the number of non-trivial n-cycles; i.e., the number of $(n+1)$-dimensional cavities in the objects (graphs) in Fig. 4.1(c). And a *cavity* is a non-filled face of dimensionality n, Fig 4.1(d).

Since cycles are chain complexes that map to $\emptyset$ by the boundary operator, *cavities* are the nullspace of this boundary operator, $null(\partial_n) = k(\partial_n) \subset C_n$.

4.3 Projecting a DNN into the Topological Space

The above is defined in binary graphs, i.e., in $\mathbb{F}_2$. For weighted graphs we will compute all the possible binary graphs. Let us define an approach to achieve this.

Let $G = (V, E)$ be the graph defining the DNN we wish to study, and $X = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^m$ the labeled training samples, with m the number of samples, $\mathbf{x}_i \in \mathbb{R}^p$, and $\mathbf{y}_i \in \mathbb{Z}$ in classification and $\mathbf{y}_i \in \mathbb{R}^q$ in regression ($q \geq 1$).

Passing the sample vectors $\mathbf{x}_i$ through the network ($i = 1, \dots, m$), allows us to compute the correlation c_{ij} between the activation (a_i, a_j) of each pair of nodes (v_i, v_j) of G. That is given by,

$$c_{ij} = \frac{m(m-1)}{2} \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} \frac{(a_i - \mu_{a_i})(a_j - \mu_{a_j})}{\varsigma_{a_i} \varsigma_{a_j}}, \tag{4.5}$$

where μ and ς indicate the mean and standard deviation over X on the given node.

Let us now order the absolute values of these correlations from largest to smallest: $|c_{i_1 j_1}| \geq |c_{i_2 j_2}| \geq \cdots \geq |c_{i_r j_r}| > T$, where r the number of correlations larger than T in G given X. In this notation, the subscripts of each correlation indicate the node pairs in that correlation: $(v_{i_1}, v_{j_1}), \dots, (v_{i_r}, v_{j_r})$.

Binary graphs are obtained by iteratively adding a node pair at a time. We start by adding the nodes with largest correlation, (v_{i_1}, v_{j_1}), and continue adding node pairs up to the last one, (v_{i_r}, v_{j_r}).

We will refer to the number of edges included in a binary graph as its density, $\rho_k = k/r$ where k is the number of non-zero correlations ($|c_{ij}| > 0$, $\forall i$ and $j > i$) and r is the total possible number of correlations. Thus, the density of our binary graph will increase at each iteration (as we add more and more nodes).

This process allows us to investigate how the homology of the graph defining the DNN of interest evolves as a function of the density, Fig. 4.1(e).

This approach is summarized in Algorithm 2. This algorithm maps a DNN defined by a weighted graph onto the set of binary graphs $\{G_1, \dots, G_r\}$. And, these binary graphs are topological objects as shown in Fig. 4.1.

Algorithm 1 allows us to compute any topological property of a DNN given X as a function of learning.

Algorithm 2 Weighted to binary graph mapping.

```
Let $G = (V, E)$, $X = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{m}$, and define $T > 0$.
Let $r = \{|c_{i_1j_1}|, |c_{i_2j_2}|, \ldots, |c_{i_rj_r}|\}$, s.t. $|c_{i_1j_1}| \geq |c_{i_2j_2|} \geq \cdots \geq |c_{i_rj_r}| > T$.
Let $G_0 = \emptyset$ and k=0.
**repeat**
    $k = k + 1$.
    $G_k = G_{k-1} \bigcup \{v_{i_k}, \hat{e}_{ij} v_{j_k}\}$, with $\hat{e}_{ij}$ the undirected edge defining $v_{i_k}$— $v_{j_k}$.
    $\rho_k = k/r$.
**until** $k = r$.
```

Chapter 5

What Does It Mean to Learn in Deep Neural Networks?

The flexibility and high-accuracy of Deep Neural Networks (DNNs) has transformed computer vision. But, the fact that we do not know when a specific DNN will work and when it will fail has resulted in a lack of trust. A clear example is self-driving cars; people are uncomfortable sitting in a car driven by algorithms that may fail under some unknown, unpredictable conditions. Interpretable and explainable approaches attempt to address this by uncovering what a DNN models, i.e, what each node (cell) in the network represents and what images are most likely to activate it. This can be used to generate, for example, adversarial attacks. But these approaches do not generally allow us to determine where a DNN will succeed or fail and *why* does this learned representation *generalize* to unseen samples. In this chapter, we use the theoretical framework proposed in Chapter 4 to define what it means to learn in deep networks, and how to use this knowledge to detect adversarial attacks. We show how this defines the ability of a network to generalize to unseen testing samples and, most importantly, *why* this is the case.

5.1 Introduction

Deep Neural Networks (DNNs) have enough capacity to learn from very large datasets, without the need to define hand-crafted features, models or hypotheses for every problem. This has revolutionized computer vision and other areas of Artificial Intelligence (AI) [116].

Using high-capacity DNNs to learn from huge datasets, however, does not tell us how the network learns what, it does and why. This so called “black-box” problem has resulted in a lack of trust [27, 213]. For example, why did a given DNN identified a specific deep representation as more appropriate than other possible representations?

Interpretability and explainability methods [210] allow us to see what a DNN has learned, but not *how* and *why* it learned it. For example, if the goal is to know what type

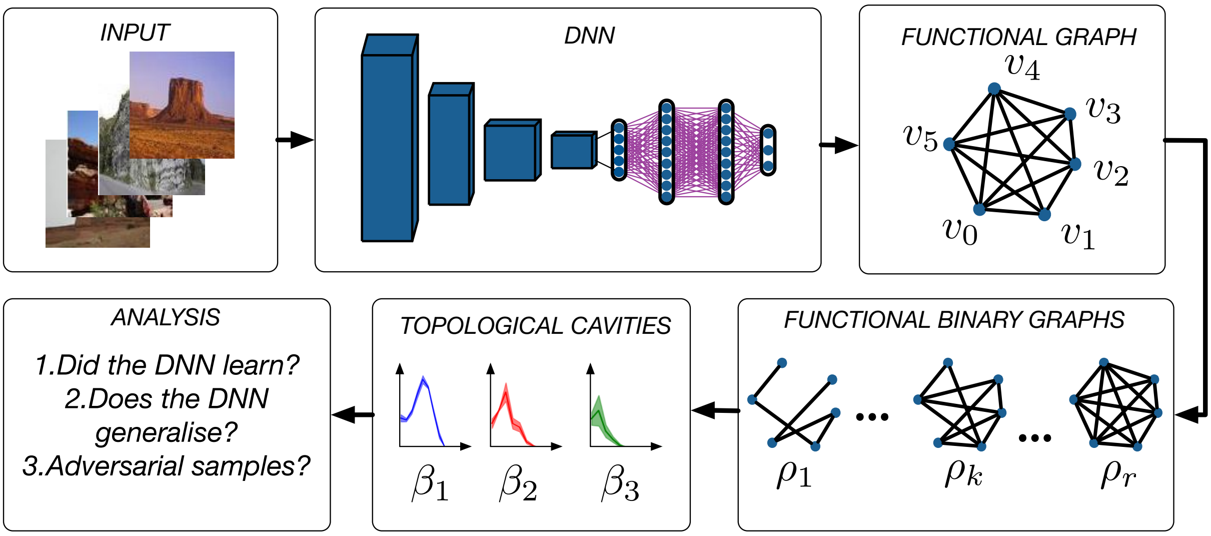


Figure 5.1: Given a DNN its functional graph is defined by the correlation of activation of distant nodes. This allows us to compute binary graphs defining local and global topological properties of the network (Algorithm 2). Global topological properties define generalization, while local properties identify overfitting (Algorithm 3). The same topological properties are used to detect adversarial attacks.

of image activates a specific node (cell) of the network, one can optimize the DeepDream or other related criteria [2]. When our goal is to generate plausible class output images, optimizing softmax and related objectives may be performed [170]. But neither method tells us how and why these deep features (representations) were chosen by the network and *where* they may fail.

The main problems of lack of interpretability and explainability are: 1. If we do not know how DNNs learn, we can only improve them by trail-and-error. This is slow and does not solve the trustability problem. And, 2. If we do not know why DNNs learn a specific deep representation or model, we will not know where and why the network fails, e.g., detect an adversarial attack.

With the help of theoretical framework presented in Chapter 4, we derive a set of algorithms that help us address the problems stated above. Simple local properties of the network (e.g., the degree of a node) as well as global properties (e.g., the path length between nodes) have been found to be insufficient to explain how and why DNNs learn [23].

This chapter shows how topological properties (e.g., n-cliques and nD holes) of the functionality of the network describe how a DNN learns, Figure 6.2. We use these properties to demonstrate we can predict when a DNN successfully learns, and when and why it is likely to misclassify an unseen test sample (Algorithms 2 and 3). We also show how to use the derived approach to successfully detect adversarial attacks.

5.2 Related Work

In recent years, high capacity Deep Neural Networks (DNNs) have outperformed many other machine learning algorithms in a number of applications like object recognition [111, 201] and speech recognition [85, 152], especially when large labeled datasets [171, 120] and/or means of obtaining information from large datasets were available [68].

Why or how DNNs achieve this feat and where and why DNNs fail remains a mystery. This makes DNNs untrustworthy "black-box" models to many [213], with some researchers claiming DNNs are too unpredictable to make them a general solution to most AI problems [169, 59, 149].

To solve this problem, researchers are trying to understand what DNNs do. Two approaches are *interpretability* and *explainability*, with an emphasis on defining what the nodes (cells) of the network represent and how they respond to different inputs [97, 113, 210, 166, 240, 141, 156].

For example, DNNs used in object recognition generally learn to extract low-level, Gabor-like features in the first layers, combine these low-level features to represent parts of an object in mid-level layers, and finally combine those to generate highly complex representation of objects that are invariant to image changes, e.g., pose, illumination and affine transformations [176, 157, 256].

Another group of methods focuses on the analysis of these features. One way is by exploring the semantic meaning of filters [200] or computing feature distributions of different attributes [12]. Saliency maps, heatmaps, sensitivity maps, and attention maps display class relevant information that is used by the model to make its predictions [63, 121, 63, 257].

The goal of this chapter is to go beyond these approaches of interpretability and explainability and ask, instead, what does it mean to learn in DNNs? Specifically, how does the graph defining the network evolve during the learning process? And, how can we use this knowledge to know where the network has successfully learned and where it will fail?

Making "black-box" models like DNNs interpretable is of great importance for several reasons, as for example to: *i*) increase trust; *ii*) facilitate transferability to other problems, e.g., use a network pre-trained on a specific problem in a different problem [239]; *iii*) predict where a network is likely to work and where it will most probably fail; *iv*) help researchers and practitioners design better networks, because we now know what works and what does not; *v*) derive unsupervised learning algorithms [225, 256], because we will know what the network needs to do to generalize to unseen samples; and *vi*) better prepare our algorithms for fair and ethical decision makings (i.e., unbiased performance) [24, 69].

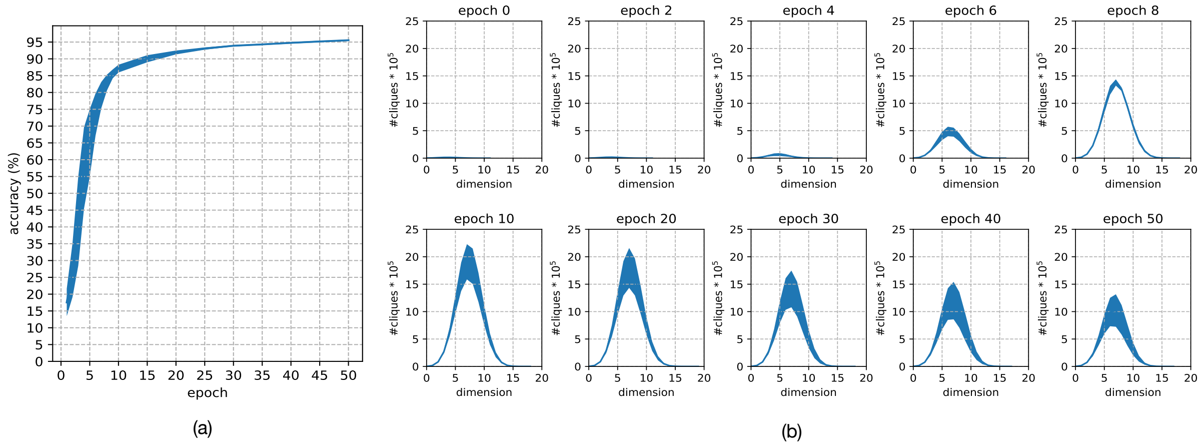


Figure 5.2: Testing accuracy for LeNet training on MNIST (a) vs the number of cliques (b). Mean and standard deviation given by 5-fold cross-validation.

5.3 Experimental Analysis

The structure of a DNN is generally defined by a weighted graph. Similarly, the “behavior” of a DNN (the activation of its nodes) are given by a functional graph, the correlations between the activation of each pair of nodes.

We study the topological changes of this functional graphs during training and testing. The reader is referred to Chapter 4 for the detailed methodology. *We show that networks that generalize to unseen testing samples converge to a common topology, and that this topology is differential from that of networks that fail to generalize.* We then demonstrate how we can exploit this new knowledge to determine whether an unseen testing sample will be correctly analyzed by the network and when it will not. Specifically, we focus on adversarial attacks.

We start with the analysis of the number of n-cliques. To illustrate this, we will use the LeNet derived in [115]. LeNet is a historical network and will be used here as a proof-of-concept. This network was originally derived to recognize written digits, i.e, 0-10. We use the MNIST dataset, consisting of $60,000$ training images and $10,000$ test images, as X [115].

The cross-validation (CV) classification accuracy of this network is shown in Fig. 5.2(a), where the x-axis defines the epoch and the y-axis the 5-fold CV accuracy, and the thickens of the blue curve defines the variance over this CV procedure.

Given the graph of LeNet G_{LeNet} and the training set X, we use Algorithm 2 to get G_k s.t. $\rho_k = .25$. The number of n-cliques ($n \in [0, 20]$) in G_k is shown in Fig. 5.2(b), where each plot represents the results at the indicated epoch, the x-axis specifies the value of n, and the y-axis the number of n-cliques.

Note that the number of n-cliques in an untrained DNN must be zero or very close to zero. That is because the number of nodes working together (correlated, as given by c_{ij}) before any training is performed must be zero or tiny. This is shown in the first plot of Fig. 5.2(b). But as the network *learns*, the nodes in the graph start to work together to solve the problem of mapping $\mathbf{x}_i$ onto $\mathbf{y}$.

Note that by the time the learning process has made its major gains (by about epoch 10), the number of n-cliques is maximum. As the learning process continuous tightening the knobs of the network (adjusting its parameters), the number of cliques starts to decrease. This is the result of overfitting to X.

