Universitat Autònoma de Barcelona

Towards Robust Neural Models for Fine-Grained Image Recognition

Pau Rodríguez López

Universitat Autònoma
de Barcelona

Towards Robust Neural Models for Fine-Grained Image Recognition

A dissertation submitted by **Pau Rodríguez López** at Universitat Autònoma de Barcelona to fulfil the degree of **Doctor of Philosophy**.

Bellaterra, January 17, 2019

Co-Director	**Dr. Jordi Gonzàlez Sabaté** Dept. Ciències de la computació & Centre de Visió per Computador
Co-Director	**Dr. Josep M. Gonfaus** Centre de Visió per Computador
Co-Director	**Dr. F. Xavier Roca Marvà** Dept. Ciències de la computació & Centre de Visió per Computador
Thesis committee	**Dr. Gregory Rogez** INRIA, Grenoble, France **Dr. Carles Fernandez Tena** Herta Security, Barcelona, Spain **Dr. David Masip Rodó** Universitat Oberta de Catalunya, Barcelona, Spain
International evaluators	**El-Hadi Zahzah** University of La Rochelle, La Rochelle, France **Dr. Marina Ivasic-Kos** University of Rijeka, Rijeka, Croatia

This document was typeset by the author using LaTeX 2$_\varepsilon$.

ISBN: 978-84-948531-3-5

Printed by Ediciones Gráficas Rey, S.L.

I was born not knowing and I had only a little time to change that here and there.
— Richard P. Feynman

The universe is not required to be in perfect harmony with human ambition.
— Carl Sagan

A la familia, amics, i a l'Aida ...

Acknowledgements

M'agradaria començar agraint al Jordi Gonzàlez, el director d'aquesta tesi, per a creure en mi des del principi, fins i tot quan jo no ho feia, per transmetre'm la seva passió centrada la humanitat i la visió *wabi-sabi* del món – acceptant-ne la imperfecció i trobant-ne la bellesa. En comptes de limitar-se a dirigir-me per un camí, el Jordi m'ha cedit totes les eines i recursos de què ha disposat perquè jo trobi la meva direcció, i això el fa probablement el millor director que podria haver tingut. També voldria agrair al Pep Gonfaus, qui admiro i qui ha estat el meu model a seguir durant tota la tesi, tant dins com fora de l'àmbit acadèmic. Li vull agrair la paciència, discussions i el fet de mantenir-me amb els peus a terra. Cada cop que em sento satisfet amb un projecte penso – què diria el Pep? – i sé que ell tindria raó, o com ell diria, estaria menys equivocat que jo. Gràcies també al Xavi Roca per fer-me sentir com a casa al grup ISELAB des del primer moment i sé que també en el futur. Finalment, gràcies al Guillem Cucurull, desitjaria haver-lo ajudat a ell tant com ell em va ajudar a mi.

I also want to mention my friends and colleagues from the CVC: Pau Riba, Edgar Riba, Arnau Baró, Albert Berenguel, Diego Velazquez, and all the others for all the coffee breaks, the discussions – not only about research. Thanks to the CVC IT team, who gracefully handled the deep learning explosion. I would also like to thank Joan Masoliver and Ramon Baldrich for the countless hours at SAF. Thanks also to all the people from the CVC administration that have helped me one way or another.

My sincere appreciation to the Element AI team. Especially I give thanks to David Vazquez for his hospitality, and Alexandre Lacoste for adopting me on his team, for transmitting his positiveness, and for all the discussions. Thanks also to Boris Oreshkin for accepting my help on his project. I would like to mention all the friends I made there: Alex Drouin, Issam Laradji, Olivier Mastropietro – if you had coffee or hung out with me, you can also be included to the list. Thanks also Catherine Martin and Valerie Becaert who made me feel at home.

Words are not enough to thank Aida, my wife, who has always been there even when I was not. Thanks for all the patience, for the support, and for encouraging me to take hard decisions, even those that affected herself. I think she is the one who has worked the hardest during this period, and I am the one taking credit.

Les últimes paraules, tot i que no les menys importants, les adreço a la meva familia. Ells son realment els que m'han posat les portes per a arribar fins aqui, jo només he hagut d'obrir-les. Al meu pare li vull agrair especialment el despertar-me l'interés en la ciència i a la meva mare per ajudar-me a posar un límit a l'autoexigència. Gràcies al meu germà, per ensenyar-me que hi ha més d'un tipus de porta.

Abstract

Fine-grained recognition, i.e. identifying similar subcategories of the same superclass, is central to human activity. Recognizing a friend, finding bacteria in microscopic imagery, or discovering a new kind of galaxy, are just but few examples. However, fine-grained image recognition is still a challenging computer vision task since the differences between two images of the same category can overwhelm the differences between two images of different fine-grained categories. In this regime, where the difference between two categories resides on subtle input changes, excessively invariant CNNs discard those details that help to discriminate between categories and focus on more obvious changes, yielding poor classification performance. On the other hand, CNNs with too much capacity tend to memorize instance-specific details, thus causing overfitting. In this thesis, motivated by the potential impact of automatic fine-grained image recognition, we tackle the previous challenges and demonstrate that proper alignment of the inputs, multiple levels of attention, regularization, and explicit modeling of the output space, results in more accurate fine-grained recognition models, that generalize better, and are more robust to intra-class variation. Concretely, we study the different stages of the neural network pipeline: input pre-processing, attention to regions, feature activations, and the label space. In each stage, we address different issues that hinder the recognition performance on various fine-grained tasks, and devise solutions in each chapter: i) We deal with the sensitivity to input alignment on fine-grained human facial motion such as pain. ii) We introduce an attention mechanism to allow CNNs to choose and process in detail the most discriminate regions of the image. iii) We further extend attention mechanisms to act on the network activations, thus allowing them to correct their predictions by looking back at certain regions, at different levels of abstraction. iv) We propose a regularization loss to prevent high-capacity neural networks to memorize instance details by means of almost-identical feature detectors. v) We finally study the advantages of explicitly modeling the output space within the error-correcting framework. As a result, in this thesis we demonstrate that attention and regularization seem promising directions to overcome the problems of fine-grained image recognition, as well as proper treatment of the input and the output space.

Key words: *computer vision, machine learning, fine-grained image recognition*

Resumen

Reconocer e identificar diferentes subcategorías en nuestro entorno es una actividad crucial en nuestras vidas. Reconocer un amigo, encontrar cierta bacteria en imágenes de microscopio, o descubrir un nuevo tipo de galaxia son solo algunos ejemplos. Sin embargo, el reconocimiento de subcategorías en imágenes aún es una tarea ardua en el campo de la visión por computador, ya que las diferencias entre dos imágenes de la misma subcategoría eclipsan los detalles que distinguen dos subcategorías diferentes. En este tipo de problema, en que la distinción entre categorías radica en diferencias sutiles, las redes neuronales más insensibles a perturbaciones se centran en los cambios más obvios y tienden a errar, ya que ignoran aquellos detalles que permiten desambiguar entre diferentes categorías. Por otro lado, los modelos con demasiada capacidad tienden a memorizar detalles únicos de imágenes concretas, por lo que fallan al generalizar con nuevas imágenes nunca vistas. En esta tesis doctoral, motivados por el impacto potencial del reconocimiento automático de subcategorías, abordamos los desafíos presentados y demostramos que es posible obtener modelos generales y robustos. Concretamente, estudiamos las diferentes fases de los algoritmos de reconocimiento de imágenes: preproceso de los datos, atención a diferentes regiones, actividad de las neuronas y el espacio de categorías. En cada fase, abordamos diferentes problemas que merman la precisión de los modelos al clasificar diferentes tipos de datos, y proponemos diferentes soluciones en cada capítulo: i) Primero abordamos el problema de la sensibilidad al alineamiento de las imágenes en el reconocimiento de expresiones faciales, como el dolor. ii) A continuación, proponemos un mecanismo de atención que permite a las redes neuronales centrarse y procesar en detalle las partes más informativas de las imágenes. iii) Extendemos los mecanismos de atención más allá de los píxeles, permitiendo las redes atender su propia actividad neuronal para corregir las predicciones finales. iv) Después proponemos una nueva función de coste para regularizar las conexiones de las capas de neuronas, incentivando el aprendizaje de patrones distintos y, por lo tanto, previniendo la memorización de detalles únicos en objetos. v) Finalmente, estudiamos las ventajas de modelar explícitamente el espacio de categorías usando la teoría de códigos correctores de errores. Como resultado, en esta tesis demostramos que los mecanismos de atención y regularización pueden ser la clave para solucionar los problemas del reconocimiento de subcategorías, así como una buena modelización del espacio de entrada y salida de los modelos.

Palabras clave: *visión por computador, aprendizaje computacional, clasificación de imágenes*

Resum

Reconèixer i identificar diverses subcategories en el nostre entorn és una activitat crucial a les nostres vides. Reconèixer un amic, trobar cert bacteri en imatges de microscopi, o descobrir un nou tipus de galàxia en són només alguns exemples. Malgrat això, el reconeixement de subcategories en imatges encara és una tasca costosa en el camp de la visió per computador, ja que les diferències entre dues imatges de la mateixa subcategoria eclipsen els detalls que distingeixen dues subcategories diferents. En aquest tipus de problema, en què la distinció entre categories radica en diferències subtils, les xarxes neuronals més robustes a pertorbacions se centren en els canvis més obvis i solen fallar, ja que ignoren els detalls que permeten distingir entre diferents categories. Per altra banda, els models amb massa capacitat tendeixen a memoritzar detalls únics d'imatges concretes, pel que fallen en generalitzar amb noves imatges mai vistes. En aquesta tesi doctoral, motivats per l'impacte potencial del reconeixement automàtic de subcategories, abordem els desafiaments presentats i demostrem que és possible obtenir models generals i robustos. Concretament, estudiem les diferents fases dels algorismes de reconeixement d'imatges: preprocessament de les dades, atenció a diferents regions, activitat de les neurones, i l'espai de categories. A cada fase abordem diferents problemes que redueixen la precisió dels models al classificar diferents tipus de dades i proposem diferents solucions a cada capítol: i) Abordem el problema de la sensibilitat a l'alineament de les imatges en el reconeixement d'expressions facials, com el dolor. ii) Proposem un mecanisme d'atenció que permet a les xarxes neuronals centrar-se i processar en detall les parts més informatives de les imatges. iii) Estenem els mecanismes d'atenció més enllà dels píxels, permetent les xarxes atendre la seva pròpia activitat neuronal per a corregir les prediccions finals. iv) Després proposem una nova funció de cost per a regularitzar les connexions de les capes de neurones, incentivant l'aprenentatge de patrons diferents i, per tant, prevenint la memorització de detalls únics. v) Estudiem els avantatges de modelar explícitament l'espai de categories utilitzant la teoria de codis correctors d'errors. Com a resultat, en aquesta tesi demostrem que els mecanismes d'atenció i regularització poden ser la clau per a solucionar els problemes de reconeixement de subcategories, així com una bona modelització de l'espai d'entrada i sortida dels models.

Paraules clau: *visió per computador, aprenentatge computacional, classificació d'imatges*

Contents

List of Figures

List of Tables

1 Introduction

All birds are different, yet we call them with the generic term *bird*. In fact, Greek philosophers such as Plato (~428 BCE) already noticed the importance of this phenomena, stating that all objects are instances of a perfect abstract entity. This theory was later adapted by conceptualist philosophers, such as William Ockham (1287), rejecting the existence of abstract entities, explaining them as a process of the mind. In the same vein, Emmanuel Kant (1724) proposed that the world is made of real objects, that we experience through our senses, and organize them in abstract *categories* to put coherence in an otherwise chaotic world. Inspired by David Hume (1711), Gestalt philosophers (~1890) state that we tend to order our stimuli into abstract categories in order to find the simplest explanation, hence perceiving whole objects instead of their individual parts. For instance, we tend to perceive a bird instead of a bunch of feathers and a beak.

In other fields, such as information theory, categorization is a computational mechanism for compressing information and saving space. For instance, it takes less effort to just remember *bird*, than remembering all the particularities of that bird. Psychologists such as Steven Pinker (1954) propose a more practical view, where we categorize because the world is ruled by laws such as physics and mathematics, allowing us to infer properties between similar objects, which is crucial for survival. For instance, if you know that a red mushroom with white circular spots is poisonous, you will avoid similar red mushrooms with white squared spots. In fact, to *know* or *recognize* a mushroom, not only involves identifying its general appearance, such as the parts (cap, stem, texture), but also the particular appearance of each of these parts (size and color of the cap, ringed stem, etc). This particularly challenging task is called fine-grained image recognition, and it is the subject of this thesis.

In image recognition, we aim to find the identity and properties of the different objects that compose an image with the final goal of understanding its content. For instance, given a set of images with cars, people, and animals, an image recognition system would output which of the three categories are present in the image. Computer vision researchers usually model these image recognition systems as a machine learning problem, where the images and corresponding categories are used to fit a classification function. However, the visual world is extremely complex, with small perturbations, such as the movement or deformation of an object, caus-

(a) Dhole

(b) Dingo

Figure 1.1: **Fine-grained recognition problem.** Row (a) shows images of dogs of the *dhole* breed, while (b) shows dogs of the *dingo* breed. As it can be seen, de differences in pose have more impact on the appearance than the breed.

ing changes in the data that are unpredictable by most of the traditional machine learning algorithms.

Throughout the past decade, neural networks have become the main approach to tackle this task since they can deal with the high dimensionality and complexity of the visual world. Concretely, neural networks can "learn" to detect intricate interactions between the pixels of an image in order to map those pixels into the desired target. As a result, researchers have focused on building increasingly powerful neural networks, attaining impressive scores on many image recognition benchmarks, even when compared against human performance. These advances are appealing for numerous industrial applications, and therefore are progressively being used to assist complex repetitive tasks, such as spotting cancer on x-rays, or identifying traffic signs. However, there are cases where neural networks still fall behind the human image recognition skills. For instance, while a human may be able to recognize an object by seeing it once, a neural network sill needs many examples of the object in many situations. Moreover, while humans do not have to obliterate previously seen objects to learn a new one, neural networks tend to

forget previously seen classes when retrained for a new one. Another particular case where neural networks perform worse than humans is on fine-grained recognition. Namely, when the task is to discriminate between many subcategories of the same object. For instance, to differentiate between bird species, car models, cell types, and the like. This is a particularly difficult task because the difference between objects of the same fine-grained category can eclipse de variation between objects of different categories. For example, imagine two birds of similar species flying or posing in a branch, the difference between the posing and the flying birds greater than the difference between the two posing birds. Despite the inherent challenges of fine-grained recognition, most approaches are based on coarse-grained recognition models, *i.e.* neural networks with many layers (deep) specialized for image processing. The next section provides detail on this kind of architectures, and how to enhance them for fine-grained recognition.

1.1 Deep Learning and Fine-Grained Recognition

Since the great success at the Imagenet Large Scale Visual Recognition Competition (ILSVRC2012) [1], Convolutional Neural Networks (CNN) [2] have become the main approach to image recognition. Furthermore, CNNs are now the main workhorse for most computer vision task such as motion analysis [3], 3D reconstruction [4], and image generation [5, 6].

Regarding the task of image recognition, it is classically framed as a supervised machine learning problem. That is, given a set of images X, and a set of corresponding labels Y, we aim to find a function $F : X \rightarrow Y$ that minimizes a target error criterion $J : X \times Y \rightarrow \mathbb{R}$. In the case of artificial neural networks, F is parameterized by a set of neuron connections or weights θ. Thus, we aim to find the optimal value of θ so that the function $\mathbf{y} = f(\mathbf{x};\boldsymbol{\theta})$ minimizes our criterion J.

Neural networks with multiple layers are universal approximators [7], *i.e.* they can represent a large variety of functions. This is particularly useful for image recognition since it requires modeling complex interactions between pixels to identify objects and categories. However, when connected to all the pixels of an image, the high capacity of neural networks makes them very sensitive to noise, small perturbations, and prone to memorize the input. As a result, when applied to image recognition, vanilla artificial neural networks tend to exhibit poor performance on unseen images (overfitting). Differently, the neurons of a CNN are not connected to all the pixels but to a sliding subset or window, outputting one value for each position, see Figure 1.2. This sliding-window operation is called convolution, and it is represented by the $*$ operator. Since the same weights are applied at different regions of the image, the network is forced to "learn" universal patterns, *i.e.* patterns

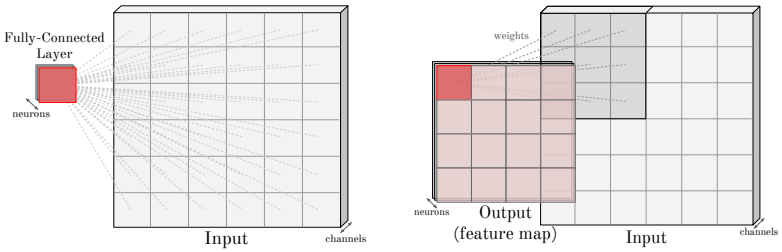


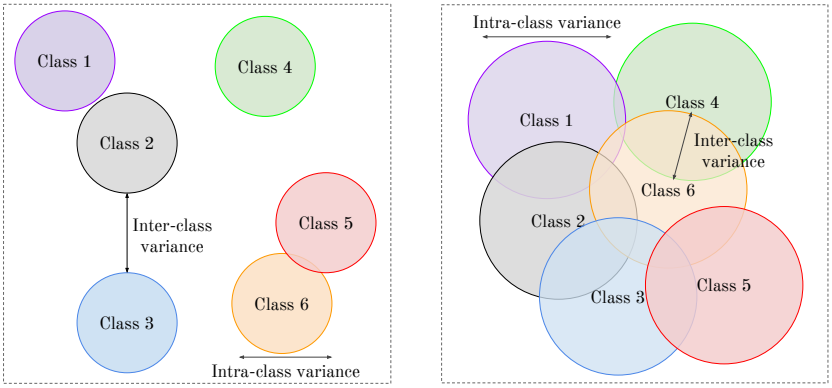
Figure 1.2: A fully connected (a), and a convolutional layer (b)

that repeat through all the image, such as edges. An additional useful property of the convolution is locality. Namely, pixels that are distant from each other are not connected, hence avoiding modeling correlations between unrelated parts of the image (for instance, the sun and a road). These local features keep being aggregated layer after layer until a global description of the image is attained in the end. As a result, CNNs are robust to translation and slight perturbations of individual pixels.[1] Moreover, convolutional layers are usually followed by a max pooling operation that provides additional translation invariance. For instance, max pooling outputs the window maximum at each position of the image, thus as long as a maximal pixel falls into a given window position, the output will remain invariant to small translations of the input.

In fine-grained recognition, the differences between two images of the same category can overwhelm the differences between two images of different fine-grained categories, see Figure 1.3. This is often referenced in the literature as the "high intra-class variance" and "low inter-class variance" problem [9]. In this regime, where the difference between two categories resides on subtle input changes, excessively invariant CNNs discard those details that help to discriminate between categories and focus on more obvious changes, yielding poor classification performance. Therefore, most of the literature on fine-grained recognition focuses on making the most discriminative regions of the image explicit to the classifier [9]. This can be done at different stages of the image recognition pipeline depicted in Figure 1.4:

- **Image level**: a classic approach to tackle fine-grained recognition tasks is to reduce intra-class variance at the input so that the classifier can focus on the inter-class variance. For instance, we can align the object parts to prevent the

[1] When not done on purpose to alter the original behavior of the model [8].



(a) Coarse-grained object recognition (b) Fine-grained recognition

Figure 1.3: **Comparison between a fine-grained and a coarse-grained recognition.**

classifier from focusing on pose differences but on their appearance [10, 11]. This approach is particularly useful for face verification, since (frontal) faces have a direct correspondence between all parts (eyes, ears, mouth), and can be easily aligned by affine transformations, (rotation, translation, scaling, and shear) [12, 13].

- **Model input**: inspired by human attention, the recent literature focuses on models that find and select the most discriminative parts of the image to process them in detail [14, 15]. Iterative approaches allow the network to dynamically look for the most discriminative region of the image [16].

- **Model activations**: in *spatial transformer networks* [17], the authors proposed a layer that dynamically performs an affine transformation to the input, thus aligning or cropping certain regions of the image. This layer can be placed at any place of the network, so the network can focus on certain parts of its own feature maps. Similarly, residual attention networks [18], enhance or weaken certain regions of its own feature maps.

- **Model output**: differently from coarse-grained recognition, it is not safe to assume that fine-grained output categories are equidistant. For instance, hummingbird subspecies resemble each other more than penguin sub-species. Therefore, a number of approaches focus on modeling the output space to

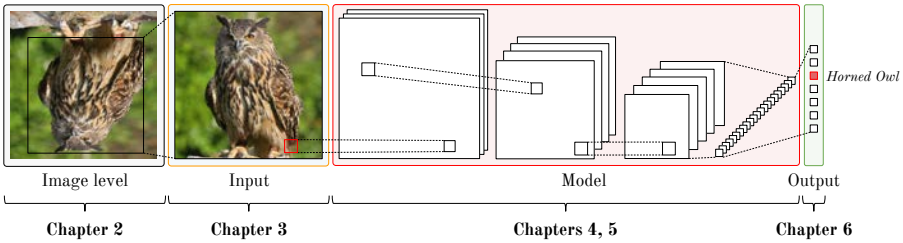


Figure 1.4: **Example of fine-grained recognition pipeline.** Each stage corresponds to a chapter in this work.

account for this fact. For instance, ensembles or mixtures can be used to assign subsets of similar categories to specialized CNNs [19, 20].

1.2 Thesis contributions

In this thesis, we analyze different stages of the neural network pipeline, identifying issues that hinder the performance on various fine-grained recognition tasks, and propose solutions to them in each chapter:

- In chapter 2, we show that alignment and frontalization have a great impact on facial image processing. Concretely, we propose a deep learning pipeline to recognize pain from facial images. Since pain is recognized from subtle facial muscle movements, removing the variability introduced by head movement is crucial to let the model focus on muscle dynamics. Moreover, we show that too much alignment, such as frontalization, can cause too much deformation and introduce noise into muscle dynamics, thus making simple frontalization more effective than other complex approaches.

- In chapter 3, we propose an attention mechanism to allow neural networks to focus on the most discriminative parts of the image to perform age and gender recognition. Thus, the network spends more processing time on the most discriminative regions of the image, in higher resolution, while ignoring clutter. The resulting models obtain competitive performance on three different benchmarks.

- In chapter 4, we focus on the intermediate network feature activations. Concretely, we propose an efficient self-attention mechanism for convolutional neural networks that augments them with the capacity of correcting the

original predictions based on highly discriminative regions of the image at different levels of abstraction. For instance, correcting "horse" for "zebra" by attending at the stripes. When augmented with the proposed approach, all the models show improved classification performance, obtaining excellent results in all benchmarks.

- In chapter 5, we deal with the problem of overfitting. Concretely, since fine-grained recognition relies on learning subtle differences between images, overparametrized neural networks have the risk of memorizing noise. For instance, almost-identical filters could be learned to identify two noisy variations of the same pattern in the data. In order to address this issue, we propose a new regularization loss that enforces orthogonality between neuron weights. As a result, regularized neural networks show better generalization yielding higher validation accuracy.

- In chapter 6, we pay attention to the output space of neural networks. Concretely, we find that representing classes as a vector with a zero for each class and a one for the target class (one-hot) is highly inefficient for fine-grained recognition settings, with potentially thousands of classes. Thus, we propose to use error correcting output codes (ECOC) to represent the class space. Since ECOC can have an arbitrary size and are denser than one-hot vectors, they are more efficient either in space and gradient signal. Moreover, we show that the distance between codes can be adjusted to account for class confusion, adding extra correction capacity to very similar classes than to highly dissimilar ones.

Summarizing, in this thesis, we show that proper alignment of the inputs, multiple levels of attention, regularization, and modeling of the output space, results in accurate models, that generalize better, and stride towards robust fine-grained image recognition.

1.3 First Published Appearances

This thesis follows the format of a publications compendium. Hence, each chapter corresponds to an article published in a journal or conference:

- Chapter 2: **Pau Rodríguez**, Guillem Cucurull, Jordi Gonzalez, Josep M Gonfaus, Kamal Nasrollahi, Thomas B Moeslund, and F Xavier Roca. Deep pain: Exploiting long short-term memory networks for facial expression classification. *IEEE cybernetics*, (99):1–11, 2017

- Chapter 3: **Pau Rodríguez**, Guillem Cucurull, Josep M Gonfaus, F Xavier Roca, and Jordi Gonzalez. Age and gender recognition in the wild with deep attention. *PR*, 2017
- Chapter 4: **Pau Rodríguez**, Jordi Gonzalez, Guillem Cucurull, Josep M Gonfaus, and Xavier Roca. Regularizing cnns with locally constrained decorrelations. In *ICLR*, 2017
- Chapter 5: **Pau Rodríguez**, Josep M Gonfaus, Guillem Cucurull, F Xavier Roca, and Jordi Gonzalez. Attend and rectify: a gated attention mechanism for fine-grained recovery. In *ECCV*, pages 349–364, 2018
- Chapter 6: **Pau Rodríguez**, Miguel A Bautista, Jordi Gonzàlez, and Sergio Escalera. Beyond one-hot encoding: Lower dimensional target embedding. *IMAVIS*, 75:21–31, 2018

2 Pain Recognition from Facial Images

Pain is an unpleasant feeling that has been shown to be an important factor for the recovery of patients. Since this is costly in human resources and difficult to do objectively, there is the need for automatic systems to measure it. In this paper, contrary to current state-of-the-art techniques in pain assessment, which are based on facial features only, we suggest that the performance can be enhanced by feeding the raw frames to deep learning models, outperforming the latest state-of-the-art results while also directly facing the problem of imbalanced data. As a baseline, our approach first uses convolutional neural networks (CNN) to learned facial features from VGG_Faces, which are then linked to a Long Short-Term Memory (LSTM) to exploit the temporal relation between video frames. We further compare the performances of using the so popular schema based on the canonically normalized appearance versus taking into account the whole image: As a result, we outperform current state-of-the-art AUC performance in the UNBC-McMaster Shoulder Pain Expression Archive Database. In addition, to evaluate the generalization properties of our proposed methodology on facial motion recognition, we also report competitive results in the Cohn Kanade+ facial expression database.

2.1 Motivation

The automatic detection of pain is a subject of high interest in the health domain since it is not only an important indicator for medical diagnosis, but has also been shown to be an obstacle for patient recuperation in Intensive Care Units [26] and after surgery [27]. In [28], it is shown how good pain assessment is crucial for a good pain control, which is usually verbally checked by professional nurses, known as self-report. However, this is not always possible due to the age of the patient, the particular illness or language impairments. Moreover, pain is a subjective feeling which can be described differently across cultures [29]. Thus, pain assessment could be highly benefited from automatic tools.

Indeed this goal has been already addressed several times in the past, for example in 2011 the authors of [30] tackle the problem using brain activity imaging.

Figure 2.1: **Examples of pain and no pain frames.** This figure shows how hard it can be to distinguish between pain and no pain frames. The subject was not in pain in the frames of the first row (a), whereas it was suffering pain in all frames of row (b). At first glance it is very hard to determine which row contains pain frames and which one contains frames labeled as zero pain level, demonstrating that the task of pain detection is not trivial and that the proposed model faces a lot of difficult cases like the ones being shown.

So pain detection is also an important task from the point of view of computer vision, since it is a clear step towards an automatic detector of spontaneous face expressions [31], [32], [33], [34] and [35]. In particular, it was of high importance for the computer vision community the release of a database published by Lucey *et al.* in [36], in order to alleviate the lack of representative data of the other existing databases. Their UNBC-McMaster database consists of 200 video sequences taken from 25 patients who were suffering from shoulder pain. The frames were labeled using the validated Prkachin and Solomon metric [37] (PSPI) based on the Facial Action Coding System (FACS) [38], which codes different movements of the face muscles with different intensity levels. It is a very challenging dataset, and as it can be seen in Fig. 2.1, in some cases it can be very hard to determine whether a subject is in pain or not, even for clinical professionals.

So the UNBC-McMaster Painful dataset has been used to propose new models for facial pain detection. In the first place, Lucey *et al.* in [36] already released a baseline along with the dataset, using Support Vector Machines (SVMs) on top of the pixel and landmark features extracted using Active Appearance Models (AAM) [39] in order to predict painful Action Units (AUs) and the PSPI for the presence of pain. [35] proposed a late fusion of shape and appearance features in order to predict the continuous PSPI scores of the Painful data.

In fact, facial Action Units have been typically used to encode facial motion

corresponding to different facial expressions such as pain or anger. As stated by Rudovic *et al.* [40], the task of AU intensity estimation is very challenging, due to the high variability in facial expressions depending on the context, such as illumination, head movements or subject-specific expressions. Being a complex task, Action Unit intensity estimation has received a lot of attention over two decades for generic facial motion analysis. It has been approached by Kim *et al.* in [41], where they use a dynamic ranking model to overcome the difficulty of the emotion intensities differing substantially across subjects. Valstar *et al.* [42] also tackle the task of facial AUs recognition by using a facial point detector to localize 20 facial fiducial points. Then these points are tracked through a sequence of images and then a combination of GentleBoost, SVMs and hidden Markov models (HMM) is used for AU recognition. According to [43] most of the temporal graphical models such as HMM or conditional random fields (CRF) used for AU recognition fail to jointly model different emotions. To overcome this issue, they propose the use of a Hidden Conditional Ordinal Random Field (H-CORF) to achieve both intensity estimation of facial expressions and dynamic recognition of multiple emotions at the same time. Ming *et al.* [44] proposed a method based on multi-kernel SVM and feature fusion to approach AUs intensity estimation.

Focused on facial landmark estimation for pain detection, Rudovic *et al.* [45] proposed to use a heteroscedastic Conditional Ordinal Random Field (CORF) model in order to deal with the inter-subject variability of the pain expressions. Authors in [46] and [47] used weakly supervised learning and multiple instance learning to predict pain only using sequence-level annotations. Khan *et al.* [48] also used the referenced dataset for pain/no-pain recognition using shape information extracted with a pyramid histogram of orientation gradients (PROG) and appearance information using a pyramid local binary patterns (PLBP). Subsequently, Zafar and Khan's [49] used a K-NN classifier to classify AUs using 22 facial characteristic points. In 2015, Irani *et al.* [50] use Spatiotemporal Feature Extraction in order to model the exploits the released energy of the facial muscles in the spatial and temporal domains. They applied their system to both RGB [51] and RGB-Thermal-Depth [50] facial images. Presti and Cascia [52] use Hankel Matrices to represent the temporal dynamics of a sequence of Face Image Descriptors. Pedersen [53] addressed the identity bias of the dataset using autoencoders, ensuring the presence of discriminative features by training with a combined loss function that balances the reconstruction error and the classification error. Later in the same year, Neshov and Manolova [54] used SVMs on top of Scale Invariant Feature Transform (SIFT) features for continuous and discrete PSPI prediction. Rathee and Ganotra [55] proposed the use of Thin Plate Spline (TPS) mapping [56] for modeling the deformation of facial features and a Distance Metric Learning (DML) method to ensure the distance between features belonging to different levels of pain. The recent work of Zhao *et al.* [57]

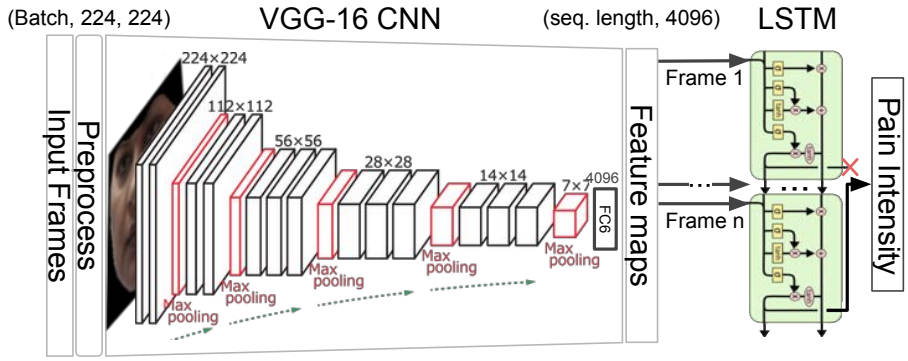


Figure 2.2: **Proposed framework.** Schematic depicting the different stages of our proposed pain detection model.

proposes the novel Alternating Direction Method of Multipliers (ADMM) to solve Ordinal Support Vector Regression (OSVR) achieving competitive performances in supervised, semi-supervised and unsupervised prediction of PSPI scores.

