Universitat Autònoma de Barcelona

Color in Visual Recognition: from flat to deep representations and some biological parallelisms

Ivet Rafegas Fonoll

Universitat Autònoma
de Barcelona

Color in Visual Recognition: from flat to deep representations and some biological parallelisms

A dissertation submitted by **Ivet Rafegas Fonoll** at Universitat Autònoma de Barcelona to fulfill the degree of **Doctor of Philosophy**.

Bellaterra, September 28, 2017

Director	**Maria Vanrell Martorell** Dpt. de Ciències de la Computació Centre de Visió per Computador Universitat Autònoma de Barcelona
Thesis committee	**Dra. Sophie Wuerger** University of Liverpool Dpt. of Psychological Sciences **Dr. Gustau Camps-Valls** Dpt. d'Enginyeria Electrònica Universtitat de València **Dr. Carlo Gatta** Vintra Inc.
International evaluators	**Dra. Sophie Werger** University of Liverpool Dpt. of Psychological Sciences **Dr. Luís A. Alexandre** Universidade da Beira Interior Full professor

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

ISBN: 978-84-945373-7-0

Printed by Ediciones Gráficas Rey, S.L.

Al meu avi, iaia, pares, al Borja i al Roger ...

The art and science of asking questions
is the source of all knowledge
— Thomas Berger

Acknowledgements

Fer una tesi doctoral és un projecte personal però que alhora no s'aconsegueix sense tenir suport extern. Sense aquest suport ben segur que el desenvolupament d'aquesta tesi no hagués estat possible, i és per això que m'agradaria dedicar unes paraules a tots aquests que m'han ajudat o acompanyat, d'una manera o altra, en aquest camí intens i enriquidor.

En primer lloc, agrair a la Universitat Autònoma de Barcelona, així com també al Centre de Visió per Computador, per haver-me donat l'oportunitat de poder entrar al món de la recerca i formar-me en l'àmbit de la visió per computador que tant em fascina.

Un especial agraïment a la directora de la meva tesi, la Dra. Maria Vanrell. És evident que la seva professionalitat, implicació, supervisió, orientació, idees i la transferència del seu coneixement han estat claus per al desenvolupament d'aquesta tesi. Però no només això, sinó per transmetre'm tota la seva passió per aquest camp. He après molt al seu costat, i tots aquests anys amb ella m'han fet créixer tant a caràcter personal, com professional i cultural. Crec que l'optimisme que sempre m'ha transmès ha compensat els moments en què m'apagava i em donava forces a enfrontar-me als obstacles. Ens vam endinsar en un tema que en aquell moment era emergent i desconeixíem, però que hem acabat enfrontant-nos-hi i guanyant coneixement conjuntament. He gaudit moltíssim. Maria, moltes gràcies!

També agrair al Dr. Robert Benavente i al Dr. Ramon Baldrich. La vostra col•laboració també ha deixat, indiscutiblement, el vostre granet de sorra en aquesta tesi. Gràcies per totes les idees i consells que m'heu fet; com també les crítiques o discussions constructives que ens han fet aprendre cada dia alguna cosa més. Finalment també agrair al Dr. Javier Vázquez, amb qui vaig compartir els meus inicis en la recerca i per sempre transmetre'm la visió de tenir "el got mig ple".

I would also thank Dr. Luís A. Alexandre for giving me the opportunity of collaborating with him in the Universidade da Beira Interior in Portugal. I really appreciate his professionalism as well as all of his suggestions and guide, that made me learn every day. It was a nice experience that I also enjoyed with the company of Joaõ and Miguel in the lab.

Entrant ja en el terreny més personal, vull fer un fort agraïment a la meva família. Pare, mare i Borja, mil gràcies pel vostre suport incondicional, per donar-me consells i forces cada dia, com també per ajudar-me en tot el que he necessitat. Sé que heu valorat molt cada una de les petites coses que he anat fent, us heu alegrat d'allò que m'aportava felicitat i heu intentat comprendre allò que em feia tenir el

cap ben ocupat cada dia. No hi ha res que ompli més com veure que tens sempre a la teva família al teu costat. Estic profundament agraïda per tot el que heu fet per mi. Avi i iaia, gràcies també per mostrar interès i tenir sempre el vostre carinyo. Entre tots vosaltres que m'heu ensenyat que cal intentar lluitar pels objectius i somnis de cadascú, i que si algú pot aconseguir alguna meta, sols cal una mica d'ambició i esforç per a què un mateix pugui aconseguir-ho. Gràcies de tot cor a tots vosaltres, us estimo molt.

I evidentment també agrair a l'Agustí, Carme, a l'Arnau i la Petya. M'heu obert la porta de la vostra casa sempre amb una rialla i heu aconseguit que em sentís una més. A més, m'heu acompanyat durant tota aquesta època de doctorat enviant-me sempre energia i alegria. Carme, gràcies per tenir la paciència d'anar-me resolent dubtes d'anglès. I a tots vosaltres, així com també a la tieta Dolors, donar-vos les gràcies per fer-me sentir recolzada i mostrar que confieu plenament en mi. Gràcies.

Roger. Crec que no tinc paraules per agrair-te la paciència que has tingut amb mi. Recordo com vas animar-me a endinsar-me en aquest camí i has fet que sempre m'hagi sentit acompanyada. Sé que per tu ha estat molt difícil i que has hagut d'adaptar-te en molts aspectes. Però sempre ho has fet amb una bona rialla i ajudant-me amb tot el que he necessitat. Has estat la persona més important aquests darrers anys, gràcies per tot el que hem viscut. Sense tu això sí que no ho hauria aconseguit. Aquest treball és gràcies a tu i junts l'hem fet possible. Gràcies per ser com ets, gràcies per fer-me una millor persona cada dia. Per fer-me tan feliç. T'estimo.

Tampoc pot faltar un agraïment als *#imprescindibles*. Entre tots heu fet que aquesta època hagi estat plena de felicitat i de moments inoblidables. Gràcies per haver-me acceptat, inclòs i trobar-hi sempre els braços oberts. Especialment, mencionar a la Maria C., pel seu optimisme i ànims, a qui també aprofito per donar-li molta energia en el seu projecte de recerca, sé que te'n sortiràs; a la Sílvia, que sempre hi ha sigut per tot i m'ha fet treure una bona rialla; a la Irene, per fer-me pujar l'autoestima; a la Mercè B., que evidentment la seva empenta m'ha fet arribar fins aquí; a la Marta C., per aquells correus durant la meva estada a l'estranger que em feien sentir més a prop vostre; indiscutiblement a la Victòria, no només pel millor dinar de l'estiu, sinó per mostrar-se sempre al meu costat i haver-me ajudat en tantes ocasions (tant a nivell personal com professional); i al Patx, per sempre estar disposat a donar-me un cop de mà amb l'estadística. Mil gràcies!