Are cavities an even better indicator of how well the network learns a dataset X? We explore this next.

5.3.1 Cavities in DNNs

The rank of the n-homology group H_n in a topological space is called the n^{th} Betti number. In other words, Betti numbers compute the maximum amount of cuts that must be made to separate a surface into two k-cycles, $k = 1, 2, \ldots$.

Hence, the first Betti number, β_0, computes the number of connected elements in $S(G)$; the second Betti number, β_1, calculates 1D cavities (or holes); β_2 gives the number of 2D cavities; and, more generally, β_n computes the number of nD cavities, Figure 4.2.

Key to understanding Figure 4.2 is to note that all kD faces of the simplicies of a n-clique are filled, e.g., see the last object in the second row and the last two in the last row in Figure 4.2; note how these kD faces eliminate cavities of lower dimensionality and add cavities of higher dimensionality.

To further clarify this important point, consider the second example in the second row in Figure 4.2. The functional representation of this DNN indicates that node v_0 and v_1 work together to solve the classification or regression problem the network is tasked to address (their activation patterns) are highly correlated (Algorithm 2). Similarly, v_i and v_{i+1} $(i = 1, \ldots, 6)$ and v_6 and v_0 also work together to solve the problem the network is faced with. This creates a 1D cavity, $\beta_1 = 1$, because ∂ maps this chain to $\emptyset$. But if v_0 and v_4 and v_0 and v_5 are also highly correlated, this forms cliques with filled simplicies, yielding two additional 1D cavities, $\beta_1 = 3$; as shown in the last objects in the second row in Fig. 4.2.

The Betti numbers of a clique complex S of a graph G are formally given by,

$$\beta_n(S) = null(\partial_n) - rank(\partial_{n+1}). \tag{5.1}$$

Note that in order to compute β_n, we need the null space and the rank of the matrix formed by cliques of dimension n and dimension $n-1$. This means that we only need to compute the cliques in the first n dimensions to calculate $\beta_1, \ldots, \beta_n$. Since, in most

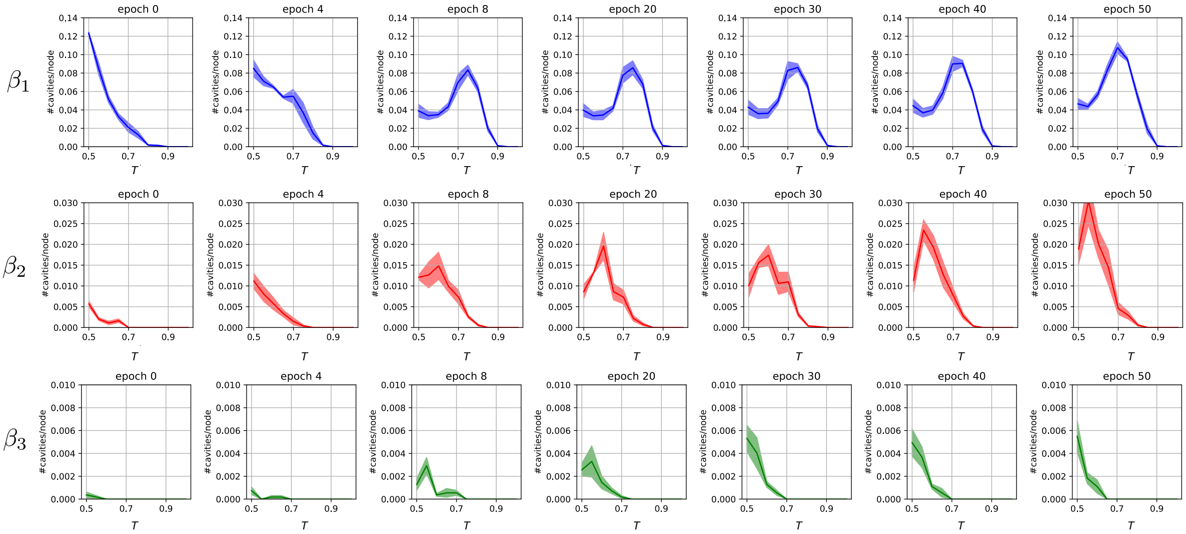


Figure 5.3: Betti number dynamics during LeNet [115] training on MNIST. Mean and standard deviation given by 5-fold cross-validation. Refer to Fig. 5.2(a) for testing accuracy.

DNNs, $\beta_n = 0$ for n larger than 4 or 5, this means that the computations can be performed efficiently.

5.3.2 Learning and generalization

The evolution of the Betti numbers of the LeNet network as a function of T and epochs is shown in Fig. 5.3. Recall, T defines the density our approach explores, i.e., T is inversely proportional to ρ (see Algorithm 2). The y-axis in the plots indicates the number of cavities properly normalized by the number of nodes (i.e., number of cavities/node).

As we can see in this *key* figure, the number of 1D cavities moves from higher to lower densities as the DNN learns. That is, as the DNN learns, the correlation pattern of activation of the nodes in the network that are further apart as well as those that generate larger simplicial complexes start to appear.[1] This demonstrates that the network is constructing *global* structures to learn the function that maps the input samples to their corresponding outputs, $\mathbf{y}_i = g(\mathbf{x}_i)$, i.e., the network is learning to generalize by using large portions of the graph.

But when the network can no longer generalize, it starts to memorize the samples that do not abide by the learned functional mapping $g(\cdot)$. This results in a decrease of global structure and an increase in *local* chain complexes. This is clearly visible in Figure 5.3:

[1] This effect is due to the fact that nodes that work together start to form cliques whose simplices fill in the holes of the functional graph. That is, rather than adding additional nodes when increasing the density from ρ_k to ρ_{k+1}, we add edges between the nodes already available at density ρ_k. We refer to this as a *global* property, because rather than having a chain of semi-related nodes (e.g., $v_1 \to v_2 \to \cdots \to v_r$), we have n-cliques (i.e., all nodes connected to all other nodes). Note that the cliques appearing at higher densities delete the cavities we observe at lower densities, Figure 4.2.

Algorithm 3 Generalization.

```
Let $G = (V, E)$ define a DNN with $\theta$ its parameters.
Let $X$ be the training set, and set $T > 0$.
Set t=0, and $n$ to either 1, 2, or 3.
repeat
    Use $X$ and the selected loss to optimize $\theta$.
    t=t+1.
    Use Algorithm 2 to obtain $G_k$, $k = 1, \dots, r$.
    Compute the clique complexes $S_k$ of $G_k$.
    $\widehat{k}_t = \arg\max_k \beta_n(S_k)$.
until $\widehat{k}_t > \widehat{k}_{t-1}$.
```

note the maximum number of cavities starts to move toward higher densities. As seen in Figure 5.3, this happens after about epoch = 40, when the network has done all it can learn to generalize, Figure 5.2(a). An example of this topological effect was illustrated in the second and third rows of Fig. 4.2.

This means *learning in DNNs is equivalent to finding the smallest density $\rho_{\widehat{k}}$ of nD cavities in the functional binary graphs that define the network.* This peak density allows us to identify when a network has reach its limits of global learnability and generalization to unseen samples.

This approach is summarized in Algorithm 3.

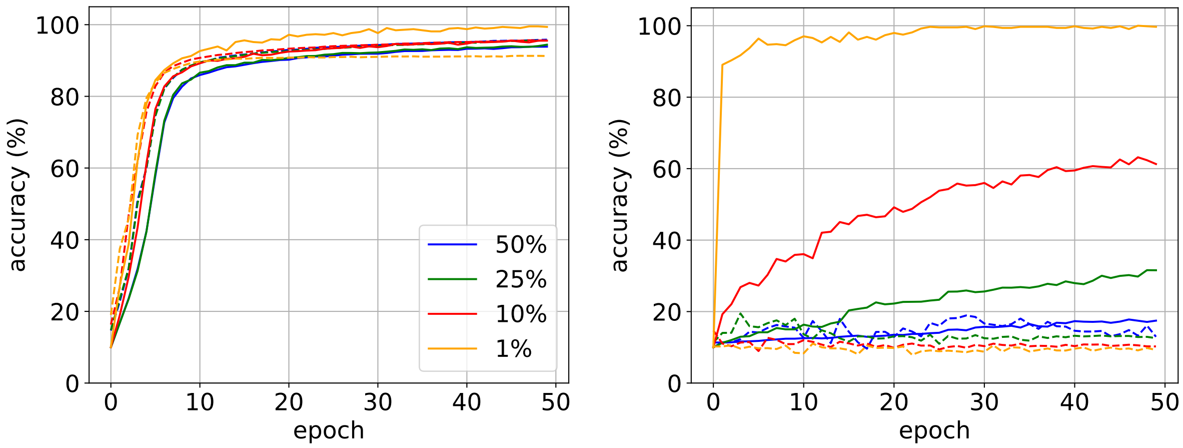
If we wish to find global properties with Algorithm 3, we set $n = 1$ to detect 1D cavities. But if we wish to identify even larger topological connections, we set $n = 2$ or 3. Hence, larger n values mean we are interested in increasingly general properties of the underlying unknown function $g(\cdot)$ our DNN needs to learn.

The smaller n is, the more we allow the DNN to adapt to our specific dataset. Thus, a smaller n will yield more accurate results on a testing set that is representative of the training, but less accurate results on test sets that diverge from the training set.

5.3.3 Learning vs. failing to learn

We can also use the above defined topological properties of our DNN to determine where the network fails to learn to generalize to unseen samples.

Let us illustrate this in an example using LeNet. In this example, we trained the network with either 50%, 25%, 10% or 1% random selection of the training samples (plus the data augmentation typically used in LeNet). The training and testing accuracies are shown in Figure 5.4(a). Solid lines indicate training accuracy; dashed lines testing accuracy. As it can be appreciated in the figure, for LeNet the testing accuracy follows the same pattern as the training, regardless of the percentage of training data used.



(a)

(b)

Figure 5.4: Training/testing accuracy with decreasing proportion of training data and permuted/non-permuted labels.

But, when we redo the above experiment with permuted labels,[2] the results are very different, Figure 5.4(b).

Note how the network is able to learn from small nonsensical (permuted) datasets during training. But the testing accuracy curves show this is just an illusion, i.e., the network is able to memorize these nonsensical sample pairs $\{\mathbf{x}_i, \tilde{\mathbf{y}}_i\}$, where $\tilde{\mathbf{y}}_i$ are the permuted labels, but this serves no purpose when it comes to testing the performance of the system on independent samples. This means we cannot relay on the training accuracy as a measure of *generalization*; a well-known fact.

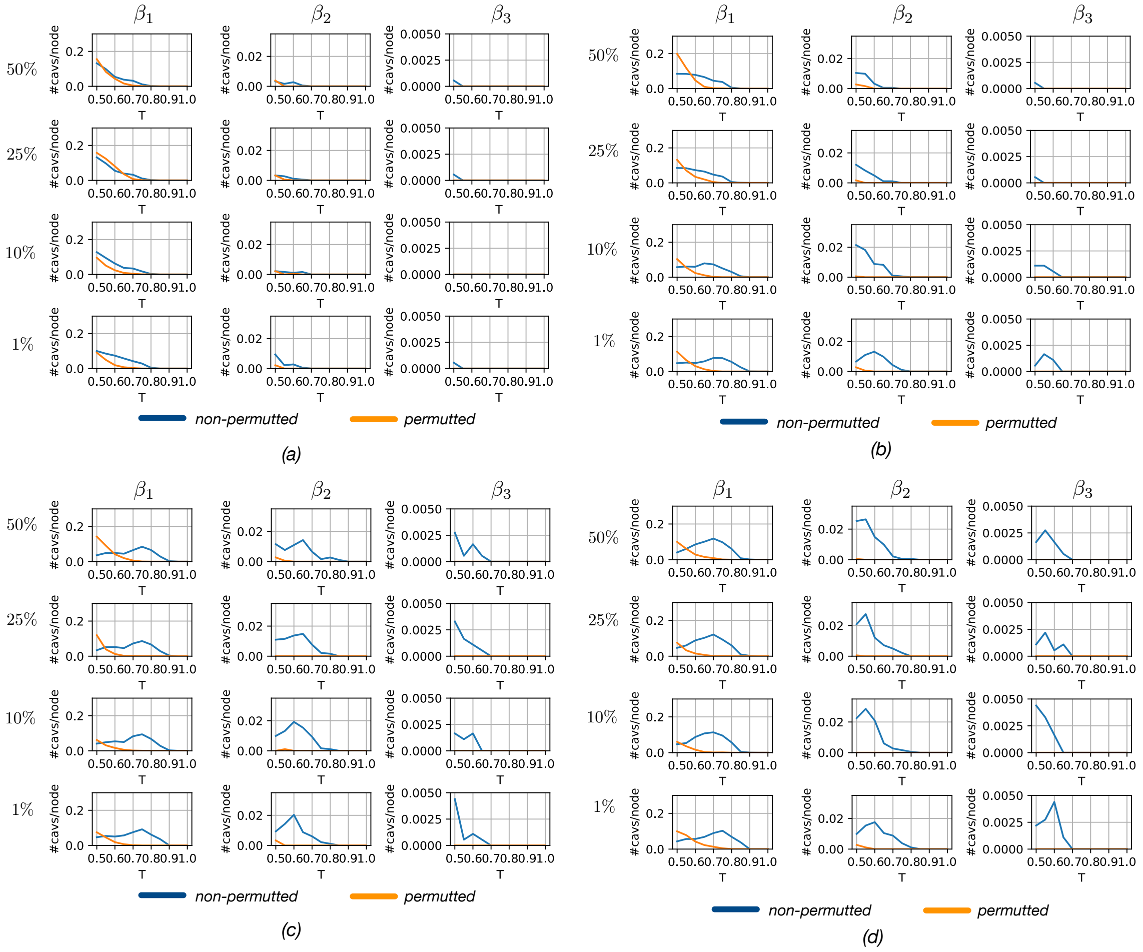
Fortunately, from our previous section, we know that nD cavities move toward lower densities of our functional binary graphs when the network learns to generalize. We will now use this insight to derive a simple algorithm that knows whether the network is learning or not.

Figure 5.5 shows the Bettis at different densities, percentages of training data (50 to 1%), and epochs (epoch = 2, 10 and 50). The blue curves correspond to the Bettis computed with Algorithm 3 and non-permuted data. Orange curves show the Bettis obtained when using the permuted labels $\tilde{\mathbf{y}}_i$. Figure 5.5(a) are the results at epoch = 2, Figure 5.5(b) at epoch = 4, Figure 5.5(c) at epoch = 10 and Figure 5.5(d) at epoch = 50.

As expected, the maximum number in β_1 moves to lower densities when the labels are *non-permuted*, but does *not* when the labels are *permuted*. Even more telling is the lack of 2D and 3D cavities when using *permuted* labels.