In this paper we continue current trends on deep learning [58][59][60][61][62] applied to pain estimation [63]. Similarly to [63], we also perform regression with Deep Convolutional Neural Networks (DCNNs) in order to predict the PSPI score for each frame. Subsequently, we adapt the resulting CNN model for pain classification inspired by [36]. In order to alleviate the problem of data scarcity, we use VGG_Faces, i.e. a VGG-16 CNN [62] pre-trained with millions of faces [64], which already obtains state-of-the-art scores compared with other leave-one-subject-out methods.

Differently to [63], we follow the ideas exposed in [51], by directly exploiting the temporal axis information using Long Short-Term Memory (LSTM) [65][66] on top of the previously-learned deep features, boosting our scores even more. So the main difference of our Deep Learning methodology as described above and the Recurrent Convolutional Neural Networks used in [63] is that we leverage the temporal information without renouncing to the representational power of generic pre-trained CNN features like the ones learned from VGG_Faces, i.e. we link the VGG_Faces features to the LSTM Recurrent Network. In other words, the approach of [63] either discards the temporal information of the data when considering pre-trained features from VGG_Faces or considers temporal information but using less-discriminative features, since the RCNN is learned from scratch.

In addition, differently to [63], we consider the raw image as the input of the CNN, rather than using facial landmarks. By doing so, the proposed method is able to outperform current state-of-the-art in pain intensity estimation.

As pain detection is a form of facial expression recognition, similar methods can be applied to the more general task of emotion recognition. For example Lucey *et al.* [67] used an SVM on top of features extracted using AAM to build a facial emotion classifier. Based on the observation that only a few facial patches are important for expression recognition, Zhong *et al.* [68] use a two-stage approach. First LBP features are used to describe every patch on a grid of 8×8 over the images of 96×96 pixels. Then Multi-task sparse learning (MTSL) is used to learn common patches across expressions. Similar to this idea, Liu *et al.* [69] propose a method which adapts 3D Convolutional Neural Networks (3D CNN) to detect facial action parts under spatial constraints. In the work by Liu *et al.* [70] they propose to use a Boosted Deep Belief Network to jointly learn the best set of features to describe expression related facial appearance and a classifier on top of these features to perform emotion recognition. Jung *et al.* [71] approached the task by using deep learning techniques. Specifically, their method combines two deep networks: the deep temporal appearance network (DTAN) and the deep temporal geometry network (DTGN). The DTAN receives as input raw images, whereas the DTGN receives the position of the facial landmarks points. Thus, the DTAN learns to extract appearance features and the DTGN extracts geometrical features. Mollahosseini *et al.* also used a deep learning approach, but in this case, they use only one CNN, with the difference that it has several Inception modules. In the work by Zhao *et al.* [57] they propose the Peak-Piloted Deep Network (PPDN) to use the peak samples (frames with maximum expression) to supervise the feature responses for the non-peak frames of the same emotion and the same subject. Their approach is to minimize both the classification error and the difference in the representations of both frames, and at the same time, they propose the usage of Peak Gradient Suppression (PGS) to prevent the representations of peak-frames driving towards the representations of non-peak frames.

2.2 Proposed Approach

The block-diagram of the proposed system is shown in Fig. 2.2. We use the same data registration as the one used by Lucey *et al.* [36] for fair comparison: images are cropped using the provided landmarks and then frontalized. Then, we apply global contrast normalization before feeding the images to a deep convolutional neural network pre-trained with faces [64]. Contrary to most of the approaches and in the same line as Kaltwang *et al.* [35], we try to solve the regression task because it fits best to this problem. However, we finally threshold the predictions in order to get performance metrics so that we can compare to [72] or [36] as previously seen in the introduction. The following sub-sections go through the steps of the system.

- **Data Pre-processing.** As it can be seen in Figures 2.3 and 2.4, we use the provided landmarks in order to crop and frontalize the faces. Following the procedure in [36], we use Generalized Procrustes Analysis (GPA) to align the landmarks [73]. This method is no more than an extension of the Procrustes Analysis for comparing more than two ordered sets of landmarks. For the simple case, in order to align two sets $X = \{x_1, x_2, ..., x_n\}$, $Y = \{y_1, y_2, ..., y_n\}$ of N landmarks, one has to (i) move their centroids $\bar{x}, \bar{y}$ to the origin (ii) find their scaling factor s:

$$s = \frac{\sqrt{\sum (x_i - \bar{x})^2 + (y_i - \bar{y})^2}}{N} \forall x_i, y_i \in X, Y, \tag{2.1}$$

so that we can remove it from the landmarks by dividing them by s. Then, one can find the rotation θ between two sets of landmarks by optimizing the rotation angle needed to minimize the mean squared distance between the two sets. This leads to the following equation:

$$\theta = \tan^{-1}\left(\frac{\sum_{i=1}^{N}(w_i y_i - z_i x_i)}{\sum_{i=1}^{N}(w_i y_i + z_i x_i)}\right). \tag{2.2}$$

Then, for K sets of points, the GPA consists in choosing one of the sets as a reference in order to align the rest, use the mean of the alignment as a new reference and repeat the process until the Procrustes distance $d = \sqrt{\sum (x_i - y_i)^2}$ between the new reference and the previous one are below a threshold. Once the final reference is obtained, the images are aligned so that their respective landmarks are aligned to it. Then, Delaunay triangulation is used to create a mesh corresponding to the dual graph of the Voronoi diagram of the points so that piecewise-affine warping can be used to get the so called *canonical normalized appearance*. As it can be seen in Fig. 2.4, we did not use all the provided landmarks since it forces too much the facial expression, i.e. eliminates mouth gestures and closed eyes, and we did not want to lose any pain-related information.

Contrary to the procedure described in [36] and followed by others, e.g. [53], we do not grayscale the image and we warp it to 224×244 because it is the common input size for most deep neural network models after cropping. We do not crop patches during training due to the fact that faces are already aligned so there is no need for translation invariance.

Finally, per-pixel mean subtraction is performed in order to pass real zeros for the black areas to the neural network. Global contrast normalization is then applied to ease the training of the model.

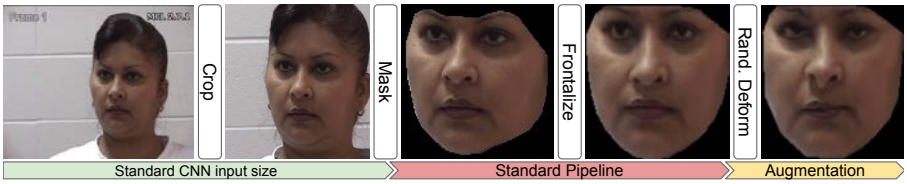


Figure 2.3: **Pre-processing pipeline.** The different stages of data pre-processing that were applied to the image. First, the image is aligned and cropped so as to fit the standard CNN input size. Then, it is masked and frontalized using piece-wise affine warping to match the standard pipeline proposed in [72]. Finally we perform data augmentation by applying landmark-based random deformations.

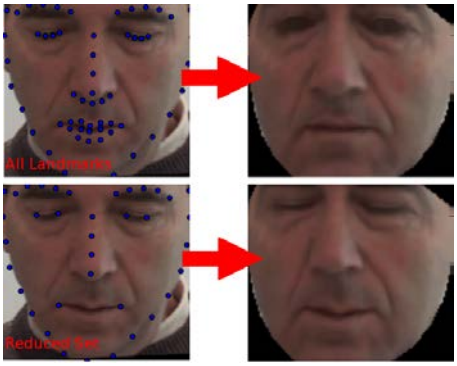


Figure 2.4: **Frontalized Images.** This figure shows the difference between frontalizing using all the provided landmarks or a coarser subset. It can be seen that using a smaller subset, the eyes preserve their state and the line of the mouth is more similar to the original frame.

- **Facing imbalanced data** Since there are about 8K pain frames and about 40K labeled as no pain in PSPI score, it is probable that any model gets biased towards the prediction of no-pain at the cost of missing pain frames. There are two common approaches to overcome this problem: (i) balancing data, (ii) using weighted loss functions. In this work we balance the training data (i) and validate the original validation data, but we also complement the results by giving normalized scores, as proposed by [74] (i.e. balancing the validation dataset). To balance the data, we randomly under-sample the majority class, i.e the no-pain class, so that both pain and no-pain categories have the same

probability to be randomly picked by the training algorithm. To create the training sequences for the RNN we also need to balance the data, but instead of balancing at the frame level, we balance at the sequence level so that there are no frame skips. This means that we sort the frames in time, split them in sequences, and discard entire sequences with no pain in all their frames until they match the number of sequences with pain inside.

- **Target pre-processing** Because MSE is very sensitive and most suited for the cases where Gaussian noise is present, it is good practice to standardize the labels, i.e. the pain levels, before training.

After data is pre-processed, it is used to train a CNN to perform the pain level recognition task. This is achieved by fine-tunning a VGG-16 CNN pre-trained with Faces [64]. Instead of using the log-likelihood objective function, we used the $L2$ between the predicted label $\hat{Y}$ and the actual label Y in an attempt to make the model get a better insight on pain detection since it is not binary and it actually proved to perform slightly better:

$$E = \frac{1}{2N} \sum_{n=1}^{N} ||\hat{y}_n - y_n||_2^2. \tag{2.3}$$

In order to improve the model generalization, data augmentation is used. This is done by (i) flipping images with 50% probability, and (ii), adding random noise to the reference landmarks before performing piece-wise affine warping in order to introduce small deformations to faces, see Fig. 2.3.

The masking and the frontalization performed during the pre-process alter the original face, resulting in an image considerably different from a non-processed face like the ones that the CNN used has been pre-trained with. These differences between the pre-training data and the fine-tunning data could affect the results obtained, because the network has learned to extract specific features from raw face images, and it may not be able to extract them from the processed faces. Thus, we also provide results with a network trained with raw faces, similar to the ones used during pre-training, and each frame is processed only to extract a crop around the face, see Fig. 2.3, and then the mean pixel value is subtracted to each image.

2.2.1 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are an architecture of neural networks proposed by LeCun *et al.* that localize local features in images to extract information of the visual content [75]. CNNs are made of different types of layers, stacked on top of each other. The basic layer of a CNN is the convolution layer, which convolves a

given tensor of size

$$W \times H \times D,$$

with K different filters of size

$$F \times F \times D,$$

with a stride of S between convolutions and padding the input with P zeros. This convolution of the input by K filters outputs a tensor with dimensions:

$$W' \times H' \times D',$$

where

$$W' = (W - F + 2P)/S + 1,$$

$$H' = (H - F + 2P)/S + 1,$$

$$D' = K.$$

The values of the convolution filters are learned by initializing them randomly and updating them by performing gradient descent using the backpropagation algorithm [76]. To compute the error for a given input to the network, the last layer of the network is a loss layer which computes the error between the ground truth label of an input image and the predicted output for that image. This error at the output is backpropagated to previous layers in order to compute the gradients for the weights of previous layers.
This architecture is specially designed to capture 2D information, so it performs very well on images, where pixels intensities are related to their neighbors. The recent increase in computational power provided by GPUs and the availability of large datasets like Imagenet [77] have made the initial CNN implementations evolve to very deep networks [61], [62]. These deep networks have been proven to perform very well in a variety of computer vision tasks such as human action recognition [78], handwritten digit recognition [79] or automatic face detection [80].

2.2.2 Using temporal information

Although we are using video data, the previous sections only deal with the problem of labeling isolated frames. Thus, temporal information can still be used in order to improve the model. In order to take it into account, similarly to the work of [63], the features from the `fc6` layer are extracted and used to feed a recurrent neural network (RNN). This kind of neural nets is especially suited for sequential data since their neurons do not only have connections (weights) between the next layers but to themselves, which are used to keep information from previous inputs. Since they

have to be unrolled, the training of this kind of networks is done with an extension of the back-propagation algorithm [76], called back-propagation through time BPTT [81].

In this work we use LSTM, a type of RNN which is capable of learning long-term dependencies present on sequential data. Standard RNNs are theoretically capable of learning long-term dependencies, but in practice, it is difficult to train them because the gradients tend to either explode or vanish [82]. LSTM differs from standard RNN because it has a cell state controlled by 3 gates, which decide how much information should be let through. These gates are known as forget, input and output gates, see 2.2. The amount of information that is let through each gate is controlled by a point-wise multiplication and sigmoid function, as the sigmoid function output is between 0 and 1, indicating how much of the information should let through the gate.

At each time-step, the input gate is computed depending on the input to the LSTM for that time-step and the previously hidden state. The cell state candidate is also computed by:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i), \tag{2.4}$$

$$\hat{C}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c). \tag{2.5}$$

Then output of the forget get is computed as:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f). \tag{2.6}$$

And when the forget and input gates have determined how much information of the previous cell state C_{t-1} and the new cell state candidate $\hat{C}_t$ should be let through, the cell state for the current time-step is computed:

$$C_t = f_t * C_{t-1} + i_t * \hat{C}_t. \tag{2.7}$$

Then, the state can be used in order to predict the output of the cell:

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o), \tag{2.8}$$

$$h_t = o_t * \tanh C_t. \tag{2.9}$$

In order to train the RNN for pain detection, we used the MSE loss since it better

Table 2.1: **Summary of previous approaches.** This table compares the experimental setup of previous approaches to solve the task of automatic pain detection. We compare our method against the previous approaches that have used a subject-exclusive leave-one-subject-out performance measure and do not discard any painful image. * Custom split, different than leave-one-subject-out.

	Feature descriptors	Classifier	Metric	Score	All images
Lucey [83]	PTS, APP,	SVM	AUC	78%	Yes
Lucey [84]	PTS, APP,	SVM	AUC	78.4%	Yes
Lucey [36]	SPTS, CAPP	SVM	AUC	83.9%	Yes
Lucey [72]	SPTS, SAPP, CAPP	SVM	AUC	84.7	Yes
Kaltwang [35]	PTS, DCT, LBP	RVR	MSE, PCC, ICC	1.39, 0.59, 0.50	Yes
Florea [85]	HoT	SVR	MSE, PCC	1.21, 0.53	Yes
Zhou [63]	learnt	RCNN	MSE, PCC	1.54, 0.65	Yes
Zhao [57]	LBP,Gabor	OSVR-L1,L2	MAE,PCC,ICC	0.81, 0.60, 0.56	Yes
Ashraf [34]	S-PTS, S-APP, C-APP	SVM	Hit rate	82%	No
Hammal [86]	CAPP	SVM	Recall, F1 Precision	61%, 57% 65%	No (Only 15%)
Rudovic [45]	LBP	KCORF	F1*	40.2%	No
Khan [48]	PHOG, PLBP	SVM, DT RF, 2NN	10 fold CV, Accuracy	96.4%	Yes
Rathee [55]	TPS	SVM	Accuracy*	96.0%	Yes
Pedersen [53]	Custom features	SVM	AUC, Accuracy	96.5, 86.1%	No

suits the nature of the problem, where pain levels have distances in the output space. In case we need to compare in terms of binary accuracy, we can just use a binary threshold. In fact, we empirically found that using the cross-entropy error for binary classification yielded worse performance than just using a threshold after regression. Concretely, we could only reach 81% of accuracy on the test set with the initial settings shown in Table 2.2, which presents a 83.1% for the same model after regression and thresholding.

To train the LSTM, first a feature vector has to be extracted for each image, being this vector the input to the LSTM. We can think of this feature vector as a low-dimensional representation of the image in the feature space. To create this vector for each frame, the frame is processed through the VGG-16 CNN fine-tuned to perform pain level detection and the outputs of the a fully-connected layer are used as the encoding for that frame. As it can be seen in Table 2.3, we found that the outputs in the `fc6` layer had less temporal invariability than the ones from the `fc7` and thus, the former yielded better performance when fed to the LSTM. Hence, the `fc6` is always used for comparison with the state-of-the-art. This process results in M feature vectors v where M is the number of frames and $v \in \mathbb{R}^{4096}$ since the `fc6` layer of the VGG-16 network has 4096 units. Then, the M feature vectors have to be grouped together in sequences of length ρ. The

sequences are created so that each frame is the last of a sequence once, e.g if the first sequence is $s_0 = \{v_0, v_1, ..., v_{n-1}, v_n\}$, the next sequence is $s_1 = \{v_1, v_2, ..., v_n, v_{n+1}\}$. Each sequence s is labeled with one label t, corresponding to the label of the last frame of the sequence. In the classification task, t is a binary one-hot vector $t \in \{0,1\}^2$, and for the regression task t is a real number $t \in \mathbb{R}$. As each sequence has only one label, only the hidden state of the last time-step h_{t_n} is used to compute the output of the network.

Hence, the label of a frame is predicted taking into account the past ρ frames. For this problem, we found that $\rho = 16$ worked well, and an LSTM [65] RNN is used in order to avoid the problem of gradient vanishing for long sequences. The network is optimized with ADAM since it has proved to be more stable than SGD with momentum [87].

2.3 Experiments and Results

As said in the previous sections, we center our experimentation on The UNBC-McMaster Shoulder Pain Expression Archive Database [36]. In addition, we prove the generality of our model by testing it on the Cohn Kanade+ face emotion detection dataset [67] and obtaining competitive results.

2.3.1 Results on Pain Recognition

A quick skim through the pain detection literature concerning the database will show the reader that there are multiple benchmark procedures. While the original paper [36] and some posterior ones [53] use leave-one-subject-out cross-validation, others like [45], [48], and [55] use k-fold cross-validation or even leave-one-frame-out cross-validation. In addition, Jeni *et al.* face the problem of data imbalance in [74], proposing normalized metrics that take the skew into account.

In Table 2.1 there is a summary of previous approaches to performing pain detection on the same dataset, indicating the method used to extract features and the classifier or regressor trained with those features. It also shows the metric used to evaluate their approach, along with the score obtained and the performance measure. The main difference between most of the listed previous approaches and our approach is that they manually extract a set of features, and then train a model with them, whereas we use an end-to-end deep learning model which learns to extract features from the data and how to combine them to give the correct output. Our approach is also based on Convolution Neural Networks as in [63], but in contrast, we apply temporal modeling using LSTM onto the features learned from the VGG_faces network. This is different from the method proposed

Table 2.2: **Unbalanced and normalized scores.** This table reports the accuracy and area under the ROC curve obtained by different versions of our method.

	Normalized [74]		Unbalanced	
Metric	**Accuracy**	**AUC**	**Accuracy**	**AUC**
Align	77.1	83.2	83.1	83.1
Align + Fron.	83.2	82.4	86.4	82.1
Align + Front. + Data aug.	85.9	89.9	88.8	89.9
Aligned Crop	80.8	90.0	87.5	89.6
Aligned Crop + LSTM	83.8	90.1	90.3	91.3

in [63], which discards the temporal information of the data when considering pre-trained features from VGG_Faces, and considers temporal information on low-discriminative features, since the RCNN is learnt from scratch in an unbalanced, quite small dataset (even smaller in [63], since no data augmentation pre-processing is applied).

In this work, we compare within the dataset authors' scheme: AUC score on leave-one-subject-out cross-validation, since subject-exclusiveness increases the confidence that the model will behave similarly with new data. In addition to comparing our model in a binary setting by using the AUC score, we also test it against other state-of-the-art continuous prediction models with the Intraclass Correlation Coefficient (ICC), Pearson Correlation Coefficient (PCC), the Mean Square Error (MSE), and the Mean Absolute Error (MAE). For the continuous setting, we aggregated the pain levels as indicated in [57] so that the levels 4 and 5 are merged, as well as 6+, that become the 5th level.

In our case, and only for comparison purposes, we also trained on aligned and canonical normalized faces but including data augmentation to add robustness to the model predictions. In Table 2.2, we show the effect of the different stages of pre-processing shown in Fig. 2.3 on the performance of the model. Specifically, it can be seen that the aligned frontalized facial landmarks proposed in [36] already provides a good performance, but the VGG_faces model is not pre-trained with similar kind of images [64]. In fact, it is interesting that with canonically normalized appearance, the position and translation invariances of the faces are not enough to compensate their difference with the pre-trained model. We also found very important the mean subtraction step since the pre-trained model was trained with faces with some background and the canonically normalized appearance contains a black background. Hence, subtracting the global pixel mean was making all

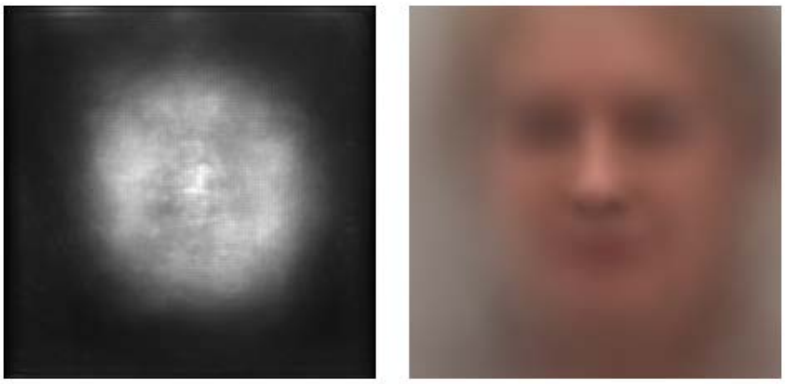


Figure 2.5: **Average saliency map and average face.** The first picture (a) corresponds to the average saliency map computed for each image as described by Simonyan *et al.* [88]. the second picture is the average of all the training images. The saliency map shows where the CNN is looking to decide the level of pain of a frame.

those zeros to be non-zero and thus lower the performance. The solution was to subtract the per-pixel mean. The best score for the AUC metric, 89.9 is achieved by considering the so popular pre-processing step, as used in [36].

The last 2 rows in Table 2.2 show the results obtained by our model when it is trained with centrally cropped Procrustes aligned faces. With this different setting, the performance of the model is enhanced, only matched by the canonically normalized setting when heavy data augmentation was used (face deformations). The main reason to this gain is due to the fact that VGG_Faces is pretrained with millions of raw images.

A possible drawback of keeping the image background could be that the CNN is helped by non-facial information (such as the arms) to improve its performance. In order to verify that the model is ignoring the background and that it is using only face information, we performed a class saliency visualization as described by Simonyan *et al.* [88]. In Fig. 2.5 it can be seen the average saliency map compared to the mean face, and by comparing both pictures it can be seen that the network bases its decision looking at the face region, without using background information. The average saliency map has been obtained by computing the saliency map of all the images and averaging them. According to [88] the saliency map of an image can

be thought as the magnitude of the derivative of the output S_c with respect to the input image I, because the magnitude of the derivative indicates which pixels need to be changed the least to affect the output the most, and therefore those pixels correspond to the region of the image that the network is using to give its output. The derivative is computed as following:

$$w = \frac{\partial S_c}{\partial I}, \tag{2.10}$$

and the saliency map $M \in \mathbb{R}^{m \times n}$ for an image $I \in \mathbb{R}^{m \times n}$ is computed as:

$$M_{ij} = max_c |w_{h(i,j,c)}|, \tag{2.11}$$

where $h(i, j, c)$ is the index of the element in w that corresponds to the i-th row, j-th column and c-th colour channel value of the image I. As the saliency map does not have a color dimension, the maximum magnitude of w across all colour channels is selected to create the map.

The UNBC-McMaster Shoulder Pain Expression Archive Database is unbalanced, meaning that there are a lot more frames labeled as zero pain than frames labeled with some level of pain. There is a total of 48398 frames coded with a pain intensity, 40029 of them being labeled as zero pain-intensity. This means that the 83.6% of the examples of the dataset belong to the same class, whereas only the other 16.4% examples have some level of pain [36]. As stated by the authors in [74] the results of the accuracy metric is influenced by the skew in the testing data, whereas the AUC metric is not affected that much. Therefore, to avoid providing a score which is influenced by the skew in the data set, in Table 2.2 the first two columns correspond to the accuracy and AUC obtained when the score is skew normalized to mitigate the effect of imbalanced data. The last two columns correspond to the scores obtained testing the models with an unbalanced distribution. In the same way as the authors in [74], to calculate the skew normalized scores shown in Table 2.2, we under-sample the majority class at test time. This means that we randomly choose a set of no-pain samples (the majority class) that has as many images as the pain class (the minority class). Then, the normalized scores provided are calculated based on those samples. As stated by [74] the results of the accuracy metric are influenced by the skew in the testing data, whereas the AUC metric is not affected that much. That is why the accuracy scores change significantly when score normalization is applied and the AUC scores don't differ much. Accuracies are reported with a threshold interval of $[0, 1)$ for no-pain and $[1, \infty)$ for pain. It is important to remark that just a square crop centered on the nose of the subjects already performed very good in terms of AUC. However, for a fair comparison with

Table 2.3: **Comparison against binary leave-one-subject-out methods with AUC scores.**

	AUC
Lucey *et al.* [36]	83.9
Lucey *et al.* [72]	84.7
Aligned crop (Ours)	89.6
Frontalization (Ours)	89.9
Aligned crop + LSTM on fc7 (Ours)	**91.3**
Aligned crop + LSTM on fc6 (Ours)	**93.3**

Table 2.4: **Comparison against continuous leave-one-subject-out methods with MAE, MSE, PCC, and Intraclass Correlation (ICC).**

	MAE	**MSE**	**PCC**	**ICC**
Kaltwang *et al.* [35]	-	1.39	0.59	0.50
Florea *et al.* [85]	-	1.21	0.53	-
Zhou *et al.* [63]	-	1.54	0.64	-
Zhao *et al.* [57]	0.81	-	0.60	**0.56**
Aligned crop + LSTM	**0.5**	**0.74**	**0.78**	0.45

previous work, scores for cut faces are also provided. Fig. 2.2 shows a fragment of the ground-truth data compared to the predictions of our model. It can be seen the model is highly correlated with the data and most of the mistakes are due to frontier effects. E.g. when a subject just stopped to feel pain, muscles relax with some lag. A similar effect happens when a subject reported pain before the facial expression completely changed.

Tables 2.3 and 2.4 show the achieved model is competitive enough to achieve state-of-the-art results using the thorough leave-one-subject-out setting. A more detailed analysis of the binary performance of our model has been conducted, evaluating the results on each subject. Table 2.5 shows the number of pain frames and no-pain frames per subject, indicating how many of them have been correctly classified by our model. As it can be seen in Table 2.3, using the same preprocessing as [72], our model already outperforms the previous state-of-the-art AUC scores. Namely, Lucey *et al.* [72] train a model to detect the presence of facial action units (AUs) from a set of facial features, while our model tries to directly find the best hierarchy of features to infer pain from the pixel level. Then, [72] use these features

Table 2.5: **Number of correctly classified pain and no-pain frames for each subject.** This table shows the number of pain and no-pain frames per subject, and how many of them are correctly classified. It can be seen that the main source of classification error is subject 20.

	Not pain		Pain	
Subject	**Correct**	**Total**	**Correct**	**Total**
0	1807	1827	122	221
1	354	408	15	40
2	547	571	60	133
3	1461	1472	57	64
4	1867	2059	158	181
5	2148	2171	463	517
6	876	1000	339	408
7	2344	2403	45	93
8	2486	2699	539	821
9	1060	1116	55	100
10	2277	2361	350	455
11	1371	1396	42	76
12	1564	1863	468	505
13	913	944	59	80
14	3034	3116	45	148
15	2026	2164	267	524
16	428	641	784	959
17	713	734	183	354
18	1376	1376	71	160
20	806	844	494	1076
21	1421	1478	218	442
22	1603	1613	103	179
23	634	684	37	84
24	300	311	376	393

to train an SVM to detect each AU while the neural network is end-to-end, i.e. it learns to extract the features and also learns to use them to predict the level of pain. Furthermore, when frames are just aligned using Procrustes analysis, we leverage all the potential of the pre-trained model, not only outperforming previous AUC scores by a large margin, but achieving state-of-the-art results in terms of MAE, MSE, and PCC; when compared with the most recent literature (as it can be seen in

Figure 2.6: **Examples of emotion frames.** This figure shows one frame of each of the seven emotions. From left to right: anger, contempt, disgust, fear, happiness, sadness, surprise.

Table 2.4).

Summarizing, we have demonstrated that considering the raw image and temporal information at the pixel level allows our model to outperform the results obtained by previous canonical normalized appearance [36] approaches.

2.3.2 Results on Emotion Recognition

Pain recognition from facial gestures is a specific task within the broader task of facial expression recognition. In order to evaluate the effectiveness and robustness of our proposed method, we apply it to the task of emotion recognition from facial pictures. Facial expressions can show different human emotions such as anger, disgust or happiness [89] so the task of emotion recognition from pictures of faces can be approached as a facial expression recognition task. Our method for pain recognition can be adapted to perform facial expression recognition very easily. For pain detection we perform a regression task, i.e. predicting the pain intensity of a face picture. To switch to emotion detection, we must now perform a classification task. To do so, we changed the number of output units in the output layer of the CNN from 1 output unit to N, where N is the number of emotions we want to recognize in one-hot encoding. The loss function was also be changed to the cross-entropy error between the correct output y and the predicted output $\hat{y}$ as defined by the equation 2.12:

$$E(y,\hat{y}) = \sum_{n=1}^{N} y_n log(\hat{y}_n) \tag{2.12}$$

The output of the network $\hat{y}$ is the result of applying the softmax function to the outputs of the last layer, and the true label y, which is the one-hot representation of the emotion label assigned to a sample. To test our method on emotion recognition we used the Extended Cohn-Kanade Emotion Dataset (CK+) [67].

CK+ Dataset

The emotion recognition CK+ dataset [67] has 593 sequences of 123 subjects which are FACS coded at the peak frame. In each sequence, the subject face evolves from a neutral face to a peak facial expression. Only 327 of the sequences are labeled with one of the following seven emotions: anger, contempt, disgust, fear, happy, sadness, surprise. In Fig. 2.6 there is an example of a peak frame for each of the seven emotions present in the dataset. Following the trend in other works [68, 90, 91], we split the sequences into 10 subject-exclusive folds in order to perform a leave-one-fold-out cross-validation to test our method on this dataset. To make sure that the classes are evenly distributed among folds, the subjects are randomly separated into 10 groups. In the same way as in other works [70, 90] we select the last three frames of each sequence to train the CNN. To train the LSTM we must provide fixed-length sequential inputs, and as the videos vary in duration, from 10 to 60 frames approximately, we have chosen the length of the sequences to be 10. For each video, we generate three different sequences of length 10, each sequence ending in one of the last three frames. If there aren't enough frames in the video to build a sequence of length 10, the first frame is repeated at the beginning of the sequence. The results provided for the CK+ dataset are obtained by training on 9 of the 10 folds and leaving one out for testing, and repeating the process until each fold has been used for testing at least one. The accuracy provided is the average within the 10 folds.