D'entre els amics també agrair al Quim, dotze anys de sincera amistat compartint infinitat de coses. Quim, també has estat una peça clau que m'ha acompanyat en aquest projecte. Gràcies! Afegir també a la Raquel Guardia per haver cercat un foradet a la seva ocupadíssima agenda i sempre mostrar interès. A la Gina, que m'ha entès més que ningú, a la Marina, que des de la llunyania també m'ha animat i, finalment, un fort agraïment a la Mercè Capdevila, per sempre transmetre tanta

felicitat i encomanar-me-la per a tirar endavant, per la seva preocupació, interès i per donar-me un punt de suport admirable. Moltes gràcies per ser-hi, Mercè.

Per acabar, a tota la família d'amistat trobada al centre amb qui hem compartit bons moments, hem après un dels altres i hem fet que aquesta etapa hagi estat divertida. Carles, gràcies per aguantar-me tantes vegades; Camp, gràcies per tot; Joan, per les nostres converses; Toni, per tants bons moments i suport; David, Jon, Lluís, Fran, Alejandro, Anjan, Sounak, Bojana, German, Ricard, Gisel•la, Chloe i Claire, gràcies per tota aquesta època, companyia i suport.

Abstract

Visual recognition is one of the main problems of computer vision that tries to solve for image understanding. It pursuits to answer the question of what objects are in images. This problem can be computationally solved by using relevant sets of visual features, such as edges, corners, color or more complex object parts. This thesis contributes on how color features have to be represented for recognition tasks.

Image features can be extracted following two different approaches. A first approach is defining handcrafted descriptors of images followed by a learning scheme to classify the content (named flat schemes in [71]. In this approach, perceptual considerations are habitually used to define efficient color features. Here we propose a new flat color descriptor based on the extension of color channels to boost the representation of spatio-chromatic contrast that overcomes state-of-the-art approaches. However, flat schemes present a lack of generality far away from the capabilities of biological systems. A second approach proposes to evolve these flat schemes to a hierarchical process, as it is performed in the visual cortex. This includes an automatic process to learn optimal features. These deep schemes, and more specially Convolutional Neural Networks (CNNs), have shown an impressive performance to solve different vision problems. However, there is a lack of understanding about the internal representation obtained, as a result of the automatic learning. In this thesis we propose a new methodology to explore the internal representation of trained CNNs by defining the Neuron Feature as a visualization of the intrinsic features encoded in each individual neuron. Additionally, and inspired by physiological techniques, we propose to compute different neuron selectivity indexes (*e.g.*, color, class, orientation or symmetry, amongst others) to label and classify the full CNN neuron population to understand learned representations.

Finally, using the proposed methodology, we show an in-depth study on how color is represented on a specific CNN, trained for object recognition, that competes with primate representational abilities [17]. We found several parallelisms with biological visual systems: (a) an important number of color selectivity neurons through all the layers; (b) an opponent and low frequency representation of color oriented edges and a higher sampling of frequency selectivity in brightness than in color in 1st layer like in V1; (c) a higher sampling of color hue in the second layer

aligned to observed hue maps in V2; (d) a strong color and shape entanglement in all layers going from basic features in shallower layers (V1 and V2) to object and background shapes in deeper layers (V4 and IT); and (e) a strong correlation between neuron color selectivities and color dataset bias.

Key words: *computer vision, flat schemes, hierarchical schemes, deep learning, convolutional neural networks, color, selectivity indexes*

Resumen

El reconocimiento visual es uno de los principales problemas de la visión por computador que intenta resolver para la comprensión de imágenes. Persigue responder a la pregunta de qué objetos hay en una imagen. Este problema puede ser resuelto computacionalmente, usando conjuntos de características visuales relevantes como bordes, esquinas, color u otras partes más complejas de los objetos. Esta tesis contribuye en averiguar cómo las características de color tienen que ser representadas para las tareas de reconocimiento.

Las características de las imágenes pueden ser extraídas mediante dos enfoques distintos. Una primera estrategia es definir manualmente descriptores de imágenes y posteriormente usar una técnica de aprendizaje para clasificar el contenido (conocido como esquema llano [71]). En esta estrategia habitualmente se usan consideraciones perceptuales para definir unas características de color eficientes. En esta tesis proponemos un nuevo descriptor de color llano basado en la extensión de canales de color para promover la representación del contraste *espaciocromático* que supera los métodos del estado de arte. No obstante, estos esquemas llanos escasean de generalidad, alejándose así de las capacidades de los sistemas biológicos. Una segunda estrategia propone evolucionar de los esquemas llanos a procesos jerárquicos, tal y como se desempeña en la corteza visual. Además incluye un proceso automático para el aprendizaje de características óptimas. Los esquemas profundos, y especialmente las redes neuronales convolucionales (CNNs), han demostrado un impactante desempeño para solventar distintos problemas visuales. Aun así, se carece de entender las representaciones internas obtenidas como resultado del aprendizaje automático. En esta tesis proponemos una nueva metodología para explorar la representación interna de CNNs entrenadas, mediante la definición de la *Neuron Feature* como visualización de las propiedades intrínsecas codificadas en cada una de las neuronas. De manera adicional, e inspirándonos en técnicas fisiológicas, proponemos obtener diferentes índices de selectividad de las neuronas (por ejemplo, color, clase, orientación o simetría, entre otros) para etiquetar y clasificar la población de neuronas de la CNN y comprender las representaciones aprendidas.

Finalmente, utilizando la metodología propuesta, mostramos un profundo estudio sobre cómo el color es representado en una red específica entrenada para

el reconocimiento de objetos y que compite con las capacidades de representación de los primates [17]. Encontramos diversos paralelismos con los sistemas visuales biológicos: (a) un importante número de neuronas selectivas al color a través de todas las capas; (b) una representación de baja frecuencia y de colores oponentes para bordes de color; mientras que hay una mayor muestra de frecuencias para las neuronas de luminosidad (comparado con las de color) tal y como se efectúa en V1; (c) una mayor representación de tonalidades de color en la segunda capa que se alinea con los mapas de tonos observados en V2; (d) un fuerte vínculo entre las características de color y de forma en todas las capas, yendo des de características básicas en las primeras capas (V1 y V2) hasta formas relacionadas con el objeto y el fondo en capas más profundas (V3 y V4); y (e) una fuerte correlación entre las neuronas selectivas al color y la tendencia de la base de datos.

Palabras clave: *visión por computador, esquemas llanos, estructuras jerárquicas, aprendizaje profundo, redes convolucionales neuronales, color, índices de selectividad*

Resum

El reconeixement visual és un dels principals problemes que la visió per computador que intenta resoldre per a la comprensió d'imatges. Persegueix respondre a la pregunta de quins objectes hi ha en una imatge. Aquest problema pot ser resolt computacionalment, usant conjunts de característiques visuals rellevants com ara fronteres, cantonades, color o altres parts més complexes dels objectes. Aquesta tesi contribueix en esbrinar com les característiques del color han de ser representades per a les tasques de reconeixement.