Our algorithm to detect lack of training is thus simple: *a.* Lack of 2D and 3D cavities, and *b.* $1D$ cavities move toward lower densities.

[2] This means that the labels $\mathbf{y}_i$ have been permuted by multiplying the vector $(\mathbf{y}_1, \ldots, \mathbf{y}_m)^T$ with a randomly generated $m \times m$ permutation matrix $\mathbf{P}$. A permutation matrix is obtained by permuting the rows of an identity matrix.



(a)

(b)

(c)

(d)

Figure 5.5: Betti numbers obtained when using 50%, 25%, 10% or 1% of the training data (top to bottom row, respectively). Blue curves indicate non-permuted labels; orange curves indicate permuted labels. 1D cavities (β_1) are shown in the first column; 2D cavities (β_2) in the second column; and 3D cavities (β_3) in the third column. Results plotted at the following epochs (a) 2, (b) 4, (c) 10 and (d) 50.

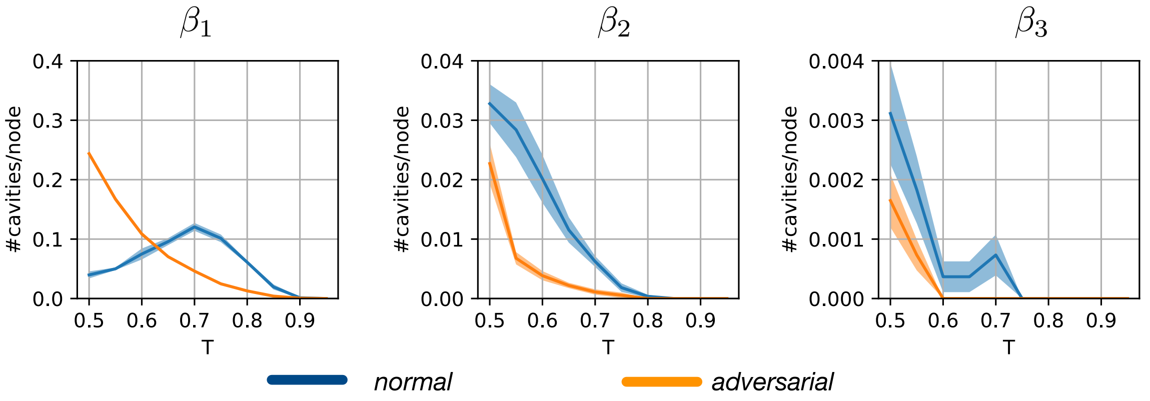


Figure 5.6: Betti numbers obtained when using unaltered and adversarial testing samples. LeNet trained on MNIST.

What does it mean to learn in DNN? Learning to generalize in DNN is defined by the creation of 2 and 3D cavities in the functional binary graphs representing the correlations of activation of distant nodes of the DNN, and the movement of 1D cavities from higher to lower graph density ρ. Overfitting is indicated by a regression of these cavities toward higher densities in these functional binary graphs.

5.3.4 Detecting adversarial attacks

We can also use the approach derived in the previous sections to know where our network is likely to fail during testing. We illustrate this by demonstrating how to detect an adversarial attack, which is a guaranteed, easy way to make a DNN fail [200].

To do this, we use the algorithm of [150] to generate images for an adversarial attack on a trained LeNet. As above, LeNet was trained on MNIST.

For testing, we used the independent MNIST testing set and the set of images prepared for the adversarial attack.

As above, we expect 1D cavities to increase and move to lower densities as well as an increase in the number of 2 and 3D cavities at lower densities on unaltered testing data. But for the data in the adversarial set, we expect only 1, 2 and 3D cavities at the highest densities, indicating local processing but a lack of global engagement of the network.

The results are in Fig. 5.6, with the blue curves indicating the cavities on the functional binary graphs G_k when using the unaltered testing sample, and the orange curve those observed when processing the samples in the adversarial attack set. Thus, our predictions are confirmed, and we know that the images in the adversarial attack set cannot be correctly classified by LeNet.

This approach, for the first time allows us to identify *test* images that the network is bound to misclassify.

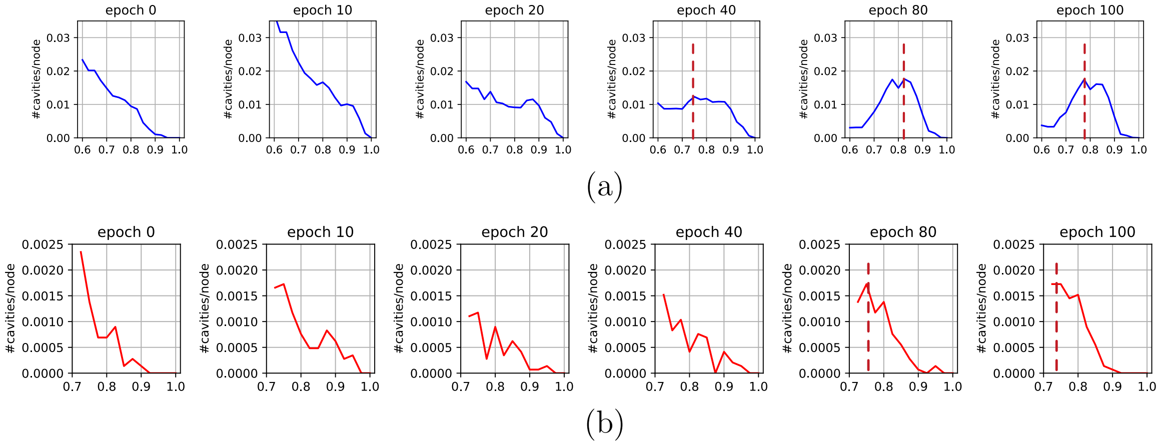


Figure 5.7: Results of the VGG16 on CIFAR. (a) Number of normalized 1D cavities (y-axis) as a function of graph density (x-axis) and number of epochs (given at the top of each plot). (b) Same as (a) but for 2D cavities.

5.3.5 Additional results

In the previous sections we illustrated the use of our approach on the LeNet, a historical DNN. In this section, we replicate results on a recent DNN: VGG16 [194].

In this case, we used the CIFAR10 [110] dataset instead. CIFAR10 consists of 60,000 training images and 10,000 separate test images labelled with 10 different classes. These are natural images representing common objects (, dog, cat, airplane). For training, data was augmented using random horizontal flips and random crops of equal image size with padding $= 4$. All data was resized to 32×32 pixels and mean and variance were normalized.

Fig. 5.7 shows the results of our approach. The first row in this figure shows the number of 1D cavities as a function of the density of the functional binary graph representing VGG16. Different plots specify these number at distinct epochs. The second row does the same but for 2D cavities.

Note how the maximum number of 1D and 2D cavities (indicated by a dashed red vertical line) moves toward less dense functional representations of the VGG16, as described the previous sections and Algorithm 2. The lowest density is achieved at epoch 80. After that, there is a small regression toward higher densities, suggestive of overfitting. These results are confirmed by the plot of the testing accuracy shown in Fig. 5.8. In this figure we see that while the training accuracy keeps growing after epoch 80, the testing accuracy does not. Instead, the testing accuracy slightly decreases after epoch 80, as suggested by our approach.

In Fig. 5.9, we show the plots of 1D and 2D cavities for the original (unaltered) training samples (blue curve) and for the adversarial samples (red curve). The plots are given as a function of the density. As we can see in the figure, the results mimic those reported in the previous sections, allowing us to detect the adversarial attack.

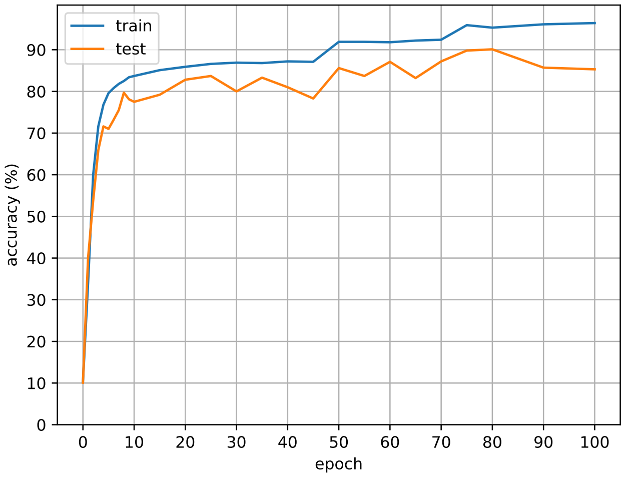


Figure 5.8: VGG16 training and testing accuracy on CIFAR10 as a function of training epochs.

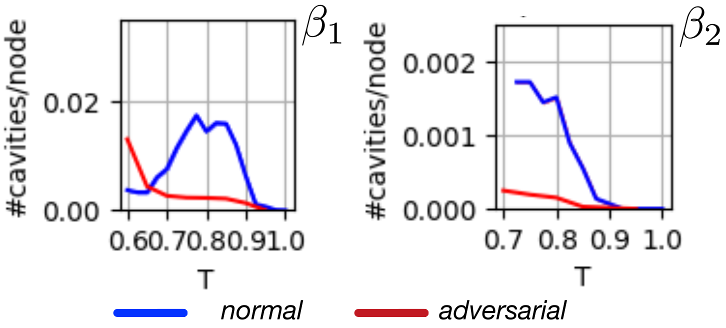


Figure 5.9: Betti numbers obtained when using unaltered and adversarial testing samples. VGG16 trained on CIFAR10.

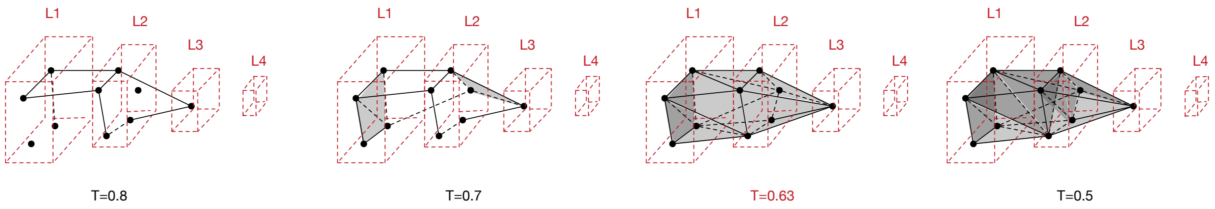


Figure 5.10: Example of 1D, 2D and 3D cavity formation in LeNet trained on MINST.

5.3.5.1 Cavity formation: an example

In this chapter, we derived a theory that relates cavity formation with generalization. These cavities are given by the binary graphs defining the correlation of activation of the nodes of the DNN.

To clarify the formation of these cavities, let us illustrate this on a selected set of nodes of the LeNet DNN, Fig. 5.10.

In this figure, we can see the formation of 1D cavities first, then the formation of 2D cavities, and, finally, the formation of a 3D cavity (second to last binary graph). Note how these cavities are formed as the density of the binary graph is increased. Then, in the right-most graph, we see how further increasing the density eliminates all the cavities.

This particular example is for the activation graph formed at epoch 20. Layers are over-imposed to show location in the network. The location within the layer is chosen for the best visual effect. By increasing density (i.e, decreasing T), more and more edges are added to the graph. Specifically, at $T = .8$, a couple of 2D cavities form. These contribute to an increase in β_1. At exactly $T = .63$ a 3D cavity is realized between these, resulting in an increase to β_2. This cavity is filled at higher densities ($T = .5$).

5.3.5.2 Training with Bettis

As this chapter details, we can use our approach as a measure of generalization. Hence, Algorithm 2 can be used in lieu of the training or verification error to determine when the network has learned to generalize and before the network overfits to the training data.

This result is illustrated in Fig. 5.11. This figure shows the testing classification accuracy (in blue) and the difference of the peak densities at epoch t and $t-1$ (in black). Also note that classification accuracies are given on the left y-axis and values of $\Delta k = \hat{k}_t - \hat{k}_{t-1}$ are on the right y-axis.

When Δk is beyond a small threshold $-\epsilon$, Algorithm 2 decides to stop training. As shown in the figure that coincides with the point where generalization is achieved (thick red line in the figure). After that, the network starts to memorize the samples that do not generalize.

Specifically, given Δk, we can clearly observe three dynamic regimes for $\hat{k}$. Initially, in the first epochs $\hat{k}$ increases rapidly, it then reaches a maximum value between epochs 8 and

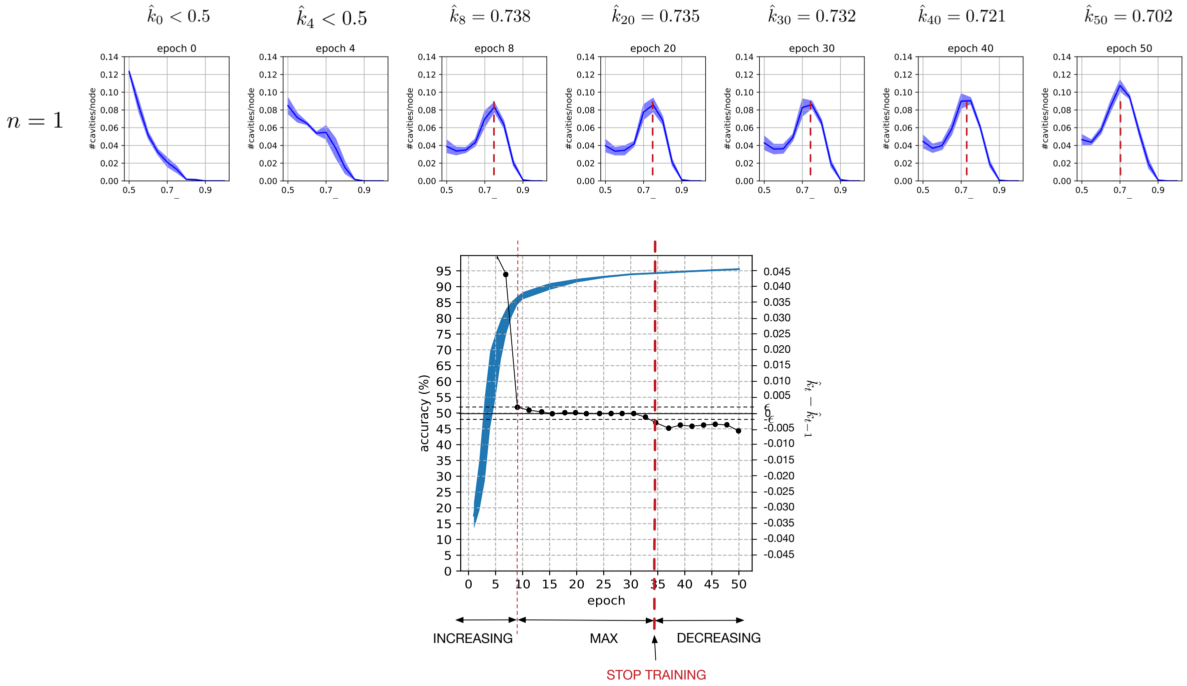


Figure 5.11: An illustration of Algorithm 2 for $n = 1$.