Results on CK+

We provide two results for the CK+ dataset, the baseline accuracy obtained by the emotion classifier built on top of the CNN and the accuracy obtained by the LSTM model. In Table 2.6 a comparison of our method scores against other state-of-the-art procedures reported in the literature can be seen. The results shown in the table are from seven emotion classes: anger, contempt, disgust, fear, happy, sadness, and surprise. The confusion matrix of the predictions on the test folds can be seen at Fig. 2.7. Other works [92] provide scores for the eight class problem where the neutral emotion is added. We can not construct sequences ending in a neutral frame because the neutral frame is always the first one, so we do not provide results for this task.

2.4 Discussion

Pain recognition has been proved to be an important task for health-care. In this work we have faced the task of binary pain recognition on facial images from

Table 2.6: **Results on the CK+ dataset**

	Accuracy (%)
Zhong *et al.* [68]	89.9
Liu *et al.* [69]	92.4
Mollahosseini *et al.* [93]	93.2
Liu *et al.* [94]	94.2
Sikka *et al.* [91]	95.1
Liu *et al.* [70]	96.7
Jung *et al.* [71]	96.9
Zhao *et al.* [90]	**97.3**
Aligned crop (Ours)	94.5
Aligned crop + LSTM (Ours)	97.2

the deep learning perspective achieving state-of-the-art results when compared to leave-one-subject-out setups. This, however, has also exposed the problem of stating which is the correct comparison methodology since results from other works have been provided in terms of accuracy, AUC, subject exclusive and non-exclusive settings. We believe subject-exclusiveness is crucial and thus, provided all the results computed this way. Our approach of training a deep CNN for pain-level estimation already provided good results, and we have proved that using an RNN to exploit the temporal relation between frames improves the results even more. By training a CNN end-to-end to perform pain-level estimation our approach obtained an AUC of 89.6, increasing up to 93.3 when that same CNN is used to the extract features to train the RNN. Moreover, we prove the generality of our method by obtaining an accuracy of 97.2% on the CK+ facial emotion recognition dataset, a competitive score when compared to the state-of-the-art (97.3% in [90]).

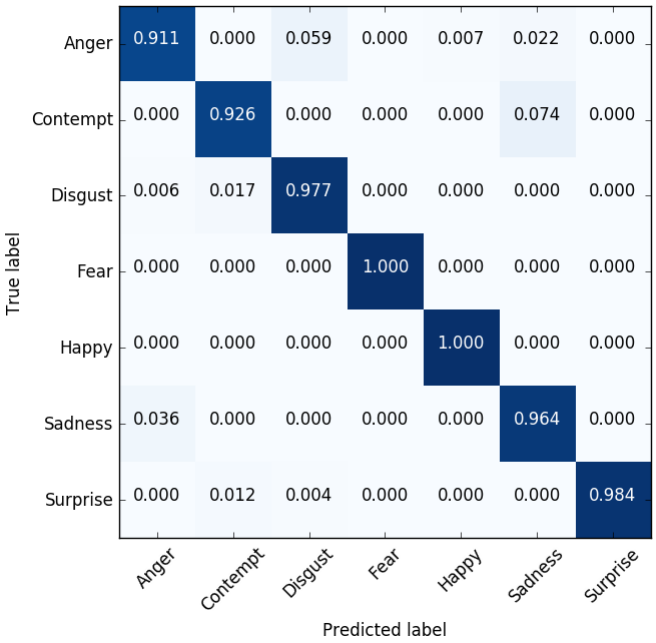


Figure 2.7: **Emotion detection confusion matrix.** Confusion matrix for the task of emotion detection in the CK+ dataset for the seven classes.

3 Age and Gender Recognition in the Wild with Deep Attention

Face analysis in images in the wild still pose a challenge for automatic age and gender recognition tasks, mainly due to their high variability in resolution, deformation, and occlusion. Although the performance has highly increased thanks to Convolutional Neural Networks (CNNs), it is still far from optimal when compared to other image recognition tasks, mainly because of the high sensitiveness of CNNs to facial variations. In this paper, inspired by biology and the recent success of attention mechanisms on visual question answering and fine-grained recognition, we propose a novel feedforward attention mechanism that is able to discover the most informative and reliable parts of a given face for improving age and gender classification. In particular, given a downsampled facial image, the proposed model is trained based on a novel end-to-end learning framework to extract the most discriminative patches from the original high-resolution image. Experimental validation on the standard Adience, Images of Groups, and MORPH II benchmarks show that including attention mechanisms enhances the performance of CNNs in terms of robustness and accuracy.

3.1 Motivation

Human face analysis constitutes one of the most important tasks in computer vision, since the automatic analysis of such a deformable object is of great importance [95]: the characterization of age, gender, facial attributes, expressions, garment, and even personality, to cite but a few, are crucial in several applications, like user identification, social interaction, face tracking, and behavior recognition [96, 97]. Regarding age and gender classification, although these two tasks have been largely addressed in the past, the reported performances are far from optimal [98, 99].

In the last few years, Convolutional Neural Networks (CNN) [75] have become the main workhorse for age and gender estimation. CNNs have been proven to perform very well in a variety of computer vision tasks such as human action recognition [78], handwritten digit recognition [79], face verification [100] or automatic

face detection [80]. In relation to the task of soft-biometrics analysis, CNNs have been recently applied to the task of apparent age estimation [101–103], gender and smile classification [104], and real age and gender prediction [105]. However, due to the high variability of facial images in the wild, i.e. for example collected from the web, the low performance of CNNs in tasks like age recognition shows that there is still room for improvement.

The main contribution of this paper is a novel feedforward attention mechanism that enhances current CNNs' robustness for highly variable unconstrained recognition tasks. Thus, inspired by biology, and the recent success of attention mechanisms [106], we propose a feedforward attention mechanism to discover the most discriminative patches of low resolution unconstrained facial images in order to process them in high resolution. So, beyond the increase in resolution, our method allows the network to assign more importance to the least occluded or deformed parts of the image, thus becoming the model more robust to noise and distractors. We perform a thorough evaluation on standard age and gender recognition benchmarks [107], proving that our attention pipeline is more robust than any previous state-of-the-art CNNs pretrained for facial recognition.

In particular, including attention on CNNs shows an increases in performance for standard CNNs such as VGG-16 [108] when applied to the Adience [107], Images of Groups (IoG) [109] and MORPH II datasets [110] for the tasks of age and gender recognition. Thus, on one hand, Adience and IoG consist of unconstrained facial images captured in the wild, showing that our model is capable of detecting soft biometric traits such as age and gender from facial pictures captured in uncontrolled environments, with distractions, deformations, and occlusions. Moreover, on the other hand, the proposed mechanism also shows improvement in controlled environments such as the MORPH II databaset, thanks to using higher resolution fixations.

3.2 Related Work

This section discusses other work that is relevant to understand our approach, together with the context and historical evolution in the use of neural networks for gender and age recognition.

3.2.1 Age Recognition

Not only the first studies in the 90s used the analysis of facial geometry [111] to estimate the age of a person, but also more recent techniques like the pipeline used in [112], presenting a combination of Biologically Inspired Features (BIF) and then

using Canonical Correlation Analysis (CCA) and Partial Least Square (PLS) based methods. Indeed BIF were already used in [113] to represent face images, paving the way to works like [98, 114], showing that the automatic approach had matched the human performance. In fact, most of the approaches previous to CNNs were based on a two-stage pipeline, i.e. extracting features such as Local Binary Patterns (LBP) [107], and then classifying with a Support Vector Machine (SVM), or a Multilayer Perceptron (MLP) [115, 116]. On the contrary, CNN based methods typically implements the two-step pipeline described above in just one step: the network learns both extracting the best features and either classifying such features into age categories [105, 117] or performing age regression [118, 119]. Deeper CNN models have been also applied to age and gender recognition [120], although most of them depend on domain-specific pre-training [121, 122]. Cascaded combinations of deep models were also considered in [123].

CNNs for facial images analysis have not been restricted to age estimation, but also to face verification, facial attribute estimation, and gender recognition. One illustrative example is the method presented in [124] which achieves a 99.2% face verification accuracy on the challenging Labeled Faces in the Wild dataset [125]. Unfortunately, this so impressive performance has not been yet achieved in other facial analysis tasks like gender recognition, for example, as shown next.

3.2.2 Gender Recognition

Regarding gender recognition, in contrast to age analysis, there is work from the early 90s where neural networks were already proposed, like the pioneering approach presented in [126]: authors proposed two neural network structures, an autoencoder and a classifier whose input was the encoded output layer of the autoencoder. The drawback of this method was that it relied on manual cropping, scaling and rotating the face of the picture, which was taken in a controlled environment.

Inspired from the age estimation methodology, pipelines based on a feature extractor and a stacked classifier were also proposed like in [127], [128], and [129]. On the other hand, the same CNN-based methods used for age were also applied to gender [118], [105], demonstrating that CNNs are truly capable of learning how to perform different tasks without any modification besides the data used for learning. For example, in [130] a CNN is trained to perform gender recognition by fine-tuning a pre-trained network, and then an SVM is trained using the deep features computed by the CNN.

3.2.3 Neural Networks with Attention

Attention is a powerful mechanism that allows neural networks to look in more detail into particular regions of the input image to reduce the task complexity and discard irrelevant information, mildly inspired in the eye fixations performed by the human visual system [106].

Previous approaches for applying bioinspired visual attention mechanisms rely on finding visually salient regions on the image for processing them in a posterior step [131, 132]. In the context of neural networks, Larochelle and Hinton [133] proposed a third-order Restricted Boltzmann Machine (RBM) to combine high resolution "glimpses" of a sequence of fixations for image classification. Likewise, Denil *et al.* perform image tracking with an RBM fed with foveated images selected by a control pathway [134]. A simpler model was proposed by Ranzato [135], which predicts a glimpse location from a downsampled image and then uses it to extract a high-resolution patch. Spatial Transformer Networks (STN) [17] can be also considered as a form of attention, however, differently from other attention approaches like the one presented in this paper, they focus on a single spatially continuous region of the image. In all these proposed papers, attention is shown specially well-suited for images in the Wild, with multiple occlusions and distractors, which is the case of Adience and IoG datasets.

More recently, RNNs have become central to attention mechanisms since they naturally integrate the information extracted from glimpses at different time-steps [106]. The ability to look "into the past" has made RNN-based attention mechanisms ideal for Natural Language Processing (NLP) tasks such as Neural Machine Translation [136], text-based question answering [137], image captioning [138], and Visual Question Answering (VQA) [139, 140].

In our work, we assume faces have already been detected, cropped, and aligned, and thus, there is no need to do a sequential search through the image with an RNN so as to find the most relevant image regions. However, since the main hypothesis of all the aforementioned papers is that CNN models can not give the same importance to all the regions of an image, mainly due to the high variability of unconstrained environments, attention mechanisms can be suitable in our case to automatically select specific regions of a face for further processing them in more detail, while ignoring background clutter. Based on these findings, we next describe our proposed attention-based CNN models.

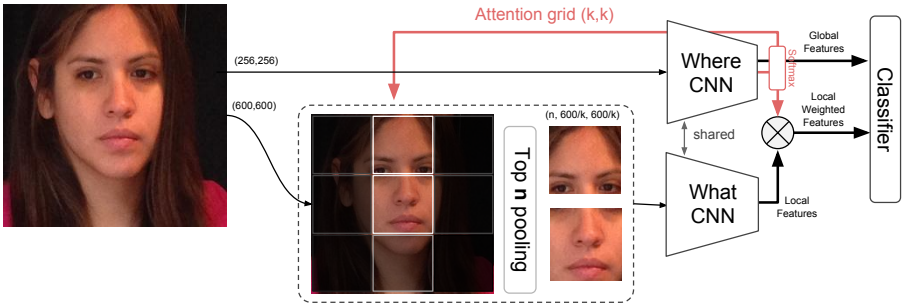


Figure 3.1: **The proposed attention model.** A lower resolution image is fed to the "where" CNN, which predicts a $k \times k$ attention grid. This grid is then used to extract high-resolution patches, from which the top n are pooled. The patches are then fed to the "what" CNN, whose output is weighted by the attention grid. Finally, feature maps from the "where" and "what" streams are concatenated and fed to an MLP classifier.

3.3 Proposed Approach

The proposed model consists of three basic modules, see Figure 3.1: (i) an attention CNN ("where") that predicts the best attention grid to perform the glimpses, (ii) a patch CNN ("what") that evaluates the higher resolution patches based on their importance predicted by the attention grid, and (iii) a Multi Layer Perceptron (MLP) that integrates the information obtained from both CNNs and performs the final classification. We detail these modules next.

The attention CNN is fed with all training images. We used the VGG-16 model since it has become the well-performing standard CNN that is supported in most of the deep learning programming frameworks [108], but any other CNN could be considered instead. This CNN is specifically trained to predict an attention $k \times k$ grid $\mathbf{G}$:

$$\mathbf{G}^{(k \times k)} \in \mathbb{R}_{\geq 0}, \qquad \sum_{i,j} \mathbf{G}_{i,j} = 1,$$

where k is an arbitrary number, and the values $\mathbf{G}_{i,j}$ represent the (normalized) importance of each patch. Then, a high-resolution version of the input image is divided in $k \times k$ patches, and fed to the patch CNN.

The patch CNN is fed with high resolution patches of the faces. Similarly to the attention CNN, any model could be used for this task, however, to reduce the computational requirements of this architecture, we reuse the first convolutional layers of the attention CNN. The output of this module consists of a matrix:

$$\mathbf{P}^{(k^2 \times d)} \in \mathbb{R},$$

where k^2 is the number of patches and d is the output dimension of the last convolutional layer. Global Average Pooling (GAP) is then used to reduce the spatial dimension of $\mathbf{P}$ to one, thus making it possible to feed images in their original resolution. These feature maps are subsequently weighted by $\mathbf{G}$ to reflect the importance of each patch of the grid.

In the literature, such weighting can be performed using either a weighted sum, denoted as "soft attention" in [138]:

$$\mathbf{P}^* = \mathbf{g} \cdot \mathbf{P}_{\mathbf{GAP}}, \tag{3.1}$$

or alternatively the element-wise product, also called the Hadamard product:

$$\mathbf{P}^* = \mathbf{g} \circ \mathbf{P}_{\mathbf{GAP}}; \tag{3.2}$$

where $\mathbf{g}^{(\mathbf{1} \times \mathbf{k^2})} \in \mathbb{R}_{\geq 0}$ is a flat view of $\mathbf{G}$. In the experimental section, we show that both strategies yield similar results. Then, on one hand, the Hadamard product can be chosen to reduce the computational time complexity at the expense of memory. And, on the other hand, the weighted sum can be chosen in limited memory scenarios but conveying higher computational time cost.

Additionally, for further reducing the computational cost, we pool the top n patches before feeding them to the patch CNN, thus using a "hard attention" mechanism instead. It is important to note that the gradients will not propagate to those grid positions outside the top n, but since the importance given to those discarded positions are zero or very close to zero, the network is still able to learn. Additionally, as it is usually done in the literature, we also performed random patch sampling given the distribution of the attention grid, however. the difference in performance when compared to sampling the top n patches is not statistically significant.

The classifier is fed with features from the `pool5` layer of the attention CNN, and the weighted features of either the `pool4` or `pool3` layers of the patch CNN. Lower level features maps from the patch CNN are preferred because they correspond to local-level image features.

We also consider two strategies for merging the feature maps of both CNNs: (i)

Figure 3.2: **Adience sample.** Sample of each age group and gender from the fourth fold of the Adience dataset.

concatenate them after an L2 normalization, and (ii) learn a projection of the patch CNN feature maps to the attention CNN feature map space, and simply add them. In the next section, we demonstrate that the normed concatenation yielded slightly better results than the project-and-add strategy.

The resulting feature maps are then fed to the final classifier, which consists of the `fc6`, `fc7`, and `fc8` layers of the VGG-16, as typically done in the CNN literature. In the following sections, several experiments are presented for testing the robustness and accuracy of the whole attention-based architecture.

3.4 Benchmark Datasets

To evaluate the performance of our approach on unconstrained facial images, we test it on the Adience dataset proposed in [107], and following the same evaluation benchmark. This dataset consists of 26.5K images distributed in eight age categories (0-2, 4-6, 8-13, 15-20, 25-32, 38-43, 48-53, 60+) with the corresponding gender label.

The Adience benchmark measures both the accuracy in gender and age recognition tasks using 5-fold cross-validation in which the provided folds are subject-exclusive. The final score is given by the mean of the accuracies of the five folds. The same subject-exclusive folds are used for age and gender, and there is a subset of only nearly frontal faces which we have not used since faces in real world images present a higher diversity on the pose. This dataset is designed to be as similar as possible as real-world challenging face images, therefore, the faces present many changes in pose, rotation, appearance, light and noise. Figure 3.2 shows some samples of the dataset presenting significant differences between them. It is important to remark that the eight age range groups and the number of samples per age class are not equally distributed since some classes have more samples than the others.

Table 3.1: **Previous results on the Adience dataset.**

Reference	**Features**	**Classifier**	**Task**	**Accuracy** (%)
Levi [105]	Learned	CNN	Age	50.7
Chen [123]	Learned	CNN	Age	52.9
Rothe [122]	Learned	CNN	Age	55.6
Ozbulak [121]	Learned	CNN	Age	57.9
Rothe [122]	Learned	CNN	Age	64.0
Eidinger [107]	LBP+FPLBP	SVM	Gender	76.1
Tapia [99]	LBP	SVM	Gender	79.8
Levi [105]	Learned	CNN	Gender	86.8
Wolfshaar [130]	Learned	CNN	Gender	87.2
Ozbulak [121]	Learned	CNN	Gender	92.0

Table 3.2: **Mapping between Adience and IoG.** This Table shows the mapping between Adience and Images of Groups age categories to perform cross-dataset evaluation.

Adience [107]	0-2	4-6	8-13	15-20	25-32	{38-43, 48-53}	60+
IoG [109]	0-2	3-7	8-12	13-19	20-36	37-65	66+

In Table 3.1 the previous results published on the Adience dataset for age and gender estimation are listed. The results can be divided between the ones using LBP features (and variants) and the ones using a deep learning approach. As expected, CNNs yield significantly better results than SVMs on LBP features extracted from facial images.

To test the generalization capability of our models on age recognition, we have tested them using the Images of Groups (IoG) dataset presented in [109]. This dataset consists of $5.1K$ images of groups of people where $28.2K$ faces have been annotated with gender and age group labels. Like the images from the Adience dataset, the ones from IoG dataset present several differences in pose, appearance, and light, and they are even more challenging because the size of the faces is much smaller than Adience faces.

The 7 age groups from this dataset are quite similar to the 8 groups used in Adience dataset, so we can train models on the Adience dataset and then evaluate them on the IoG dataset: the mapping between Adience and IoG age categories is defined in Table 3.2.

Table 3.3: **Previous results on the Images of Groups Dataset.** [1], [2], [3]: Different data splits used by the authors.

Ref.	Features	Classifier	Task	Data Split	Acc. (%)
[141]	ML-LPQ	SVM	age	original	56.0
[142]	OHLG	SVM	age	custom1	59.5
[143]	LBP+SIFT+CH	SVM	age	Dago's [144]	63.0
[107]	LBP+FPLBP	SVM	age	Dago's [144]	66.6
[129]	BIF	SVM	age	5 fold	68.1
[145]	HOG+LBP+LTP+WLD	SVM	gender	Dago's [144]	92.5
[146]	ASR+	SRC	gender	custom3	93.3
[99]	LBP	SVM	gender	custom3	94.6
[120]	Local-DNN	MLP	gender	Dago's [144]	96.3
[147]	CNN+HOG+LBP+LOSIB	SVM+CNN	gender	Dago's [144]	97.2

Table 3.4: **Previous results on the MORPH II dataset** using the same data split. For the sake of clarity and fairness in the comparison, all these reported MAEs use the same data split and data subsets, only [119] provide a more complete evaluation procedure, using different data splits of 44K (MAE 3.31) and 55K (MAE 3.88).

Reference	**Method**	**Classifier**	**Task**	**MAE**
Chang [149]	AAM	OHRank	age	5.69
Wang [117]	CNN	DLA	age	4.77
Rothe [150]	CNN	SVR	age	3.45
Huerta [119]	CNN	MLP	age	3.31
Rothe [122]	CNN	DEX	age	2.68

In Table 3.3 there is a listing of all the previous results on the Images of groups dataset, indicating the methods used to tackle the problems of age or gender classification.

Deep Learning was also used in this dataset by Mansanet et al. [120] and Dong et al. [148], achieving good results in both age and gender recognition. In these deep learning approaches, similarly to our proposed network, the best features to perform age or gender recognition are not hand-crafted but learned from the data.

In order to evaluate the advantages of the proposed mechanism in a controlled environment (i.e. centered, unoccluded faces with common background), we test it on the MORPH II dataset, which consists of more than 50K mug shots. We follow a

well-known experimental setup in the literature [101, 122, 149, 151] consisting of a subset of 5474 pictures with ages comprised between 16 and 77 years old. From the subset, a 80% of them for training and a 20% for validation. The performance of previous approaches on the same data split of the MORPH II dataset are listed in Table 3.4. Performance is reported in Mean Average Error (MAE), the standard error measure for age regression in the literature:

$$MAE = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n},\ \{Y, \hat{Y}\} \in \mathbb{R} \tag{3.3}$$

where y_i is the ground truth value corresponding to the ith example, and $\hat{y}_i$ is the predicted value for that example.

3.5 Experiments and Results

Our model is based on the VGG-16 CNN [108], and it is implemented with Tensorflow [152]. We use domain-specific pre-training for initializing the CNN weights since it has been proved to achieve better performance than pre-training in general tasks such as Imagenet [121, 122]. Parameters are initialized with the standard VGG-16 architecture trained for face recognition on 2.6M images of 2.6K people [64] since, unlike [101], (i) it has been also tested for gender recognition, (ii) it uses the base VGG-16 model (without DEX), (iii) it focuses in a higher variety of facial analysis tasks than IMDB-WIKI, and (iv) it would deviate from the main purpose of this work, which is to evaluate the effects of attention mechanisms of CNNs for facial recognition tasks. The fully-connected layers are initialized with the Xavier initializer [153]. Models are optimized with `sgd` for 30 epochs, or until they reach a plateau. The learning rate is initially set to 0.0001 and divided by 10 every ten epochs. All the other hyper-parameters are found by random search unless we explicitly specify otherwise.

Next, we evaluate the influence of the different design decisions in the proposed model, namely the attention mode, weight sharing, merge mode, attention grid and patch depth. Following the same procedure as in the related works applied to this dataset, any possible design decision is firstly evaluated on a random fold to make the experimentation tractable, since results are proven consistent between folds.

Attention mode. We found that performing the Hadamard product between the attention weights and the patch feature maps was slightly better than the weighted sum ("Project." in Table 3.5). No weighting at all (no attention) resulted in the worst performance.

Weight sharing. As it can be seen in Table 3.5, using independent weights for the patch network yields small improvements in average. However, we observed the

Table 3.5: **Average validation error rate** when fixating different properties of the attention mechanism and random combinations of the rest, on a random fold of the Adience dataset. The proposed attention mechanism is proven robust to the weight sharing, different strategies of weighting the patches, and merging the feature maps from the attention and patch streams.

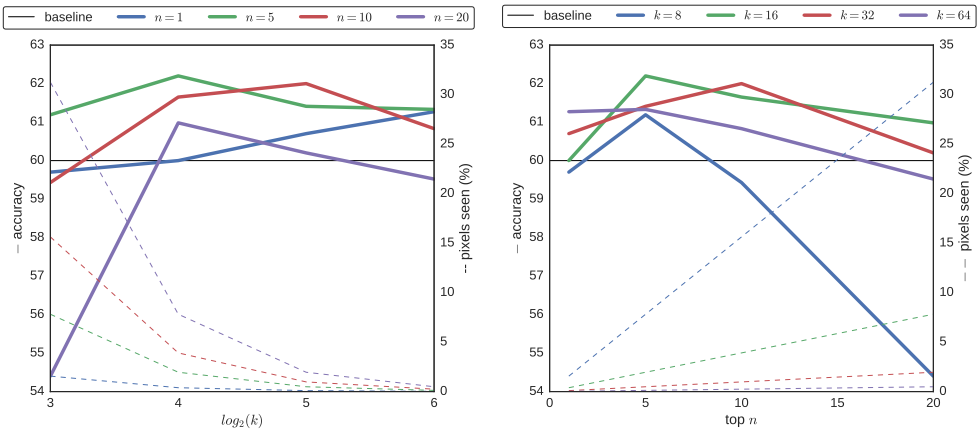
		Accuracy (%)
Baseline VGG-16 (no Attention)		57.80
VGG-16 + SVM [121]		57.90
Attention mode	No grid	59.41
	Project.	61.42
	Eltwise	61.78
Weight Sharing	No	61.37
	Yes	61.55
Merge mode	Add	61.35
	Concat.	61.78

opposite effect for big patch network inputs due to overfitting, *e.g.* $n = 20$, $k = 8$. Thus, since the proposed model is robust to changes in weight sharing, we decided to keep the weights shared to reduce the memory consumption of the network.

Merge mode. Compared to projecting the patch feature maps to the attention stream output space and adding them, feeding the classifier with the normed concatenation both streams resulted in the best performance by a small margin.

The Attention Grid. As it was shown in Figure 3.1, a $k \times k$ weight grid is predicted on the low-resolution input image. Since this grid is used to extract n patches, we can control the portion of the image that will be fed to the patch CNN by tuning n and k. Figure 3.3 shows the impact on performance of choosing different combinations of k, and n. As it can be seen, for most combinations, our approach outperforms the baseline score by a 2.4% margin with $n = 5$, and $k = 16$. Samples of the attention grid are shown in Figure 3.4, which corresponds to the predicted attention grid for $k = 4, n = 4$, and $k = 16, n = 5$.

Patch CNN Depth. Given that the convolutional layers of the attention path are reused, the depth of the patch CNN is conditioned to the five convolutional modules of the VGG-16. From this CNN, we compared features from the `pool3`, and `pool4` because they shrink the input size, and they are less invariant than features from `pool5`. We empirically found that using `pool3` yielded a 0.6% improvement over `pool4`. And this is consistent with the fact that lowest level features maps from the patch CNN are preferred since they better correspond to local image features.



(a) Fixate top-n

(b) Fixate grid k

Figure 3.3: **Maximum accuracy for different combinations of grid size** (k), and choosing n patches. Dashed lines represent the accumulated percentage of patch pixels with respect to the whole image. Dividing the image in 16 regions, and choosing the top-5 patches results in best performance by just processing a 2% of the high-resolution pixels.

Summarizing, including attention in CNN models is robust to any possible design configurations as presented in this paper, *i.e.* the attention grid application mode, the feature merging, and weight sharing. In fact, the most critical hyperparameters are the grid size k, and the number n of patches to feed to the patch network. For the next section, the parameters were fixed to `attention mode=eltwise, weight sharing=yes, merge mode=concat, k=16, n=5`, and the high-resolution images size is 600×600 before random cropping and random flip.

3.6 Evaluation on age and gender recognition

As it can be seen in Table 3.6, implementing the proposed attention mechanism on VGG-Faces [64] increases the accuracy in 4% on age recognition and 0.6% on gender recognition when not considering attention [121]. For the age classification problem, 1-off accuracy is also reported, indicating the accuracy of our model considering a one error distance prediction as correct. As expected, the 1-off accuracy of our model is 2.3% higher than the best-reported accuracy with VGG-16 pretrained

Figure 3.4: **The predicted attention grid.** The top row corresponds to a grid learned with $k = 4, n = 4$, and the bottom row to $k = 16, n = 5$. High attention is shown in red, and low attention in blue. As it can be seen, the attention grid predicts low values for background, glasses, and rings

Table 3.6: **Accuracies obtained on the Adience dataset for the 5 folds.** The VGG-16 model pre-trained on $> 3M$ faces [64] obtains the best performance when the attention mechanism is included.

	Accuracy (%)		
Model	**Age**	**1-off**	**Gender**
Eidinger [107]	45.1	80.7	77.8
Tapia [99]	-	-	79.8
Levi [105]	50.7	84.7	86.8
Wolfshaar [130]	-	-	87.2
Chen [123]	52.9	-	-
Rothe (VGG16-DEX) [122]	55.6	89.7	-
Ozbulak (VGG16-Faces + SVM) [121]	57.9	-	92.0
Rothe (VGG16-DEX-IMDB) [122]	**64.0**	**96.6**	-
Ours			
VGG16-Faces	57.8 ± 4.9	92.8 ± 1.8	92.4 ± 1.9
VGG16-Faces + Attention	61.8 ± 2.1	95.1 ± 0.03	**93.0** ± 1.8

with Faces.

Table 3.7: **Age confusion matrix for the Adience dataset.** This Table represents the confusion matrix of our model predictions over the whole dataset.

		Predicted							
		0-2	**4-6**	**8-12**	**15-20**	**25-32**	**38-43**	**48-53**	**+60**
Real	**0-2**	**64.2**	34.8	0.6	0.1	0.1	0.0	0.1	0.1
	4-6	15.8	**70.1**	12.2	1.3	0.3	0.1	0.1	0.1
	8-12	1.9	17.2	**59.1**	14.9	5.8	0.6	0.4	0.1
	15-20	0.1	1.0	12.8	**40.7**	41.1	4.2	0.2	0.1
	25-32	0.0	0.3	1.7	11.4	**69.2**	16.3	1.0	0.0
	38-43	0.0	0.1	0.5	2.1	39.8	**47.4**	8.0	2.2
	48-53	0.0	0.0	0.2	0.8	7.8	41.6	**31.1**	18.5
	+60	0.1	0.0	0.2	0.2	4.1	14.8	27.0	**53.6**

Table 3.8: **Accuracies obtained on the Images of Groups dataset.** This Table shows the accuracies obtained on the IoG dataset by averaging the predictions of the models trained on the Adience dataset. Our results are compared with the previous work where the same age balanced test set has been used, as proposed in [109].

	Accuracy (%)		
Model name	**Age**	**1-off**	**Gender**
Gallagher [109]	42.9	78.1	74.1
Li [154]	48.5	88.0	-
Shan [155]	50.3	87.1	74.9
Ylioinas [156]	51.7	88.7	-
Dong [148]	54.0	91.0	-
Bekhouche [141]	56.0	88.8	79.1
VGG-16 (Adience) + Attention	**60.0**	**94.5**	**86.9**

To the best of our knowledge, the top accuracy score obtained in gender classification Adience benchmark is 92.0% [121], and their results on age estimation are 57.9%. When our proposed method is trained to perform gender recognition it also achieves state-of-the-art performance on facial gender classification. In contrast, [122] do not apply their approach to gender analysis.

In order to evaluate how well the proposed attention model generalizes, a cross-dataset experiment was performed on IoG, see Table 3.8. The 5 models trained on the Adience dataset for gender recognition were used to classify the 1,050 test

Table 3.9: **MAE on the MORPH II dataset.** Adding attention to [122] decreases the MAE.

Reference	Method	Classifier	Task	MAE
Rothe [150]	CNN	SVR	age	3.45
Rothe [122]	CNN	DEX	age	2.68
VGG-16 [122] + Attention	CNN	DEX	age	**2.56**

Table 3.10: **Gender accuracy** per age group for the Adience and IoG datasets.

	0-2	4-6	8-12	15-20	25-32	38-43	48-53	+60
Acc.(%)	84.6	82.2	89.0	96.1	98.2	98.6	97.3	94.2

(a) Adience dataset.