Les característiques de les imatges poden ser extretes mitjançant dos enfocaments diferents. Una primera estratègia és definir manualment descriptors d'imatges i posteriorment utilitzar una tècnica d'aprenentatge per tal de classificar-ne el contingut (conegut com esquema pla [71]). En aquesta estratègia habitualment s'usen consideracions de percepció per definir unes característiques de color eficients. En aquesta tesi proposem un nou descriptor de color pla basat en la extensió de canals de color per tal de promoure la representació *espaciocromàtica* del contrast que supera els mètodes de l'estat de l'art. No obstant això, aquests esquemes plans flaquegen de generalitat, allunyant-se així de les capacitats dels sistemes biològics. Una segona estratègia proposa evolucionar aquests esquemes plans cap a processos jeràrquics, tal i com es desenvolupa en el còrtex visual. A més, inclou un procés automàtic per l'aprenentatge de característiques òptimes. Els esquemes profunds, i especialment les xarxes neuronals convolucionals (CNNs), han demostrat una impactant capacitat en solucionar diferents problemes visuals. No obstant això, s'escasseja de comprendre les representacions internes obtingudes com a resultat de l'aprenentatge automàtic. En aquesta tesis proposem una nova metodologia per a explorar la representació interna de les CNNs entrenades, mitjançant la definició de la *Neuron Feature* com a visualització de les propietats intrínseques codificades en cada una de les neurones. De manera addicional, i inspirant-nos en tècniques fisiològiques, proposem obtenir diferents índexs de selectivitat de les neurones (per exemple, color, classe, orientació o simetria, entre altres) per tal d'etiquetar i classificar la població de neurones de la CNN i comprendre les representacions apreses.

Finalment, utilitzant la metodologia proposada, mostrem un estudi profund sobre com el color és representat en una xarxa específica entrenada per al reconei-

xement d'objectes, que competeix amb les capacitats de representació dels primats [17]. Trobem diversos paral•lelismes amb els sistemes visuals biològics: (a) un important nombre de neurones selectives al color a través de totes les capes; (b) una representació de baixa freqüència i de colors oponents per a les fronteres de color; mentre que hi ha una major mostra de freqüències per a les neurones de lluminositat (comparat amb les de color) tal i com s'efectua a V1; (c) una major representació de tonalitats de color a la segona capa que s'alinea amb els mapes de tons observats a V2; (d) un fort lligam entre les característiques de color i de forma en totes les capes, anant des de característiques bàsiques a les primeres capes (V1 i V2) fins a formes relacionades amb l'objecte i el fons a les capes més profundes (V3 i V4); i (e) una forta correlació entre les neurones selectives al color i la tendència de la base de dades.

Paraules clau: *visió per computador, esquemes plans, estructures jeràrquiques, aprenentatge profund, xarxes convolucionals neuronals, color, índexs de selectivitat*

Contents

List of Figures

List of Tables

1 Introduction

1.1 A brief introduction to computer vision

Primate visual system shows what seems an effortlessly ability to recognize objects. Understanding its procedure became a target in several research fields in the last decades, ranging from psychophysics, neurophysiology to computer vision (among others) [32].

The visual system is one of the main input sensors that allow to understand the world surrounding us. It is a nervous system that provides with the capability to process visual details, starting from a physical stimuli in form of light impacting the eyes and successively processed in the brain as electrical signals, building an extraordinary representation that lets to identify and characterize signals as image content. Within biological systems, the human is one of the most complex, its complexity is irrefutable, since it is able to perform difficult tasks such as recognizing object in the crowd and under different and variable conditions, and this is probably the reason why more than a half of our brain is involved in vision [38].

Computer vision is a science that has its focus of attention on understanding and simulating the Human Visual System (HVS). Its goals can be described in two ways [54]: (a) from a biological point of view, it pursues to model the human visual system; and (b) from an engineering point of view, it seeks to design computational systems to solve tasks that HVS carries out. Nevertheless, both interpretations follow a scheme that mimics the HVS: a camera (eye) captures the light source in form of an image and it is processed through an artificial algorithm (brain) that allows to understand the information in the image.

This field was born in 1966 when S. Papert proposed it as an artificial intelligence summer project [106] to link a camera to a computer to describe what it saw. The complexity of the idea behind this project incited the growth of the Computer Vision science that it is still growing and improving solutions for several visual tasks. Fundamental tasks pursued by computer vision can be comprised in three main topics: (a) visual recognition, (b) 3D reconstruction and (c) motion analysis. The first one is mainly looking for a general solution for finding a specific object or feature from an image (set of pixels) and characterizing it by an attribution of a semantic category. The second pursuits to recover the original three-dimensional

structure or geometry shown in one or more images. Finally, the aim of the motion problem is to identify a position change of an object in a scene. However, all these problems coincide in a first stage, in which image information is transformed into a compact representation (or description) based on several visual features. Therefore, these visual features become a key point for any computer vision problem. But, *what is a visual feature?* A general definition can be found in [37] where it is described as a portion of an image that characterize relevant properties for solving a visual task, *i.e.*, a basic element of an image. The type of these features is (or should be) directly dependent on the specific task to be solved. Due to the significance of these elements, there is a huge set of different features used in computer vision, related to color, shape, texture, disparity or motion, among others. Thus, the description of an object inside an image is carried out through a particular set of features that characterizes the *visual descriptor* used for the computational algorithm that tries to understand the content of an image.

In spite of the youth of the field, it has gone through two different eras, distinguishable from two points of view: (a) from the features used to describe images (handcrafted versus learned [98, 115]), and (b) from the artificial architecture where the features are used (flat versus hierarchical schemes [71]). The evolution of going from flat and handcrafted descriptors to hierarchical schemes having automatically learned features has come in parallel with the outstanding technological achievements in three main areas: machine learning, image-specific hardware and software and in the construction of big labeled-image datasets. This promoted a strong link between both areas (Computer Vision and Biological Vision [69]) encouraging to develop algorithms to achieve the aim of S. Papert of making computers able to describe the content of an image.

1.2 From flat descriptors to deep learning architectures

Robustness and generalization of human vision system are remarkable, since it is able to solve the vision recognition problem despite being under several difficult conditions (occlusions, illumination changes, specularities,...). It demonstrates the powerful representation achieved in the brain, which has been addressed as a source of inspiration in the development of artificial algorithms in computer vision dealing with the problem of vision recognition. But, *which features are characterizing each object allowing its recognition with a robust and generalized mechanism?* Its answer is not universal and it implied the designing of several artificial feature detectors in computer vision to compute an abstraction of the image information.