32 and then it starts decreasing in the later stage of training. Even where no verification set is available, one can use this knowledge to deduce when the training should stop. One can also imagine how this approach may one day be applied to unsupervised learning, but additional research will be necessary to make this a reality.

5.3.6 Algorithmic complexity

Finally, we give a formal analysis of the computational complexity of the derived algorithm.

Let the binomial coefficient $p = \binom{N+1}{n+1}$ be the number of n-simplices of a simplicial complex S. In order to compute $\beta_n(S)$, one has to compute $rank(\partial_{n+1})$ where $\partial_{n+1} \in \mathbb{R}^{p \times q}$, p is the number of n-simplices, and q the number of $(n+1)$-simplices. This has polynomial complexity $O(q^a)$, $a > 1$.

Fortunately, the graphs of a network with N nodes are far from complete (i.e, N-simplices). We take advantage of the graph sparsity to reduce computational complexity. This means that for typical DNNs, the number of n-simplices is way lower than the binomial coefficient defined above, unless one is interested in $\beta_n(S)$ for $n > 3$, or $\rho = 1$ (i.e, $T = 0$).

In Table 5.1, we show the actual time (in minutes) it takes to compute cavities for a network with 1,820 nodes for $T > .5$. This corresponds to a single step of Algorithm 2, excluding training, on a single Intel Xeon, 2.2 GHz CPU.

dimension	$n = 1$	$n \leq 3$
LeNet	.66	2.71
VGG16	5.15	20.24

Table 5.1: Time (in minutes) it takes to compute Betti numbers for LeNet and VGG16.

5.3.7 Experimental details

The networks used in the experiments reported above were trained using stochastic gradient descent with momentum of .9 and weight decay 5×10^{-4}. Learning rate was initialized at 10^{-4} and reduced by half when accuracy stagnated. For training, data was augmented using random horizontal flips, random rotations of ±5 and random crops of equal image size with padding = 4. The images of the adversarial attack were generated using the approach of [150]. This approach computes minimal perturbations, that are mostly indistinguishable to a human observer but have devastating effects on DNNs. In our case, the initial testing accuracy of above 90% (Fig. 5.2(a)) dropped to random decision (10%) after the adversarial perturbations were used.

5.4 Conclusion

DNNs are used in a large number of real life products, including face recognition, facial expression analysis, object recognition, speech recognition, and language translation, to name but a few. Many companies depend on DNNs to design their products.

The main problem with DNNs is that they behave in a "black-box" manner, i.e, we do not know how they learn, what they learn, or where these learned deep representations will fail. This has led to a loss of trust by many [27, 213, 169, 59, 149]. Clear examples of these are self-driving cars and medical diagnoses, with variants of the following argument: If we do not know what the network learned and why and where it will fail, how can I trust it to drive my car or give me a medical diagnosis.

In this chapter, we have introduced a set of tools and algorithms to address these problems. We accomplished this by defining what learning in deep networks means. Specifically, we demonstrated that learning to generalize to unseen (test) samples is equivalent to creating cliques among large number of nodes, whereas memorizing specific training samples (i.e, a type of learning that does not translate to good generalizations) is equivalent to topological changes between low-correlated sets of nodes (i.e, nodes that are not highly cooperative among themselves). We then showed how we can use these same principles to determine when the network works correctly during testing. Specifically, we showed we can reliably identify adversarial attacks.

Beyond what we have demonstrated in this chapter, it is worth mentioning that our approach is general and by no means limited to feedforward neural networks or the use of backpropagation. Our approach works equally well on networks that have all types of directed and undirected edges, e.g., [72], alternative to backpropagation [175], or different types of activation functions.

Next chapter brings us back to the initial problem of facial action unit recognition. We will show that by using the newly proposed analytically framework, we can now train DSIN with increased control and performance and most importantly without the need of any validation data. This is an important advancement especially for a problem where labels are very expensive.

Chapter 6

Improving the Deep Structure Inference Network Through Topological Early Stopping

A common technique to avoid overfitting when training deep neural networks (DNN) is to monitor the performance in a dedicated validation data partition and to stop training as soon as it saturates. This not only completely ignores what happens inside the model by focusing only on what it does but it also requires additional labelled data. In this chapter we use the analytical framework from Chapter 4 and the derived early stopping algorithm (Alg. 3), which we will call topological early stopping (TES), to the problem of facial Action Unit (AU) recognition [1]. This is particularly useful as labelling AUs is costly which makes robust training of highly parametrized models like DNNs problematic. We exemplify the benefits of using TES on the Deep Structure Inference Network (DSIN), the network proposed in Chapter 3. We show that TES is superior in performance to the early stopping with patience, the standard early stopping algorithm in the literature. This proves beneficial for AU recognition performance and provides new insights into how learning and memorizing of AUs occurs in DNNs.

6.1 Introduction

As many other computer vision problems, in recent years, automatic AU recognition has been mainly performed using deep neural networks (DNN). Training DNNs through supervised learning requires a vast number of labelled examples. Unfortunately, AU annotation is an expensive and laborious task: labeling one minute of video can require one hour for a specially trained coder and a laymen may need a long training in order to have the appropriate level of expertise. Even with this set aside, quite often the labels are noisy

[1] The reader is referred to Fig. 6.1 for an illustration of the AUs targeted here.

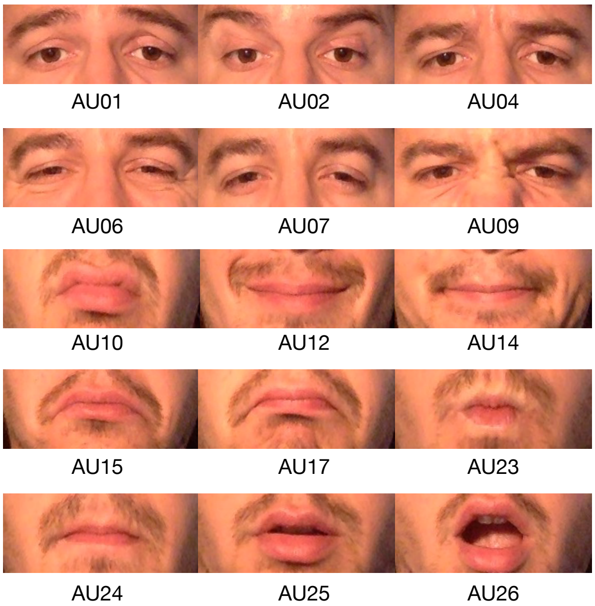


Figure 6.1: Illustration of the Action Units targeted in this chapter. AU01: inner brow raiser, AU02: outer brow raiser, AU04: brow lowerer, AU06: cheek raiser, AU07: lid tightener, AU09: nose wrinkler, AU10: upper lip raiser, AU12: lip corner puller, AU14: dimpler, AU15: lip corner depressor, AU17: chin raiser, AU23: lip tightener, AU25: lips part, AU26: jaw drop.

or simply biased which affects the quality of what the models will learn. Furthermore, as they became increasingly successful, DNNs also grew in complexity and became more opaque, being commonly regarded as "black-box" models. This lead to a paradoxical situation in which, we have models that perform very well, but we do not know how and why.

A desirable property of a transparent DNN is to be *interpretable*. Interpretability is the science of comprehending what a model did (or might have done) [73]. The main focus of this chapter is to improve interpretability of DSIN as it learns to recognize facial AUs. This is beneficial to performance as well, as new insights and increased control allow us to perform early stopping in a novel way. Our main contributions in this chapter are: 1) *Increasing interpretability of DSIN*, by looking into the behaviour of the network to *decide when it transitions from learning to memorizing AUs.* 2) Harness this new insight to *train* the DSIN for facial action recognition *without the need of any validation data* and with improved performance. An overview of the approach is depicted in Fig. 6.2.

The rest of this chapter is structured as follows: we present the experimental setting and discuss results in Sec. 6.2 and we conclude in Sec. 6.3.

[2] Similar to Chapter 3, throughout this chapter we will denote any specific patch prediction network by $PP(< patch >)$, e.g., $PP(beye)$ stands for the network trained on the *between eye* patch.

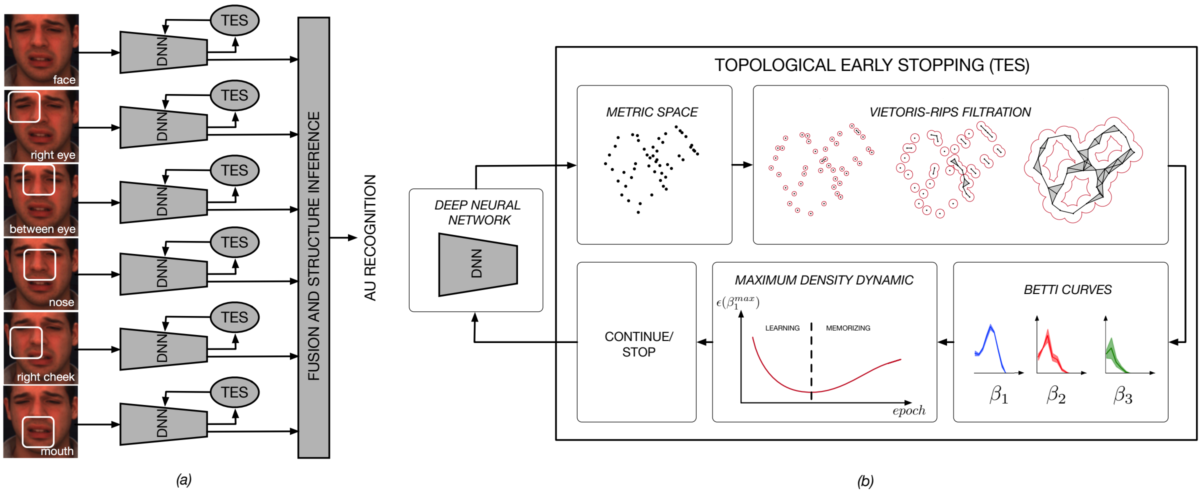


Figure 6.2: (a) DSIN consists of an ensemble of patch prediction (PP) networks [2], that in the first phase learn local facial representations and in a second phase model joint probabilities of labels through fusion and structure inference [34]. (b) We improve control and transparency of the optimization process by performing topological early stopping (TES) with Alg. 3.

6.2 Experimental Settings and Results

We first briefly comment experimental setup followed by a discussion of the main results.

6.2.1 Experimental Settings

We use randomly initialized VGG16 [194] as patch prediction network for DSIN. Even though this is different to the original DSIN from Chapter 3 where a custom network was employed, we show experimentally that the difference in performance is negligible. For the experimental analysis we used BP4D and DISFA, the same two datasets from Chapter 3. A detailed description can be found there. To make things comparable we stick to the training setting already presented in Chapter 3. All experiments are the result of a 3-fold cross-validation. The DSIN is trained incrementally. Each of the PP networks are trained first independently, then together with the immediately superior modules and so on until all the network is trained jointly (see Alg. 1 in Chapter 3). We use two different strategies to decide when to stop the training of each PP network. First we use early stopping with patience (ESP) [16], the standard technique for practitioners of deep learning. We set the patience factor $p \in \{10, 40\}$ and we denote the results by DSIN$_p^{ESP}$. The first value is commonly used in practice, the second was already used in Chapter 3 and it serves for comparison. Second we use Alg. 3 to stop each training. The result is denoted with $DSIN^{TES}$.

At each epoch during the training, we compute $\beta_1(\epsilon)$, and the coordinates of its maximum β_1^{max} and $\epsilon(\beta_1^{max})$. The decision to stop early is based on the dynamic of the

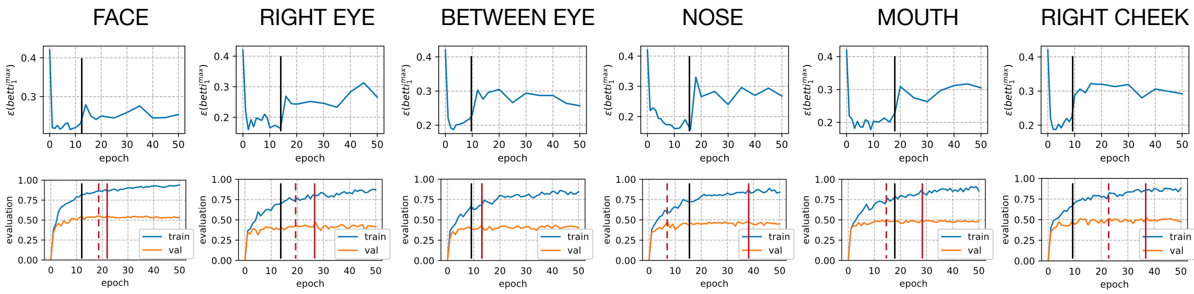


Figure 6.3: Topology dynamic and TES stopping decision for subnetworks of DSIN (upper row) and training/validation evaluation dynamic (bottom row) on the BP4D. On the evaluation dynamic, along with the TES decision (black line) we also mark ESP$_{40}$ (red line) and ESP$_{10}$ (red dashed line) decision.

later, namely the network starts overfitting (transitions from learning to memorizing) when $\epsilon(\beta_1^{max})$ starts regressing back to higher values. An actual illustration of it in practice is shown in Fig. 6.3.

6.2.2 Results and Discussion

Patch Prediction. In Table 6.1 we show the performance per AU of each of the patch prediction (PP) subnetworks of DSIN. As expected PP networks tend to perform better on AUs they see and considerably worse on the ones they do not. It is just a confirmation of the locality characteristic of AUs. In general, each model, infers labels not directly visible by taking into account label structure. While, as expected the best average performance is obtained by learning from the whole face, some PP networks are better for specific AUs. This is the case for five of the seven AUs targeted in BP4D. PP(beye) is best on AU01 (inner brow raiser), PP(reye) is considerably better on AU02, PP(rcheek) on AU14, AU15 and AU25.