	0-2	3-7	8-12	13-19	20-36	37-65	+66
Acc.(%)	67.3	82.0	82.7	91.3	96.0	98.0	90.6

(b) IoG dataset.

images from the IoG dataset and their predictions were averaged. As shown in Table 3.8, this ensemble obtained a gender classification accuracy of 86.9% on the IoG dataset, surpassing the state-of-the-art score from [141], which is 79.1% for this test split, thus confirming the generality of our approach. Table 3.9, shows the results on the MORPH II dataset. As it can be seen, adding attention results in 2.56 MAE, a relative 4.47% improvement with respect to the state of the art [122]. Additionally, in Table 3.10, it is shown that the proposed approach performs very well on adults, whereas it fails more frequently when classifying very young subjects. This performance is expected as even for humans estimating the gender of young children is harder than the gender of adults.

3.7 Discussion

A novel feedforward CNN pipeline which incorporates an attention mechanism for automatic age and gender recognition for face analysis has been proposed. The presented model consists of an attention network which estimates the most informative patches in the low-resolution image, which are further processed in a

Figure 3.5: **Gender misclassifications.** This figure shows several subjects whose gender have been misclassified. The first row contains females that were wrongly classified as males, whereas the second row contains males that were misclassified as females.

patch network in higher resolution. As a result, the attention-based CNN is proven to be more robust to clutter and deformation, inherent in deformable objects like faces. Alternative design choices for implementing the attention pipeline (i.e. attention mode, weight sharing, merge mode, attention grid, and patch network depth) have been proposed and compared, thus proving the robustness of the whole approach and consistently outperforming the model without attention.

Experiments show that networks enhanced with the proposed mechanism are more robust in in-the-wild tasks such as age and gender recognition in the Adience and IoG datasets. Concretely, enhanced models experienced a relative improvement of 8.75% for age recognition and a 7.89% on age classification with the Adience benchmark. The generality of the proposed model has also been demonstrated by performing a cross-dataset experiment, resulting in state-of-the-art performance on the IoG dataset. Moreover, experiments on MORPH II demonstrate that the proposed model enhances CNNs even in constrained environments with centered faces and gray backgrounds, resulting in a 4.47% relative improvement with respect to a state-of-the-art model [122]. An explanation for this effect is that the enhanced CNN has the ability to perform detailed fixations in the most discriminative patches depending on the context (for instance gender).

Qualitative results are shown in Figure 3.6, where images wrongly classified by VGG-16 (pre-trained on faces) are correctly classified by the proposed attention model. Also, it is shown how the attention mechanism is able to ignore clutter. Extreme rotations and occluding attributes (like fancy dressings) are the main reason of misclassifications, together with the presence of multiple people of different ages in the same image, or simply people who seem younger or older than their real age. This rises the interesting problem of apparent age estimation, as recently addressed in [157].

Figure 3.6: **Corrected miss-classifications with our attention mechanism.** Each row corresponds to an age group. Note that our approach is more robust to clutter.

For the case of gender recognition, the proposed model mostly fails with the youngest ages, difficult to be distinguished even by humans, see Fig. 3.5.

4 Attend and Rectify: a Gated Attention Mechanism for Fine-Grained Recovery

We propose a novel attention mechanism to enhance Convolutional Neural Networks for fine-grained recognition. It learns to attend to lower-level feature activations without requiring part annotations and uses these activations to update and rectify the output likelihood distribution. In contrast to other approaches, the proposed mechanism is modular, architecture-independent and efficient both in terms of parameters and computation required. Experiments show that networks augmented with our approach systematically improve their classification accuracy and become more robust to clutter. As a result, Wide Residual Networks augmented with our proposal surpasses the state of the art classification accuracies in CIFAR-10, the Adience gender recognition task, Stanford dogs, and UEC Food-100.

4.1 Motivation

Humans and animals process vasts amounts of information with limited computational resources thanks to attention mechanisms which allow them to focus resources on the most informative chunks of information [158–160]

This work is inspired by the advantages of visual and biological attention mechanisms, for tackling fine-grained visual recognition with Convolutional Neural Networks (CNN) [2]. This is a particularly difficult task since it involves looking for details in large amounts of data (images) while remaining robust to deformation and clutter. In this sense, different attention mechanisms for fine-grained recognition exist in the literature: (i) iterative methods that process images using "glimpses" with recurrent neural networks (RNN) or long short-term memory (LSTM) [161, 162], (ii) feed-forward attention mechanisms that augment vanilla CNNs, such as the Spatial Transformer Networks (STN) [17], or top-down feed-forward attention mechanisms (FAM) [22]. Although it is not applied to fine-grained recognition, the Residual Attention introduced by [18] is another example of feed-forward attention mechanism that takes advantage of residual connections [163] to enhance or dampen

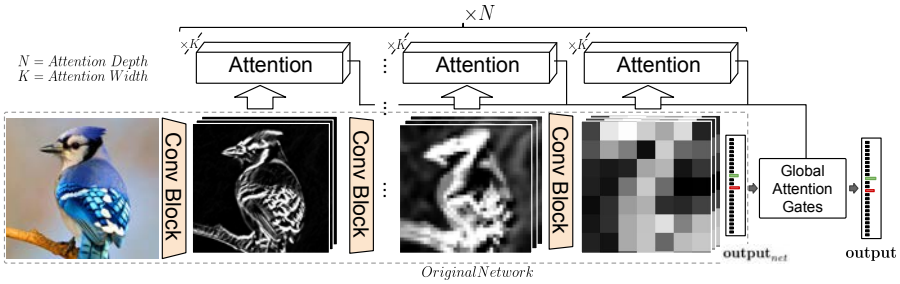


Figure 4.1: **The proposed mechanism.** The original CNN is augmented with N attention modules at N different depths. Each attention module applies K attention heads to the network feature maps to make a class prediction based on local information. The original network **output**$_{net}$ is then corrected based on the local features by means of the global attention gates, resulting in the final **output**.

certain regions of the feature maps in an incremental manner.

Thus, most of the existing attention mechanisms are either limited by having to perform multiple passes through the data [161], by carefully designed architectures that should be trained from scratch [17], or by considerably increasing the needed amount of memory and computation, thus introducing computational bottlenecks [164]. Hence, there is still the need of models with the following learning properties: (i) Detect and process in detail the most informative parts of an image for learning models more robust to deformation and clutter [106]; (ii) feed-forward trainable with SGD for achieving faster inference than iterative models [161, 162], together with faster convergence rate than Reinforcement Learning-based (RL) methods [161, 165]; (iii) preserve low-level detail for a direct access to local low-level features before they are modified by residual identity mappings. This is important for fine-grained recognition, where low-level patterns such as textures can help to distinguish two similar classes. This is not fulfilled by Residual Attention, where low-level features are subject to noise after traversing multiple residual connections [18].

In addition, desirable properties for attention mechanisms applied to CNNs would be: (i) **Modular and incremental**, since the same structure can be applied at each layer on any convolutional architecture, and it is easy to adapt to the task at hand; (ii) **Architecture independent**, that is, being able to adapt any pre-trained architecture such as VGG [166] or ResNet [163]; (iii) **Low computational impact** implying that it does not result in a significant increase in memory and computation; and (iv) **Simple** in the sense that it can be implemented in few lines of code, making

it appealing to be used in future work.

Based on all these properties, we propose a novel attention mechanism that learns to attend low-level features from a standard CNN architecture through a set of replicable Attention Modules and gating mechanisms (see Section 4.3). Concretely, as it can be seen in Figure 4.1, any existing architecture can be augmented by applying the proposed model at different depths, and replacing the original loss by the proposed one. It is remarkable that the modules are independent of the original path of the network, so in practice, it can be computed in parallel to the rest of the network. The proposed attention mechanism has been included in a strong baseline like Wide Residual Networks (WRN) [167], and applied on CIFAR-10, CIFAR-100 [168], and five challenging fine-grained recognition datasets. The resulting network, called Wide Attentional Residual Network (WARN) systematically enhances the performance of WRNs and surpasses the state of the art in various classification benchmarks.

4.2 Related Work

There are different approaches to fine-grained recognition [9]: (i) vanilla deep CNNs, (ii) CNNs as feature extractors for localizing parts and do alignment, (iii) ensembles, (iv) attention mechanisms. In this work, we focus on (iv), the attention mechanisms, which aim to discover the most discriminative parts of an image to be processed in greater detail, thus ignoring clutter and focusing on the most distinctive traits. These parts are central for fine-grained recognition, where the inter-class variance is small and the intra-class variance is high.

Different fine-grained attention mechanisms can be found in the literature. [15] proposed a *two-level attention* mechanism for fine-grained classification on different subsets of the ILSVRC [169] dataset, and the CUB200_2011. In this model, images are first processed by a bottom-up object proposal network based on R-CNN [14] and selective search [170]. Then, the softmax scores of another ILSVRC2012 pre-trained CNN, which they call *FilterNet*, are thresholded to prune the patches with the lowest parent class score. These patches are then classified to fine-grained categories with a *DomainNet*. Spectral clustering is also used on the *DomainNet* filters in order to extract parts (head, neck, body, etc.), which are classified with an SVM. Finally, the part- and object-based classifier scores are merged to get the final prediction. The *two-level attention* obtained state of the art results on CUB200-2011 with only class-level supervision. However, the pipeline must be carefully fine-tuned since many stages are involved with many hyper-parameters.

Differently from *two-level attention*, which consists of independent processing and it is not end-to-end, Sermanet *et al.* proposed to use a deep CNN and a Re-

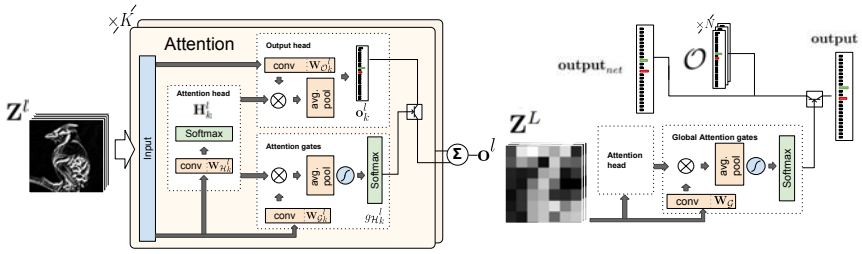
current Neural Network (RNN) to accumulate high multi-resolution "glimpses" of an image to make a final prediction [161]. However, reinforcement learning slows down convergence and the RNN adds extra computation steps and parameters.

A more efficient approach was presented by Liu *et al.* [165], where a fully-convolutional network is trained with reinforcement learning to generate confidence maps on the image and use them to extract the parts for the final classifiers whose scores are averaged. Compared to previous approaches, in the work done by [165], multiple image regions are proposed in a single timestep thus, speeding up the computation. A greedy reward strategy is also proposed in order to increase the training speed. The recent approach presented by [16] uses a classification network and a recurrent attention proposal network that iteratively refines the center and scale of the input (RA-CNN). A ranking loss is used to enforce incremental performance at each iteration.

Zhao *et al.* proposed to enforce multiple non-overlapped attention regions [162]. The overall architecture consists of an attention canvas generator, which extracts patches of different regions and scales from the original image; a VGG-16 [166] CNN is then used to extract features from the patches, which are aggregated with a long short-term memory [171] that attends to non-overlapping regions of the patches. Classification is performed with the average prediction in each region. Similarly, in [172], they proposed the Multi-Attention CNN (MA-CNN) to learn to localize informative patches from the output of a VGG-19 and use them to train an ensemble of part classifiers.

In [164], they propose to extract global features from the last layers of a CNN, just before the classifier and use them to attend relevant regions in lower level feature activations. The attended activations from each level are then spatially averaged, channel-wise concatenated, and fed to the final classifier. The main differences with [164] are: (i) attention maps are computed in parallel to the base model, while the model in [164] requires output features for computing attention maps; (ii) WARN uses fewer parameters, so dropout is not needed to obtain competitive performance (these two factors clearly reflect in gain of speed); and (iii) gates allow our model to ignore/attend different information to improve the performance of the original model, while in [164] the full output function is replaced. As a result, WARN obtains 3.44% error on CIFAR10, outperforming [164] while being 7 times faster w/o parallelization.

All the previously described methods involve multi-stage pipelines and most of them are trained using reinforcement learning (which requires sampling and makes them slow to train). In contrast, STNs, FAM, the model in [164], and our approach jointly propose the attention regions and classify them in a single pass. Moreover, different from STNs and FAM our approach only uses one CNN stream, it can be used on pre-trained models, and it is far more computationally efficient than STNs,



(a) Attention module

(b) Attention gates

Figure 4.2: **Depiction of the attention modules and heads.** (a) Attention Module: K attention heads $\mathbf{H}_k^l$ are applied to a feature map $\mathbf{Z}^l$, and information is aggregated with the layer attention gates. (b) Global attention: global information from the last feature map $\mathbf{Z}^L$ is used to compute the gating scores that produce the final **output** as the weighted average of the outputs of the attention modules and the original network **output**$_{net}$

FAM, and [164] as described next.

4.3 Proposed Approach

Our approach consists of a universal attention module that can be added after each convolutional layer without altering pre-defined information pathways of any architecture (see Figure 4.1). This is helpful since it seamlessly augments any architecture such as VGG and ResNet with no extra supervision, *i.e.* no part labels are necessary. Furthermore, it also allows being plugged into any existing trained network to quickly perform transfer learning approaches.

The attention module consists of three main submodules depicted in Figure 4.2 (a): (i) the attention heads $\mathcal{H}$, which define the most relevant regions of a feature map, (ii) the output heads $\mathcal{O}$, generate an hypothesis given the attended information, and (iii) the confidence gates $\mathcal{G}$, which output a confidence score for each attention head. Each of these modules is described in detail in the following subsections.

4.3.1 Overview

As it can be seen in Figure 4.1, a convolution layer is applied to the output of the augmented layer, producing K attentional heatmaps. These attentional maps are then used to spatially average the local class probability scores for each of the

feature maps, and produce the final class probability vector. This process is applied to an arbitrary number N of layers, producing N class probability vectors. Then, the model learns to correct the initial prediction by attending the lower-level class predictions. This is the final combined prediction of the network. In terms of probability, the network corrects the initial likelihood by updating the prior with local information.

4.3.2 Attention head

Inspired by [162] and the *transformer* architecture presented by [173], and following the notation established by [167], we have identified two main dimensions to define attentional mechanisms: (i) the number of layers using the attention mechanism, which we call *attention depth* (AD), and (ii) the number of attention heads in each attention module, which we call *attention width* (AW). Thus, a desirable property for any universal attention mechanism is to be able to be deployed at any arbitrary *depth* and *width*.

This property is fulfilled by including K attention heads $\mathcal{H}_k$ (width), depicted in Figure 4.1, into each attention module (depth)[1]. Then, the attention heads at layer $l \in [1..L]$, receive the feature activations $\mathbf{Z}^l \in \mathbb{R}^{c \times h \times w}$ of that layer as input, and output K attention masks:

$$\mathbf{H}^l = spatial\_softmax(\mathbf{W}_{\mathcal{H}}{}^l * \mathbf{Z}^l), \tag{4.1}$$

where $\mathbf{H}^l \in \mathbb{R}^{K \times h \times w}$ is the output matrix of the l^{th} attention module, $\mathbf{W}_{\mathcal{H}}{}^l : \mathbb{R}^{c \times h \times w} \rightarrow \mathbb{R}^{K \times h \times w}$ is a convolution kernel with output dimensionality K used to compute the attention masks corresponding to the attention heads $\mathbf{H}_k$, and $*$ denotes the convolution operator. The *spatial_softmax*, which performs the *softmax* operation channel-wise on the spatial dimensions of the input, is used to enforce the model to learn the most relevant region of the image. Sigmoid units could also be used at the risk of degeneration to all-zeros or all-ones. To prevent the attention heads at the same depth to collapse into the same region, we apply the regularizer proposed in [162].

4.3.3 Output head

To obtain the class probability scores, the input feature map $\mathbf{Z}_k^l$ is convolved with a kernel:

[1] Notation: $\mathcal{H}, \mathcal{O}, \mathcal{G}$ are the set of attention heads, output heads, and attention gates respectively. Uppercase letters refer to functions or constants, and lowercase ones to indices. Bold uppercase letters represent matrices and bold lowercase ones vectors.

$$\mathbf{W}_{\mathcal{O}}{}_k^l \in \mathbb{R}^{channels \times h \times w} \rightarrow \mathbb{R}^{\#labels \times h \times w},$$

h, w represent the spatial dimensions, and $channels$ is the number of input channels to the module. This results on a spatial map of class probability scores:

$$\mathbf{O}_k^l = \mathbf{W}_{\mathcal{O}}{}_k^l * \mathbf{Z}^l. \tag{4.2}$$

Note that this operation can be done in a single pass for all the K heads by setting the number of output channels to $\#labels \cdot K$. Then, class probability vectors $\mathbf{O}_k^l$ are weighted by the spatial attention scores and spatially averaged:

$$\mathbf{o}_k^l = \sum_{x,y} \mathbf{H}_k^l \odot \mathbf{O}_k^l, \tag{4.3}$$

where $\odot$ is the element-wise product, and $x \in \{1..width\}$, $y \in \{1..height\}$. The attention scores $\mathbf{H}_k^l$ are a 2d flat mask and the product with each of the input channels of $\mathbf{Z^l}$ is done by broadcasting, *i.e.* repeating $\mathbf{H}_k^l$ for each of the channels of $\mathbf{Z^l}$.

4.3.4 Layered attention gates

The final output $\mathbf{o}^l$ of an attention module is obtained by a weighted average of the K output probability vectors, through the use of head attention gates $\mathbf{g}_{\mathcal{H}}{}^l \in \mathbb{R}^{|\mathcal{H}|}$, $\sum_k g_{\mathcal{H}}{}_k^l = 1$.

$$\mathbf{o}^l = \sum_k g_{\mathcal{H}}{}_k^l o_k^l. \tag{4.4}$$

Where $g_{\mathcal{H}}$ is obtained by first convolving $\mathbf{Z}^l$ with

$$\mathbf{W}_{\mathbf{g}}{}^l \in \mathbb{R}^{channels \times h \times w} \rightarrow \mathbb{R}^{|\mathcal{H}| \times h \times w},$$

and then performing a spatial weighted average:

$$\mathbf{g}_{\mathcal{H}}{}^l = softmax(tanh(\sum_{x,y}(\mathbf{W}_{\mathbf{g}}{}^l * \mathbf{Z}^l) \odot \mathbf{H}_l)). \tag{4.5}$$

This way, the model learns to choose the attention head that provides the most meaningful output for a given attention module.

4.3.5 Global attention gates

In order to let the model learn to choose the most discriminative features at each depth to disambiguate the output prediction, a set of relevance scores **c** are predicted at the model output, one for each attention module, and one for the final prediction. This way, through a series of gates, the model can learn to query information from each level of the network conditioned to the global context. Note that, unlike in [164], the final predictions do not act as a bottleneck to compute the output of the attention modules.

The relevance scores are obtained with an inner product between the last feature activation of the network $\mathbf{Z}^L$ and the gate weight matrix $\mathbf{W}_\mathcal{G}$:

$$\mathbf{c} = tanh(\mathbf{W}_\mathcal{G}\mathbf{Z}^L). \tag{4.6}$$

The gate values $\mathbf{g}_\mathcal{O}$ are then obtained by normalizing the set of scores by means of a $softmax$ function:

$$g_\mathcal{O}{}_k^l = \frac{e^{c_k^l}}{\sum_{i=1}^{|\mathcal{G}|} e^{c_i}}, \tag{4.7}$$

where $|\mathcal{G}|$ is the total number of gates, and c_i is the i^{th} confidence score from the set of all confidence scores. The final output of the network is the weighted sum of the attention modules:

$$\mathbf{output} = g_{net} \cdot \mathbf{output}_{net} + \sum_{l \in \{1..|\mathcal{O}|\}} g_\mathcal{O}^l \cdot \mathbf{o}^l, \tag{4.8}$$

where g_{net} is the gate value for the original network output ($\mathbf{output_{net}}$), and **output** is the final output taking the attentional predictions $\mathbf{o}^l$ into consideration. Note that setting the output of $\mathcal{G}$ to $\frac{1}{|\mathcal{O}|}$, corresponds to averaging all the outputs. Likewise, setting $\{\mathcal{G} \setminus G_{output}\} = 0, G_{output} = 1$, *i.e.* the set of attention gates is set to zero and the output gate to one, corresponds to the original pre-trained model without attention.

It is worth noting that all the operations that use $\mathbf{Z}^l$ can be aggregated into a single convolution operation. Likewise, the fact that the attention mask is generated by just one convolution operation, and that most masking operations are directly performed in the label space, or can be projected into a smaller dimensionality space, makes the implementation highly efficient. Additionally, the direct access to the output gradients makes the module fast to learn, thus being able to generate foreground masks from the beginning of the training and refining them during the following epochs.

(a) (b) (c) (d) (e)

Figure 4.3: **Samples from the five fine-grained datasets.** (a) Adience, (b) CUB200 Birds, (c) Stanford Cars, (d) Stanford Dogs, (e) UEC-Food100

4.4 Experiments and Results

We empirically demonstrate the impact on the accuracy and robustness of the different modules in our model on Cluttered Translated MNIST and then compare it with state-of-the-art models such as DenseNets and ResNeXt. Finally, we demonstrate the universality of our method for fine-grained recognition through a set of experiments on five fine-grained recognition datasets, as detailed next.

4.4.1 Datasets

Cluttered Translated MNIST[2] Consists of 40×40 images containing a randomly placed MNIST [174] digit and a set of D randomly placed distractors, see Figure 4.5b. The distractors are random 8×8 patches from other MNIST digits.

CIFAR[3] The CIFAR dataset consists of 60K 32x32 images in 10 classes for CIFAR-10, and 100 for CIFAR-100. There are 50K training and 10K test images.

Stanford Dogs [175]. The Stanford Dogs dataset consists of 20.5K images of 120 breeds of dogs, see Figure 4.3d. The dataset splits are fixed and they consist of 12k training images and 8.5K validation images.

UEC Food 100 [176]. A Japanese food dataset with 14K images of 100 different dishes, see Figure 4.3e. In order to follow the standard procedure (*e.g.* [177, 178]), bounding boxes are used to crop the images before training.

Adience dataset [107]. The adience dataset consists of 26.5 K images distributed in eight age categories (0–2, 4–6, 8–13, 15–20, 25–32, 38–43, 48–53, 60+), and gender labels. A sample is shown in Figure 4.3a. The performance on this dataset is measured using 5-fold cross-validation.

[2] https://github.com/deepmind/mnist-cluttered
[3] https://www.cs.toronto.edu/~kriz/cifar.html

Stanford Cars [179]. The Cars dataset contains 16K images of 196 classes of cars, see Figure 4.3c. The data is split into 8K training and 8K testing images.

Caltech-UCSD Birds 200 [180]. The CUB200-2011 birds dataset (see Figure 4.3b) consists of 6K train and 5.8K test bird images distributed in 200 categories. Although bounding box, segmentation, and attributes are provided, we perform raw classification as done by [17].

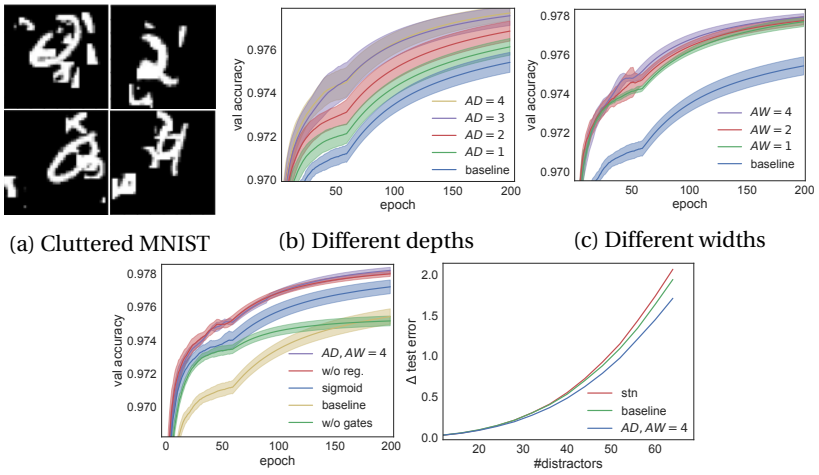
4.4.2 Ablation study

We evaluate the submodules of our method on the Cluttered Translated MNIST following the same procedure as in [106]. The proposed attention mechanism is used to augment a CNN with five 3×3 convolutional layers and two fully-connected layers in the end. The three first convolution layers are followed by Batch-normalization and a spatial pooling. Attention modules are placed starting from the fifth convolution (or pooling instead) backward until AD is reached. Training is performed with SGD for 200 epochs, and a learning rate of 0.1, which is divided by 10 after epoch 60. Models are trained on a $200k$ images train set, validated on a $100k$ images validation set, and tested on $100k$ test images. Weights are initialized using He *et al.* [181]. Figure 4.4 shows the effects of the different hyperparameters of the proposed model. The performance without attention is labeled as `baseline`. Attention models are trained with softmax attention gates and regularized with [162], unless explicitly specified.

First, we test the importance of AD for our model by increasingly adding attention layers with $AW = 1$ after each pooling layer. As it can be seen in Figure 4.4b, greater AD results in better accuracy, reaching saturation at $AD = 4$, note that for this value the receptive field of the attention module is 5×5 px, and thus the performance improvement from such small regions is limited. Figure 4.4c shows training curves for different values of AW, and $AD = 4$. As it can be seen, small performance increments are obtained by increasing the number of attention heads even with a single object present in the image.

Then, we use the best AD and AW, *i.e.* $AD, AW = 4$, to verify the importance of using softmax on the attention masks instead of sigmoid (4.1), the effect of using gates (Eq. 4.7), and the benefits of regularization [162]. Figure 4.4d confirms that ordered by importance: gates, softmax, and regularization result in accuracy improvement, reaching 97.8%. In particular, gates play an important role in discarding the distractors, especially for high AW and high AD

Finally, in order to verify that attention masks are not overfitting on the data, and thus generalize to any amount of clutter, we run our best model so far (Figure 4.4d) on the test set with an increasing number of distractors (from 4 to 64). For the



(a) Cluttered MNIST (b) Different depths (c) Different widths

(d) Softmax, gates, reg. (e) Overfitting

Figure 4.4: **Ablation experiments on Cluttered Translated MNIST.** `baseline` indicates the original model before being augmented with attention. (a) shows a sample of the cluttered MNIST dataset. (b) the effect of increasing the attention depth (AD), for attention width $AW = 1$. (c) effect of increasing AW, for AD=4. (d) best performing model ($AD, AW = 4$, softmax attention gates, and regularization [162]) vs unregularized, sigmoid attention, and without gates. (e) test error of the baseline, attention ($AD, AW = 4$), and spatial transformer networks (stn), when trained with different amounts of distractors.

.

comparison, we included the baseline model before applying our approach and the same baseline augmented with an STN [17] that reached comparable performance as our best model in the validation set. All three models were trained with the same dataset with eight distractors. Remarkably, as it can be seen in Figure 4.4e, the attention augmented model demonstrates better generalization than the baseline and the STN.

4.4.3 Training from scratch

We benchmark the proposed attention mechanism on CIFAR-10 and CIFAR-100, and compare it with the state of the art. As a base model, we choose Wide Residual Networks, a strong baseline with a large number of parameters so that the additional parameters introduced by our model (WARN) could be considered negligible.

The same WRN baseline is used to train an `att2` model [164], we refer to this model as WRN-att2. Models are initialized and optimized following the same procedure as in [167]. Attention Modules are systematically placed after each of the three convolutional groups starting by the last one until the attention depth has been reached in order to capture information at different levels of abstraction and fine-grained resolution, this same procedure is followed in [164]. The model is implemented with pytorch [182] and run on a single workstation with two NVIDIA 1080Ti.[4]

First, the same ablation study performed in Section 4.4.2 is repeated on CIFAR100. We consistently reached the same conclusions as in Cluttered-MNIST: accuracy improves 1.5% by increasing attention depth from 1 to #residual_blocks, and width from 1 to 4. Gating performs 4% better than a simpler linear projection, and 3% with respect to simply averaging the output vectors. A 0.6% improvement is also observed when regularization is activated. Interestingly, we found sigmoid attention to perform similarly to softmax. With this setting, WARN reaches 17.82% error on CIFAR100. In addition, we perform an experiment blocking the gradients from the proposed attention modules to the original network to analyze whether the observed improvement is due to the attention mechanism or an optimization effect due to introducing shortcut paths to the loss function [183]. Interestingly, we observed a 0.2% drop on CIFAR10, and 0.4% on CIFAR100, which are still better than the baseline. Note that a performance drop should be expected, even without taking optimization into account, since backpropagation makes intermediate layers learn to gather more discriminative features for the attention layers. It is also worth noting that fine-grained accuracy improves even when fine-tuning (gradients are multiplied by 0.1 in the base model), see Section 4.4.4. In contrast, the approach in [164] does not converge when gradients are not sent to the base model since classification is directly performed on intermediate feature maps (which continuously shift during training).

As seen in Table 4.1, the proposed Wide Attentional Residual Network (WARN) improves the baseline model for CIFAR-10 and CIFAR-100 even without the use of Dropout and outperforms the rest of the state of the art in CIFAR-10 while being remarkably faster, as it can be seen in Table 4.2. Remarkably, the performance on CIFAR-100 makes WARN competitive when compared with Densenet and Resnext, while being up to 36 times faster. We hypothesize that the increase in accuracy of the augmented model is limited by the base network and even better results could be obtained when applied on the best performing baseline.

Interestingly, WARN shows superior performance even without the use of dropout; this was not possible with [164], which requires dropout to achieve competitive performances, since they introduce more parameters to the augmented network. The

[4] https://github.com/prlz77/attend-and-rectify

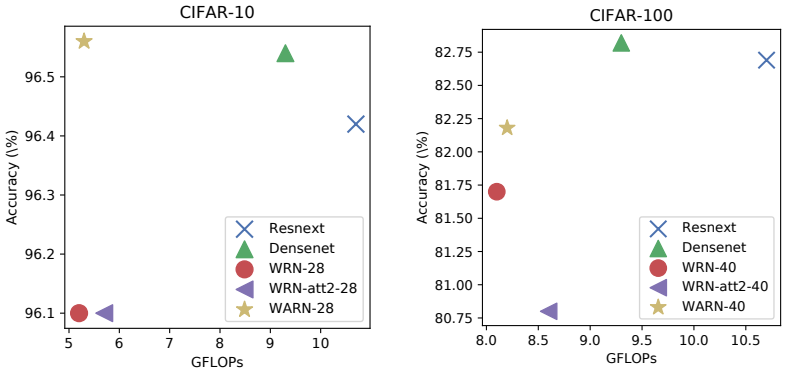
Table 4.1: **Error rate on CIFAR-10 and CIFAR-100 (%).** Results that surpass all other methods are in blue, results that surpass the baseline are in black bold font. Total network depth, attention depth, attention width, the usage of dropout, and the amount of floating point operations (Flop) are provided in columns 1-5 for fair comparison

	Depth	AD	AW	Dropout	GFlop	CIFAR-10	CIFAR-100
Resnext [184]	29	-	-		10.7	3.58	17.31
Densenet [185]	250	-	-		5.4	3.62	17.60
	190	-	-		9.3	3.46	17.18
WRN [167]	28	-	-		5.2	4	19.25
	28	-	-	✓	5.2	3.89	18.85
	40	-	-	✓	8.1	3.8	18.3
WRN-att2 [164]	28	2	-		5.7	4.10	21.20
	28	2	-	✓	5.7	**3.60**	20.00
	40	2	-	✓	8.6	3.90	19.20
WARN	28	2	4		5.2	**3.60**	18.72
	28	3	4		5.3	3.45	18.61
	28	3	4	✓	5.3	3.44	18.26
	40	3	4	✓	8.2	**3.46**	**17.82**

Table 4.2: **Number of parameters, floating point operations (Flop), time (s) per validation epoch, and error rates** (%) on CIFAR-10 and CIFAR-100. The "Time" column shows the amount of seconds to forward the validation dataset with batch size 256 on a single GPU

	Depth	Params	GFlop	Time	CIFAR-10	CIFAR-100
ResNext	29	68M	10.7	5.02s	3.58	17.31
Densenet	190	**26M**	9.3	6.41s	3.46	**17.18**
WRN	40	56M	8.1	0.18s	3.80	18.30
WRN-att2	40	64M	8.6	0.24s	3.90	19.20
WARN	28	37M	**5.3**	**0.17s**	**3.44**	18.26
WARN	40	56M	8.2	0.18s	3.46	17.82

computing efficiency of the top performing models is shown in Figure 4.5. WARN provides the highest accuracy per GFlop on CIFAR-10, and is more competitive than



(a) CIFAR-10 (b) CIFAR-100

Figure 4.5: **Performance comparison on CIFAR.** The best performing Resnext, Densenet, WRN, WRN-att2, and WARN are compared on the CIFAR-10 and CIFAR-100. Validation accuracy is reported as a function of the number of GFLOPs.