Taking inspiration from early stages of ventral stream where edges, contours and corners have shown a fundamental key for visual perception [120] several detection

algorithms have been designed using handcrafted features to find where these features appear on images (*e.g.* Canny detector for edges [18] or Harris for corners [50]). Such feature detectors, normally based on computing partial derivations, were used in visual descriptors to characterize each image, either as descriptors based on local features (*e.g.* SIFT [84], SURF [8], LBP [103]) or global features (*e.g.* HOG [28], GIST [104]). However, the success of their performance was based mainly on the features used, which were human engineered as was the design of the descriptors. With these approaches, computer vision algorithms were able to extract important features from images and to build a representation (visual descriptor) to train a classifier (such as SVM [27]) to solve a specific task, moving away from finding the general solution for vision problems. Krüger *et al.* in [71] defined these kind of approaches as *flat processing schemes*, where the extraction of features is carried out in a single layer and, afterwards, used in a learning algorithm to solve a certain task.

In spite of the progress of flat schemes during the first decades of the computer vision history, these computational algorithms were still failing on generality, as achieved with the primate visual system. As introduced by Hubel and Wiesel [55, 56], and afterwards observed from the neuron population responses [67], visual features are organized through a hierarchical scheme which increases progressively the complexity of the receptive field properties. With the desire of building systems like the primate visual system, jointly with advances on the knowledge of the visual cortex as well as the increase of computational resources on computers, the design of algorithms following a *deep hierarchy scheme* [71] gained interest. These are architectures composed of multiple stacked levels, where information of the detected features is transmitted from one level to its next and composing several features to achieve an increasing complexity. Several works have pointed out the benefits of these hierarchical models [10, 71]. They surpass the modeling and representation limitations presented in flat schemes giving more potential on dealing with more complicated visual problems and also they present a high generalization capability.

Therefore, computer vision evolved on building approaches adopting a hierarchical scheme and trying to provide methods to achieve the same capability for pattern recognition as a human being. For this purpose, the invariance to transformations (*e.g.* with respect to the position, size or view) requires to be tackled [109, 140]. Authors in [109] suggested that this invariance could be achieved through pooling operations over different descriptions of the same stimulus. Following this idea, authors in [111, 119] proposed the well-known algorithm HMAX which combines a hierarchical scheme with maxpooling operations providing a biologically inspired method. Although achieving promising results on the recognition problem, the features used by this method for describing input images were handcrafted (Gabor functions) and a new trend started on believing that engineered features

would lack on providing generality. Researches in machine learning were, therefore, aiming for substituting these handcraft features for trainable feature extraction giving rise to the beginning of the deep learning era.

Deep learning techniques allow to learn automatically optimal representations from unprocessed data on hierarchical models with several layers or levels of abstractions. Using statistical methods, each layer is trained to focus on selecting the best features and forwarding them to the following layer. They are able to learn complex functions through composing non-linear transformations [73]. The use of these techniques have impressively improved the state-of-the-art and conventional methods [20]. Although they have become popular recently, mathematical fundamentals behind these techniques as well as simple deep learning methods were published around 80s [29, 42, 113, 144]. One of the most factors that deprived the use of deep learning techniques on that time was the insufficient computational resources. Nowadays, this problem has been minimized thanks to the new developments on computer sciences (GPU's and multi-core computer systems) that clearly influenced in the popularity of deep learning methods. A second limitation was the nonexistence of large datasets required to deal with the huge parameter space present in these approaches. Thus, efforts on creating large datasets with a large variety of categories to get closer to the real world diversity (*e.g.* ImageNet [114]) have also influenced in the appearance of this trend. Even so, training deep architectures becomes a hard task due to the difficulty in optimizing them [11].

Nevertheless, among all the algorithms that are included under the label of deep learning, there is a particular technique which has shown easier to be trained and capable to achieve a generalization for several visual tasks [73]. This is what is known as Convolutional Neural Network (CNN). Due to its impressive performance has recently become a popular technique in the computer vision community. CNNs are inspired in the visual system following the neuroscience notions of the existence of simple and complex cells [55]. The parallelism with biological vision is derived from the fact that a CNN presents a deep hierarchy similar to the stages in ventral stream of the human visual system. Moreover, these layers are mainly based on two kind of operations: (a) *a bank of convolution operations followed by a non-linearity*, which allows encoding translation-invariance of features across the visual field; and (b) *a max-pooling operation* that is a sub-sampling step that inserts some local tolerance and also introduce scale invariance along the hierarchy. Therefore, two main layers are considered: convolutional layers and pooling layers. First act as local feature detectors while pooling gather similar features [73]. The interest on these techniques is growing up not only for the performance achieved with them but also for the understanding of the learned features along the hierarchy as a source of inspiration on the understanding of the visual system [69]. Deeper details of these techniques can be found in Section 3.2.

1.3 Color feature for visual recognition

In previous sections we have introduced the importance of defining an appropriate feature to describe the image content on designing artificial mechanisms (ideally) capable to solve visual tasks, likewise in the Human Visual System. Taking into account that this system perceives color images of the world surrounding us and this property is used to discriminate between objects, materials, textures, food, among others, color has naturally become a feature that is relevant for the models of computer vision dealing with the image understanding problem.

Although most of the first algorithms were designed to work on gray-scale images (single-channel), ignoring color properties from images complicates the resolution of some visual tasks or simply makes it impossible. With the acquisition of color images computer vision researches started to evolve to use color images and take profit of this feature.

In color images the values of pixels encode the spectral information of the light reflected by the surfaces in the scene. These values are represented in a k-dimensional color space (usually $k = 3$, inspired on the three types of receptors of the HVS), and a common formulation of this representation is written as

$$\rho_k = \int_{\omega} R_k(\lambda) E(\lambda) S(\lambda) d\lambda, \qquad k = 1,2,3, \tag{1.1}$$

where $E(\lambda)$ is the illuminant of the scene, $S(\lambda)$ is the surface reflectance we are viewing, $R_k(\lambda)$ is the sensitivity function of the k-th sensor defining an axis of the color space, and ω is the visible spectrum usually ranging between 400 and 700 nanometers.

However, introducing color information to computer vision algorithms requires to deal with the use an appropriate color model to specify a particular color, *i.e.*, to define the coordinate system in which each pixel is represented. This is fundamental, taking into account that, whichever is the descriptor used for solving the visual task, any feature should be easily detected from this representation.

Although equation 1.1 tells us that color in the physical world is mathematically modeled as a point-based phenomenon, when we face the problem of solving higher level visual tasks, such as automatic image classification, the building of efficient color descriptors requires a definition of color in the surrounding context. This involves the representation of spatio-chromatic information, which is a difficult problem to overcome. It has been tackled in previous works from different points of view [2, 88, 133, 143].