Early stopping In Figure 6.3 we illustrate the stopping decisions made by TES, ESP$_{10}$ and ESP$_{40}$. On the BP4D, several interesting observations can be made. On average, ESP$_{10}$ stops 3.33 epochs later than TES, while ESP$_{40}$ 14.7 epochs later. A valid early stopping can be performed without the use of validation data which is reflected by the absolute mean difference in accuracy on the validation dataset between TES and ESP$_{10}$ which is 0.75% nominal and the absolute mean difference between TES and ESP$_{40}$ which is 0.91% nominal. On the other hand, if we consider the performance on the training set, on average the training accuracy when ESP$_{10}$ stops is 6.03% higher then when TES stops, while for ESP$_{40}$ it is 9.85%. The main exception is for ESP$_{10}$ in the case of PP(nose) and PP(mouth). Because there is no significant difference in validation performance between ESP and TES decision to stop, but considerable training performance differences, we argue that the extra training iterations are used for memorising training samples. In Figure 6.4 we give a few examples of such cases. Observe that most of the memorized samples are

method	AU01	AU02	AU04	AU06	AU07	AU10	AU12	AU14	AU15	AU17	AU23	AU24	avg
PP(faces)	[48.1]	[28.2]	**46.4**	**75.6**	**71.2**	**82.4**	**84.6**	53.8	30.7	**62.8**	**39.9**	[36.0]	**55.0**
PP(reye)	37.4	**35.2**	39.0	65.3	[70.0]	71.6	73.9	60.2	22.0	48.4	17.8	20.7	46.8
PP(beye)	**49.4**	25.5	[46.1]	63.0	68.5	65.9	69.3	52.6	18.2	36.0	16.9	8.7	43.4
PP(mouth)	30.0	16.1	28.4	65.2	69.1	[80.7]	81.2	[62.0]	[32.3]	[60.8]	[37.8]	34.1	49.8
PP(nose)	43.7	19.7	41.6	69.4	69.9	78.1	79.5	57.2	19.4	44.3	26.6	25.2	47.9
PP(rcheek)	36.5	25.3	38.7	[71.9]	67.7	79.9	[83.0]	**63.6**	**33.8**	50.9	36.7	**41.2**	[52.4]

Table 6.1: Recognition results on BP4D per patch. For each AU, best result is marked in bold and second best result marked between square brackets.

especially for difficult AUs, where in any case the model had problems learning either because of noisy labels, noisy input or unbalanced classes. For example AU04 in Fig. 6.4(a,c,d), AU17 in Fig. 6.4(b), AU10 in Fig. 6.4(e) or AU1 in Fig. 6.4(a) are good examples of noisy labels. There is no visual cue that could indicate these AUs. Refer to Fig. 6.1 for comparison. The only way that the DSIN could predict these labels is by memorizing the particular mapping between the input and the label. Another interesting example can be found in Fig. 6.4(f) where the input itself is very noisy due to poor illumination.

Comparison With State-of-the-Art. In Table 6.2 and Table 6.3 we show final results of DSIN trained using TES and ESP compared to some state-of-the-art methods. Notice that even without the use of validation data DSIN$_{TES}$ obtains competitive or better results. Because of the addition of fusion and structure inference on top of the PP networks there is considerable benefit to some AUs like AU04 and AU14 for BP4D. These are AUs that benefit from the structure inference as they are highly correlated (positively or negatively with other) with other AUs (for the pairwise correlations refer to Figure 3.11) and from fusing the PPs as is the case of AU4 where the dedicated network PP(mouth) has considerable better performance than the holistic PP(face). If we were to compare the two ways of training a new set of observations could be made. For example for BP4D (Table 6.2), it is interesting to notice that in the case of AU02 the improvement is significant. We hypothesise that this might be the result of the earlier stopping especially in the case of PP(face), PP(reye) and PP(rcheek) that saves unnecessary memorization in the networks and permits better learning in the later stages of fusion and structure learning. Compare this with the result for AU04, where DSIN$_{10}^{ESP}$ is outperforming both DSIN$_{40}^{ESP}$ and DSIN^{TES}. This might be related to the fact that ESP stops earlier than the other two for PP(nose) patch related to this AU. In the case of DISFA on the other hand (Table 6.3), there is significant improvement especially for AU02 which again might corresponds to the fact that networks trained on upper face patches are stopped from memorizing earlier by the TES. Finally, in the case of DSIN^{TES} it is important to keep in mind that the validation data is not needed. Instead, this could be used for increasing the training partition with further additional benefits for performance and generalization.

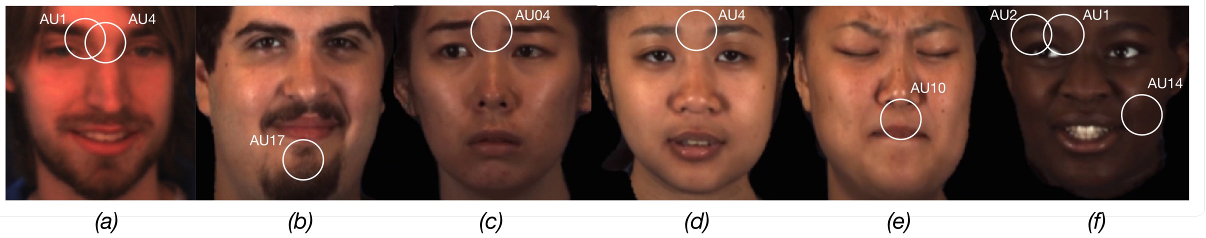


Figure 6.4: After a sufficient number of iterations, if the training continues, the generalization gap increases without any significant improvement on the validation set. We show here some examples of noisy labels that the dedicated networks of DSIN memorize between the epoch TES would stop and the epoch ESP would stop.

method	AU01	AU02	AU04	AU06	AU07	AU10	AU12	AU14	AU15	AU17	AU23	AU24	AVG
JPML [249]	32.6	25.6	37.4	42.3	50.5	72.2	74.1	**65.7**	38.1	40.0	30.4	42.3	45.9
DRML [250]	36.4	41.8	43.0	55.0	67.0	66.3	65.8	54.1	33.2	48.0	31.7	30.0	48.3
CPM [241]	43.4	40.7	43.3	59.2	61.3	62.1	68.5	52.5	36.7	54.3	39.5	37.8	50.0
ROI [119]	36.2	31.6	43.4	**77.1**	73.7	**85.0**	**87.0**	62.6	**45.7**	58.0	38.3	37.4	56.4
DSIN$^{ESP}_{40}$	[49.9]	41.2	54.1	73.1	73.4	80.0	[84.9]	[63.5]	35.2	63.1	[42.1]	41.6	58.5
DSIN$^{ESP}_{10}$	49.7	[42.5]	**56.6**	72.0	[74.7]	[81.1]	82.2	62.2	36.5	[63.9]	40.1	**43.3**	[58.7]
DSIN^{TES}	**50.4**	**44.3**	[56.2]	[73.3]	**75.6**	79.3	83.2	61.2	[42.7]	**65.2**	**44.2**	[43.1]	**59.9**

Table 6.2: AU recognition results on BP4D. Best results are shown in bold. Second best results are shown in brackets.

method	AU01	AU02	AU04	AU06	AU09	AU12	AU25	AU26	avg
APL[255]	11.4	12.0	30.1	12.4	10.1	65.9	21.4	26.0	23.8
DRML [250]	17.3	17.7	37.4	29.0	10.7	37.7	38.5	20.1	26.7
ROI [119]	41.5	26.4	[66.4]	**50.7**	8.5	**89.3**	88.9	15.6	48.5
DSIN$^{ESP}_{40}$	**45,3**	38.0	65.2	29.4	[42.8]	[73.8]	**90.2**	**41.5**	[53.3]
DSIN$^{ESP}_{10}$	43,2	[39.1]	**67.3**	31.2	42.6	73.5	[89.1]	40.3	[53.3]
DSIN^{TES}	[44.4]	**43.6**	64.8	[33.1]	**43.1**	72.2	88.0	[41.3]	**53.8**

Table 6.3: AU recognition results on DISFA. Best results are shown in bold. Second best results are shown in brackets.

6.3 Conclusion

In this final chapter, we studied the functional topology of the DSIN while it learns to recognize facial AU recognition. We show that even without a separated validation data partition, the DSIN can be stopped correctly during training with significant benefits for the performance of the network. Also, it is a first glimpse into what a DNN does when learning AUs, an important step towards increased transparency and interpretability in facial AU recognition with deep neural networks.

Chapter 7

Conclusion

Facial expressions are fundamental for human communication in social contexts. One of the most important ways of coding facial expressions is by using facial action units, a set of local micro-expressions defined by muscular facial activity. In this thesis we tackle the problem of automatic facial action unit recognition from images. We first proposed the Deep Structure Inference Network, a custom deep neural network specially designed to learn local facial representations and to model output structure. We then define a novel generic theoretical framework based on algebraic topology for analyzing deep neural networks. We show that networks that learn are topologically different from networks that memorize and we propose an early stopping algorithm that does not need validation data. Finally, we applied these new insights to the Deep Structure Inference Network for more performant and more transparent facial action unit recognition.

7.1 Contributions

The complete set of contributions in this thesis are the following:

1. **Automatic Facial Expression Recognition. General Framework, Evolutionary Perspective and Trends.**
   (a) An evolutionary perspective of affect inference from the face (Chapter 2.2).
   (b) Definition of comprehensive taxonomy of automatic computer vision approaches to automatic facial expression recognition (Chapter 2.3).
   (c) Extended survey of historical and current trends in AFER (Chapter. 2.4).
2. **Performance in Facial Expression Recognition.**
   (a) Proposal of a model that learns representation, patch and output structure of the face end-to-end (Chapter 3.3).

(b) Introduction of a structure inference topology that replicates inference algorithm in probabilistic graphical models by using a recurrent neural network (Chapter 3.3).

(c) Extended ablation study and experimental analysis of the newly proposed architecture (Chapter 3.4).

3. **Interpretability in Facial Expression Recognition.**

(a) Formulation of novel general framework for analysis of deep neural networks based on algebraic topology (Chapter 4).

(b) Analysis of fundamental topological differences between DNNs that learn and DNNs that memorize (Chapter 5).

(c) Analyze and improving performance of the previously proposed architecture for facial expression architecture using the new theoretical framework (Chapter 6.)

7.2 Future Work

The Deep Structure Inference Network could be improved in several ways. First, at least initial parts of the network are redundant, there is a lot of room to optimize the size of the model while keeping its performance. Second, there is also considerable redundancy in the fusion modules, a slimmer, more simple architecture could be easily envisioned. Third, several additions would probably make it more robust. For example facial geometry or person specific features could be used. With the analytical framework proposed we have barely scratched the surface of what could be done. First, there is a lot of room for improvement in terms of computation cost. Any advance would greatly increase its usability in real world scenarios. Then, one could ask very interesting additional questions. For example, if we can predict transition towards overfitting using topology, could we get even further by regressing the generalization gap? If we know that deep networks that learn have specific topological properties, could a topological criterion guide optimization of neural networks without the need of labelled observations? Finally, in terms of explainability of facial action unit recognition, it is still not clear when a network is biased, and why it is taking a certain decision or not. Hopefully, future research will shed light on at least some of these questions.

Bibliography

[1] www.affectiva.com.

[2] https://deepdreamgenerator.com. Accessed: 2018-11-16.

[3] www.emotient.com.

[4] http://www.vcipl.okstate.edu/otcbvs/bench/.

[5] www.kairos.com.

[6] http://www.equinoxsensors.com/.

[7] www.realeyesit.com.

[8] what-when-how.com.

[9] Mojtaba Khomami Abadi, Juan Abdón Miranda Correa, Julia Wache, Heng Yang, Ioannis Patras, and Nicu Sebe. Inference of personality traits and affect schedule by analysis of spontaneous reactions to affective videos. *FG*, 2015.

[10] P. S. Aleksic and A. K. Katsaggelos. Automatic facial expression recognition using facial animation parameters and multistream hmms. *TIFS*, 1(1):3–11, 2006.

[11] Ahmed B. Ashraf, Simon Lucey, Jeffrey F. Cohn, Tsuhan Chen, Zara Ambadar, Kenneth M. Prkachin, and Patricia E. Solomon. The painful face–pain expression recognition using active appearance models. *IVC*, 27(12):1788–1796, 2009.

[12] Mathieu Aubry and Bryan C Russell. Understanding deep features with computer-generated imagery. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2875–2883. IEEE, 2015.

[13] Sander Bakkes, Chek Tien Tan, and Yusuf Pisan. Personalised gaming. *JCT*, 3, 2012.

[14] L. F. Barrett. Was Darwin wrong about emotional expressions? *CDPS*, 20(6): 400–406, 2011.

[15] Ligia Maria Batrinca, Nadia Mana, Bruno Lepri, Fabio Pianesi, and Nicu Sebe. Please, tell me about yourself: automatic personality assessment using short self-presentations. In *ICMI*, pages 255–262, 2011.

[16] Yoshua Bengio. Practical recommendations for gradient-based training of deep architectures. *Neural Networks: Tricks of the Trade*, pages 437–478, 2012.

[17] Stefano Berretti, Boulbaba Ben Amor, Mohamed Daoudi, and Alberto Del Bimbo. 3D facial expression recognition using sift descriptors of automatically detected keypoints. *TVC*, 27(11):1021–1036, 2011.

[18] Joan-Isaac Biel and Daniel Gatica-Perez. The youtube lens: Crowdsourced personality impressions and audiovisual analysis of vlogs. *Multimedia*, 15(1):41–55, 2013.

[19] Joan-Isaac Biel, Lucía Teijeiro-Mosquera, and Daniel Gatica-Perez. Facetube: predicting personality from facial expressions of emotion in online conversational video. In *ICMI*, pages 53–56, 2012.

[20] Joan-Isaac Biel, Vagia Tsiminaki, John Dines, and Daniel Gatica-Perez. Hi youtube!: Personality impressions and verbal content in social video. In *ICMI*, pages 119–126, 2013.

[21] Paris Mavromoustakos Blom, Sander Bakkes, Chek Tien Tan, Shimon Whiteson, Diederik Roijers, Roberto Valenti, and Theo Gevers. Towards personalised gaming via facial expression recognition. In *AIIDE*, 2014.

[22] Nigel Bosch, Sidney D'Mello, Ryan Baker, Jaclyn Ocumpaugh, Valerie Shute, Matthew Ventura, Lubin Wang, and Weinan Zhao. Automatic detection of learning-centered affective states in the wild. In *Proceedings IUI*, pages 379–388, 2015.

[23] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.

[24] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on Fairness, Accountability and Transparency*, pages 77–91, 2018.

[25] A. Burrows and J. F. Cohn. Comparative anatomy of the face. In *Handbook of biometrics*, pages 1–10. Springer, 2nd edition, 2014.

[26] George Caridakis, Lori Malatesta, Loic Kessous, Noam Amir, Amaryllis Raouzaiou, and Kostas Karpouzis. Modeling naturalistic affective states via facial and vocal expressions recognition. In *ICMI*, pages 146–154, 2006.

[27] Davide Castelvecchi. Can we open the black box of ai? *Nature News*, 538(7623):20, 2016.

[28] Ya Chang, Marcelo Vieira, Matthew Turk, and Luiz Velho. Automatic 3D facial expression analysis in videos. *AMFG*, pages 293–307, 2005.