WRN, and WRN-att2 on CIFAR-100.

4.4.4 Transfer Learning

We fine-tune an augmented WRN-50-4 pre-trained on Imagenet [169] and report higher accuracy on five different fine-grained datasets: Stanford Dogs, UEC Food-100, Adience, Stanford Cars, CUB200-2001 compared to the WRN baseline. All the experiments are trained for 100 epochs, with a batch size of 64. The learning rate is first set to 10^{-3} to all layers except the attention modules and the classifier, for which it ten times higher. The learning rate is reduced by a factor of 0.1 every 30 iterations and the experiment is automatically stopped if a plateau is reached. The network is trained with standard data augmentation, *i.e.* random 224×224 patches are extracted from 256×256 images with random horizontal flips. Since the aim of this work is to demonstrate that the proposed mechanism universally improves the baseline CNNs for fine-grained recognition, we follow the same training procedure in all datasets. Thus, we do not use 512×512 images, which are central for state-of-the-art methods such as RA-CNNs, MA-CNNs, or color jitter [178] for food recognition. The proposed method is able to obtain state of the art results in Adience Gender, Stanford dogs and UEC Food-100 even when trained with lower resolution.

As seen in table 4.3, WRN substantially increase their accuracy on all bench-

Table 4.3: **Results on six fine-grained recognition tasks**. *DSP* means that the cited model uses Domain Specific Pre-training. *HR* means the cited model uses high-resolution images. Accuracies that improve the baseline model are in black bold font, and highest accuracies are in blue

	Dogs	**Food**	**Cars**	**Gender**	**Age**	**Birds**
SotA	RA-CNN [16]	Inception [178]	MA-CNN [172]	FAM [22]	DEX [186]	MA-CNN [172]
DSP				✓	✓	
HR	✓		✓			✓
Acc.	87.3	81.5	92.8	93.0	64.0	86.5
WRN	89.6	84.3	88.5	93.9	57.4	84.3
WARN	92.9	85.5	**90.0**	94.6	**59.7**	**85.6**

Table 4.4: Increment of accuracy (%) per Million of parameters

	Dogs	**Food**	**Cars**	**Gender**	**Age**	**Birds**	**Average**
WRN	1.3	1.2	1.3	1.4	0.8	1.2	1.2
WARN	**6.9**	**2.5**	**3.1**	**1.5**	**4.0**	**2.5**	**3.4**

marks by just fine-tuning them with the proposed attention mechanism. Moreover, we report the highest accuracy scores on Stanford Dogs, UEC Food, and Gender recognition, and obtain competitive scores when compared with models that use high resolution images, or domain-specific pre-training. For instance, in [186] a domain-specific model pre-trained on millions of faces is used for age recognition, while our baseline is a general-purpose WRN pre-trained on the Imagenet. It is also worth noting that the performance increase on CUB200-2011 (+1.3%) is higher than the one obtained in STNs with 224×224 images (+0.8%) even though we are augmenting a stronger baseline. This points out that the proposed mechanism might be extracting complementary information that is not extracted by the main convolutional stream. As seen in table 4.4, WARN not only increases the absolute accuracy, but it provides a high efficiency per introduced parameter. A sample of the attention masks for each dataset is shown on Figure 4.6. As it can be seen, the attention heads learn to ignore the background and to attend the most discriminative parts of the objects. This matches the conclusions of Section 4.4.2.

Figure 4.6: **Attention masks for each dataset**: (a) Stanford dogs, (b) Stanford cars, (c) Adience gender, (d) CUB birds, (e) Adience age, (f) UEC food. As it can be seen, the masks help to focus on the foreground object. In (c), the attention mask focuses on ears for gender recognition, possibly looking for earrings

4.5 Discussion

We have presented a novel attention mechanism to improve CNNs. The proposed model learns to attend the most informative parts of the CNN feature maps at different depth levels and combines them with a gating function to update the output distribution.

We suggest that attention helps to discard noisy uninformative regions, avoiding the network to memorize them. Unlike previous work, the proposed mechanism is modular, architecture independent, fast, simple, and yet WRN augmented with it obtain state-of-the-art results on highly competitive datasets while being 37 times faster than DenseNet, 30 times faster than ResNeXt, and making the augmented model more parameter-efficient. When fine-tuning on a transfer learning task, the attention augmented model showed superior performance in each recognition dataset. Moreover, state of the art performance is obtained on dogs, gender, and food. Results indicate that the model learns to extract local discriminative information that is otherwise lost when traversing the layers of the baseline architecture.

5 Regularizing CNNs with Locally Constrained Decorrelations

Regularization is key for deep learning since it allows training more complex models while keeping lower levels of overfitting. However, the most prevalent regularizations do not leverage all the capacity of the models since they rely on reducing the effective number of parameters. Feature decorrelation is an alternative for using the full capacity of the models but the overfitting reduction margins are too narrow given the overhead it introduces. In this thesis, we show that regularizing negatively correlated features is an obstacle for effective decorrelation and present OrthoReg, a novel regularization technique that locally enforces feature orthogonality. As a result, imposing locality constraints in feature decorrelation removes interferences between negatively correlated feature weights, allowing the regularizer to reach higher decorrelation bounds, and reducing the overfitting more effectively. In particular, we show that the models regularized with OrthoReg have higher accuracy bounds even when batch normalization and dropout are present. Moreover, since our regularization is directly performed on the weights, it is especially suitable for fully convolutional neural networks, where the weight space is constant compared to the feature map space. As a result, we are able to reduce the overfitting of state-of-the-art CNNs on CIFAR-10, CIFAR-100, and SVHN.

5.1 Motivation

Neural networks perform really well in numerous tasks even when initialized randomly and trained with Stochastic Gradient Descent (SGD) (see [59]). Deeper models, like Googlenet ([187]) and Deep Residual Networks ([187, 188]) are released each year, providing impressive results and even surpassing human performances in well-known datasets such as the Imagenet ([169]). This would not have been possible without the help of regularization and initialization techniques which solve the overfitting and convergence problems that are usually caused by data scarcity and the growth of the architectures.

From the literature, two different regularization strategies can be defined. The first ones consist in reducing the complexity of the model by (i) reducing the effective number of parameters with weight decay ([189]), and (ii) randomly dropping activations with Dropout ([190]) or dropping weights with DropConnect ([191]) so as to prevent feature co-adaptation. Due to their nature, although this set of strategies have proved to be very effective, they do not leverage all the capacity of the models they regularize.

The second group of regularizations is those which improve the effectiveness and generality of the trained model without reducing its capacity. In this second group, the most relevant approaches decorrelate the weights or feature maps, e.g. [192] introduced a new criterion so as to learn slow decorrelated features while pre-training models. In the same line [193] presented "incoherent training", a regularizer for reducing the decorrelation of the network activations or feature maps in the context of speech recognition. Although regularizations in the second group are promising and have already been used to reduce the overfitting in different tasks, even with the presence of Dropout (as shown by [194]), they are seldom used in the large scale image recognition domain because of the small improvement margins they provide together with the computational overhead they introduce.

Although they are not directly presented as regularizers, there are other strategies to reduce the overfitting such as Batch Normalization ([195]), which decreases the overfitting by reducing the internal covariance shift. In the same line, initialization strategies such as "Xavier" ([153]) or "He" ([181]), also keep the same variance at both input and output of the layers in order to preserve propagated signals in deep neural networks. Orthogonal initialization techniques are another family which set the weights in a decorrelated initial state so as to condition the network training to converge into better representations. For instance, [196] propose to initialize the network with decorrelated features using orthonormal initialization ([197]) while normalizing the variance of the outputs as well.

In this work we hypothesize that regularizing negatively correlated features is an obstacle for achieving better results and we introduce OrhoReg, a novel regularization technique that addresses the performance margin issue by only regularizing positively correlated feature weights. Moreover, OrthoReg is computationally efficient since it only regularizes the feature weights, which makes it very suitable for the latest CNN models. We verify our hypothesis through a series of experiments: first using MNIST as a proof of concept, secondly we regularize wide residual networks on CIFAR-10, CIFAR-100, and SVHN ([198]) achieving the lowest error rates in the dataset to the best of our knowledge.

5.2 Proposed Approach

5.2.1 Orthogonal weight regularization

This section introduces the orthogonal weight regularization, a regularization technique that aims to reduce feature detector correlation enforcing local orthogonality between all pairs of weight vectors. In order to keep the magnitudes of the detectors unaffected, we have chosen the cosine similarity between the vector pairs in order to solely focus on the vectors angle $\beta \in [-\pi, \pi]$. Then, given any pair of feature vectors of the same size θ_1, θ_2 the cosine of their relative angle is:

$$\cos(\theta_1, \theta_2) = \frac{\langle \theta_1, \theta_2 \rangle}{||\theta_1|| ||\theta_2||} \tag{5.1}$$

Where $\langle \theta_1, \theta_2 \rangle$ denotes the inner product between θ_1 and θ_2. We then square the cosine similarity in order to define a regularization cost function for steepest descent that has its local minima when vectors are orthogonal:

$$C(\theta) = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n} \cos^2(\theta_i, \theta_j) = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n} \left(\frac{\langle \theta_i, \theta_j \rangle}{||\theta_i|| ||\theta_j||} \right)^2 \tag{5.2}$$

Where θ_i are the weights connecting the output of the layer $l-1$ to the neuron i of the layer l, which has n hidden units. Interestingly, minimizing this cost function relates to the minimization of the Frobenius norm of the cross-covariance matrix without the diagonal. This cost will be added to the global cost of the model $J(\theta; X, y)$, where X are the inputs and y are the labels or targets, obtaining $\tilde{J}(\theta; X, y) = J(\theta; X, y) + \gamma C(\theta)$. Note that γ is an hyperparameter that weights the relative contribution of the regularization term. We can now define the gradient with respect to the parameters:

$$\frac{\delta}{\delta \theta_{(i,j)}} C(\theta) = \sum_{k=1, k \neq i}^{n} \frac{\theta_{(k,j)} \langle \theta_i, \theta_k \rangle}{\langle \theta_i, \theta_i \rangle \langle \theta_k, \theta_k \rangle} - \frac{\theta_{(i,j)} \langle \theta_i, \theta_k \rangle^2}{\langle \theta_i, \theta_i \rangle^2 \langle \theta_k, \theta_k \rangle} \tag{5.3}$$

The second term is introduced by the magnitude normalization. As magnitudes are not relevant for the vector angle problem, this equation can be simplified just by assuming normalized feature detectors:

$$\frac{\delta}{\delta \theta_{(i,j)}} C(\theta) = \sum_{k=1, k \neq i}^{n} \theta_{(k,j)} \langle \theta_i, \theta_k \rangle \tag{5.4}$$

Algorithm 1 Orthogonal Regularization Step.

Require: Layer parameter matrices Θ^l, regularization coefficient γ, global learning rate α.

1: **for** each layer l **to** regularize **do**
2: $\eta_1 = norm\_rows(\Theta^l)$ {Keep norm of the rows of Θ^l.}
3: $\Theta_1^l = div\_rows(\Theta^l, \eta_1)$ {Keep a Θ_1^l with normalized rows.}
4: $innerProdMat = \Theta_1^l transpose(\Theta_1^l)$
5: $\nabla\Theta_1^l = \gamma(innerProdMat - diag(innerProdMat))\Theta_1^l$ {Second term in eq. 5.6}
6: $\Delta\Theta^l = -\alpha(\nabla J_{\Theta^l} + \gamma\nabla\Theta_1^l)$ {Complete eq. 5.6.}
7: **end for**

We then add eq. 5.4 to the backpropagation gradient:

$$\Delta\theta_{(i,j)} = -\alpha\Big(\nabla J_{\theta_{(i,j)}} + \gamma \sum_{k=1,k\neq i}^{n} \theta_{(k,j)}\langle\theta_i, \theta_k\rangle\Big) \tag{5.5}$$

Where α is the global learning rate coefficient, J any target loss function for the backpropagation algorithm.

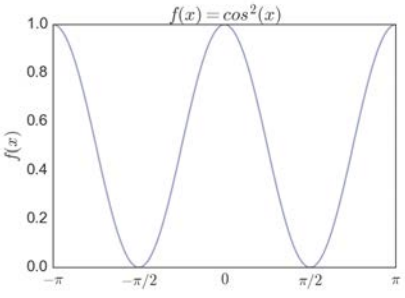
Although this update can be done sequentially for each feature-detector pair, it can be vectorized to speedup computations. Let Θ be a matrix where each row is a feature detector $\theta_{(I,j)}$ corresponding to the normalized weights connecting the whole input I of the layer to the neuron j. Then, $\Theta\Theta^t$ contains the inner product of each pair of vectors i and j in each position i, j. Subsequently, we subtract the diagonal so as to ignore the angle from each feature with respect to itself and multiply by Θ to compute the final value corresponding to the sum in eq. 5.5:

$$\Delta\Theta = -\alpha\Big(\nabla J_\Theta + \gamma(\Theta\Theta^t - diag(\Theta\Theta^t))\Theta\Big) \tag{5.6}$$

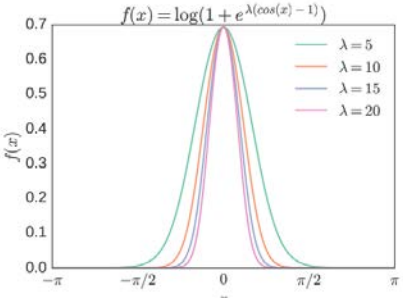
Where the second term is ∇C_Θ. Algorithm 1 summarizes the steps in order to apply OrthoReg.

5.2.2 Negative Correlations

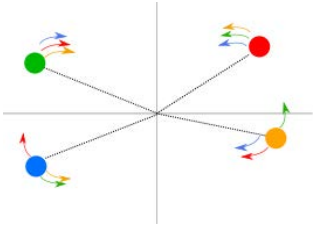
Note that the presented algorithm, based on the cosine similarity, penalizes any kind of correlation between all pairs of feature detectors, i.e. the positive and the negative correlations, see Figure 5.1a. However, negative correlations are related to inhibitory connections, competitive learning, and self-organization. In fact, there is evidence that negative correlations can help a neural population to increase the



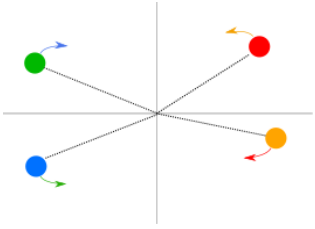
(a) Global loss plot (eq. 5.2)



(b) Local loss plot (eq. 5.7)



(c) Direction of gradients for the loss in (a).



(d) Direction of gradients for loss in (b).

Figure 5.1: **Comparison between the two loss functions** represented by eq. 5.2 and 5.7. (a) is the original loss, (b) is the new loss that discards negative correlations given for different λ values. It can be seen $\lambda = 10$ reaches a plateau when approximating to $\frac{\pi}{2}$. (c) and (d) shows the directions of the gradients for the two loss functions above. For instance, a red arrow coming from a green ball represents the gradient of the loss between the red and green balls with respect to the green one. In (d) most of the arrows disappear since the loss in (b) only applies to angles smaller than $\frac{\pi}{2}$.

signal-to-noise ratio ([199]) in the V1. In order to find out the advantages of keeping negative correlations, we propose to use an exponential to squash the gradients for angles greater than $\frac{\pi}{2}(orthogonal)$:

$$C(\theta) = \sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} \log(1+e^{\lambda(cos(\theta_i,\theta_j)-1)}) = \log(1+e^{\lambda(\langle\theta_i,\theta_j\rangle-1)}),\ ||\theta_i|| = ||\theta_j|| = 1 \quad (5.7)$$

Where λ is a coefficient that controls the minimum angle-of-influence of the regularizer, i.e. the minimum angle between two feature weights so that there exists a gradient pushing them apart, see Figure 5.1b. We empirically found that

the regularizer worked well for $\lambda = 10$, see Figure 5.2b. Note that when $\lambda \simeq 10$ the loss and the gradients approximate to zero when vectors are at more than $\frac{\pi}{2}$ (orthogonal). As a result of incorporating the squashing function on the cosine similarity, negatively correlated feature weights will not be regularized. This is different from all previous approaches and the loss presented in eq. 5.2, where all pairs of weight vectors influence each other. Thus, from now on, the loss in eq. 5.2 is named as global loss and the loss in eq. 5.7 is named as local loss.

The derivative of eq. 5.7 is:

$$\frac{\delta}{\delta\theta_{(i,j)}} C(\theta) = \sum_{k=1,k\neq i}^{n} \lambda \frac{e^{\lambda\langle\theta_i,\theta_k\rangle}\theta_{(k,j)}}{e^{\lambda\langle\theta_i,\theta_k\rangle} + e^{\lambda}} \tag{5.8}$$

Then, given the element-wise exponential operator exp, we define the following expression in order to simplify the formulas:

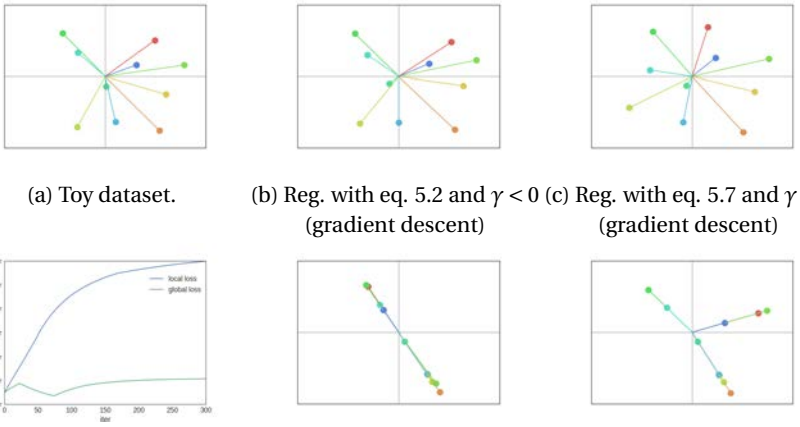
$$\hat{\Theta} = \exp(\lambda(\Theta\Theta^t)) \tag{5.9}$$

and thus, the Δ in vectorial form can be formulated as:

$$\nabla C_\Theta = \lambda \frac{(\hat{\Theta} - diag(\hat{\Theta}))\Theta}{\hat{\Theta} - diag(\hat{\Theta}) + e^{\lambda}} \tag{5.10}$$

In order to provide a visual example, we have created a $2D$ toy dataset and used the previous equations for positive and negative γ values, see Figure 5.2. As expected, it can be seen that the angle between all pairs of adjacent feature weights becomes more uniform after regularization. Note that Figure 5.2b shows that regularization with the global loss (eq. 5.2) results in less uniform angles than using the local loss as shown in 5.2c (which corresponds to the local loss presented in eq. 5.7) because vectors in opposite quadrants still influence each other. This is why in Figure 5.2d, it can be seen that the mean nearest neighbor angle using the global loss (b) is more unstable than the local loss (c). As a proof of concept, we also performed gradient ascent, which minimizes the angle between the vectors. Thus, in Figures 5.2e and 5.2f, it can be seen that the locality introduced by the local loss reaches a stable configuration where feature weights with angle $\frac{\pi}{2}$ are too far to attract each other.

The effects of global and local regularizations on Alexnet, VGG-16 and a 50-layer ResNet are shown on Figure 5.3. As it can be seen, OrthoReg reaches higher decorrelation bounds. Lower decorrelation peaks are still observed when the input dimensionality of the layers is smaller than the output since all vectors cannot be orthogonal at the same time. In this case, local regularization largely outperforms global regularization since it removes interferences caused by negatively correlated



(a) Toy dataset. (b) Reg. with eq. 5.2 and $\gamma < 0$ (gradient descent) (c) Reg. with eq. 5.7 and $\gamma < 0$ (gradient descent)

(d) Nearest neighbor angle average. (e) Reg. with eq. 5.2 and $\gamma > 0$ (gradient ascent) (f) Reg. with eq. 5.7 and $\gamma > 0$ (gradient ascent)

Figure 5.2: **Toy experiments.** A 2d dataset is regularized with global loss (eq. 5.2) and local loss (eq. 5.7). (a) shows the initial 2D randomly generated dataset. (b) the dataset after 300 regularization steps using the global loss and (c) using the local loss. (d) the evolution of the mean nearest neighbor angle for the global loss (b) and the local loss (c). (e) and (f) correspond to (b) and (c) but using gradient ascent instead of gradient descent as a sanity-check.

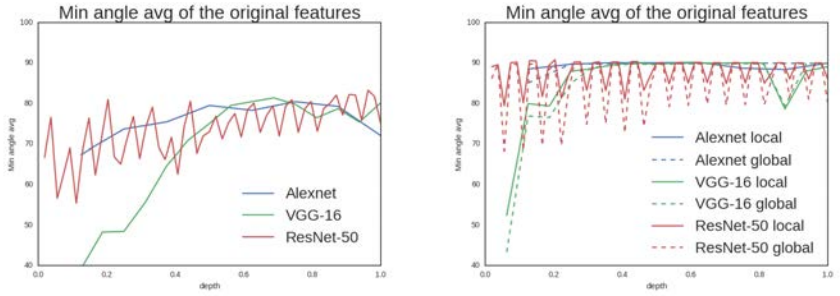
feature weights. This suggests why increasing fully connected layers' size has not improved networks performance.

5.3 Experiments and Results

In this section we provide a set of experiments that verify that (i) training with the proposed regularization increases the performance of naive unregularized models, (ii) negatively correlated feature weights are useful, and (iii) the proposed regularization improves the performance of state-of-the-art models.

5.3.1 Verification experiments

As a sanity check, we first train a three-hidden-layer Multi-Layer Perceptron (MLP) with `ReLU` non-liniarities on the MNIST dataset ([2]). Our code is based in the



(a) Unregularized feature weights.

(b) Regularized feature weights.

Figure 5.3: **Effects of local and global regularization** on the Alexnet, VGG-16 and 50-layer-ResNet weights. The regularized versions reach higher decorrelation bounds (in terms of minimum angle) than the unregularized counterparts.

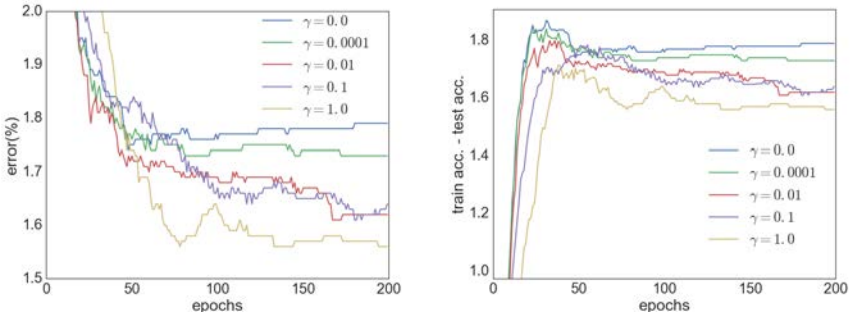
`train-a-digit-classifier` example included in `torch/demos`[1], which uses an upsampled version of the dataset (32×32). The only pre-processing applied to the data is a global standardization. The model is trained with SGD and a batch size of 200 during 200 epochs. No momentum neither weight decay was applied. By default, the magnitude of the weights of this experiments is recovered after each regularization step in order to prove the regularization only affects their angle.

Sensitivity to hyperparameters. We train a three-hidden-layer MLP with 1024 hidden units, and different γ and λ values so as to verify how they affect the performance of the model. Figure 5.4a shows that the model effectively achieves the best error rate for the highest gamma value ($\gamma = 1$), thus proving the advantages of the regularization. On Figure 5.4b, we verify that higher regularization rates produce more general models. Figure 5.5a depicts the sensitivity of the model to λ. As expected, the best value is found when lambda corresponds to Orthogonality ($\lambda \simeq 10$).

Negative Correlations. Figure 5.5b highlights the difference between regularizing with the global or the local regularizer. Although both regularizations reach better error rates than the unregularized counterpart, the local regularization is better than the global. This confirms the hypothesis that negative correlations are useful and thus, performance decreases when we reduce them.

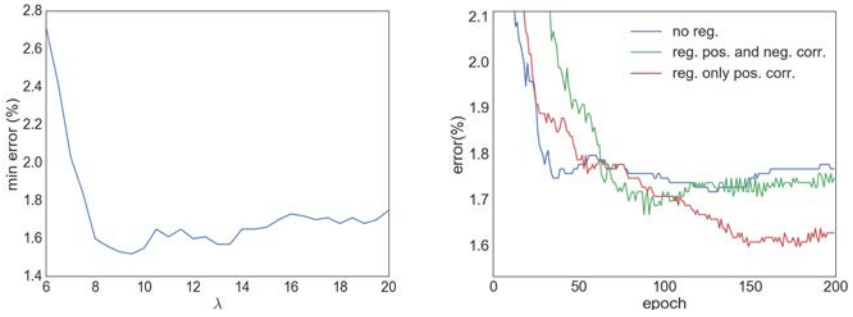
Compatibility with initialization and dropout. To demonstrate the proposed regularization can help even when other regularizations are present, we trained a

[1] https://github.com/torch/demos



(a) (b)

Figure 5.4: **Sensitivity to γ.** (a) The evolution of the error rate on the MNIST validation set for different regularization magnitudes. It can be seen that for $\gamma = 1$ it reaches the best error rate (1.45%) while the unregularized counterpart ($\gamma = 0$) is 1.74%. (b) Measures the overfitting of the model for different γ, confirming that higher regularization rates decrease overfitting.



(a) (b)

Figure 5.5: **Sensitivity to λ.** (a) Shows the minimum error rate for different λ values. (b) Classification error on MNIST for different loss functions. Not regularizing negative correlated feature weights (eq. 5.7) results in better test error than regularizing them (eq.5.2).

CNN with (i) dropout (`c32-c64-l512-d0.5-l10`)[2] or (ii) LSUV initialization ([196]).

[2] cxx = convolution with xx filters. lxx = fully-connected with xx units. dxx = dropout with prob

Table 5.1: **Error rates for a small CNN trained with the MNIST dataset.** OrthoReg leads to much better results when no other improvements such as Dropout and LSUV are present but it can still make small accuracy increments when these two techniques are present.

OrthoReg	Base	Base+Dropout	Base+LSUV
None	0.92	0.70 ± 0.01	0.86
Conv Layers	0.75	0.69 ± 0.03	0.83
All Layers	**0.75**	$\mathbf{0.66 \pm 0.03}$	**0.79**

In Table 5.1, we show that best results are obtained when orthogonal regularization is present. The results are consistent with the hypothesis that OrthoReg, as well as Dropout and LSUV, focuses on reducing the model redundancy. Thus, when one of them is present, the margin of improvement for the others is reduced.

5.3.2 Regularization on CIFAR-10 and CIFAR-100

We show that the proposed OrthoReg can help to improve the performance of state-of-the-art models such as deep residual networks ([188]). In order to show the regularization is suitable for deep CNNs, we successfuly regularize a 110-layer ResNet[3] on CIFAR-10, decreasing its error from 6.55% to 6.29% without data augmentation.

In order to compare with the most recent state-of-the-art, we train a wide residual network ([200]) on CIFAR-10 and CIFAR-100. The experiment is based on a `torch` implementation of the 28-layer and 10th width factor wide deep residual model, for which the median error rate on CIFAR-10 is 3.89% and 18.85% on CIFAR-100[4]. As it can be seen in Figure 5.6, regularizing with OrthoReg yields the best test error rates compared to the baselines.

The regularization coefficient γ was chosen using grid search although similar values were found for all the experiments, specially if regularization gradients are normalized before adding them to the weights. The regularization was equally applied to all the convolution layers of the (wide) ResNet. We found that, although the regularized models were already using weight decay, dropout, and batch normalization, best error rates were always achieved with OrthoReg.

Table 5.2 compares the performance of the regularized models with other state-

xx.

[3] https://github.com/gcr/torch-residual-networks

[4] https://github.com/szagoruyko/wide-residual-networks

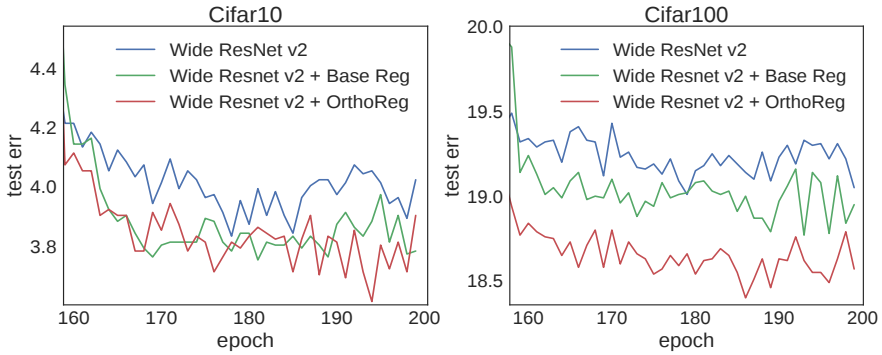


Figure 5.6: **Wide ResNet error rate on Cifar10 and Cifar100.** OrthoReg shows better performance than the base regularizer (all feature maps), and the unregularized counterparts.

Table 5.2: **Comparison with other CNNs on CIFAR-10 and CIFAR-100 (Test error %).** Orthogonally regularized residual networks achieve the best results to the best of our knowldege. Only single-crop results are reported for fairness of comparison. *Median over 5 runs as reported by [200].

Network	**CIFAR-10**	**CIFAR-100**	**Augmented**
Maxout ([201])	9.38	38.57	YES
NiN ([202])	8.81	35.68	YES
DSN ([183])	7.97	34.57	YES
Highway Network ([203])	7.60	32.24	YES
All-CNN ([204])	7.25	33.71	NO
110-Layer ResNet ([188])	6.61	28.4	NO
ELU-Network ([205])	6.55	**24.28**	NO
OrthoReg on 110-Layer ResNet*	**6.29** $\pm$ 0.19	28.33 $\pm$ 0.5	NO
LSUV ([196])	5.84	-	YES
Fract. Max-Pooling ([206])	4.50	27.62	YES
Wide ResNet ([200])*	3.89	18.85	YES
OrthoReg on Wide ResNet*	**3.69** $\pm$ **0.01**	**18.56** $\pm$ **0.12**	YES

of-the-art results. As it can be seen the regularized model surpasses the state of the art, with a 5.1% relative error improvement on CIFAR-10, and a 1.5% relative error improvement on CIFAR-100.

Table 5.3: **Comparison with other CNNs on SVHN.** Wide Resnets regularized with OrthoReg show better performance.