The first one, generalized by Weickert [143], is based on the consideration of color differences as partial derivatives computed on each RGB color channel. This

was first addressed by Di Zenzo [31] who introduced the idea of color tensor. It provided a way to combine channel gradients to obtain the orientation of the color variation in a local spatial neighborhood. Subsequently, this idea was further developed by Kass and Witkin [63] for oriented patterns, and it was finally established by Weickert [143], who introduced an additional integration scale that increases the color-spatial coherence.

A second approach by Mäenpää and Pietikäinen [88] is based on computing image descriptors in different color spaces and using the best space for each specific application. This idea led van de Sande *et al.* [133] to study which combinations of color representation and descriptor were the most appropriate for recognition tasks. They considered well-known three-dimensional color spaces such as device-dependent RGB, colorimetric XYZ, perceptually uniform CIELab and CIELuv, cylindrical hue-saturation-lightness (HSL) and hue-saturation-value (HSV), and physiologically-based opponent space. These spaces were combined with common image descriptors, such as scale invariant feature transform (SIFT) [85] and GIST [104]. In this direction, Zhang *et al.* [152] proposed a biologically inspired descriptor that extends the 3D color space with a fourth opponent channel. Recently, Cernadas *et al.* [19] searched for the best combination of color spaces, normalization methods and features for texture classification, and González-Rufino *et al.* [46] studied different colour-texture features to diferentiate cells in histological images.

The third approach is based on the direct extraction of color blobs (*i.e.*, homogeneous color regions) from trichromatic representations. In particular, Alvarez and Vanrell [2] describe an image in terms of shape and color attributes of the image blobs. In this case, the blobs are obtained from each channel of the opponent space by using Lindeberg's blob detector [81]. Khanina *et al.* [65, 66] adapted the scale-space technique for color images and proposed to use the Hessian matrix. Ming and Ma [93] proposed a weighted multi-scale blob detector using a hybrid operator that combines the Laplacian and the determinant of the Hessian. The results of this operator are later processed by a blob filter that includes a color-based Förstner operator and a hue-based histogram.

On the other hand, within the convolutional neural networks approaches the trend is to train these architectures from RGB color images, but the learned color representation through their intermediate representational spaces, as well as if this feature is learned, is unknown.

1.4 Objectives

The ambition of having a better understanding of Convolutional Neural Networks techniques and their optimal learned feature representation, as well as the belief that knowledge derived from these architectures may give insights on the functional mechanisms of the visual cortex, give rise to this thesis.

Concretely we focus on the problem of visual recognition and, specially, on the color representation. We pursuit to answer the following two questions:

- Which color representation properties benefit to the detection (extraction) of features for solving the visual recognition problem?
- Is color a feature learned in a Convolutional Neural Network trained for solving the object recognition task?

The "black-box" nature involved in Convolutional Neural Networks as a result of an automatic feature learning without imposing any insight on the structure promotes a lack of understanding of their intrinsic features. And it opens new goals to be addressed:

- How can we understand (and visualize) the internal representations in a trained CNN?
- How can we characterize the functionality of each neuron in the representation?
- How can we relate a neuron activation with a specific property, such as color?

Finally, attending that these deep learning techniques are inspired in biological processes, we deal with the following question:

- Can we find parallelisms between learned features in CNNs and the representation done in the HVS?

We believe that the studies reported in this thesis can be useful for computer and biological vision areas or, at least, to open new research lines seeking the identification of which kind of features are useful for representing objects and helping on the understanding of the visual cortex encoding.

1.5 Outline

The rest of this PhD thesis is structured in six chapters:

- In Chapter 2 we enhance the importance of representing images, and therefore, their features, in an appropriate color representation holding specific properties. We propose a new color representation which is somehow inspired on the existence of multiple hue maps in the HVS [107, 131] which maximizes the intra-channel contrast and minimizes the inter-channel correlation. This is analyzed under the perspective of using handcrafted descriptors in a flat scheme for solving the scene recognition problem.

- In Chapter 3 we introduce Convolutional Neural Network techniques by providing basic definitions, our notation and a summary of the main parts involved in these approaches.

- In Chapter 4 we review the state-of-the-art in visualizing intermediate features on trained architectures. Furthermore, we propose our Neuron Feature as a visualization of the intrinsic properties encoded by each neuron in the CNN and we provide examples of these visualizations on the network VGG-M trained by Chatefield *et al.* [20] on the ImageNet dataset [114], also presented in this chapter.

- In Chapter 5 we introduce a methodology to characterize each neuron activity with a specific property, concretely to the color. Our proposal is based on the definition of a selectivity index that allows to classify the neuron population regarding the studied property. This chapter contains a deep study of how color is tackled through all the network and, moreover, how the CNN encodes color. The analysis is done including some parallelisms with the HVS.

- In Chapter 6 we extend the neuron selectivity indexes to other properties, such as class or orientation, and we also propose a metric to find similar neurons within a specific layer.

- Finally, in Chapter 7 we conclude this dissertation, highlighting the benefits of the usage of a methodology capable to characterize neuron properties and inspect learned features. Further research lines are also exposed, inspired on seeking for other properties beyond individual neurons but focused on neuron relationships; as well as some insights on how to take profit of the neuron exploration to adapt a trained network into a new task.

1.6 Contributions

In this PhD dissertation we have made contributions in both flat and hierarchical algorithms, mostly based on the color representation of both perspectives. In the

following lines we expose these contributions, relating them to the corresponding chapter.

- A new color representation based on the extension of color channels is presented. This is based on maximizing local intra-channel and minimizing inter-channel correlation, providing a color representation which is adapted to the specific content of each image. (Chapter 2)
- The introduction of the selectivity measurement (inspired on the neuroscience) to characterize neuron properties. (Chapter 5)
- A package to explore trained Convolutional Neural Networks: *NefeSi package: Neuron Feature and Selectivity index for CNN visualization*. This introduces:
  - A generic visualization of individual neuron activity based on top activation images: the *Neuron Feature*. (Chapter 4)
  - A hierarchical feature composition visualization of deeper neurons. (Chapter 4)
  - A classification of neurons considering their selectivity indexes to specific properties. (Chapter 5 and Chapter 6)
- An in-depth study of the color selectivity index on a specific network trained for object recognition that allows to show some parallelisms with primate visual systems (chapter 5):
  - An important number of color selectivity neurons through all the layers.
  - An opponent and low frequency representation of color edges in 1st layer as in V1.
  - A higher sampling of frequency selectivity in brightness than in color of neurons in first layer, as in V1.
  - A higher sampling of color neurons in the hue dimension is found in the second layer, aligned to observed hue maps in V2.
  - A strong color and shape entanglement in all layers, going from basic features in shallower layers (V1 and V2) to object and background shapes in deeper layers (V4 and IT)
  - A strong correlation between neuron color selectivities and color dataset bias.