[29] Andre Chastel. *Leonardo on Art and the Artist*. Courier Corporation, 2002.

[30] Xiao Chu, Wanli Ouyang, Xiaogang Wang, et al. Crf-cnn: Modeling structured information in human pose estimation. In *Advances in Neural Information Processing Systems*, pages 316–324, 2016.

[31] I. Cohen, N. Sebe, F. G. Gozman, M. C. Cirelo, and T. S. Huang. Learning bayesian network classifiers for facial expression recognition both labeled and unlabeled data. In *CVPR*, pages I–595–I–601, 2003.

[32] Ira Cohen, Nicu Sebe, Larry Chen, Ashutosh Garg, and Thomas S. Huang. Facial expression recognition from video sequences: Temporal and static modelling. In *CVIU*, pages 160–187, 2003.

[33] J. F. Cohn, T. S. Kruez, I. Matthews, Ying Yang, Minh H. Nguyen, M. T. Padilla, Feng Zhou, and F. De la Torre. Detecting depression from facial actions and vocal prosody. In *ACII*, pages 1–7, 2009.

[34] Ciprian Corneanu, Meysam Madadi, and Sergio Escalera. Deep structure inference network for facial action unit recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 298–313, 2018.

[35] Arnaud Dapogny, Kevin Bailly, and Severine Dubuisson. Dynamic facial expression recognition by joint static and multi-time gap transition classification. In *FG*, 2015.

[36] C. Darwin. *The expression of emotion in man and animals*. Oxford University Press, 1872.

[37] G-B Duchenne de Boulogne and R Andrew Cuthbertson. *The Mechanism of Human Facial Expression*. Cambridge University Press, 1990.

[38] Zhiwei Deng, Arash Vahdat, Hexiang Hu, and Greg Mori. Structure inference machines: Recurrent neural networks for analyzing relations in group activity recognition. In *IEEE CVPR*, pages 4772–4781, 2016.

[39] D. DeVault, R. Artstein, G. Benn, T. Dey, E. Fast, A. Gainer, and L.-P. Morency. A virtual human interviewer for healthcare decision support. *AAMAS*, 2014.

[40] Abhinav Dhall, Akshay Asthana, Roland Goecke, and Tom Gedeon. Emotion recognition using phog and lpq features. In *FG*, pages 878–883, 2011.

[41] Abhinav Dhall, Roland Goecke, Simon Lucey, and Tom Gedeon. Acted facial expressions in the wild database. Technical report, Australian Nat. U., 2011.

[42] Abhinav Dhall, Roland Goecke, Simon Lucey, and Tom Gedeon. Static facial expression analysis in tough conditions: Data, evaluation protocol and benchmark. In *ICCV Workshops*, pages 2106–2112, 2011.

[43] Abhinav Dhall, Roland Goecke, Jyoti Joshi, Karan Sikka, and Tom Gedeon. Emotion recognition in the wild challenge 2014: Baseline, data and protocol. In *ICMI*, pages 461–466, 2014.

[44] Abhinav Dhall, Jyoti Joshi, Karan Sikka, Roland Goecke, and Nicu Sebe. The more the merrier: Analysing the affect of a group of people in images. In *FG*, 2015.

[45] Zoran Duric, Wayne D Gray, Ric Heishman, Fayin Li, Azriel Rosenfeld, Michael J Schoelles, Christian Schunn, and Harry Wechsler. Integrating perceptual and cognitive modeling for adaptive and intelligent human-computer interaction. *Proceedings of the IEEE*, 90(7):1272–1289, 2002.

[46] I. Eibl-Eibesfeldt. *Human ethology.* 1989.

[47] I. Eibl-Eibesfeldt. An argument for basic emotions. In *Cogn. Emot.*, pages 169–200. 1992.

[48] P. Ekman. Universal and cultural differences in facial expression of emotion. *Nebr. Sym. Motiv.*, 19:207–283, 1971.

[49] P. Ekman. Strong evidence for universals in facial expressions: A reply to Russell's mistaken critique. *Psychol. Bull.*, 115(2):268–287, 1994.

[50] P. Ekman and H. Oster. Facial expressions of emotion. *Annu. Rev. Psychol.*, (30): 527–554, 1979.

[51] P. Ekman and E. Rosenberg. *What the face reveals.* 2nd edition, 2005.

[52] P. Ekman, R. J. Davidson, and W. V. Friesen. The Duchenne smile: Emotional expression and brain psychology ii. *JPSP*, 58(2):342–353, 1990.

[53] Paul Ekman and Wallace V. Friesen. *Facial Action Coding System: A Technique for the Measurement of Facial Movement.* Consulting Psychologists Press, 1978.

[54] Paul Ekman, Thomas S Huang, Terrence J Sejnowski, and Joseph C Hager. Final report to NSF of the planning workshop on facial expression understanding. *Human Interaction Lab*, 378, 1993.

[55] Paul Ekman, W. Friesen, and J. Hager. Facs manual. a human face. 2002.

[56] Paul Ekman, Wallace V. Friesen, and Joseph C. Hager. *Facial Action Coding System: The Manual on CD ROM. A Human Face*, 2002.

[57] Rana El Kaliouby and Peter Robinson. Real-time inference of complex mental states from facial expressions and head gestures. In *Real-time vision for human-computer interaction*, pages 181–200. Springer, 2005.

[58] Stefanos Eleftheriadis, Ognjen Rudovic, and Maja Pantic. Multi-conditional latent variable model for joint facial action unit detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3792–3800, 2015.

[59] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1625–1634, 2018.

[60] C. Fabian Benitez-Quiroz, Yan Wang, and Aleix M. Martinez. Recognition of action units in the wild with deep nets and a new global-local loss. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[61] Tianhong Fang, Xi Zhao, S. K. Shah, and I. A. Kakadiaris. 4d facial expression recognition. In *ICCV*, pages 1594–1601, 2011.

[62] B. Fasel and Juergen Luettin. Automatic facial expression analysis: a survey. *PR*, 36(1):259–275, 2003.

[63] Ruth C Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. *arXiv preprint arXiv:1704.03296*, 2017.

[64] N Fragopanagos and John G Taylor. Emotion recognition in human–computer interaction. *Neural Net.*, 18(4):389–405, 2005.

[65] A. J. Fridlund. The behavioral ecology and sociality of human faces. In *Emotion*, pages 90–121. 1997.

[66] W. V. Friesen and P. Ekman. Emfacs-7: Emotional facial action coding system. *U. California*, 2:36, 1983.

[67] N. H. Frijda and A. Tcherkassof. Facial expressions as modes of action readiness. In *The psychology of facial expression*, pages 78–102. 2nd edition, 1997.

[68] Siddha Ganju, Olga Russakovsky, and Abhinav Gupta. Whats in a question: Using visual questions as a form of supervision. *arXiv preprint arXiv:1704.03895*, 2017.

[69] Nikhil Garg, Londa Schiebinger, Dan Jurafsky, and James Zou. Word embeddings quantify 100 years of gender and ethnic stereotypes. *Proceedings of the National Academy of Sciences*, 115(16):E3635–E3644, 2018.

[70] A Geetha, Vennila Ramalingam, S Palanivel, and B Palaniappan. Facial expression recognition–a real time approach. *Expert Syst. Appl.*, 36(1):303–308, 2009.

[71] Tobias Gehrig and Hazım Kemal Ekenel. Why is facial expression analysis in the wild challenging? In *ICMI Workshops*, pages 9–16, 2013.

[72] Dileep George, Wolfgang Lehrach, Ken Kansky, Miguel Lázaro-Gredilla, Christopher Laan, Bhaskara Marthi, Xinghua Lou, Zhaoshi Meng, Yi Liu, Huayan Wang, et al. A generative vision model that trains with high data efficiency and breaks text-based captchas. *Science*, 358(6368):eaag2612, 2017.

[73] Leilani H Gilpin, David Bau, Ben Z Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*, pages 80–89. IEEE, 2018.

[74] J. M. Girard, J. F. Cohn, M. A. Sayette, L. A. Jeni, and F. De la Torre. Spontaneous facial expression can be measured automatically. *Beh. Res. Meth.*, 2014.

[75] Jeffrey M Girard, Jeffrey F Cohn, Mohammad H Mahoor, S Mohammad Mavadati, Zakia Hammal, and Dean P Rosenwald. Nonverbal social withdrawal in depression: Evidence from manual and automatic analyses. *IVC*, 32(10):641–647, 2014.

[76] Boqing Gong, Yueming Wang, Jianzhuang Liu, and Xiaoou Tang. Automatic facial expression recognition on a single 3D face by exploring shape deformation. In *ICM*, pages 569–572, 2009.

[77] H. Gray and C. M. Goss. *Anatomy of the human body.* Lea & Febiger, 28th edition, 1966.

[78] Stephen Greenblatt et al. Toward a universal language of motion: reflections on a seventeenth century muscle man. 1994.

[79] M. Greenwald, E. Cook, and P. Lang. Affective judgment and psychophysiological response: Dimensional covariation in the evaluation of pictorial stimuli. *J. Psychophysiology*, (3):51–64, 1989.

[80] Wenfei Gu, Cheng Xiang, YV Venkatesh, Dong Huang, and Hai Lin. Facial expression recognition using radial encoding of local gabor features and classifier synthesis. *PR*, 45(1):80–91, 2012.

[81] Munawar Hayat, Mohammed Bennamoun, and Amar A El-Sallam. Clustering of video-patches on grassmannian manifold for facial expression recognition from 3D videos. In *WACV*, pages 83–88, 2013.

[82] Lang He, Dongmei Jiang, Le Yang, Ercheng Pei, Peng Wu, and Hichem Sahli. Multimodal affective dimension prediction using deep bidirectional long short-term memory recurrent neural networks. In *AVEC*, pages 73–80, 2015.

[83] Shan He, Shangfei Wang, Wuwei Lan, Huan Fu, and Qiang Ji. Facial expression recognition using deep boltzmann machine from thermal infrared images. In *ACII*, pages 239–244, 2013.

[84] Benjamín Hernández, Gustavo Olague, Riad Hammoud, Leonardo Trujillo, and Eva Romero. Visual learning of texture descriptors for facial expression recognition in thermal imagery. *CVIU*, 106(2):258–269, 2007.

[85] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6):82–97, 2012.

[86] Geoffrey E. Hinton, Simon Osindero, and Yee W. Teh. A fast learning algorithm for deep belief nets. *Neural Comput.*, 18(7):1527–1554, 2006.

[87] Earnest Paul Ijjina and C Krishna Mohan. Facial expression recognition using kinect depth sensor and convolutional neural networks. In *ICMLA*, pages 392–396, 2014.

[88] Ramin Irani, Kamal Nasrollahi, Marc O Simon, Ciprian A Corneanu, Sergio Escalera, Chris Bahnsen, Dennis H Lundtoft, Thomas B Moeslund, Tanja L Pedersen, Maria-Louise Klitgaard, et al. Spatiotemporal analysis of rgb-dt facial images for multimodal pain level recognition. *CVPR Workshops*, 2015.

[89] Hiroshi Ishiguro, Tetsuo Ono, Michita Imai, Takeshi Maeda, Takayuki Kanda, and Ryohei Nakatsu. Robovie: an interactive humanoid robot. *Industrial robot: An international journal*, 28(6):498–504, 2001.

[90] C. E. Izard. *The face of emotion.* 1971.

[91] C. E. Izard. *Maximally discriminative facial movement coding system (MAX).* Instructional Resources Center, University of Delaware, 1983.

[92] Dougherty L. M. Hembree E. A. Izard, C. E. *A system for identifying affect expressions by holistic judgments.* Instructional Resources Center, University of Delaware, 1983.

[93] R. E. Jack, C. Blais, C. Scheepers, P. G. Schyns, and R. Caldara. Cultural confusions show that facial expressions are not universal. *Current Biology*, 19:1–6, 2009.

[94] Varun Jain, James L Crowley, Anind K Dey, and Augustin Lux. Depression estimation using audiovisual features and fisher vector encoding. In *AVEC*, pages 87–91, 2014.

[95] Shashank Jaiswal and Michel Valstar. Deep learning the dynamic appearance and shape of facial action units. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pages 1–8. IEEE, 2016.

[96] Qiang Ji, Peilin Lan, and Carl Looney. A probabilistic framework for modeling and real-time monitoring human fatigue. *SMC-A*, 36(5):862–875, 2006.

[97] Ulf Johansson, Cecilia Sönströd, Ulf Norinder, and Henrik Boström. Trade-off between accuracy and interpretability for predictive in silico modeling. *Future medicinal chemistry*, 3(6):647–663, 2011.

[98] Jyoti Joshi, Abhinav Dhall, Roland Goecke, Michael Breakspear, and Gordon Parker. Neural-net classification for spatio-temporal descriptor based depression analysis. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 2634–2638. IEEE, 2012.

[99] Mase K. and Pentland A. Automatic lipreading by optical-flow analysis. *SCJ*, 22, 1991.

[100] Samira Ebrahimi Kahou, Christopher Pal, Xavier Bouthillier, Pierre Froumenty, Çaglar Gülçehre, Roland Memisevic, Pascal Vincent, Aaron Courville, Yoshua Bengio, Raul Chandias Ferrari, et al. Combining modality specific deep neural networks for emotion recognition in video. In *ICMI*, pages 543–550, 2013.

[101] Sebastian Kaltwang, Ognjen Rudovic, and Maja Pantic. Continuous pain intensity estimation from facial expressions. *ISVC*, pages 368–377, 2012.

[102] Takeo Kanade, Jeffrey F Cohn, and Yingli Tian. Comprehensive database for facial expression analysis. In *FG*, pages 46–53, 2000.

[103] Ashish Kapoor, Winslow Burleson, and Rosalind W Picard. Automatic prediction of frustration. *IJHCS*, 65(8):724–736, 2007.

[104] Vahid Kazemi and Josephine Sullivan. One millisecond face alignment with an ensemble of regression trees. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1867–1874, 2014.

[105] D. Keltner and P. Ekman. Facial expression of emotion. In *Handbook of emotions*, pages 236–249. 2nd edition, 2000.

[106] Yasunari Koda, Yasunari Yoshitomi, Mari Nakano, and Masayoshi Tabuse. A facial expression recognition for a speaker of a phoneme of vowel using thermal image processing and a speech recognition system. In *RO-MAN*, pages 955–960, 2009.

[107] S. Koelstra, M. Pantic, and I. Patras. A dynamic texture-based approach to recognition of facial actions and their temporal models. *TPAMI*, 32(11):1940–1954, 2010.

[108] Christian G Kohler, Elizabeth A Martin, Neal Stolar, Fred S Barrett, Ragini Verma, Colleen Brensinger, Warren Bilker, Raquel E Gur, and Ruben C Gur. Static posed and evoked facial expressions of emotions in schizophrenia. *Schizophr. Res.*, 105 (1-3):49–60, 2008.