Model	Error rate
NiN ([202])	2.35
DSN ([183])	1.92
Stochastic Depth ResNet ([207])	1.75
Wide Resnet ([200])	1.64
OrthoReg on Wide Resnet	**1.54**

5.3.3 Regularization on SVHN

For SVHN we follow the procedure depicted in [200], training a wide residual network of `depth=28`, `width=4`, and dropout. Results are shown in Table 5.3. As it can be seen, we reduce the error rate from 1.64% to 1.54%, which is the lowest value reported on this dataset to the best of our knowledge.

5.4 Discussion

Regularization by feature decorrelation can reduce Neural Networks overfitting even in the presence of other kinds of regularizations. However, especially when the number of feature detectors is higher than the input dimensionality, its decorrelation capacity is limited due to the effects of negatively correlated features. We showed that imposing locality constraints in feature decorrelation removes interferences between negatively correlated feature weights, allowing regularizers to reach higher decorrelation bounds, and reducing the overfitting more effectively.

In particular, we show that the models regularized with the constrained regularization present lower overfitting even when batch normalization and dropout are present. Moreover, since our regularization is directly performed on the weights, it is especially suitable for fully convolutional neural networks, where the weight space is constant compared to the feature map space. As a result, we are able to reduce the overfitting of 110-layer ResNets and wide ResNets on CIFAR-10, CIFAR-100, and SVHN improving their performance. Note that despite OrthoReg consistently improves state of the art ReLU networks, the choice of the activation function could affect regularizers like the one presented in this work. In this sense, the effect of asymmetrical activations on feature correlations and regularizers should be further investigated in the future.

6 Beyond One-hot Encoding: lower dimensional target embedding

Target encoding plays a central role when learning Convolutional Neural Networks. In this realm, One-hot encoding is the most prevalent strategy due to its simplicity. However, this so widespread encoding schema assumes a flat label space, thus ignoring rich relationships existing among labels that can be exploited during training. In large-scale datasets, data does not span the full label space, but instead lies in a low-dimensional output manifold. Following this observation, we embed the targets into a low-dimensional space, drastically improving convergence speed while preserving accuracy. Our contribution is two fold: *(i)* We show that random projections of the label space are a valid tool to find such lower dimensional embeddings, boosting dramatically convergence rates at zero computational cost; and *(ii)* we propose a normalized eigenrepresentation of the class manifold that encodes the targets with minimal information loss, improving the accuracy of random projections encoding while enjoying the same convergence rates. Experiments on CIFAR-100, CUB200-2011, Imagenet, and MIT Places demonstrate that the proposed approach drastically improves convergence speed while reaching very competitive accuracy rates.

6.1 Motivation

Convolutional Neural Networks lie at the core of the latest breakthroughs in large-scale image recognition [169, 208], at present even surpassing human performance [181], applied to the classification of objects [209], faces [210], or scenes [211]. Due to its effectiveness and simplicity, one-hot encoding is still the most prevalent procedure for addressing such multi-class classification tasks: in essence, a function $f : \mathbb{R}^p \rightarrow \mathbb{Z}_2^n$ is modeled, that maps image samples to a probability distribution over a discrete set of the n labels of target categories.

Unfortunately, when the output space grows, class labels do not properly span the full label space, mainly due to existing label cross-correlations. Consequently, one-hot encoding might result inadequate for fine-grained classification tasks,

since the projection of the outputs into a higher dimensional (orthogonal) space dramatically increases the parameter space of computed models. In addition, for datasets with a large number of labels, the ratio of samples per label is typically reduced. This constitutes an additional challenge for training CNN models in large output spaces, and the reason of slow convergence rates [212].

In order to address the aforementioned limitations, output embeddings have been proposed as an alternative to the one-hot encoding for training in large output spaces [213]: depending on the specific classification task at hand, using different output embeddings captures different aspects of the structure of the output space. Indeed, since embeddings use weight sharing during training for finding simpler (and more natural) partitions of classes, the latent relationships between categories are included in the modeling process.

According to Akata *et al.* [214], output embeddings can be categorized as:

- Data-independent embeddings, such as drawing rows or columns from a Hadamard matrix [215]: data-independent embeddings produce strong baselines [216], since embedded classes are equidistant due to the lack of prior knowledge;
- Embeddings based on a priori information, like attributes [217], or hierarchies [218]: unfortunately, learning from attributes requires expert knowledge or extra labeling effort and hierarchies require a prior understanding of a taxonomy of classes, and in addition, approaches that use textual data as prior do not guarantee visual similarity [216]; and
- Learned embeddings, for capturing the semantic structure of word sequences (i.e. annotations) and images jointly [219]. The main drawbacks of learning output embeddings are the need of a high amount of data, and a slow training performance.

Thus, in cases where there exist high quality attributes, methods with prior information are preferred, while in cases of a known equidistant label space, data-independent embeddings are a more suitable alternative. Unfortunately, the architectural design of a model is bound to the particular choice among the above-mentioned embeddings. Thus, once a model is chosen and trained using an specific output embedding, it is hard to reuse it for another tasks requiring a different type of embedding.

In this paper, Error-Correcting Output Codes (ECOC) are proven to be a better alternative to one-hot encoding for image recognition, since ECOCs are a generalization of the three embedding categories [220], so a change in the ECOC matrix will not constitute a change in the chosen architecture. In addition, ECOCs naturally

enable error-correction, low dimensional embedding spaces [221], and bias and variance error reduction [222].

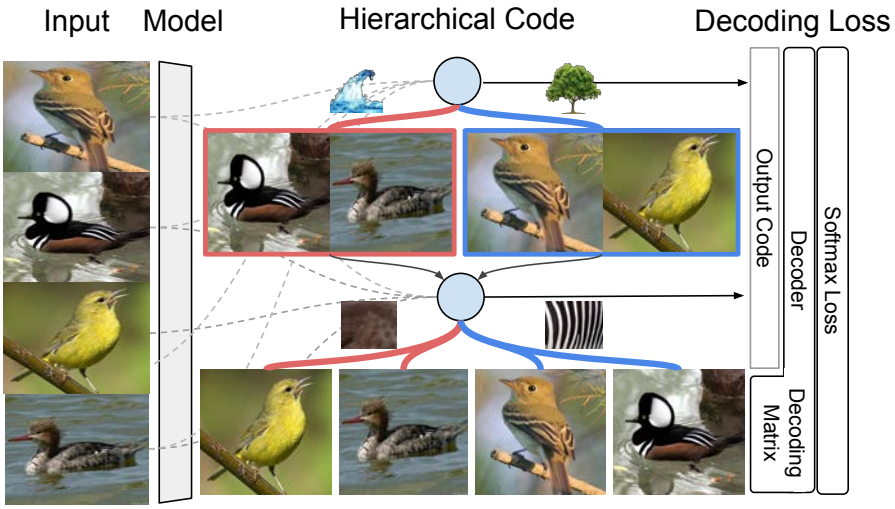


Figure 6.1: **Hierarchical coding scheme.** This paper proposes to replace the traditional one-hot output scheme of CNNs with a reduced scheme with at least $log_2(k)$ outputs. In addition, when using a hierarchical representation of the data labels, outputs show that the most discriminative attributes to split the target classes have been learned. In essence, a decoder computes the similarities of the "predicted code" in a "code-matrix", and subsequently the output label is then obtained through a softmax layer. The internal code representation is depicted in a tree structure, where each bit of the code corresponds to the actual learned partition from the data, from lower partition cost (aquatic) to higher (stripped).

Inspired by the latest advances on ECOCs, we circumvent one-hot encoding by integrating the Error-Correcting Output Codes into CNNs, as a generalization of output embedding. As a result, a best-of-both-worlds approach is indeed proposed: compact outputs, data-based hierarchies, and error correction. Using our approach, training models in low-dimensional spaces drastically improves convergence speed in comparison to one-hot encoding. Figure 6.1 shows an overview of the proposed model.

The rest of the paper is organized as follows: Section 6.2 reviews the existing work most closely related to this paper. Section 6.3 presents the contribution of the proposed embedding technique, which is two fold: *(i)* we show that random projections of the label space are suitable for finding useful lower dimensional

embeddings, while boosting dramatically convergence rates at zero computational cost; and *(ii)* In order to generate partitions of the label space that are more discriminative than the random encoding (which generates random partitions of the label space), we also propose a normalized eigenrepresentation of the class manifold to encode the targets with minimal information loss, thus improving the accuracy of random projections encoding while enjoying the same convergence rates. Subsequently, the experimental results on CIFAR-100 [168], CUB200-2011 [223], MIT Places [211], and ImageNet [169] presented in Section 6.4 show that our approach drastically improves convergence speed while maintaining a competitive accuracy. Lastly, Section 6.5 concludes the paper discussing how, when gradient sparsity on the output neurons is highly reduced, more robust gradient estimates and better representations can be found.

6.2 Related work

This section reviews those works on output embeddings most related to ours, in particular those using ECOC.

Output Embeddings Most of the related literature addresses the challenge of zero-shot learning, i.e. training a classifier in the absence of labels. Often, the proposed approaches take into account the attributes of objects [214, 224–226] related to the different classes through well-known, shared object features.

Due to their computing efficiency based on a divide-and-conquer strategy, output embeddings have been also proven useful for those multi-class classification problems in which testing all possible class labels and hierarchical structures is not feasible [213, 219, 223, 227]. Given a large output space, most labels are usually considered instances of a superior category e.g., sunflower and violet are flower plants. In this sense, the inherent hierarchical structure of the data makes divide-and-conquer hierarchical output spaces a suitable alternative to the traditionally flat 1-of-N classifiers. Likewise in the context of language processing, Mikolov *et al.* combine Huffman binary codes and hierarchical soft-max in order to map the most frequent codes to shorter paths in a tree [228].

Because output embeddings enforce weight sharing, they have been also used when the number of classes is rather large, with no clear inter-class boundaries, and a decaying ratio of the number of examples per class. In this context, in order to reduce the output space, Weston *et al.* proposed WSABIE, an online learning-to-rank algorithm to find an embedding for the labels based on images [229].

In the field of large-scale recognition, hierarchical approaches such as using tree-based priors [230], label relational graphs [231], CNN hierarchies [232], and HD-CNNs [233] have been proposed. For example in [234] binary hash codes are

used for fast image retrieval. However, such hierarchical approaches need to be learned, and cannot be easily interchanged with other embeddings. In addition, for approaches learning codes as latent variables, to find the optimal ones in terms of class separability or error correction is not guaranteed [234]. Due to all this, ECOC constitute a better alternative for seamless integration with CNNs, as detailed next.

Error-Correcting Output Codes[1] ECOC have been applied in multiple fields such as medical imaging [235], face and facial-feature recognition [236, 237], and segmentation of human limbs citesanchez2015hupba8k+. ECOCs are a generic divide-and-conquer framework that combines binary partitions to achieve multi-class recognition [238]. Their core property is the capability to correct errors of binary classifiers using redundancy, while reducing the bias and variance of the ensemble [222]. Advanced approaches propose to use them as intermediate representations [239].

ECOC consist of two main steps: *coding* and *decoding*. The *coding* step consists in assigning a codeword of arbitrary length k to each of the n classes. Codewords are organized in a "code matrix" $\mathbf{M}_{k,n} \in \{-1, 1\}$, where each column is a binary partition on the label space in meta-classes. Since there are many possible bi-partitions, the design of the code is central for obtaining discriminative ones. Indeed there are several approaches for generating ECOCs: Exhaustive codes [238], BCH codes [240], random codes [241], and circular ECOC [242] are few examples of methods that generate codes independently from the inherent structure of the data.

Although ECOCs can be data-independent and even randomly generated, they can also be learnt from data: Pujol *et al.* propose a discriminant ECOC approach based on hierarchical partitions of the output space [243]. Subsequently, Escalera *et al.* [244] proposed to split complex problems into easier subclasses, embedded as binary dichotomizers in the ECOC framework, easier to optimize. In [245], it is also shown Optimal continuous ECOCs can be found by gradient descent. Griffin & Perona [246] use trees to efficiently handle multi-class problems, which posteriorly Zhang *et al.* improved by finding optimal partitions with spectral ECOCs [247].

In the decoding step, a sample x can be decoded as the output of k binary classifiers $\{f_1(x), f_2(x), ..., f_k(x)\}$. Given the predicted code, the class label y corresponds to the closest row in $M_{k,n}$. The most common decoding methods are the Hamming and Euclidean distances but there are more sophisticated approaches such as probabilistic-based decoding, especially with ternary codes [220].

Inspired from latest ECOC advances, we propose to integrate output codes in large-scale deep learning problems. In this context, few approaches in the literature have been presented: in [231, 248], CNNs are also used to directly predict the code

[1] We use the standard notation in ECOCs: bold capital letters denote matrices (e.g. $\mathbf{X}$) and bold lower-case letters represent vectors (e.g., $\mathbf{z}$). All non-bold letters denote scalar variables.

bits for Optical Character recognition (OCR). We go a step further by: *(i)* showing that the convergence speed in large scale settings with millions of images can be dramatically improved; *(ii)* instead of directly predicting the code bits, we integrate the euclidean decoding with the cross-entropy loss, so that the network does not only optimize individual bits independently but also inter-code distances, which results in error-correction.

Our approach enhances the convergence of CNNs using random codes, i.e. when the inter-class relationships are not considered. We achieve even lower error rates with data-dependent codes, due to using more efficient data partitions. Similarly, Yang *et al.* also used CNNs to integrate data-independent Hadamard Codes with the Euclidean loss [249]. But due to the efficiency of data-dependent codes, our encoding proposal is shown more efficient than [249], by halving the required CNN output size, and eliminating the need of training multiple CNNs to predict code chunks.

6.3 Proposed Approach

Figure 6.1 depicts our proposed model inspired by the ECOC framework [238] and applied for deep supervised learning. Given a set of n classes, an ECOC consists of a set of k binary partitions of the label space (groups of classes) representing each of the n classes in the dataset. The codes are usually arranged in a design matrix $\mathbf{M} \in \{-1,1\}^{n\times k}$.

Let's define the output of the last layer of a neural network as z^l, with l the depth of the network. For the sake of clarity the identity non-linearity $\phi(\cdot)$ is used so that $\mathbf{z}^l = \phi(\mathbf{z}^l)$. Thus, given the weights of the previous layer $\Theta^{(l-1)}$, and the corresponding bias $\mathbf{b}^{(l-1)}$, $\mathbf{z}^l$ can be computed as $\Theta^{(l-1)}\mathbf{z}^{(l-1)} + \mathbf{b}^{(l-1)}$.

In our case, we reduce the output dimensionality of a CNN, i.e. the dimensionality of $\mathbf{z}^l$, from n (the number of classes) to k, an arbitrary number of partitions. Then, given a design matrix $\mathbf{M}^{n\times k}$, where each row encodes a class label, the predicted class is obtained by finding the distance of the output with each row of the design matrix $\mathbf{D} = \mathbf{M} - \mathbb{1}^\top \mathbf{z}^l$, with $\mathbb{1}^\top$ a column vector constituted by ones, and obtaining the label with $argmin(\mathbf{D})$. Then, we seamlessly integrate our proposal in the traditional log-likelihood and softmax loss layer.

6.3.1 Embedding output codes in CNNs

Given a training set $\{x_i, y_i\}$ $i = 1 : s$, of image-label pairs, CNNs constitute the state-of-the-art at finding good local minima by empirical risk minimization (ERM) using the cross-entropy as the loss function J by means of backpropagation [250]:

$$J(X,Y;\Theta) = -\frac{1}{s}\sum_{i=1}^{s}[y_i log(\hat{y})_i + (1-y_i) log(1-\hat{y}_i)],$$

where $\hat{y}_i = argmax(\mathbf{h}(\mathbf{z}^l(\mathbf{x}_i))) \in \mathbb{R}$ is the predicted label for the i^{th} example and $y_i \in \{0,1\}$ the ground truth label. Since cross-entropy requires probability distributions, the output of the network $\mathbf{z}^l$ is fed to a softmax layer that assigns a probability score to each of the n possible classes:

$$h(\mathbf{z}^l)_j = \frac{e^{z_j^l}}{\sum_{i=1}^{N} e^{z_i^l}}, j \in \{1,2,...,n\}.$$

The derivative of the loss function J for gradient descent through backpropagation is known to be:

$$\frac{\delta J}{\delta z_i^l} = y_i - \hat{y}_i.$$

The decoder is introduced between the output $\mathbf{z}^l$ of the network and the softmax function $\mathbf{h}(\mathbf{z}^l)$. Concretely, the negative normalized Euclidean distance $\mathbf{D}(\mathbf{z}^l|\mathbf{M})$ between $\mathbf{z}^l$ and the rows in $\mathbf{M}$ is used, so that the output of the softmax represents the probability of the output of the CNN to be decoded as the $i^{th}, i \in \{1,2,3..,n\}$ output word.

We reformulate the softmax function $\mathbf{h}(\mathbf{z}^l)$ as $\mathbf{h}(\mathbf{D}(\frac{\mathbf{z}^l}{||\mathbf{z}^l||_2}))$, with the variable change of $\mathbf{D}(\frac{\mathbf{z}^l}{||\mathbf{z}^l||_2})$ by $\mathbf{D}(\mathbf{U})$ (with $\mathbf{U}(\mathbf{z})$ the normalized vector). The derivative of the loss can be computed using the chain rule:

$$\frac{\delta J(\mathbf{D},Y;\Theta)}{\delta \mathbf{z}} = \frac{\delta J(\mathbf{D},Y;\Theta)}{\delta \mathbf{D}} \frac{\delta \mathbf{D}}{\delta \mathbf{U}}^{(1)} \frac{\delta \mathbf{U}}{\delta \mathbf{z}^l}^{(2)}.$$

We now calculate:

$$\frac{\delta \mathbf{D}}{\delta \mathbf{U}} = \frac{\delta}{\delta \mathbf{U}} \frac{-1}{2}(\mathbf{M} - \mathbb{1}^\top \mathbf{U})(\mathbf{M} - \mathbb{1}^\top \mathbf{U})^\top = \mathbf{M} - \mathbb{1}^\top \mathbf{U}, \tag{6.1}$$

$$\frac{\delta \mathbf{U}}{\delta \mathbf{z}} = \frac{\delta}{\delta \mathbf{z}} \frac{\mathbf{z}}{||\mathbf{z}||_2} = \frac{\mathbf{I}||\mathbf{z}||_2 - \mathbf{z}\mathbf{U}^\top}{||\mathbf{z}||_2}. \tag{6.2}$$

Given eq. 6.1 and 6.2, it is possible to compute the derivative of the cross-entropy with the new decoding loss $\hat{J}$:

$$\frac{\delta \hat{J}}{\delta \mathbf{z}^l} = (\mathbf{Y} - \hat{\mathbf{Y}})[(\mathbf{M} - \mathbb{1}^\top \mathbf{U}) \frac{\mathbf{I}||\mathbf{z}^l||_2 - \mathbf{z}^l \mathbf{U}^\top}{||\mathbf{z}^l||_2}]^\top. \tag{6.3}$$

Provided the amount of computation that can be shared from the forward pass to the backward pass, this process does not slow-down the training phase. In fact, the cost is compensated by *(i)* the shrinkage of $\mathbf{z}$, which also results in a reduction of the number of network parameters, and *(ii)* the increase of convergence speed.

The convergence speed increases because reducing the output layer results in parameter sharing, which produces more robust gradient estimates. The explanation is that the softmax function distributes the probabilities among a high number of neurons. Thus, the the gradient $\delta J = y_i - \hat{y}_i$ is zero for most outputs because $y_i = 1$ only once in the ground truth vector, and $\mathbb{E}(\hat{y}_j) = \frac{1}{n}$. Given that the network is certain about the output i', the expected output for the rest of the outputs is even smaller $\mathbb{E}(\hat{y}_{j\neq i'}) = \frac{1-y_{i'}}{n-1}$.

In other words, output layers with huge number of outputs and smaller mini-batch size can only update the weights of few output units per iteration, since activation expected value is virtually zero. Thus, the gradients for these outputs are either zero or based on too few examples. This leads to noisy estimates to the real loss surface. As a result, reducing the output space with our method increases the ratio of activations per mini-batch, helping to obtain more robust gradient estimates and increasing convergence speed, reduces the mini-batch size, and thus the memory requirements.

6.3.2 Connections with Normalized Cuts

CNNs trained with our approach are robust and fast even when drawing codes from a normal distribution. The reason is the fact that random gaussian matrices tend to follow the coding properties described in the literature [238, 251], such as row and column orthogonality. For most large datasets the label space follows a hierarchical structure and defining random partitions of the label space is rather unnatural. In order to find the most simple partitions we use an eigenrepresentation of the class manifold based on the class similarities found in the dataset. Concretely, solving the normalized cut (Ncut) problem on the class similarity graph is a way of obtaining n uncorrelated low-cost partitions, with n the number of classes [252]. The NCut can be approximated by solving the eigendecomposition of the normalized Laplacian of the class similarity matrix $\mathbf{L_M}$:

$$\mathbf{L_G} = \mathbf{D}^{\frac{1}{2}}(\mathbf{D} - \mathbf{M})\mathbf{D}^{\frac{-1}{2}} = \lambda\mathbf{V},$$

where $\mathbf{M}$ is the class similarity matrix, $\mathbf{D}$ is the degree matrix, $\lambda_\mathbf{i}$ are the eigenvalues in ascending order and $\mathbf{v_i}$, the corresponding eigenvectors $i \in \{0, 1, 2, ..., k\}$. Given that $\lambda_0 = 0$, the eigenvectors $\mathbf{v}_i$, $i \in \{1, ..., k\}$ constitute the partitions ordered by the Ncut cost. As explained in [211], this kind of codes have desirable properties

such as balancing, orthogonality, lower error bounds due to the separability maximization, and similarity preserving, i.e. similar classes have similar codes. We show that training CNNs to predict the embedded target, together with this data-based codes, exhibit lower error rates than using random codes. Contrary to [247], we do not threshold the eigenvectors so as to obtain a binary code but we interpret the values as likelihoods.

In the following section, we provide empirical evidence confirming that CNNs trained with our proposed methodology on CIFAR-100, CUB-200, MIT Places, and Imagenet have faster convergence rates (with comparable or better recognition rates), even with smaller mini-batch size, than their one-hot counterparts.

6.4 Experiments and Results

To validate our approach, we perform a thorough analysis of the advantages of embedding output codes in CNN models over different state-of-the-art datasets. First, we describe the considered datasets, methods and evaluation.

6.4.1 Datasets

We first experiment the ImageNet 2012 Large-Scale Visual Recognition Challenge (ILSVRC-2012) [169] and the MIT Places-205 [211] datasets. ImageNet consists of 1.2M images, and 50K validation images with 10K object classes. MIT Places is constituted by 2.5M images from 205 scene categories for training, and 100 images for category for testing.

Subsequently we experiment on the CIFAR-100 [168] and the Caltech-UCSD Birds-200-2011 [253]. CIFAR-100 consists of 50K 32×32 images for training, and 10K 32×32 images for testing belonging to 10 coarse categories and 100 fine-grained categories. CUB-200 contains 11,788 images (5,994 images for training and 5,794 for test) of 200 bird species, each image annotated with 15 part locations, 312 binary attributes, and 1 Bounding Box.

6.4.2 Methods and evaluation

We use standard state-of-the-art models to evaluate the contribution of the proposed target embedding procedure instead of comparing with state-of-the-art results on the considered datasets. Note that any model, including more recent and powerful state-of-the-art architectures, can benefit from our target embedding methodology.

As a proof of concept, we first validate data-independent codes on the Imagenet

and MIT Places datasets. Concretely, we retrain with our approach the `fc7` and `fc8` layers of an Alexnet model [1] pre-trained on the respective datasets. Concretely, we randomly reinitialize their weights and train them using SGD with a global learning rate (`lr`) of 0.001, and the specific `lr` of the reinitialized layers is multiplied by 10.

Then, we demonstrate the advantages of data-dependent codes on the fine-grained CIFAR-100 and CUB-200 2011. For CIFAR-100, we use the `cifar_quick` models found in the Caffe framework [254]. The network is initialized with noise sampled from a gaussian distribution, and the model is trained for 100 epochs. Fine-tuning on CUB-200 is performed with the same pre-trained model of the Imagenet experiments for 30 epochs, and the `lr` is divided by 10 after 15 epochs.

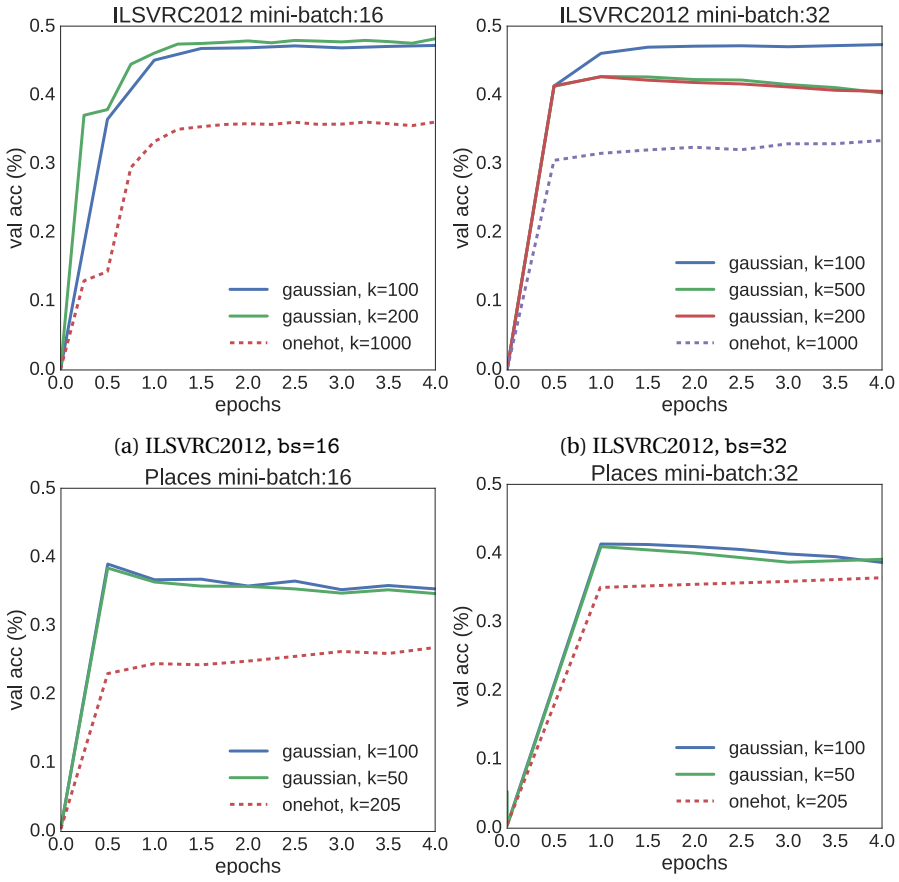
Experiments with the standard Alexnet CNN [1] (caffe version [254]) on Imagenet, and MIT Places, prove that CNNs trained with random codes and our approach show faster convergence rates than using one-hot encoding, especially for small mini-batch sizes, while matching one-hot in performance for bigger mini-batch sizes. Thus, the proposed data-dependent encoding approach performs better than using random codes for fine-grained datasets, with fuzzy inter-class boundaries, essentially because random codes alone do not take into account the correlation of attributes.

6.4.3 Random codes for faster convergence

Output encodings allow to embed sparse output spaces into compact representations. For instance, codes generated with the dense random strategy only need $k = 10log(n)$ bits [241] to encode n classes. An inherent property of one-hot encoding is the output activation sparsity for huge output spaces. Given a randomly initialized CNN with one-hot encoding, provided that the output neurons follow a uniform distribution, the probability assigned to each class will be $\frac{1}{n},\ n = \#Classes$, which tends to 0 for $n \to \infty$. In the final stages of training, the situation will persist since just an extremely small ratio of the neurons activate, i.e. a small subset of the neurons show high probability for the predicted class while the residual probability mass is spread over a much larger number of neurons.

Thus, it can be coarsely estimated that the update probability of the parameters associated to an output neuron during an SGD step is related to the ratio $\rho = \frac{bs}{n}$, with mini-batch size bs, being $\rho = 256 \cdot 10^3$ for Alexnet trained on Imagenet, provided that $p(Y = n_i) = p(Y = n_j),\ i \neq j$. In other words, given a label, sampling more images increases the probability of that label being in the set of samples, and drawing less samples than the number of labels ensures that at least $n - s$ labels will not be seen during the update.

Figure 6.2 shows the resulting validation accuracy when training Alexnet on the ILSVRC2012 and MIT Places for different mini-batches and a random code sampled



(a) ILSVRC2012, `bs=16`

(b) ILSVRC2012, `bs=32`

(c) MIT Places, `bs=16`

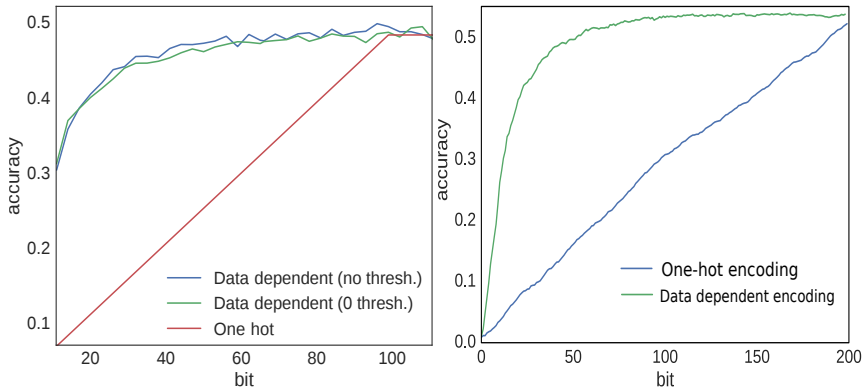
(d) MIT Places, `bs=32`

Figure 6.2: **Validation accuracy on ILSVRC2012 and MIT Places.** Using output codes randomly sampled from a normal distribution results in faster convergence, especially for small mini-batch sizes (a,c)

from $\mathcal{N}(0,1)$. As it can be seen, models trained with our approach converge faster than those trained with one-hot encoding.

6.4.4 Using data-based encodings

In order to adapt to fine-grained settings, i.e. with high inter-class correlations, and few examples per class, we propose to generate the output codes using the



(a) CIFAR100 (b) CUB200

Figure 6.3: **Classification accuracy based on the number of the code bits.** As expected, the same amount of information is encoded for each of the one-hot bits while the same results are obtained with just the 25% of the data-based codes.

eigenvectors of the normalized Laplacian of the class similarity matrix. Since this eigendecomposition generates the most discriminating, hierarchical partitions based on the data, models trained with this data-dependent codes result in higher accuracy bounds than the random counterparts.

To confirm the aforementioned advantages of using data-dependent codes we choose to experiment on the well-established CIFAR-100 and CUB-200 2011 fine-grained datasets. see Fig. 6.3. We use CIFAR-100 for fast experimentation, and then we apply the best setting to CUB-200.

CIFAR-100. First, we evaluate different procedures for generating the codes:

1. One-hot. A vector of $n-1$ zeros and a one at the target position (with n the number of classes).
2. Dense random [241]. Sampling the matrix with the most uncorrelated rows and columns from $\mathcal{U}(0,1)$.
3. Gaussian. Sampling matrices from a normal distribution.
4. Data-based. Constructing the code matrix from the eigenvalues of the class similarity Laplacian.