2 A color representation for a Flat Descriptor

Extraction of spatio-chromatic features from color images is usually performed independently on each color channel. Usual 3D color spaces, such as RGB, present a high inter-channel correlation for natural images. This correlation can be reduced using color-opponent representations, but the spatial structure of regions with small color differences is not fully captured in two generic Red-Green and Blue-Yellow channels. To overcome these problems, we propose a new color coding that is adapted to the specific content of each image. Our proposal is based on two steps: (a) setting the number of channels to the number of distinctive colors we find in each image (avoiding the problem of channel correlation), and (b) building a channel representation that maximizes contrast differences within each color channel (avoiding the problem of low local contrast). We call this approach *more-than-three color coding* (MTT) to enhance the fact that the number of channels is adapted to the image content. The higher color complexity an image has, the more channels can be used to represent it. Here we select distinctive colors as the most predominant in the image, which we call color pivots, and we build the new color coding using these color pivots as a basis. To evaluate the proposed approach we measure its efficiency in an image categorization task. We show how a generic descriptor improves its performance at the description level when applied on the MTT coding.

2.1 Introduction

Describing image content using most known descriptors is affected by the color representation, which is usually performed on each channel independently. A variety of descriptors based on local spatial features have been defined over different three-dimensional color representations (see section 1.3), mostly on RGB or opponent color spaces. However, in this chapter, we hypothesize that the performance of these descriptors for high level visual tasks, such as image classification, can be improved by using color spaces that boost the appearance of the spatio-chromatic image structure. Boosting can be achieved by overcoming two main drawbacks: (a) inter-channel correlation of RGB spaces, and (b) lack of contrast in color-homogeneous regions of opponent spaces. These two effects can be seen

in Fig. 2.1, where important edges between regions of different colors (orange-green edges) present clearer differences in color-opponent spaces with respect to the inter-channel correlated edges in RGB. However spatial structure that appears inside homogeneous-color regions is more contrasted in RGB than in opponent channels, where minor details (across the green or orange area) are lost.

To prove the previous hypothesis, we propose a new color representation that achieves decorrelation and enhancement of local color contrast based on the following ideas: (a) using more than three channels if required, *i.e.*, adapting color coding to the content of each specific image; and (b) enhancing local contrast inside channels by maximizing the contrast with respect to the most representative color of each channel. Following previous ideas, we compute a multi-channel representation of the spatio-chromatic image structure in a two-step process. First, we select the set of distinctive image colors, denoted as pivots, which capture the most relevant colors for each specific image. Second, the value of a pixel in each new channel is computed by the similarity between the trichromatic color and the corresponding pivot of the channel. We name the proposed representation *more-than-three* color coding, since the number of distinctive colors is not restricted to the usual three (although in some cases it can be three, or even two). In general, the more color diversity the image has, the greater number of color channels our representation has. We denote our approach as MTT (*more-than-three*) from now on.

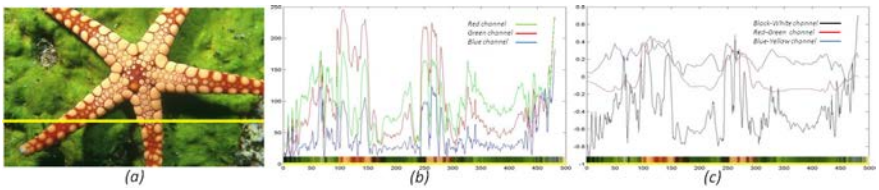


Figure 2.1 – Visualization of high RGB correlation and low local contrast of color-opponent channels for a single row of a natural image. (a) Analyzed row is highlighted in yellow. (b) and (c) Profiles of the image row for RGB and the opponent space, respectively.

To test the proposed MTT coding, we use the semi-joint texton descriptor (STD) introduced by Alvarez and Vanrell [2]. This descriptor, based on the Texton theory by Julesz and Bergen [61], decomposes the image into minimal color regions (blobs). These blobs are described in terms of their color and shape attributes, which are not conditioned by the image space. This independence from the space makes this descriptor the most adequate to be directly applied to the new color representation without any additional computation. We report our results for two different experiments. First, we compare the representation capabilities between MTT and two

trichromatic representations, namely RGB and opponent space, concluding that MTT allows a more accurate representation of the image content because it has lower correlation and higher local contrast, which allows a more carefu blob-based representation over the full image. Second, we perform an experiment on scene categorization showing that this approach gets a higher accuracy, outperforming state-of-the art results computed at the descriptor level.

Although we show good performance with the proposed approach, two criticisms to our initial hypothesis may arise. The first one refers to the increase in the number of color channels compared to usual representations. However, the use of extra channels can be linked to recent findings about the existence of multiple hue maps in the human visual system [107, 131]. These hue maps show selectivity to more colors than the primaries encoded in three-dimensional opponent spaces.[1] The second criticism refers to tuning to each specific image content. This tuning may complicate the description of images for comparison purposes. However, it ensures that a better spatio-chromatic representation is obtained for image regions that can otherwise be lost with a fixed coding, as will be shown in the experiments.

The rest of the chapter is organized as follows. In Section 2.2 we describe our new representation. In Section 2.3 we define the experimental setup and present the results obtained by our approach on the experiments. Finally, in Section 2.4, the conclusions of this new color representation are discussed.

2.2 More-than-three color coding (MTT)

Our goal is to define a color representation that has a channel for each distinctive color in the image. By distinctive colors we mean those that play an important role in understanding the image content. We use as many channels as distinctive colors an image has. For a given channel we assign, (i) the maximum value to pixels of the distinctive color, and (ii) a value inversely proportional to the distance to such distinctive color to the rest of pixels. In this way, in each channel, we are maximizing the representation of a distinctive color preserving its spatial coherence. Since all distinctive colors have their own channel, we ensure that all the important color regions of the image will be fully represented in at least one channel, and that all the region details will be maximally contrasted in the corresponding channel. We denote the distinctive color of a channel as its pivot.

Let us note here that the proposed representation is based on the content of

[1] Hue maps are defined as clusters of neurons that peak when a specific color stimuli is presented. Although a lot of research is left to be done in this area, some interesting results have begun to arise: there are more hue maps in higher levels than the six opponent colors [79, 107], and the peaks of the cell responses are given by particular hues [26, 112, 141].

each image. Color coding for each image is dependent on the color pivots computed from that particular image. For instance, an image of a forest with four distinctive colors could be represented by a channel for green leaves, a channel for brown tree trunks, another for blue sky, and a last one for white clouds. Meanwhile, an image of a beach could be represented by three channels with all the details of yellowish sand on one channel, deep blue of the sea on a second one and light blue of the sky on a third. We want to remark that this representation has not a fixed dimensionality, but it varies from one to any number representing the color complexity of a specific image scene. Nonetheless, we can state that this dimensionality usually converges to a moderate number, since natural images are typically dominated by only a few colors [105].