[109] I. Kotsia and I. Pitas. Facial expression recognition in image sequences using geometric deformation features and support vector machines. *TIP*, 16:172–187, 2007.

[110] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[111] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[112] Kaustubh Kulkarni, Ciprian Adrian Corneanu, Ikechukwu Ofodile, Sergio Escalera, Xavier Baro, Sylwia Hyniewska, Juri Allik, and Gholamreza Anbarjafari. Automatic recognition of deceptive facial expressions of emotion. *arXiv preprint arXiv:1707.04061*, 2017.

[113] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

[114] Vuong Le, Hao Tang, and T. S. Huang. Expression recognition from 3D dynamic faces using robust spatio-temporal shape features. In *FG*, pages 414–421, 2011.

[115] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[116] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553): 436, 2015.

[117] P. Lemaire, M. Ardabilian, Liming Chen, and M. Daoudi. Fully automatic 3D facial expression recognition using differential mean curvature maps and histograms of oriented gradients. In *FG*, pages 1–7, 2013.

[118] R. W. Levenson, P. Ekman, and W. V. Friesen. Voluntary facial action generates emotion-specific autonomic nervous system activity. *Psychophysiology*, 27(4):363–384, 1990.

[119] Wei Li, Farnaz Abtahi, and Zhigang Zhu. Action unit detection with region adaptation, multi-labeling learning and optimal temporal fusing. In *CVPR*, pages 6766–6775. IEEE, 2017.

[120] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[121] Zachary C Lipton. The mythos of model interpretability. *arXiv preprint arXiv:1606.03490*, 2016.

[122] Gwen Littlewort, M. S. Bartlett, Ian Fasel, J. Susskind, and J. Movellan. Dynamics of facial expression extracted automatically from video. In *CVPR Workshops*, page 80, 2004.

[123] Gwen Littlewort, Jacob Whitehill, Tingfan Wu, Ian Fasel, Mark Frank, Javier Movellan, and Marian Bartlett. The computer expression recognition toolbox (cert). In *FG*, pages 298–305, 2011.

[124] Gwen C Littlewort, Marian Stewart Bartlett, and Kang Lee. Faces of pain: automated measurement of spontaneousallfacial expressions of genuine and posed pain. In *ICMI*, pages 15–21, 2007.

[125] Gwen C. Littlewort, Marian S. Bartlett, and Kang Lee. Automatic coding of facial expressions displayed during posed and genuine pain. *IVC*, 27(12):1797–1803, 2009.

[126] Mengyi Liu, Shaoxin Li, Shiguang Shan, and Xilin Chen. Au-aware deep networks for facial expression recognition. In *FG*, pages 1–6, 2013.

[127] Mengyi Liu, Ruiping Wang, Zhiwu Huang, Shiguang Shan, and Xilin Chen. Partial least squares regression on grassmannian manifold for emotion recognition. In *ICMI*, pages 525–530, 2013.

[128] Mengyi Liu, Shiguang Shan, Ruiping Wang, and Xilin Chen. Learning expressionlets on spatio-temporal manifold for dynamic facial expression recognition. In *CVPR*, pages 1749–1756, 2014.

[129] Mengyi Liu, Ruiping Wang, Shaoxin Li, Shiguang Shan, Zhiwu Huang, and Xilin Chen. Combining multiple kernel methods on riemannian manifold for emotion recognition in the wild. In *ICMI*, pages 494–501, 2014.

[130] Peng Liu and Lijun Yin. Spontaneous facial expression analysis based on temperature changes and head motions. In *FG*, 2015.

[131] Ping Liu, Shizhong Han, Zibo Meng, and Yan Tong. Facial expression recognition via a boosted deep belief network. In *CVPR*, pages 1805–1812, 2014.

[132] Zhilei Liu and Shangfei Wang. Emotion recognition using hidden markov models from facial temperature sequence. In *ACII*, pages 240–247. 2011.

[133] Gale M Lucas, Jonathan Gratch, Stefan Scherer, Jill Boberg, and Giota Stratou. Towards an affective interface for assessment of psychological distress.

[134] Gale M Lucas, Jonathan Gratch, Stefan Scherer, Jill Boberg, and Giota Stratou. Towards an affective interface for assessment of psychological distress. *ACII*, 2015.

[135] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In *CVPR Workshops*, pages 94–101, 2010.

[136] P. Lucey, J. F. Cohn, I. Matthews, S. Lucey, S. Sridharan, J. Howlett, and K. M. Prkachin. Automatically detecting pain in video through facial action units. *SMC-B*, 41(3):664–674, 2011.

[137] Simon Lucey, Ahmed Bilal Ashraf, and Jeffrey F Cohn. *Investigating spontaneous facial action recognition through aam representations of the face*. INTECH, 2007.

[138] Michael J. Lyons, Julien Budynek, and Shigeru Akamatsu. Automatic classification of single facial images. *TPAMI*, 21(12):1357–1362, 1999.

[139] A. Maalej, Boulbaba Amor, M. Daoudi, Anuj Srivastava, and Stefano Berretti. Shape analysis of local facial patches for 3D facial expression recognition. *PR*, 44(8): 1581–1589, 2011.

[140] Ludo Maat and Maja Pantic. Gaze-x: adaptive, affective, multimodal interface for single-user office scenarios. In *Artifical Intelligence for Human Computing*, pages 251–271. Springer, 2007.

[141] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5188–5196, 2015.

[142] Aleix Martinez and Shichuan Du. A model of the perception of facial expressions of emotion by humans: Research overview and perspectives. *JMLR*, 13(1):1589–1608, 2012.

[143] David Matsumoto, Dacher Keltner, Michelle N. Shiota, Maureen O'Sullivan, and Mark Frank. Facial expressions of emotion. In *Handbook of Emotions*, chapter 13, pages 211–234. 2008.

[144] S Mohammad Mavadati, Mohammad H Mahoor, Kevin Bartlett, Philip Trinh, and Jeffrey F Cohn. Disfa: A spontaneous facial action intensity database. *IEEE Transactions on Affective Computing*, 4(2):151–160, 2013.

[145] Seyed Mohammad Mavadati, Mohammad H Mahoor, Kevin Bartlett, Philip Trinh, and Jeffrey F Cohn. Disfa: A spontaneous facial action intensity database. *TAC*, 4 (2):151–160, 2013.

[146] Daniel McDuff, Rana El Kaliouby, Thibaud Senechal, May Amr, Jeffrey F. Cohn, and Rosalind Picard. Amfed facial expression dataset: Naturalistic and spontaneous facial expressions collected "in-the-wild". In *CVPR Workshops*, pages 881–888, 2013.

[147] Daniel McDuff, Rana El Kaliouby, Thibaud Senechal, David Demirdjian, and Rosalind Picard. Automatic measurement of ad preferences from facial responses gathered over the internet. *IVC*, 32(10):630–640, 2014.

[148] Gary McKeown, Michel Valstar, Roddy Cowie, Maja Pantic, and Marc Schröder. The semaine database: Annotated multimodal records of emotionally colored conversations between a person and a limited agent. *TAC*, 3(1):5–17, 2012.

[149] Melanie Mitchell. Artificial intelligence hits the barrier of meaning. *New York Times, Op-Ed*, 2018.

[150] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2574–2582, 2016.

[151] Iordanis Mpiperis, Sotiris Malassiotis, Vassilios Petridis, and Michael G Strintzis. 3D facial expression recognition using swarm intelligence. In *ICASSP*, pages 2133–2136, 2008.

[152] Tara N. Sainath, Abdel-rahman Mohamed, Brian Kingsbury, and Bhuvana Ramabhadran. Deep convolutional neural networks for lvcsr. *Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on*, pages 8614–8618, 05 2013.

[153] Hung Nguyen, Kazunori Kotani, Fan Chen, and Bac Le. A thermal facial emotion database and its analysis. In *PSIVT*, pages 397–408. 2014.

[154] Jérémie Nicolle, Vincent Rapp, Kévin Bailly, Lionel Prevost, and Mohamed Chetouani. Robust continuous prediction of human emotions using multiscale dynamic cues. In *ICMI*, pages 501–508, 2012.

[155] Paula M Niedenthal. Embodying emotion. *Science*, 116:1002–1005, 2007.

[156] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2(11):e7, 2017.

[157] Alice J OToole, Carlos D Castillo, Connor J Parde, Matthew Q Hill, and Rama Chellappa. Face space representations in deep convolutional neural networks. *Trends in cognitive sciences*, 2018.

[158] M. Pantic and M. Bartlett. Machine analysis of facial expressions. In *Face Recognition*, pages 377–416. I-Tech Education and Publishing, 2007.

[159] M. Pantic and L. J. M. Rothkrantz. Automatic analysis of facial expressions: the state of the art. *TPAMI*, 22(12):1424–1445, 2000.

[160] Maja Pantic and Ioannis Patras. Dynamics of facial expression: recognition of facial actions and their temporal segments from face profile image sequences. *SMC-B*, 36: 433–449, 2006.

[161] Montse Pardàs and Antonio Bonafonte. Facial animation parameters extraction and expression detection using hmm. In *SPIC*, pages 675–688, 2002.

[162] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *British Machine Vision Conference*, 2015.

[163] I. R. Gross, R. Matthews, J. Cohn, T. Kanade, and S. Baker. Multi-pie. In *FG*, 2008.

[164] Subramanian Ramanathan, Ashraf Kassim, YV Venkatesh, and Wu Sin Wah. Human facial expression recognition using a 3D morphable model. In *ICIP*, pages 661–664, 2006.

[165] M Ranzato, Joshua Susskind, Volodymyr Mnih, and Geoffrey Hinton. On deep generative models with applications to recognition. In *CVPR*, pages 2857–2864, 2011.

[166] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144. ACM, 2016.

[167] Salah Rifai, Yoshua Bengio, Aaron Courville, Pascal Vincent, and Mehdi Mirza. Disentangling factors of variation for facial expression recognition. In *ECCV*, volume 7577, pages 808–822. 2012.

[168] Richard Wiseman Roger Highfield and Rob Jenkins. How your looks betray your personality. *New Scientist*, 2009.

[169] Amir Rosenfeld, Richard Zemel, and John K Tsotsos. The elephant in the room. *arXiv preprint arXiv:1808.03305*, 2018.

[170] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533, 1986.

[171] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[172] J. Russell and A. Mehrabian. Evidence for a three-factor theory of emotions. *J. Research in Personality*, 11:273–294, 1977.

[173] J. A. Russell. Is there universal recognition of emotion from facial expression? A review of the cross-cultural studies. *Psychol. Bull.*, 115(1):102–141, 1994.

[174] Andrew Ryan, Jeffery F. Cohn, Simon Lucey, Jason Saragih, Patrick Lucey, Fernando De la Torre, and Adam Ross. Automated facial expression recognition system. In *ICCST*, 2009.

[175] João Sacramento, Rui Ponte Costa, Yoshua Bengio, and Walter Senn. Dendritic cortical microcircuits approximate the backpropagation algorithm. *NIPS*, 2018.

[176] Ruslan Salakhutdinov, Joshua B Tenenbaum, and Antonio Torralba. Learning with hierarchical-deep models. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1958–1971, 2013.

[177] Ashok Samal and Prasana A. Iyengar. Automatic recognition and analysis of human faces and facial expressions: A survey. *PR*, 25(1):65–77, 1992.

[178] Dairazalia Sanchez-Cortes, Joan-Isaac Biel, Shiro Kumano, Junji Yamato, Kazuhiro Otsuka, and Daniel Gatica-Perez. Inferring mood in ubiquitous conversational video. In *MUM*, page 22, 2013.

[179] G. Sandbach, S. Zafeiriou, M. Pantic, and D. Rueckert. A dynamic approach to the recognition of 3D facial expressions and their temporal models. In *FG*, pages 406–413, 2011.

[180] Georgia Sandbach, Stefanos Zafeiriou, and Maja Pantic. Local normal binary patterns for 3D facial action unit detection. In *ICIP*, pages 1813–1816, 2012.

[181] Arman Savran, Neşe Alyüz, Hamdi Dibeklioğlu, Oya Çeliktutan, Berk Gökberk, Bülent Sankur, and Lale Akarun. Bosphorus database for 3D face analysis. In *BIOID*, volume 5372, pages 47–56. 2008.

[182] Arman Savran, Houwei Cao, Miraj Shah, Ani Nenkova, and Ragini Verma. Combining video, audio and lexical indicators of affect in spontaneous conversation via particle filtering. In *Proceedings of the 14th ACM international conference on Multimodal interaction*, pages 485–492. ACM, 2012.

[183] Arman Savran, BüLent Sankur, and M Taha Bilge. Comparative evaluation of 3D vs. 2d modality for automatic detection of facial action units. *PR*, 45(2):767–782, 2012.

[184] Arman Savran, Haichuan Cao, Ani Nenkova, and Rajesh Verma. Temporal bayesian fusion for affect sensing: Combining video, audio, and lexical modalities. *CYB*, 2014.

[185] Stefan Scherer, Giota Stratou, Mohamed Mahmoud, Jill Boberg, Jonathan Gratch, Alessandro Rizzo, and Louis-Philippe Morency. Automatic behavior descriptors for psychological disorder analysis. In *Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on*, pages 1–8. IEEE, 2013.

[186] K. L. Schmidt and J. F. Cohn. Human facial expressions as adaptations: Evolutionary perspectives in facial expression research. *Yearbook of Physical Anthropology*, 116: 8–24, 2001.

[187] Nicu Sebe, Michael S Lew, Yafei Sun, Ira Cohen, Theo Gevers, and Thomas S Huang. Authentic facial expression analysis. *IVC*, (12):1856–1863, 2007.

[188] 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. *See https://arxiv. org/abs/1610.02391 v3*, 7(8), 2016.

[189] Mohammed Senoussaoui, Milton Sarria-Paja, João F Santos, and Tiago H Falk. Model fusion for multimodal depression classification and level detection. In *AVEC*, pages 57–63, 2014.

[190] Caifeng Shan, Shaogang Gong, and Peter W. McOwan. Facial expression recognition based on local binary patterns: A comprehensive study. *IVC*, 27(6):803–816, 2009.

[191] A. F. Shariff and J. L. Tracy. What are emotion expressions for? *CDPS*, 20(6): 395–399, 2011.

[192] Maxim Sidorov and Wolfgang Minker. Emotion recognition and depression diagnosis by acoustic and visual features: A multimodal approach. In *AVEC*, pages 81–86, 2014.

[193] Karan Sikka, Karmen Dykstra, Suchitra Sathyanarayana, Gwen Littlewort, and Marian Bartlett. Multiple kernel learning for emotion recognition in the wild. In *ICMI*, pages 517–524, 2013.