Note that Gaussian and Data-based codes are composed of real numbers and a thresholding function should be applied for obtaining binary partitions. We test thresholding at zero and the median of the rows of the code matrix. Additionally,

Table 6.1: **Influence of code designs on CIFAR-100.** Dense output encodings are more robust than One-hot to the loss of bits. As expected, data-based codes outperform the rest of encodings (50%), especially when no threshold is applied to binarize the code.

Code	One-hot			Gaussian								
Binarization	-			-			Zero			Median		
Length	66	100	200	66	100	200	66	100	200	66	100	200
Accuracy (%)	32.4	49.2	-	44.9	44.8	44.8	45.0	47.1	49.1	45.6	47.8	48.4
	Dense Random			Data-dependent								
	-	-	-	-			Zero			Median		
	66	100	200	66	99	200	66	99	200	66	99	200
	43.9	44.5	44.3	48.0	50.0	49.7	46.7	49.0	48.9	47.4	47.8	49.7

(a) One-hot. (b) Dense random. (c) Data-based.

Figure 6.4: **T-sne visualization on CIFAR-100** on the ten coarse categories for the hidden fc layer of a CNN trained with (a) one-hot encoding, (b) an output code generated with the dense random strategy, and (c) a data-based code.

we test the raw values, interpreting them as the likelihood of the k^{th} metaclass to be present in the n^{th} class.

As it can be seen in table 6.1, output encodings are more robust, losing a smaller percentage of the accuracy when the number of code-bits are halved, while one-hot scales linearly with the number of bits, see 6.3a for a detailed analysis. In addition, data-based codes find the more discriminative partitions, resulting in better accuracy than the rest of the encodings. Moreover, keeping the raw values of the eigenvectors provides additional information about the likelihood of a metaclass to be present in a certain class, resulting in more robust predictions. Since output codes are based on binary partitions, they constrain the learning so that features are encoded to fall into hyperplanes.

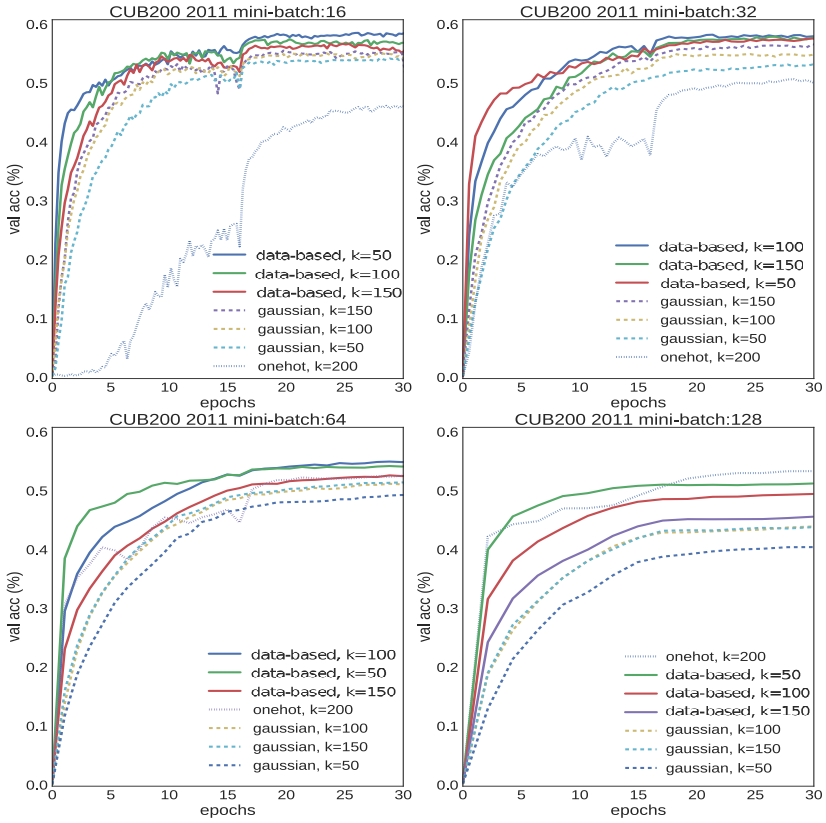


Figure 6.5: **Validation accuracy on CUB200.** Plots have been generated for different mini-batch sizes. (a) when mini-batch size is 16, the performance of one-hot encoding is dramatically reduced

In figure 6.4 we show the 2D projection of those hyperplanes using t-sne. Note the higher overlapping of samples from different classes displayed on the target embedding space of 1-hot in comparison to dense and data-dependent alternatives. In particular, the proposed eigendecomposition of the output space shows a more discriminative splitting of the data samples according to their labels.

CUB-200. Figure 6.5 shows that using small mini-batch sizes with data-based encodings largely outperforms the one-hot baseline for different code lengths when training a CNN on CUB-200 with data-dependent codes based on the raw eigenvalues of the class similarity matrix (best setting on CIFAR100). Moreover, in figure 6.3b,

it can be seen that the data-based code matches the one-hot encoding with just the 25% of the bits. As expected, the first bits correspond to the most discriminative partitions ordered by cut cost. The class similarity matrix was built with the `fc7` outputs of a pre-trained network, but any other would also work if it reflects the inter-class relationships.

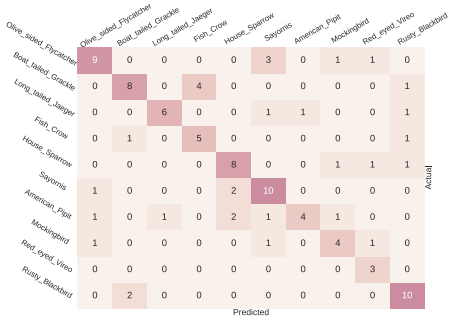
Figure 6.6 contains the confusion matrices for ten of the CUB-200 classes. Note that data-dependent encodings find low cost partitions, discriminating classes prone to be confused in the first stages of the hierarchy (the first encoding bits), and keeping those harder classification problems to the leafs. A comparison of one-hot, random and data-dependent encodings for the classification of "Fish crow" and "Grackle" is shown in figure 6.8.

We lastly verify the correspondence of the metaclasses found with data-dependent encodings by computing the Pearson Correlation Coefficient (CCP) between the columns of the code-matrix and the attributes associated to each of the CUB-200 classes, see table 6.2.

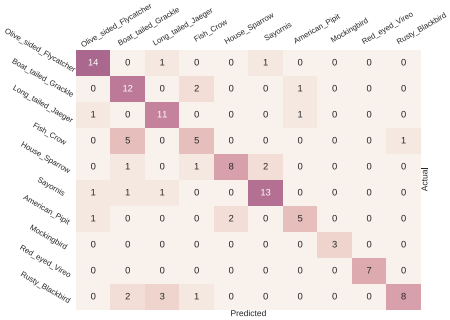
As expected, the data-dependent code finds a high-level partition that already discriminates both classes. One-hot, instead acts directly at the class level, without being explicitly based on shared attributes. On the other hand, random codes, although also based on metaclasses (attributes), do not guarantee that those metaclasses are the most discriminative ones.

6.5 Discussion

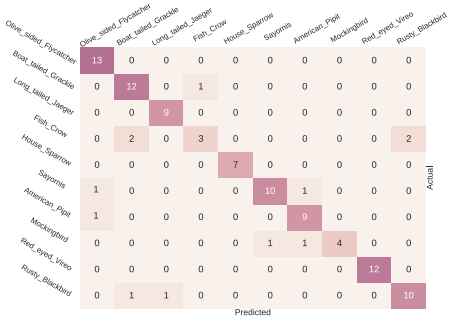
In this work, output codes are integrated with the training of deep CNNs on large-scale datasets. We found that CNNs trained on CIFAR-100, CUB200, Imagenet, and MIT Places using our approach show less sparsity at the output neurons. As a result, models trained with our approach showed more robust gradient estimates and faster convergence rates than those trained with the prevalent one-hot encoding at a small cost, especially for huge label spaces. As a side effect, CNNs trained with our approach can use smaller minibatch sizes, lowering the memory consumption. Moreover, we showed that training with data-dependent codes based on eigenrepresentations of the class space allows for more efficient, hierarchical representations, achieving lower error rates than those trained with data-independent output codes.



(a) One-hot encoding.



(b) Random encoding.



(c) Data-dependent code.

Figure 6.6: **Confusion matrices on CUB200-2011.** Alexnet trained with random codes sampled from a normal distribution (b) already advantage those trained with one-hot encoding (a) e.g., reducing the number of confusions of "Olive sided Flycatcher" with the rest of the classes. Moreover, data-dependent codes based on eigenrepresentations of the output space (d), can better discriminate even more classes, like "boat tailed Grackle" from "Fish crow". Samples for the classes in the confusion matrices are shown in figure 6.7.

Figure 6.7: **Confusion matrix classes.**

Table 6.2: **Top CUB200 attributes by correlation with the code.** Random codes do not show relevant correlations with the data attributes, while data-dependent codes are visibly correlated with the attributes. Concretely, the first bit of the code, i.e. the partition with the lowest cut cost, is highly correlated with shape and size attributes (0.79). The sign of the PPC indicates the expected side of the bi-partition associated for the attribute. As expected, the PPC coefficient decreases in absolute value as the cut cost increases, since higher bits correspond to increasingly difficult partitions.

Code bit	**1**	**2**	**3**	**4**	**5**	**6**
Attribute	Belly-color red	Head-pattern eyeline	breast-color blue	bill-color green	head-pattern unique	crown-color yellow
PCC	0.18	0.16	0.15	0.15	0.14	0.14
Attribute	Tail-shape rounded	Under-tail-color iridiscent	biill-color brown	belly-color pink	bill-shape all-purpose	tail-shape forked
PCC	-0.22	-0.17	-0.17	-0.16	-0.18	-0.18

(a) Random Code

Code bit	**1**	**2**	**3**	**4**	**5**	**6**
Attribute	shape perching-like	primary-color yellow	back-color black	bill-color black	throat-color yellow	upperpart-color white
PCC	0.79	0.64	0.50	0.44	0.53	0.55
Attribute	size:medium	upper-tail-color brown	wing-color grey	primary-color red	primary-color rufous	belly-color black
PCC	-0.73	-0.56	-0.58	-0.38	-0.42	-0.48

(b) Data-dependent code.

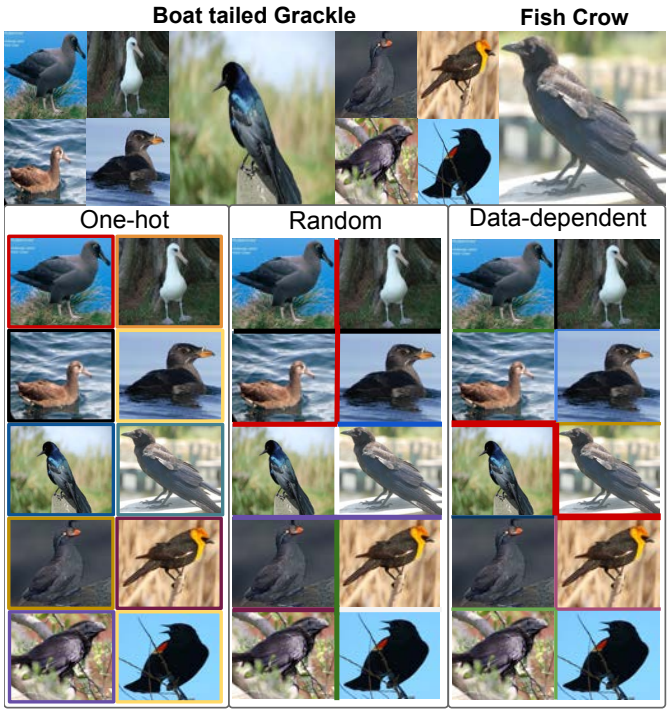


Figure 6.8: **Classifying Boat tailed Grackle and Fish Crow.** One-hot encoding directly assigns labels to each of the examples. Random encoding partitions groups of classes into meta-classes systematically. Data-depending codes first group aquatic and non-aquatic birds, eliminating posterior confusions.

7 Conclusions and Future work

7.1 Conclusions

Fine-grained image recognition is still a challenging problem in computer vision since it involves learning subtle differences between images, while ignoring more obvious variations such as the pose. In this regime, high capacity neural networks risk to memorize intra-class variations caused by pose changes, clutter, and noise, thus overfitting and being inefficient. In fact, fine-grained recognition can be arduous even for humans, which need expert training in many cases (herbology, medicine, mycology, etc), and careful observation. Thus, reaching or surpassing fine-grained human performance has the potential to bring progress across many fields. Motivated the previous reasons, in this PhD dissertation we have tackled the problem of robustness on fine-grained image recognition, showing that proper alignment of the inputs, multiple levels of attention, regularization, and explicit modeling of the output space, results in accurate models, that generalize better, and are more robust to intra-class variation.

Each stage of the fine-grained recognition pipeline (input, model, output) has been analized in a different chapter. In chapter 2, we proposed a deep learning pipeline for pain recognition from facial images. We have shown that removing pose variations on the input images helps recurrent neural networks to isolate the dynamics of facial muscles, yielding better classification performance. On the other hand, we have shown that forcing too much invariance can supersede distortions caused by muscle movement, thus hindering the performance of the models. As a result, the proposed pipeline attains higher pain recognition rates than other state of the art.

In chapter 3, we have proposed an attention mechanism for facial age and gender recognition. This mechanism allow the network to process in detail the most discriminative regions of the image such as skin wrinkles, while ignoring clutter such as eyeglasses. The resulting model is more robust to clutter and deformation, which is inherent to datasets such as facial pictures in the wild, obtaining state-of-the-art performance on three different benchmarks.

In chapter 4, motivated by the benefits of attention seen in 3, we have extended attention from the input to the network feature activations themselves, thus allowing neural networks to focus on informative regions beyond the pixel space at different levels of abstraction. Unlike previous work, the proposed mechanism is modular, architecture independent, fast, efficient, and yet we have empirically proved that when augmented with it, neural networks systematically improve their classification accuracy on seven distinct benchmarks.

In chapter 5, we have focused on the neural network weights themselves, since high-capacity networks can learn over-specialized feature detectors, memorizing certain patterns in the data, and thus overfitting. Therefore we have proposed a new regularization term that enforces feature detectors to be as different as possible, thus overcoming the problem of over-specialization. As a result, models trained with the proposed regularizer attained higher performance than the unregularized counterparts.

Finally, in 6, we have explored the effects of modeling the output space. Concretely, based on the error-correcting framework, we show that using dense output codes to represent classes instead of sparse one-hot vectors results in more robust and efficient learning. Moreover, we have showed that it is possible to adapt the correction capacity to inter-class similarities, assigning more distant codes to the most confused classes.

The take-home message from this work is that the current models used for fine-grained recognition are very sensitive to distractors, deformation, clutter, and easily overfit. Attention and regularization seem promising directions to overcome these problems, as well as proper treatment of the input and the output space.

7.2 Future Perspective

Thanks to the availability of vast amounts of data, image classification systems have reached outstanding performances on popular benchmarks such as the ImageNet [169] and MIT Places [255]. However, these are coarse-grained recognition tasks, for which it is easy to gather information. On the other hand, data is scarcer in the realm of fine-grained recognition (for instance, it is easier to take pictures of "birds" than of "bird-subspecies").

Transfer learning, *i.e.* reusing the knowledge extracted from a large pool of labeled data to classify a different smaller set of labeled samples, is the most common approach to tackle the problem of data scarcity in the literature [256, 257] and in this thesis. For example, in [21] we fine-tuned a model pre-trained with millions of facial attributes on a smaller dataset for pain recognition. However, a major challenge for transfer learning is the adaptation to new domains while retaining

the performance on the original domain. This is an important problem because training deep learning models is computationally expensive, and leveraging information obtained from other domains might help them improve performance with fewer data. This is very different from biological systems, which quickly learn even from single exposures (for instance the location of food), and generalize well on a wide range of tasks and domains, reusing previous knowledge without forgetting what was already learned. Furthermore, it would be compelling that already-trained models learn *online*, at the same time they are solving the target tasks.

Therefore, a promising research direction is continuous improvement and adaptation of deep learning models to new domains and tasks, avoiding *catastrophic forgetting* [258], *i.e.* unlearning previously-learned information. A biology-inspired promising direction to tackle this problem is memory [259, 260], since it directly deals with the problem of long-term information storage and recall. Another interesting direction for continuous learning of new tasks without forgetting is the one presented in [261], *i.e.* to keep a minimal amount of exemplars for the original classes, and ensure that the model predictions after being updated resemble those of the original one (knowledge *distillation*).

Despite fine-tuning requires less data than training from scratch, many examples are still required to attain low classification error rates. Hence, few-shot learning has emerged as an attractive alternative to achieve transfer learning from very few examples (one per class in the case of one-shot learning) [262, 263]. However, current few-shot approaches require huge amounts of labeled examples during training time in order to generalize well. Moreover, these methods can only classify small numbers of classes, usually between 5 and 20 [262, 263], and their performances are still far from fully supervised neural networks [185]. Hence, it would be interesting to scale up these models to work with an arbitrary number of classes and to leverage the availability of large amounts of unlabeled data to train them.

Few-shot learning can be framed as a meta-learning problem [264], where a model is adapted to solve different tasks. This allows to aggregate data from various fine-grained tasks to learn a single model, but it fails when those tasks are very different from each other, thus requiring specialized algorithms to reduce the impact of task disparity [265]. In this realm, modularity [266, 267] is another encouraging research direction, since it allows sharing parameters across tasks, while also learning specialized modules.

Finally, a different problem of current fine-grained recognition systems is that CNNs fail to model the relative position of the object parts, which makes them vulnerable noise and adversarial attacks [8]. Capsule neural networks have been recently proposed as a possible solution to this problem [268] since they use vectors to represent single-neuron activations and thus, they can encode the presence of a feature in the vector magnitude and its pose in the vector direction.

7.3 Scientific Articles

This PhD dissertation has led to the following publications:

7.3.1 Journals

- **Pau Rodríguez**, Miguel A Bautista, Jordi Gonzàlez, and Sergio Escalera. Beyond one-hot encoding: Lower dimensional target embedding. *IMAVIS*, 75: 21–31, 2018
- **Pau Rodríguez**, Guillem Cucurull, Josep M Gonfaus, F Xavier Roca, and Jordi Gonzalez. Age and gender recognition in the wild with deep attention. *PR*, 2017
- **Pau Rodríguez**, Guillem Cucurull, Jordi Gonzalez, Josep M Gonfaus, Kamal Nasrollahi, Thomas B Moeslund, and F Xavier Roca. Deep pain: Exploiting long short-term memory networks for facial expression classification. *IEEE cybernetics*, (99):1–11, 2017
- Farhood Negin, **Pau Rodríguez**, Michal Koperski, Adlen Kerboua, Jordi Gonzàlez, Jeremy Bourgeois, Emmanuelle Chapoulie, Philippe Robert, and Francois Bremond. Praxis: Towards automatic cognitive assessment using gesture recognition. *Expert Systems with Applications*, 106:21–35, 2018

7.3.2 International Conferences and Workshops

- **Pau Rodríguez**, Josep M Gonfaus, Guillem Cucurull, F Xavier Roca, and Jordi Gonzalez. Attend and rectify: a gated attention mechanism for fine-grained recovery. In *ECCV*, pages 349–364, 2018
- **Pau Rodríguez**, Jordi Gonzalez, Guillem Cucurull, Josep M Gonfaus, and Xavier Roca. Regularizing cnns with locally constrained decorrelations. In *ICLR*, 2017
- Boris N Oreshkin, **Pau Rodríguez**, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. In *NeurIPS*, 2018
- Marco Bellantonio, Mohammad A Haque, **Pau Rodríguez**, Kamal Nasrollahi, Taisi Telve, Sergio Escalera, Jordi Gonzalez, Thomas B Moeslund, Pejman Rasti, and Gholamreza Anbarjafari. Spatio-temporal pain recognition in cnn-based super-resolved facial images. In *ICPR FFER Workshop*, pages 151–162. Springer, 2016

7.4 Contributed Code

- **Attend and Rectify**:code to reproduce the results presented in [24] within the PyTorch framework. https://github.com/prlz77/attend-and-rectify
- **Tiny-dnn**: a header only, dependency-free deep learning framework in C++14. Contributed with LSTM and recurrent cells during the Google Summer of Code. https://github.com/tiny-dnn/tiny-dnn
- **LSTM on CNN**: code to reproduce [21] using caffe and Torch. https://github.com/prlz77/LSTM-on-CNN
- **Orthoreg**: code to reproduce the results presented in [23] within the Torch framework. https://github.com/prlz77/orthoreg

7.5 Scientific Dissemination

7.5.1 Invited Talks

- *Patrones Biométricos: cómo saber la edad y género con la IA*, at Libary Living Lab, Volpelleres, Barcelona, Spain, 2018
- *TADAM*, at DLBCN, Barcelona, Spain, 2018
- *Computer Vision and AI*, at Parc de Recerca, UAB, Bellaterra, Spain, 2018
- *Computer Vision and AI*, at AI4ALL, UAB, Bellaterra, Spain, 2018
- *Fine-grained image recognition in the wild*, at Element AI, Montreal, Canada, 2018
- *How do Machines Learn?*, at CosmoCaixa, Barcelona, Spain, 2018
- *Deep Learning 101*, at Vall d'Hebron Research Institute, Barcelona, Spain, 2015

7.5.2 In the Media

- *El teléfono se queda con tu cara*, La Vanguardia, September, 2017

Bibliography

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, pages 1097–1105, 2012.

[2] 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.

[3] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *ICCV*, pages 2758–2766, 2015.

[4] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, pages 2366–2374, 2014.

[5] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, pages 2672–2680, 2014.

[7] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.

[8] 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.

[9] Bo Zhao, Jiashi Feng, Xiao Wu, and Shuicheng Yan. A survey on deep learning-based fine-grained object classification and semantic segmentation. *IJAC*, 14 (2):119–135, 2017.

[10] Kun Duan, Devi Parikh, David Crandall, and Kristen Grauman. Discovering localized attributes for fine-grained recognition. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3474–3481. IEEE, 2012.

[11] Thomas Berg and Peter Belhumeur. Poof: Part-based one-vs.-one features for fine-grained categorization, face verification, and attribute estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 955–962, 2013.

[12] Yaniv Taigman, Lior Wolf, Tal Hassner, et al. Multiple one-shots for utilizing class label information. In *BMVC*, volume 2, pages 1–12, 2009.

[13] Neeraj Kumar, Alexander Berg, Peter N Belhumeur, and Shree Nayar. Describable visual attributes for face verification and image search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(10):1962–1977, 2011.

[14] Ning Zhang, Jeff Donahue, Ross Girshick, and Trevor Darrell. Part-based r-cnns for fine-grained category detection. In *ECCV*, pages 834–849. Springer, 2014.

[15] Tianjun Xiao, Yichong Xu, Kuiyuan Yang, Jiaxing Zhang, Yuxin Peng, and Zheng Zhang. The application of two-level attention models in deep convolutional neural network for fine-grained image classification. In *CVPR*, pages 842–850, 2015.

[16] Jianlong Fu, Heliang Zheng, and Tao Mei. Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. In *CVPR*, volume 2, page 3, 2017.

[17] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *NeurIPS*, pages 2017–2025, 2015.

[18] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3156–3164, 2017.

[19] ZongYuan Ge, Christopher McCool, Conrad Sanderson, and Peter Corke. Subset feature learning for fine-grained category classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 46–52, 2015.

[20] ZongYuan Ge, Alex Bewley, Christopher McCool, Peter Corke, Ben Upcroft, and Conrad Sanderson. Fine-grained classification via mixture of deep convolutional neural networks. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pages 1–6. IEEE, 2016.

[21] **Pau Rodríguez**, Guillem Cucurull, Jordi Gonzalez, Josep M Gonfaus, Kamal Nasrollahi, Thomas B Moeslund, and F Xavier Roca. Deep pain: Exploiting long short-term memory networks for facial expression classification. *IEEE cybernetics*, (99):1–11, 2017.

[22] **Pau Rodríguez**, Guillem Cucurull, Josep M Gonfaus, F Xavier Roca, and Jordi Gonzalez. Age and gender recognition in the wild with deep attention. *PR*, 2017.

[23] **Pau Rodríguez**, Jordi Gonzalez, Guillem Cucurull, Josep M Gonfaus, and Xavier Roca. Regularizing cnns with locally constrained decorrelations. In *ICLR*, 2017.

[24] **Pau Rodríguez**, Josep M Gonfaus, Guillem Cucurull, F Xavier Roca, and Jordi Gonzalez. Attend and rectify: a gated attention mechanism for fine-grained recovery. In *ECCV*, pages 349–364, 2018.

[25] **Pau Rodríguez**, Miguel A Bautista, Jordi Gonzàlez, and Sergio Escalera. Beyond one-hot encoding: Lower dimensional target embedding. *IMAVIS*, 75: 21–31, 2018.

[26] Atul Gawande. *The Checklist Manifesto: How to Get Things Right*. Macmillan, 2010. ISBN 978-1-4299-5338-2.

[27] Girish P. Joshi and Babatunde O. Ogunnaike. Consequences of inadequate postoperative pain relief and chronic persistent postoperative pain. *Anesthesiology Clinics of North America*, 23(1):21–36, 2005-03.

[28] K. O. Anderson, T. R. Mendoza, V. Valero, S. P. Richman, C. Russell, J. Hurley, C. DeLeon, P. Washington, G. Palos, R. Payne, and C. S. Cleeland. Minority cancer patients and their providers: pain management attitudes and practice. *Cancer*, 88(8):1929–1938, 2000-04-15.

[29] J. A. Encandela. Social science and the study of pain since zborowski: a need for a new agenda. *Social Science & Medicine (1982)*, 36(6):783–791, 1993-03.

[30] Justin E. Brown, Neil Chatterjee, Jarred Younger, and Sean Mackey. Towards a physiology-based measure of pain: patterns of human brain activity distinguish painful from non-painful thermal stimulation. *PloS One*, 6(9):e24124, 2011.

[31] Kenneth D. Craig, Kenneth M. Prkachin, and V. E. The facial expression of pain. In D. C. Turk and R. Melzack, editors, *Handbook of pain assessment*, pages 257–276. Guilford Press, 1992.

[32] Behnood Gholami, Wassim M. Haddad, and Allen R. Tannenbaum. Agitation and pain assessment using digital imaging. *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2009:2176–2179, 2009.

[33] Gwen C. Littlewort, Marian Stewart Bartlett, and Kang Lee. Automatic coding of facial expressions displayed during posed and genuine pain. *Image and Vision Computing*, 27(12):1797–1803, 2009-11.

[34] Ahmed Bilal 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. *Image and Vision Computing*, 27(12):1788–1796, 2009-11.

[35] Sebastian Kaltwang, Ognjen Rudovic, and Maja Pantic. Continuous pain intensity estimation from facial expressions. In *Advances in Visual Computing*, number 7432 in Lecture Notes in Computer Science, pages 368–377. Springer Berlin Heidelberg, 2012-07-16.

[36] P. Lucey, J.F. Cohn, K.M. Prkachin, P.E. Solomon, and I. Matthews. Painful data: The UNBC-McMaster shoulder pain expression archive database. In *2011 IEEE International Conference on Automatic Face Gesture Recognition and Workshops (FG 2011)*, pages 57–64, 2011-03.

[37] Kenneth M. Prkachin and Patricia E. Solomon. The structure, reliability and validity of pain expression: evidence from patients with shoulder pain. *Pain*, 139(2):267–274, 2008-10-15.

[38] Paul Ekman, Wallace V. Friesen, and Joseph C. Hager. *Facial Action Coding System - The Manual on CD-ROM*. A Human Face, 2nd edition edition, 2002.

[39] Timothy F. Cootes, Gareth J. Edwards, and Christopher J. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685, 2001.

[40] Ognjen Rudovic, Vladimir Pavlovic, and Maja Pantic. Context-sensitive dynamic ordinal regression for intensity estimation of facial action units. *IEEE transactions on pattern analysis and machine intelligence*, 37(5):944–958, 2015.

[41] Minyoung Kim and Vladimir Pavlovic. Structured output ordinal regression for dynamic facial emotion intensity prediction. In *European Conference on Computer Vision*, pages 649–662. Springer, 2010.

[42] Michel F Valstar and Maja Pantic. Fully automatic recognition of the temporal phases of facial actions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(1):28–43, 2012.

[43] Ognjen Rudovic, Vladimir Pavlovic, and Maja Pantic. Multi-output laplacian dynamic ordinal regression for facial expression recognition and intensity estimation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2634–2641. IEEE, 2012.

[44] Zuheng Ming, Aurélie Bugeau, Jean-Luc Rouas, and Takaaki Shochi. Facial action units intensity estimation by the fusion of features with multi-kernel support vector machine. In *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*, volume 6, pages 1–6. IEEE, 2015.

[45] Ognjen Rudovic, Vladimir Pavlovic, and Maja Pantic. Automatic pain intensity estimation with heteroscedastic conditional ordinal random fields. In *International Symposium on Visual Computing*, pages 234–243. Springer, 2013.

[46] K. Sikka, A. Dhall, and M. Bartlett. Weakly supervised pain localization using multiple instance learning. In *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, pages 1–8, 2013-04.

[47] Karan Sikka, Abhinav Dhall, and Marian Stewart Bartlett. Classification and weakly supervised pain localization using multiple segment representation. *Image and Vision Computing*, 32(10):659–670, 2014-10.

[48] R.A. Khan, A. Meyer, H. Konik, and S. Bouakaz. Pain detection through shape and appearance features. In *2013 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2013-07.

[49] Z. Zafar and N.A. Khan. Pain intensity evaluation through facial action units. In *2014 22nd International Conference on Pattern Recognition (ICPR)*, pages 4696–4701, 2014-08.

[50] 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. In *CVPR Workshops*, pages 88–95, 2015.

[51] R. Irani, K. Nasrollahi, and T.B. Moeslund. Pain recognition using spatiotemporal oriented energy of facial muscles. In *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 80–87, 2015-06.

[52] Liliana Presti and Marco Cascia. Using hankel matrices for dynamics-based facial emotion recognition and pain detection. In *CVPR Workshops*, pages 26–33, 2015.

[53] Henrik Pedersen. Learning appearance features for pain detection using the UNBC-McMaster shoulder pain expression archive database. In Lazaros Nalpantidis, Volker Krüger, Jan-Olof Eklundh, and Antonios Gasteratos, editors, *Computer Vision Systems*, number 9163 in Lecture Notes in Computer Science, pages 128–136. Springer International Publishing, 2015-07-06.

[54] N. Neshov and A. Manolova. Pain detection from facial characteristics using supervised descent method. In *2015 IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, volume 1, pages 251–256, 2015-09.

[55] Neeru Rathee and Dinesh Ganotra. A novel approach for pain intensity detection based on facial feature deformations. *Journal of Visual Communication and Image Representation*, 33:247–254, 2015-11.

[56] Fred L. Bookstein. Principal warps: thin-plate splines and the decomposition of deformations. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 11(6):567–585, Jun 1989.

[57] Rui Zhao, Quan Gan, Shangfei Wang, and Qiang Ji. Facial expression intensity estimation using ordinal information. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3466–3474, 2016.

[58] Yoshua Bengio. Learning deep architectures for AI. *Found. Trends Mach. Learn.*, 2(1):1–127, 2009-01. ISSN 1935-8237.

[59] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. In *NeurIPS*, pages 1106–1114, 2012.

[60] Yoshua Bengio. Deep learning of representations: Looking forward. *arXiv:1305.0445 [cs]*, 2013-05-02.

[61] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *arXiv:1409.4842 [cs]*, 2014-09-16.

[62] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. URL http://arxiv.org/abs/1409.1556.