The process to obtain the proposed MTT coding can be divided in two parts: (a) the selection of pivots (Section 2.2.1) and (b) the definition of the channel values (Section 2.2.2). A general scheme of this process is summarized in Fig. 2.2.

2.2.1 Selecting color pivots

As we have introduced before, color pivots must be the most distinctive colors of the image. We propose to interpret distinctive colors as the most predominant ones in the image, and we find them using the Ridge-based Analysis of Distributions (RAD) technique [135]. The RAD algorithm groups image colors according to the ridges of the histogram. Ridges are computed by extracting all the local maxima of the histogram and connecting those which are close to each other.

Although other existing approaches could be used instead, we selected RAD because it has been demonstrated to fulfill two properties that are of clear interest to our method. First, the RAD algorithm is invariant to some color distortions as ridges extract all the histogram maxima plus all their nearby similar values, therefore being robust to small changes like the ones caused by noise. Second, all the points in a ridge are connected, which means that the ridge representation is robust to shadows and highlights, since both shadow and non-shadow regions of an object are included in the same ridge. Thus, small color distortions will not affect our method, since they will be captured by the ridge algorithm obtaining always a single color pivot for each dominant color. These effects are not captured by classical clustering methods (*e.g.*, k-means) which group colors mainly based on color similarity, while RAD allows joining colors from different parts of the histogram in the same ridge, if there is a sequence of local maxima that can be connected. Next, we briefly summarize how predominant colors are extracted with this method.

Let us define an image I as a $M \times d$ matrix where M represents the number of pixels in the image and d is the dimension of the color space (RGB, Lab, etc.). In RAD, the first step is to look for local maxima on the color histogram $H(I)$ with the

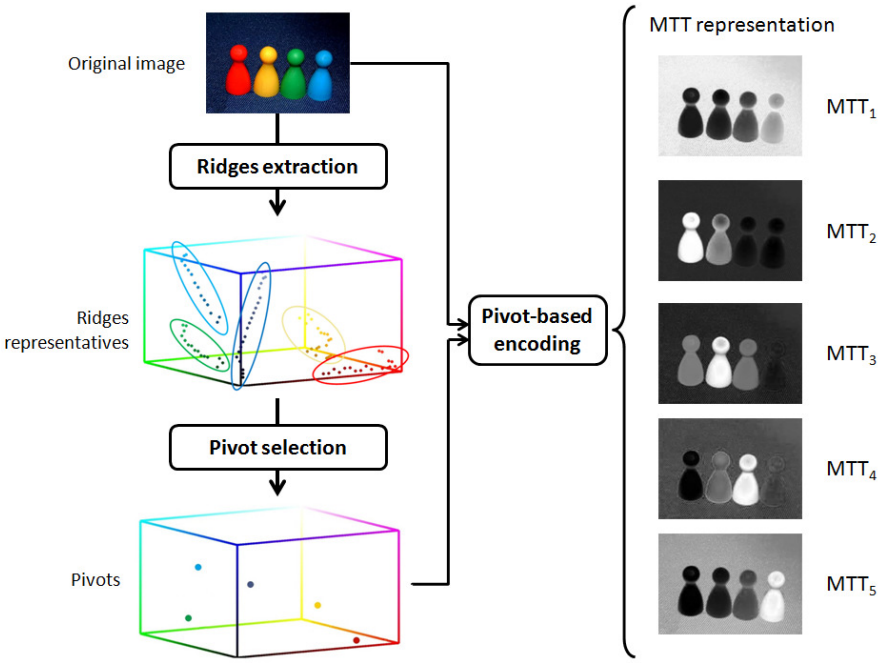


Figure 2.2 – Pipeline of the method. From the original image, we extract a set of ridges corresponding to the most distinctive colors and then we select a color pivot for each ridge. These color pivots are the basis to generate the proposed more-than-three (MTT) color coding of the image, which results in as many channels as distinctive colors of the image

multilocal creaseness measure of Lopez *et al.* [82, 83] defined as

$$\kappa(x) = -\frac{d}{r}\sum_{k=1}^{r} \bar{\omega}^t(x_k)\cdot n(x_k), \tag{2.1}$$

where x is a bin of the histogram $H(I)$, x_k is the k-th neighbor of x on an r-connected neighborhood, $\bar{\omega}(x_k)$ and $n(x_k)$ are the dominant gradient orientation and the unit normal vector to the discrete boundary of the neighborhood at each boundary site x_k, respectively, and d is the dimension of the histogram space. All mathematical details can be found in [82]. In our implementation we use the RGB color space (*i.e.*, $d = 3$) quantized in $30 \times 30 \times 30$ equally spaced bins. We use $r = 6$ to consider a 6-connected neighborhood as in the original implementation of RAD [135].

The local maxima of $\kappa(\cdot)$ which are close in the histogram are connected by following the lines of shallowest gradient descent until a flat region is reached. The sets of points contained in each of these lines are called ridges of the histogram and will be denoted by

$$C = \{c_1, \cdots, c_n\}, \tag{2.2}$$

where c_i is a color value from the image. For a particular image, the set of all the ridges extracted applying RAD will be denoted by $\{C_i^I\}_{i=1:L}$ and they will represent the most predominant colors of the image I.

Let us now focus on searching for the color pivots. For a particular ridge C_i^I of an image I, the color pivot, ρ_i^I, is defined as the one that fulfills

$$\rho_i^I = argmax_{c \in C_i^I} H(c), \tag{2.3}$$

that is, ρ_i^I is the color value of ridge C_i^I that has maximum value in the image histogram $H(\cdot)$.

2.2.2 Pivot-based encoding

After selecting the set of color pivots $\{\rho_i^I\}_{i=1:L}$ of image I, we define the new spatio-chromatic representation as the $M \times L$ matrix obtained using the similarity metric given by

$$\begin{aligned} MTT_{j,i}^I &= \max_{k \in 1:M} (\|\rho_i^I - I_{k,\cdot}\|_m) - \|\rho_i^I - I_{j,\cdot}\|_m \\ &\propto 1 - \frac{\|\rho_i^I - I_{j,\cdot}\|_m}{\max_{k \in 1:M} \|\rho_i^I - I_{k,\cdot}\|_m}, \end{aligned} \tag{2.4}$$

where $I_{j,\cdot}$ represents the vector consisting of the three color components of pixel j from the original image and $\|\cdot\|_m$ represents the m-Minkowski norm. We have used $m = 2$ that is equivalent to the Euclidean distance, although other distances, such as the perceptual CIEDE2000 [86], could also be used.