[194] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2015.

[195] Inchul Song, Hyun-Jun Kim, and Paul Barom Jeon. Deep learning for real-time robust facial expression recognition on a smartphone. In *ICCE*, pages 564–567, 2014.

[196] Hamit Soyel and Hasan Demirel. Facial expression recognition using 3D facial feature distances. In *ICIAR*, pages 831–838. 2007.

[197] Edwin H Spanier. *Algebraic topology*. Springer, 1994.

[198] Bo Sun, Liandong Li, Tian Zuo, Ying Chen, Guoyan Zhou, and Xuewen Wu. Combining multimodal features with hierarchical classifier fusion for emotion recognition in the wild. In *ICMI*, pages 481–486, 2014.

[199] Motoi Suwa, Noboru Sugie, and Keisuke Fujimora. A preliminary note on pattern recognition of human emotional expression. In *IJCPR*, pages 408–410, 1978.

[200] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[201] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[202] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701–1708, 2014.

[203] Chek Tien Tan, Daniel Rosser, Sander Bakkes, and Yusuf Pisan. A feasibility study in using facial expressions analysis to evaluate player experiences. In *IE*, page 5, 2012.

[204] Hao Tang and Thomas Huang. 3D facial expression recognition based on automatically selected features. In *CVPR*, pages 1–8, 2008.

[205] Hao Tang and Thomas S Huang. 3D facial expression recognition based on properties of line segments connecting facial feature points. In *FG*, pages 1–6, 2008.

[206] Ying-Li Tian, T. Kanade, and J. F. Cohn. Recognizing action units for facial expression analysis. *TPAMI*, 23:97–115, 2001.

[207] Leonardo Trujillo, Gustavo Olague, Riad Hammoud, and Benjamin Hernandez. Automatic feature localization in thermal images for facial expression recognition. In *CVPR Workshops*, pages 14–14, 2005.

[208] Filareti Tsalakanidou and Sotiris Malassiotis. Robust facial action recognition from real-time 3D streams. In *CVPR Workshops*, pages 4–11, 2009.

[209] Filareti Tsalakanidou and Sotiris Malassiotis. Real-time 2d+ 3D facial action and expression recognition. *PR*, 43(5):1763–1775, 2010.

[210] Ryan Turner. A model explanation system. In *Machine Learning for Signal Processing (MLSP), 2016 IEEE 26th International Workshop on*, pages 1–6. IEEE, 2016.

[211] Michel F Valstar, Maja Pantic, Zara Ambadar, and Jeffrey F Cohn. Spontaneous vs. posed facial behavior: automatic analysis of brow actions. In *ICMI*, pages 162–170, 2006.

[212] Alessandro Vinciarelli, Maja Pantic, and Hervé Bourlard. Social signal processing: Survey of an emerging domain. *IVC*, 27(12):1743–1759, 2009.

[213] Paul Voosen. How ai detectives are cracking open the black box of deep learning. *Science, July*, 2017.

[214] Nicholas Vretos, Nikos Nikolaidis, and Ioannis Pitas. 3D facial expression recognition using Zernike moments on depth images. In *ICIP*, pages 773–776, 2011.

[215] Esra Vural, Mujdat Cetin, Aytul Ercil, Gwen Littlewort, Marian Bartlett, and Javier Movellan. Drowsy driver detection through facial movement analysis. In *Human–Computer Interaction*, pages 6–18. 2007.

[216] Robert Walecki, Ognjen Rudovic, Vladimir Pavlovic, and Maja Pantic. Variable-state latent conditional random fields for facial expression recognition and action unit detection. *FG*, pages 1–8, 2015.

[217] Robert Walecki, Vladimir Pavlovic, Björn Schuller, Maja Pantic, et al. Deep structured learning for facial action unit intensity estimation. *arXiv preprint arXiv:1704.04481*, 2017.

[218] B. M. Waller, J. J. Cray, and A. M. Burrows. Selection for universal facial emotion. *Emotion*, 8(3):435–439, 2008.

[219] B. M. Waller, L. A. Parr, K. M. Gothard, A. M. Burrows, and A. J. Fuglevand. Mapping the contribution of single muscles to facial movements in the rhesus macaque. *Physiol. Behav.*, 95:93–100, 2008.

[220] B. M. Waller, Manuela Lembeck, Paul Kuchenbuch, A. M. Burrows, and K. Liebal. Gibbonfacs: A muscle-based facial movement coding system for hylobatids. *J. Primatol.*, 33:809–821, 2012.

[221] J. Wang and L. Yin. Facial expression representation and recognition from static images using topographic context. Technical report, Department of Computer Science, 2005.

[222] Jun Wang, Lijun Yin, Xiaozhou Wei, and Yi Sun. 3D facial expression recognition based on primitive surface feature distribution. In *CVPR*, pages 1399–1406, 2006.

[223] Shangfei Wang, Zhilei Liu, Siliang Lv, Yanpeng Lv, Guobing Wu, Peng Peng, Fei Chen, and Xufa Wang. A natural visible and infrared facial expression database for expression recognition and emotion inference. *T. Multimedia*, 12(7):682–691, 2010.

[224] Shangfei Wang, Menghua He, Zhen Gao, Shan He, and Qiang Ji. Emotion recognition from thermal infrared images using deep boltzmann machine. *FCS*, 8(4):609–618, 2014.

[225] Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2794–2802, 2015.

[226] Ziheng Wang, Yongqiang Li, Shangfei Wang, and Qiang Ji. Capturing global semantic relationships for facial action unit recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3304–3311, 2013.

[227] D. Watson, L. A. Clark, and A. Tellegen. Development and validation of brief measures of positive and negative affect: The PANAS scales. *JPSP*, 54:1063–1070, 1988.

[228] James R Williamson, Thomas F Quatieri, Brian S Helfer, Gregory Ciccarelli, and Daryush D Mehta. Vocal and facial biomarkers of depression based on motor incoordination and timing. In *AVEC*, pages 65–72, 2014.

[229] Martin Wöllmer, Moritz Kaiser, Florian Eyben, Björn Schuller, and Gerhard Rigoll. Lstm-modeling of continuous emotions in an audiovisual affect recognition framework. *IVC*, 31(2):153–163, 2013.

[230] Chongliang Wu, Shangfei Wang, and Qiang Ji. Multi-instance hidden markov model for facial expression recognition. In *FG*, 2015.

[231] Yue Wu and Qiang Ji. Constrained joint cascade regression framework for simultaneous facial action unit recognition and facial landmark detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3400–3408, 2016.

[232] Wen-Jing Yan, Qi Wu, Yong-Jin Liu, Su-Jing Wang, and Xiaolan Fu. Casme database: A dataset of spontaneous micro-expressions collected from neutralized faces. In *FG*, pages 1–7, 2013.

[233] Lijun Yin, Xiaozhou Wei, Peter Longo, and Abhinesh Bhuvanesh. Analyzing facial expressions using intensity-variant 3D data for human computer interaction. In *ICPR*, pages 1248–1251, 2006.

[234] Lijun Yin, Xiaozhou Wei, Yi Sun, Jun Wang, and M. J. Rosato. A 3D facial expression database for facial behavior research. In *FG*, pages 211–216, 2006.

[235] Lijun Yin, Xiaochen Chen, Yi Sun, T. Worm, and M. Reale. A high-resolution 3D dynamic facial expression database. In *FG*, pages 1–6, 2008.

[236] Y Yoshitomi, N Miyawaki, S Tomita, and S Kimura. Facial expression recognition using thermal image processing and neural network. In *RO-MAN*, pages 380–385, 1997.

[237] Yasunari Yoshitomi et al. Facial expression recognition for speaker using thermal image processing and speech recognition system. In *WSEAS*, pages 182–186, 2010.

[238] Stefanos Zafeiriou and Maria Petrou. Nonlinear nonnegative component analysis. In *CVPR*, pages 2860–2865, 2009.

[239] Amir R Zamir, Alexander Sax, William Shen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3712–3722, 2018.

[240] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[241] Jiabei Zeng, Wen-Sheng Chu, Fernando De la Torre, Jeffrey F Cohn, and Zhang Xiong. Confidence preserving machine for facial action unit detection. In *Proceedings of the IEEE ICCV*, pages 3622–3630, 2015.

[242] Wei Zeng, Huibin Li, Liming Chen, J. M. Morvan, and X. D. Gu. An automatic 3D expression recognition framework based on sparse representation of conformal images. In *FG*, pages 1–8, 2013.

[243] Z. Zeng, M. Pantic, G. I. Roisman, and T. S. Huang. A survey of affect recognition methods: Audio, visual, and spontaneous expressions. *TPAMI*, 31(1):39–58, 2009.

[244] Zhihong Zeng, Yun Fu, Glenn I Roisman, Zhen Wen, Yuxiao Hu, and Thomas S Huang. Spontaneous emotional facial expression detection. *JMM*, 1(5):1–8, 2006.

[245] Xiao Zhang and Mohammad H Mahoor. Task-dependent multi-task multiple kernel learning for facial action unit detection. *Pattern Recognition*, 51:187–196, 2016.

[246] Xing Zhang, Lijun Yin, Jeffrey F. Cohn, Shaun Canavan, Michael Reale, Andy Horowitz, Peng Liu, and Jeffrey M. Girard. Bp4d-spontaneous: a high-resolution spontaneous 3D dynamic facial expression database. *IVC*, 32(10):692–706, 2014.

[247] Xing Zhang, Lijun Yin, Jeffrey F Cohn, Shaun Canavan, Michael Reale, Andy Horowitz, Peng Liu, and Jeffrey M Girard. Bp4d-spontaneous: a high-resolution spontaneous 3d dynamic facial expression database. *Image and Vision Computing*, 32(10):692–706, 2014.

[248] Guoying Zhao and Matti Pietikainen. Dynamic texture recognition using local binary patterns with an application to facial expressions. *TPAMI*, 29(6):915–928, 2007.

[249] Kaili Zhao, Wen-Sheng Chu, Fernando De la Torre, Jeffrey F Cohn, and Honggang Zhang. Joint patch and multi-label learning for facial action unit detection. In *IEEE CVPR*, pages 2207–2216, 2015.

[250] Kaili Zhao, Wen-Sheng Chu, and Honggang Zhang. Deep region and multi-label learning for facial action unit detection. In *IEEE CVPR*, pages 3391–3399, 2016.

[251] Xi Zhao, Di Huang, Emmanuel Dellandréa, and Liming Chen. Automatic 3D facial expression recognition based on a bayesian belief net and a statistical facial feature model. In *ICPR*, pages 3724–3727, 2010.

[252] Xi Zhao, Emmanuel Dellandréa, Jianhua Zou, and Liming Chen. A unified probabilistic framework for automatic 3D facial expression analysis based on a bayesian belief inference and statistical feature models. *IVC*, 31(3):231–245, 2013.

[253] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip HS Torr. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1529–1537, 2015.

[254] Ruicong Zhi, Markus Flierl, Qiuqi Ruan, and W Bastiaan Kleijn. Graph-preserving sparse nonnegative matrix factorization with application to facial expression recognition. *SMC-B*, 41:38–52, 2011.

[255] Lin Zhong, Qingshan Liu, Peng Yang, Junzhou Huang, and Dimitris N Metaxas. Learning multiscale active facial patches for expression analysis. *IEEE transactions on cybernetics*, 45(8):1499–1510, 2015.

[256] Bolei Zhou, David Bau, Aude Oliva, and Antonio Torralba. Interpreting deep visual representations via network dissection. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[257] Luisa M Zintgraf, Taco S Cohen, Tameem Adel, and Max Welling. Visualizing deep neural network decisions: Prediction difference analysis. *arXiv preprint arXiv:1702.04595*, 2017.

Appendix A

Publications

The following publications are a consequence of the research carried out during the elaboration of this thesis.

A.1 Journals

- Corneanu, C. A., Simón, M. O., Cohn, J. F., Guerrero, S. E. (2016). Survey on rgb, 3d, thermal, and multimodal approaches for facial expression recognition: History, trends, and affect-related applications. IEEE transactions on pattern analysis and machine intelligence, 38(8), 1548-1568.

- Kulkarni, Kaustubh, Ciprian Corneanu, Ikechukwu Ofodile, Sergio Escalera, Xavier Baro, Sylwia Hyniewska, Juri Allik, and Gholamreza Anbarjafari. "Automatic recognition of facial displays of unfelt emotions." IEEE transactions on affective computing (2018).

- Noroozi, Fatemeh, Ciprian Corneanu, Dorota Kaminska, Tomasz Sapinski, Sergio Escalera, and Gholamreza Anbarjafari. "Survey on emotional body gesture recognition." IEEE transactions on affective computing (2018).

- Simón, Marc Oliu, Ciprian Corneanu, Kamal Nasrollahi, Olegs Nikisins, Sergio Escalera, Yunlian Sun, Haiqing Li, Zhenan Sun, Thomas B. Moeslund, and Modris Greitans. "Improved RGB-DT based face recognition." Iet Biometrics 5, no. 4 (2016): 297-303.

A.2 International Conferences and Workshops

- Corneanu, Ciprian, Meysam Madadi, and Sergio Escalera. "Deep structure inference network for facial action unit recognition." In Proceedings of the European Conference on Computer Vision (ECCV), pp. 298-313. 2018.

- Corneanu, Ciprian A., Meysam Madadi, Sergio Escalera, and Aleix M. Martinez. ”What Does It Mean to Learn in Deep Networks? And, How Does One Detect Adversarial Attacks?.” In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4757-4766. 2019.

- Oliu, Marc, Ciprian Corneanu, László A. Jeni, Jeffrey F. Cohn, Takeo Kanade, and Sergio Escalera. ”Continuous supervised descent method for facial landmark localisation.” In Asian Conference on Computer Vision, pp. 121-135. Springer, Cham, 2016.

- Irani, Ramin, Kamal Nasrollahi, Marc O. Simon, Ciprian A. Corneanu, Sergio Escalera, Chris Bahnsen, Dennis H. Lundtoft et al. ”Spatiotemporal analysis of RGB-DT facial images for multimodal pain level recognition.” In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 88-95. 2015.

- Escalera, Sergio, Mercedes Torres Torres, Brais Martinez, Xavier Baró, Hugo Jair Escalante, Isabelle Guyon, Georgios Tzimiropoulos et al. ”Chalearn looking at people and faces of the world: Face analysis workshop and challenge 2016.” In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 1-8. 2016.

- Ponce-López, Víctor, Baiyu Chen, Marc Oliu, Ciprian Corneanu, Albert Clapés, Isabelle Guyon, Xavier Bar, Hugo Jair Escalante, and Sergio Escalera. ”Chalearn lap 2016: First round challenge on first impressions-dataset and results.” In European Conference on Computer Vision, pp. 400-418. Springer, Cham, 2016.