[63] Jing Zhou, Xiaopeng Hong, Fei Su, and Guoying Zhao. Recurrent convolutional neural network regression for continuous pain intensity estimation in video. In *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2016, Las Vegas, NV, USA, June 26 - July 1, 2016*, pages 1535–1543, 2016. doi: 10.1109/CVPRW.2016.191. URL http://dx.doi.org/10.1109/CVPRW.2016.191.

[64] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *BMVC*, 2015.

[65] S Hochreiter and J Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997-11.

[66] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. *Neural computation*, 12(10):2451–2471, 2000.

[67] Patrick Lucey, Jeffrey F Cohn, Takeo Kanade, Jason Saragih, Zara Ambadar, and Iain Matthews. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops*, pages 94–101. IEEE, 2010.

[68] Lin Zhong, Qingshan Liu, Peng Yang, Bo Liu, Junzhou Huang, and Dimitris N Metaxas. Learning active facial patches for expression analysis. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2562–2569. IEEE, 2012.

[69] Mengyi Liu, Shaoxin Li, Shiguang Shan, Ruiping Wang, and Xilin Chen. Deeply learning deformable facial action parts model for dynamic expression analysis. In *Asian Conference on Computer Vision*, pages 143–157. Springer, 2014.

[70] Ping Liu, Shizhong Han, Zibo Meng, and Yan Tong. Facial expression recognition via a boosted deep belief network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1805–1812, 2014.

[71] Heechul Jung, Sihaeng Lee, Sunjeong Park, Injae Lee, Chunghyun Ahn, and Junmo Kim. Deep temporal appearance-geometry network for facial expression recognition. *CoRR*, abs/1503.01532, 2015. URL http://arxiv.org/abs/1503.01532.

[72] 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. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 41 (3):664–674, 2011.

[73] J. C. Gower. Generalized procrustes analysis. *Psychometrika*, 40(1):33–51, 1975-03.

[74] L.A. Jeni, J.F. Cohn, and F. De la Torre. Facing imbalanced data–recommendations for the use of performance metrics. In *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 245–251, 2013-09.

[75] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[76] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986-10-09.

[77] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. IEEE, 2009.

[78] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3D convolutional neural networks for human action recognition. *TPAMI*, 35(1):221–231, 2013.

[79] Dan Ciresan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. In *CVPR*, pages 3642–3649. IEEE, 2012.

[80] Sachin Sudhakar Farfade, Mohammad J Saberian, and Li-Jia Li. Multi-view face detection using deep convolutional neural networks. In *ICMR*, pages 643–650. ACM, 2015.

[81] P. J. Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990-10.

[82] Sepp Hochreiter, Yoshua Bengio, and Paolo Frasconi. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In J. Kolen and S. Kremer, editors, *Field Guide to Dynamical Recurrent Networks*. IEEE Press, 2001.

[83] Patrick Lucey, Jeffrey F Cohn, Simon Lucey, Sridha Sridharan, and Kenneth M Prkachin. Automatically detecting action units from faces of pain: Comparing shape and appearance features. In *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on*, pages 12–18. IEEE, 2009.

[84] Patrick Lucey, Jeffrey Cohn, Simon Lucey, Iain Matthews, Sridha Sridharan, and Kenneth M Prkachin. Automatically detecting pain using facial actions. In *Affective Computing and Intelligent Interaction and Workshops, 2009. ACII 2009. 3rd International Conference on*, pages 1–8. IEEE, 2009.

[85] Corneliu Florea, Laura Florea, and Constantin Vertan. Learning pain from emotion: transferred hot data representation for pain intensity estimation. In *Computer Vision-ECCV 2014 Workshops*, pages 778–790. Springer, 2014.

[86] Zakia Hammal and Jeffrey F Cohn. Automatic detection of pain intensity. In *ICMI*, pages 47–52. ACM, 2012.

[87] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[88] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Proceedings of the International Conference on Learning Representations (ICLR) Workshops*, 2014.

[89] Takeo Kanade, Jeffrey F Cohn, and Yingli Tian. Comprehensive database for facial expression analysis. In *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, pages 46–53. IEEE, 2000.

[90] Xiangyun Zhao, Xiaodan Liang, Luoqi Liu, Teng Li, Yugang Han, Nuno Vasconcelos, and Shuicheng Yan. Peak-piloted deep network for facial expression recognition. In *European Conference on Computer Vision*, pages 425–442. Springer, 2016.

[91] Karan Sikka, Gaurav Sharma, and Marian Bartlett. Lomo: Latent ordinal model for facial analysis in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5580–5589, 2016.

[92] Mengyi Liu, Shaoxin Li, Shiguang Shan, and Xilin Chen. Au-aware deep networks for facial expression recognition. In *Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on*, pages 1–6. IEEE, 2013.

[93] Ali Mollahosseini, David Chan, and Mohammad H Mahoor. Going deeper in facial expression recognition using deep neural networks. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–10. IEEE, 2016.

[94] Mengyi Liu, Shiguang Shan, Ruiping Wang, and Xilin Chen. Learning expressionlets on spatio-temporal manifold for dynamic facial expression recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1749–1756, 2014.

[95] Antitza Dantcheva, Petros Elia, and Arun Ross. What else does your biometric data reveal? a survey on soft biometrics. *Information Forensics and Security, IEEE Transactions on*, 11(3):441–467, 2015.

[96] Lacey Best-Rowden, Hu Han, Christina Otto, Brendan F Klare, and Anubhav K Jain. Unconstrained face recognition: Identifying a person of interest from a media collection. *Information Forensics and Security, IEEE Transactions on*, 9 (12):2144–2157, 2014.

[97] Javier Orozco, Ognjen Rudovic, Jordi Gonzàlez, and Maja Pantic. Hierarchical on-line appearance-based tracking for 3d head pose, eyebrows, lips, eyelids and irises. *Image and Vision Computing*, 32(1):14–26, 2014.

[98] Hu Han, Christina Otto, and Anubhav K Jain. Age estimation from face images: Human vs. machine performance. In *Biometrics (ICB), 2013 International Conference on*, pages 1–8. IEEE, 2013.

[99] Juan E Tapia and Claudio A Perez. Gender classification based on fusion of different spatial scale features selected by mutual information from histogram of lbp, intensity, and shape. *IEEE Transactions on Information Forensics and Security*, 8(3):488–499, 2013.

[100] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lars Wolf. Deepface: Closing the gap to human-level performance in face verification. In *CVPR*, pages 1701–1708. IEEE, 2014.

[101] Rasmus Rothe, Radu Timofte, and Luc Van Gool. Dex: Deep expectation of apparent age from a single image. In *ICCV Workshops*, pages 10–15, 2015.

[102] Xin Liu, Shaoxin Li, Meina Kan, Jie Zhang, Shuzhe Wu, Wenxian Liu, Hu Han, Shiguang Shan, and Xilin Chen. Agenet: Deeply learned regressor and classifier for robust apparent age estimation. In *ICCV Workshops*, pages 16–24, 2015.

[103] Zhanghui Kuang, Chen Huang, and Wei Zhang. Deeply learned rich coding for cross-dataset facial age estimation. In *ICCV Workshops*, pages 96–101, 2015.

[104] Kaipeng Zhang, Lianzhi Tan, Zhifeng Li, and Yu Qiao. Gender and smile classification using deep convolutional neural networks. In *CVPR Workshops*, pages 34–38, 2016.

[105] G. Levi and T. Hassncer. Age and gender classification using convolutional neural networks. In *CVPR Workshops*, pages 34–42, June 2015.

[106] Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. Recurrent models of visual attention. In *NeurIPS*, pages 2204–2212, 2014.

[107] Eran Eidinger, Roee Enbar, and Tal Hassner. Age and gender estimation of unfiltered faces. *TIFS*, 9(12):2170–2179, 2014.

[108] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.

[109] A. Gallagher and T. Chen. Understanding images of groups of people. In *Proc. CVPR*, 2009.

[110] Karl Ricanek and Tamirat Tesafaye. Morph: A longitudinal image database of normal adult age-progression. In *Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on*, pages 341–345. IEEE, 2006.

[111] Young Ho Kwon and Niels Da Vitoria Lobo. Age classification from facial images. In *CVPR*, pages 762–767. IEEE, 1994.

[112] Guodong Guo and Guowang Mu. Joint estimation of age, gender and ethnicity: Cca vs. pls. In *Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on*, pages 1–6. IEEE, 2013.

[113] Guodong Guo, Guowang Mu, Yun Fu, Charles Dyer, and Thomas Huang. A study on automatic age estimation using a large database. In *ICCV*, pages 1986–1991. IEEE, 2009.

[114] Hu Han, Charles Otto, Xiaoming Liu, and Anil K Jain. Demographic estimation from face images: Human vs. machine performance. *TPAMI*, 37(6):1148–1161, 2015.

[115] Tsuneo Kanno, Masakazu Akiba, Yasuaki Teramachi, Hiroshi Nagahashi, and AGUI Takeshi. Classification of age group based on facial images of young males by using neural networks. *IEICE TRANSACTIONS on Information and Systems*, 84(8):1094–1101, 2001.

[116] K. B. Raja K Ramesha. Feature extraction based face recognition, gender and age classification. *International Journal on Computer Science and Engineering*, 2(1), 2010. ISSN 2278-3091.

[117] Xiaolong Wang, Rui Guo, and Chandra Kambhamettu. Deeply-learned feature for age estimation. In *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*, pages 534–541. IEEE, 2015.

[118] Dong Yi, Zhen Lei, and Stan Z Li. Age estimation by multi-scale convolutional network. In *Computer Vision–ACCV 2014*, pages 144–158. Springer, 2015.

[119] Ivan Huerta, Carles Fernández, Carlos Segura, Javier Hernando, and Andrea Prati. A deep analysis on age estimation. *Pattern Recognition Letters*, 68: 239–249, 2015.

[120] Jordi Mansanet, Alberto Albiol, and Roberto Paredes. Local deep neural networks for gender recognition. *Pattern Recognition Letters*, 70:80–86, 2016.

[121] Gokhan Ozbulak, Yusuf Aytar, and Hazim Kemal Ekenel. How transferable are cnn-based features for age and gender classification? In *BIOSIG*, pages 1–6. IEEE, 2016.

[122] Rasmus Rothe, Radu Timofte, and Luc Van Gool. Deep expectation of real and apparent age from a single image without facial landmarks. *IJCV*, pages 1–14, 2016.

[123] Jun-Cheng Chen, Amit Kumar, Rajeev Ranjan, Vishal M Patel, Azadeh Alavi, and Rama Chellappa. A cascaded convolutional neural network for age estimation of unconstrained faces. In *Biometrics Theory, Applications and Systems (BTAS), 2016 IEEE 8th International Conference on*, pages 1–8. IEEE, 2016.

[124] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation by joint identification-verification. *CoRR*, abs/1406.4773, 2014.

[125] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[126] Beatrice A Golomb, David T Lawrence, and Terrence J Sejnowski. Sexnet: A neural network identifies sex from human faces. In *NeurIPS*, volume 1, page 2, 1990.

[127] Ihsan Ullah, Muhammad Hussain, Ghulam Muhammad, Hatim Aboalsamh, George Bebis, and Anwar M Mirza. Gender recognition from face images with local wld descriptor. In *Systems, Signals and Image Processing (IWSSIP), 2012 19th International Conference on*, pages 417–420. IEEE, 2012.

[128] Caifeng Shan. Learning local binary patterns for gender classification on real-world face images. *Pattern Recognition Letters*, 33(4):431–437, 2012.

[129] H Han and A Jain. Age, gender and race estimation from unconstrained face images. *Dept. Comput. Sci. Eng., Michigan State Univ., East Lansing, MI, USA, MSU Tech. Rep.(MSU-CSE-14-5)*, 2014.

[130] Jos van de Wolfshaar, Mahir F Karaaba, and Marco A Wiering. Deep convolutional neural networks and support vector machines for gender recognition. In *Computational Intelligence*, pages 188–195. IEEE, 2015.

[131] Laurent Itti, Christof Koch, Ernst Niebur, et al. A model of saliency-based visual attention for rapid scene analysis. *TPAMI*, 20(11):1254–1259, 1998.

[132] Antonio Torralba, Aude Oliva, Monica S Castelhano, and John M Henderson. Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search. *Psychological review*, 113(4):766, 2006.

[133] Hugo Larochelle and Geoffrey E Hinton. Learning to combine foveal glimpses with a third-order boltzmann machine. In *NeurIPS*, pages 1243–1251, 2010.

[134] Misha Denil, Loris Bazzani, Hugo Larochelle, and Nando de Freitas. Learning where to attend with deep architectures for image tracking. *Neural computation*, 24(8):2151–2184, 2012.

[135] Michael Connolly, Norah Jones, and David Turner. E-learning: a fresh look. *Higher Education Management and Policy*, 18(3):135, 2006.

[136] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[137] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *NeurIPS*, pages 1693–1701, 2015.

[138] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard S Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. *preprint arXiv:1502.03044*, 2015.

[139] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked attention networks for image question answering. *arXiv preprint arXiv:1511.02274*, 2015.

[140] Huijuan Xu and Kate Saenko. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. *arXiv preprint arXiv:1511.05234*, 2015.

[141] Salah Eddine Bekhouche, Abdelkrim Ouafi, Azeddine Benlamoudi, Abdelmalik Taleb-Ahmed, and Abdenour Hadid. Facial age estimation and gender classification using multi level local phase quantization. In *Control, Engineering & Information Technology (CEIT), 2015 3rd International Conference on*, pages 1–4. IEEE, 2015.

[142] Fares Alnajar, Caifeng Shan, Theo Gevers, and Jan-Mark Geusebroek. Learning-based encoding with soft assignment for age estimation under unconstrained imaging conditions. *Image and Vision Computing*, 30(12): 946–953, 2012.

[143] Ehsan Fazl-Ersi, M Esmaeel Mousa-Pasandi, Robert Laganiere, and Maher Awad. Age and gender recognition using informative features of various types. In *Image Processing (ICIP), 2014 IEEE International Conference on*, pages 5891–5895. IEEE, 2014.

[144] Pablo Dago-Casas, Daniel González-Jiménez, Long Long Yu, and José Luis Alba-Castro. Single-and cross-database benchmarks for gender classification under unconstrained settings. In *ICCV Workshops*, pages 2152–2159, 2011.

[145] Modesto Castrillón-Santana, Javier Lorenzo-Navarro, and Enrique Ramón-Balmaseda. On using periocular biometric for gender classification in the wild. *Pattern Recognition Letters*, 2015.

[146] Domingo Mery and Kevin Bowyer. Recognition of facial attributes using adaptive sparse representations of random patches. In *ECCV Workshops*, pages 778–792. Springer, 2014.

[147] M Castrillón-Santana, J Lorenzo-Navarro, and E Ramón-Balmaseda. Descriptors and regions of interest fusion for gender classification in the wild. *arXiv preprint arXiv:1507.06838*, 2015.

[148] Yuan Dong, Yinan Liu, and Shiguo Lian. Automatic age estimation based on deep learning algorithm. *Neurocomputing*, 2015.

[149] Kuang-Yu Chang, Chu-Song Chen, and Yi-Ping Hung. Ordinal hyperplanes ranker with cost sensitivities for age estimation. In *CVPR*, pages 585–592, 2011.

[150] Rasmus Rothe, Radu Timofte, and Luc Van Gool. Some like it hot-visual guidance for preference prediction. In *CVPR*, pages 5553–5561, 2016.

[151] Guodong Guo, Yun Fu, Charles R Dyer, and Thomas S Huang. Image-based human age estimation by manifold learning and locally adjusted robust regression. *Image Processing, IEEE Transactions on*, 17(7):1178–1188, 2008.

[152] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[153] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, volume 9, pages 249–256, 2010.

[154] Changsheng Li, Qingshan Liu, Jing Liu, and Hanqing Lu. Learning ordinal discriminative features for age estimation. In *CVPR*, pages 2570–2577. IEEE, 2012.

[155] Caifeng Shan. Learning local features for age estimation on real-life faces. In *Workshop on Multimodal Pervasive Video Analysis*, pages 23–28. ACM, 2010. ISBN 978-1-4503-0167-1. doi: 10.1145/1878039.1878045.

[156] Juha Ylioinas, Abdenour Hadid, and Matti Pietikainen. Age classification in unconstrained conditions using lbp variants. In *ICPR*, pages 1257–1260, 2012.

[157] Sergio Escalera, Junior Fabian, Pablo Pardo, Xavier Baró, Jordi Gonzalez, Hugo J Escalante, Dusan Misevic, Ulrich Steiner, and Isabelle Guyon. Chalearn looking at people 2015: Apparent age and cultural event recognition datasets and results. In *ICCV Workshops*, pages 1–9, 2015.

[158] John Robert Anderson. *Cognitive psychology and its implications*. New York, NY, US: WH Freeman/Times Books/Henry Holt and Co, 1985.

[159] Robert Desimone and John Duncan. Neural mechanisms of selective visual attention. *Annual review of neuroscience*, 18(1):193–222, 1995.

[160] Sabine Kastner Ungerleider and Leslie G. Mechanisms of visual attention in the human cortex. *Annual review of neuroscience*, 23(1):315–341, 2000.

[161] Pierre Sermanet, Andrea Frome, and Esteban Real. Attention for fine-grained categorization. In *ICLR*, 2015.

[162] Bo Zhao, Xiao Wu, Jiashi Feng, Qiang Peng, and Shuicheng Yan. Diversified visual attention networks for fine-grained object classification. *IEEE Transactions on Multimedia*, 19(6):1245–1256, 2017.

[163] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[164] Saumya Jetley, Nicholas A. Lord, Namhoon Lee, and Philip Torr. Learn to pay attention. In *ICLR*, 2018.

[165] Xiao Liu, Tian Xia, Jiang Wang, and Yuanqing Lin. Fully convolutional attention localization networks: Efficient attention localization for fine-grained recognition. *arXiv preprint arXiv:1603.06765*, 2016.

[166] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[167] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, 2016.

[168] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

[169] Olga Russakovsky, Jia Deng, Jonathan Krause, Alex Berg, and Li Fei-Fei. The imagenet large scale visual recognition challenge 2012 (ilsvrc2012), 2012.

[170] R Uijlings, A van de Sande, Theo Gevers, M Smeulders, et al. Selective search for object recognition. *IJCV*, 104(2):154, 2013.

[171] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[172] Heliang Zheng, Jianlong Fu, Tao Mei, and Jiebo Luo. Learning multi-attention convolutional neural network for fine-grained image recognition. In *ICCV*, 2017.

[173] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017.

[174] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[175] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Fei-Fei Li. Novel dataset for fine-grained image categorization: Stanford dogs. In *FGVC*, volume 2, page 1, 2011.

[176] Y. Matsuda, H. Hoashi, and K. Yanai. Recognition of multiple-food images by detecting candiyear regions. In *ICME*, 2012.

[177] Jingjing Chen and Chong-Wah Ngo. Deep-based ingredient recognition for cooking recipe retrieval. In *ACM MM*, pages 32–41. ACM, 2016.

[178] Hamid Hassannejad, Guido Matrella, Paolo Ciampolini, Ilaria De Munari, Monica Mordonini, and Stefano Cagnoni. Food image recognition using very deep convolutional networks. In *MADIMA Workshop*, pages 41–49. ACM, 2016.

[179] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *CVPR*, pages 554–561, 2013.

[180] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[181] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, pages 1026–1034, 2015.

[182] Adam Paszke, Sam Gross, Soumith Chintala, and Gregory Chanan. Pytorch, 2017.

[183] Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu. Deeply-supervised nets. In Guy Lebanon and S. V. N. Vishwanathan, editors, *AISTATS*, volume 38 of *JMLR Proceedings*, pages 562–570. JMLR.org, 2015.

[184] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, pages 5987–5995. IEEE, 2017.

[185] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, volume 1, page 3, 2017.

[186] Rasmus Rothe, Radu Timofte, and Luc Van Gool. Deep expectation of real and apparent age from a single image without facial landmarks. *IJCV*, July 2016.

[187] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015.

[188] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[189] Steven J. Nowlan and Geoffrey E. Hinton. Simplifying neural networks by soft weight-sharing. *Neural computation*, 4(4):473–493, 1992.

[190] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *JMLR*, 15(1):1929–1958, 2014.

[191] Li Wan, Matthew D Zeiler, Sixin Zhang, Yann L Cun, and Rob Fergus. Regularization of neural networks using dropconnect. In *ICML*, pages 1058–1066, 2013.

[192] Yoshua Bengio and James S Bergstra. Slow, decorrelated features for pretraining complex cell-like networks. In *NeurIPS*, pages 99–107, 2009.

[193] Yebo Bao, Hui Jiang, Lirong Dai, and Cong Liu. Incoherent training of deep neural networks to de-correlate bottleneck features for speech recognition. In *2013 IEEE ICASSP*, pages 6980–6984. IEEE, 2013.

[194] Michael Cogswell, Faruk Ahmed, Ross Girshick, Larry Zitnick, and Dhruv Batra. Reducing overfitting in deep networks by decorrelating representations. *ICLR*, 2016.

[195] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456, 2015.

[196] Dmytro Mishkin and Jiri Matas. All you need is a good init. *ICLR*, 2016.

[197] Andrew M. Saxe, James L. McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv:1312.6120*, December 2013.

[198] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NeurIPS workshop on deep learning and unsupervised feature learning*, volume 2011, page 5, 2011.

[199] Mircea I Chelaru and Valentin Dragoi. Negative correlations in visual cortical networks. *Cerebral Cortex*, 26(1):246–256, 2016.

[200] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, May 2016.

[201] Ian Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Maxout networks. In *ICML*, pages 1319–1327, 2013.

[202] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, March 2014.

[203] Rupesh K. Srivastava, Klaus Greff, and Jürgen Schmidhuber. Training very deep networks. In *NeurIPS*, pages 2368–2376, 2015.

[204] Jost T. Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. In *ICLR (workshop track)*, 2015.

[205] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (ELUs). *ICLR*, 2016.

[206] Benjamin Graham. Fractional max-pooling. *arXiv preprint arXiv:1412.6071*, 2014.

[207] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *ECCV*, pages 646–661. Springer, 2016.

[208] 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 *ECCV*, pages 740–755. Springer, 2014.

[209] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 111(1):98–136, 2015.

[210] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, 2015.

[211] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *NeurIPS*, pages 487–495, 2014.

[212] Sudheendra Vijayanarasimhan, Jonathon Shlens, Rajat Monga, and Jay Yagnik. Deep networks with large output spaces. *arXiv preprint arXiv:1412.7479*, 2014.

[213] Samy Bengio, Jason Weston, and David Grangier. Label embedding trees for large multi-class tasks. In *NeurIPS*, pages 163–171, 2010.

[214] Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid. Label-embedding for image classification. *TPAMI*, 38(7):1425–1438, 2016.

[215] Daniel Hsu, Sham Kakade, John Langford, and Tong Zhang. Multi-label prediction via compressed sensing. In *NeurIPS*, volume 22, pages 772–780, 2009.

[216] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Tomas Mikolov, et al. Devise: A deep visual-semantic embedding model. In *NeurIPS*, pages 2121–2129, 2013.

[217] Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid. Label-embedding for attribute-based classification. In *CVPR*, pages 819–826, 2013.

[218] Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. Large margin methods for structured and interdependent output variables. *JMLR*, 6(Sep):1453–1484, 2005.

[219] Jason Weston, Samy Bengio, and Nicolas Usunier. Large scale image annotation: learning to rank with joint word-image embeddings. *Machine learning*, 81(1):21–35, 2010.

[220] Sergio Escalera, Oriol Pujol, and Petia Radeva. On the decoding process in ternary error-correcting output codes. *TPAMI*, 32(1):120–134, 2010.

[221] Miguel Ángel Bautista, Sergio Escalera, Xavier Baró, Petia Radeva, Jordi Vitriá, and Oriol Pujol. Minimal design of error-correcting output codes. *PRL*, 33(6): 693–702, 2012.

[222] Eun Bae Kong and Thomas G Dietterich. Error-correcting output coding corrects bias and variance. In *ICML*, pages 313–321, 1995.

[223] Kilian Q Weinberger and Olivier Chapelle. Large margin taxonomy embedding for document categorization. In *NeurIPS*, pages 1737–1744, 2009.

[224] Xiaodong Yu and Yiannis Aloimonos. Attribute-based transfer learning for object categorization with zero/one training example. In *ECCV*, pages 127–140. Springer, 2010.

[225] Marcus Rohrbach, Michael Stark, and Bernt Schiele. Evaluating knowledge transfer and zero-shot learning in a large-scale setting. In *CVPR*, pages 1641–1648. IEEE, 2011.

[226] Pichai Kankuekul, Aram Kawewong, Sirinart Tangruamsub, and Osamu Hasegawa. Online incremental attribute-based zero-shot learning. In *CVPR*, pages 3657–3664. IEEE, 2012.

[227] Yonatan Amit, Michael Fink, Nathan Srebro, and Shimon Ullman. Uncovering shared structures in multiclass classification. In *ICML*, pages 17–24. ACM, 2007.

[228] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[229] Jason Weston, Samy Bengio, and Nicolas Usunier. Wsabie: Scaling up to large vocabulary image annotation. 2011.

[230] Nitish Srivastava and Ruslan R Salakhutdinov. Discriminative transfer learning with tree-based priors.

[231] Jia Deng, Nan Ding, Yangqing Jia, Andrea Frome, Kevin Murphy, Samy Bengio, Yuan Li, Hartmut Neven, and Hartwig Adam. Large-scale object classification using label relation graphs. In *ECCV*, pages 48–64. Springer, 2014.

[232] Tianjun Xiao, Jiaxing Zhang, Kuiyuan Yang, Yuxin Peng, and Zheng Zhang. Error-driven incremental learning in deep convolutional neural network for large-scale image classification. In *ACMMM*, pages 177–186. ACM, 2014.

[233] Zhicheng Yan, Hao Zhang, Robinson Piramuthu, Vignesh Jagadeesh, Dennis DeCoste, Wei Di, and Yizhou Yu. Hd-cnn: Hierarchical deep convolutional neural network for large scale visual recognition. In *ICCV*, 2015.

[234] Kevin Lin, Huei-Fang Yang, Jen-Hao Hsiao, and Chu-Song Chen. Deep learning of binary hash codes for fast image retrieval. In *CVPR Workshops*, pages 27–35, 2015.

[235] Xiaolong Bai, Swamidoss Issac Niwas, Weisi Lin, Bing-Feng Ju, Chee Keong Kwoh, Lipo Wang, Chelvin C Sng, Maria C Aquino, and Paul TK Chew. Learning ecoc code matrix for multiclass classification with application to glaucoma diagnosis. *Journal of medical systems*, 40(4):1–10, 2016.

[236] T Windeatt and G Ardeshir. Boosted ecoc ensembles for face recognition. In *IEE conference publication*, pages 165–168. Institution of Electrical Engineers, 2003.

[237] Raymond S Smith and Terry Windeatt. Facial action unit recognition using multi-class classification. *Neurocomputing*, 150:440–448, 2015.

[238] Thomas G. Dietterich and Ghulum Bakiri. Solving multiclass learning problems via error-correcting output codes. *JAIR*, 2:263–286, 1995.

[239] Zhuolin Jiang, Yaming Wang, Larry Davis, Walter Andrews, and Viktor Rozgic. Learning discriminative features via label consistent neural network. In *WACV*, pages 207–216. IEEE, 2017.

[240] Raj Chandra Bose and Dwijendra K Ray-Chaudhuri. On a class of error correcting binary group codes. *Information and control*, 3(1):68–79, 1960.

[241] Erin L Allwein, Robert E Schapire, and Yoram Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. *JMLR*, 1(Dec):113–141, 2000.

[242] Reza Ghaderi and T Windeau. Circular ecoc. a theoretical and experimental analysis. In *ICPR*, volume 2, pages 203–206. IEEE, 2000.

[243] Oriol Pujol, Petia Radeva, and Jordi Vitria. Discriminant ecoc: A heuristic method for application dependent design of error correcting output codes. *TPAMI*, 28(6):1007–1012, 2006.

[244] Sergio Escalera, David MJ Tax, Oriol Pujol, Petia Radeva, and Robert PW Duin. Subclass problem-dependent design for error-correcting output codes. *TPAMI*, 30(6):1041–1054, 2008.

[245] Koby Crammer and Yoram Singer. On the learnability and design of output codes for multiclass problems. *Machine learning*, 47(2-3):201–233, 2002.

[246] Gregory Griffin and Pietro Perona. Learning and using taxonomies for fast visual categorization. In *CVPR*, pages 1–8. IEEE, 2008.

[247] Xiao Zhang, Lin Liang, and Heung-Yeung Shum. Spectral error correcting output codes for efficient multiclass recognition. In *ICCV*, pages 1111–1118. IEEE, 2009.

[248] Huiqun Deng, George Stathopoulos, and Ching Y Suen. Applying error-correcting output coding to enhance convolutional neural network for target detection and pattern recognition. In *ICPR*, pages 4291–4294. IEEE, 2010.

[249] Shuo Yang, Ping Luo, Chen Change Loy, Kenneth W Shum, and Xiaoou Tang. Deep representation learning with target coding. In *AAAI*, pages 3848–3854, 2015.

[250] Yann LeCun and Yoshua Bengio. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10): 1995, 1995.

[251] Trevor Hastie, Robert Tibshirani, et al. Classification by pairwise coupling. *The annals of statistics*, 26(2):451–471, 1998.

[252] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *TPAMI*, 22(8):888–905, 2000.

[253] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-ucsd birds 200. 2010.

[254] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[255] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2018.

[256] Pierre Sermanet, Koray Kavukcuoglu, Soumith Chintala, and Yann LeCun. Pedestrian detection with unsupervised multi-stage feature learning. In *CVPR*, pages 3626–3633, 2013.

[257] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *NeurIPS*, pages 3320–3328, 2014.

[258] Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999.

[259] Thomas Miconi, Jeff Clune, and Kenneth O Stanley. Differentiable plasticity: training plastic neural networks with backpropagation. *arXiv preprint arXiv:1804.02464*, 2018.

[260] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. End-to-end memory networks. In *NeurIPS*, pages 2440–2448, 2015.

[261] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 233–248, 2018.

[262] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *NeurIPS*, pages 3630–3638, 2016.

[263] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, pages 4077–4087, 2017.

[264] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2017.

[265] Boris N Oreshkin, **Pau Rodríguez**, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. In *NeurIPS*, 2018.

[266] Chrisantha Fernando, Dylan Banarse, Charles Blundell, Yori Zwols, David Ha, Andrei A Rusu, Alexander Pritzel, and Daan Wierstra. Pathnet: Evolution channels gradient descent in super neural networks. *arXiv preprint arXiv:1701.08734*, 2017.

[267] Chihiro Watanabe, Kaoru Hiramatsu, and Kunio Kashino. Modular representation of layered neural networks. *Neural Networks*, 97:62–73, 2018.

[268] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems*, pages 3856–3866, 2017.

[269] Farhood Negin, **Pau Rodríguez**, Michal Koperski, Adlen Kerboua, Jordi Gonzàlez, Jeremy Bourgeois, Emmanuelle Chapoulie, Philippe Robert, and Francois Bremond. Praxis: Towards automatic cognitive assessment using gesture recognition. *Expert Systems with Applications*, 106:21–35, 2018.

[270] Marco Bellantonio, Mohammad A Haque, **Pau Rodríguez**, Kamal Nasrollahi, Taisi Telve, Sergio Escalera, Jordi Gonzalez, Thomas B Moeslund, Pejman Rasti, and Gholamreza Anbarjafari. Spatio-temporal pain recognition in cnn-based super-resolved facial images. In *ICPR FFER Workshop*, pages 151–162. Springer, 2016.