The computational complexity of our approach is linear in the number of pixels of the image for a fixed number of bins and a given dimension of the histogram space (in our case, $30 \times 30 \times 30$ and 3 respectively). Computing the MTT representation for an image of 768×768 pixels takes on average 888 ms, from which 722 ms correspond to the pivot selection (including the time of the RAD method) and 166 ms to the pivot-based encoding step. These computations were done on a Intel Xeon CPU E5-1620 processor.

In Fig. 2.3 we present the MTT representations of a set of images, and we

compare them to the RGB and the opponent representations. We can see that each MTT channel enhances different parts of the image. For example, in the first row, MTT channels emphasize different parts of the postbox. The base and the aperture are represented in the black channel, the box is in the red channel, the notice plate and the background trees are mainly enhanced on the gray channel, the grass is represented on the green channel, and the sky appears in the light gray channel. We can appreciate how color information is less correlated on these channels than in the RGB channels (please, focus on the green and blue channels of RGB) and that opponent channels present less contrast between the different objects of the image. Similarly, in the second row, the different parts of the boy's clothes (in red, blue, and orange channels), the snowman (in the white channel), and the background (in the gray channel) are all enhanced in different channels. An analogous analysis can be performed in the rest of images.

Notice that since our MTT representation is content-based we obtain a different number of channels on each image depending on the variety of colors in it. In the examples, the first two images have five channels whereas the last one showing a purple flower on a green background has only two channels. Notice also that the MTT channels represent different colors for each image. In some cases, as in the first-row image, two shades of the same color can be represented in different channels if they are sufficiently different from each other (in this example, gray and light gray).

Finally, let us explain how we can derive an inverse transform to the original space. By the construction of our space we know that for each channel: (i) the color selected as a pivot is always a trichromatic value appearing in the image. Therefore, the maximum value of the channel is equal to the maximum difference between the color of the pivot and the color of a certain pixel in the image; and (ii) there exists a pixel in the image (the one with its color at a further distance of the pivot) whose representation in the channel is 0. Mathematically,

$$\max_{k\in 1:M} MTT^I_{k,i} = \max_{k\in 1:M} \|\rho_i - I_{k,\cdot}\|_m. \tag{2.5}$$

$$min_{k\in 1:M} MTT^I_{k,i} = 0. \tag{2.6}$$

These two properties, allow us to invert Eq. 2.4 as follows:

$$\|\rho^I_i - I_{j,\cdot}\|_m = \max_{k\in 1:M} MTT^I_{k,i} - MTT^I_{j,i}. \tag{2.7}$$

Then, given $MTT^I_{j,i}$ and ρ_i this last equation defines a surface (an sphere if $m = 2$) of possible values for each $I_{j,\cdot}$. Therefore, to recover the original image we just need to know the value of three of the pivots that are linearly independent, and

use trilateration. Then, our recovered image will be given by values $I_{j,\cdot}$, that fulfill Eq. 2.7 for three values of i.

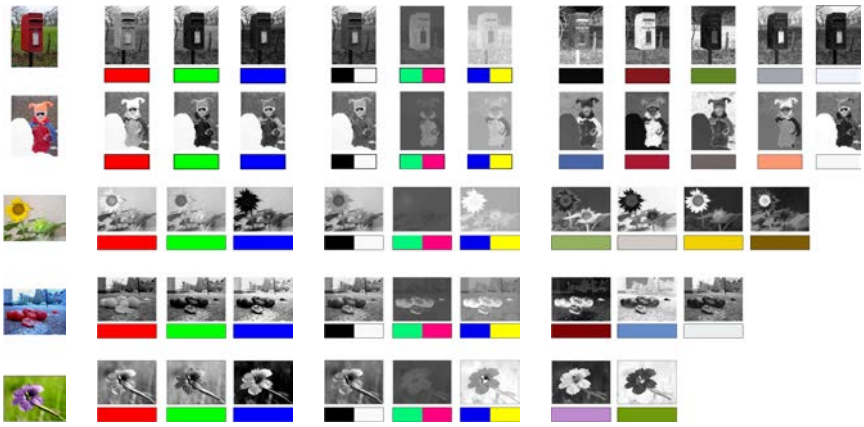


Figure 2.3 – Examples of MTT representation for several images and comparison to the RGB and opponent representations. (a) Original images. (b) RGB channels. (c) Opponent channels. (d) MTT channels. On channel images, values are represented on grayscale (black=0, white=1). The color boxes under channel images show the correspondence with RGB, opponent, and MTT channels. The proposed MTT represents images with a variable number of channels, which depends on the number of distinctive colors of the image.

2.2.3 Illumination invariance

As explained in the introduction of this chapter, pixel values of an image depend on the reflectance of the objects, the camera sensors, and the illumination of the scene. Therefore, when the illumination of the scene changes (which is usual in real images), pixel values also change, thus hindering the performance of computer vision algorithms. Different methods have been proposed to counter the illuminant variability effect, either by discounting the illuminant [6] or by performing some form of color normalization [41]. In this section, we show that our image representation can be directly used as an invariant to the illumination (therefore avoiding the need of further processing) when computed on the logRGB color space.

In RGB space, the change in illumination between two images of the same scene can be approximately modeled by a single scaling factor on each channel (*i.e.*, the Von Kries coefficient law [145]), either directly [39] or by applying the spectral

sharpening technique [40, 136]. This is, given an image I^1, an image I^2 of the same scene under a different illuminant can be defined as

$$I^2 = \mathscr{D}^{1,2} I^1, \tag{2.8}$$

where $\mathscr{D}^{1,2}$ is a 3×3 diagonal matrix containing the scaling factors for each RGB channel, transforming the colors under the first illuminant to those under the second illuminant. If we apply a logarithm operation to the RGB space, the previous equation can be rewritten as

$$\log(I^2) = \left[d^{1,2}, \cdots, d^{1,2}\right] + \log(I^1), \tag{2.9}$$

where $d^{1,2} = \left[\log(D_{11}^{1,2}), \log(D_{22}^{1,2}), \log(D_{33}^{1,2})\right]^T$. This is the case since $\mathscr{D}^{1,2}$ is a diagonal matrix, and thus the channels of I^1 are treated independently. Equation 2.9 tells us that an illumination change can be modeled by a translation in logRGB space. Therefore, for any color value $x \in$ logRGB we have

$$H^2(x) = H^1(x + d^{1,2}), \tag{2.10}$$

where $H^1(\cdot)$ and $H^2(\cdot)$ denote the histograms of $\log(I^1)$ and $\log(I^2)$, respectively. Consequently, following Section 2.2.1, we have that the color pivots of $\log(I^1)$ and $\log(I^2)$ are also related by

$$\rho_i^{\log(I^2)} = \rho_i^{\log(I^1)} + d^{1,2}. \tag{2.11}$$

From Eq. 2.11 and Eq. 2.4 we have

$$\begin{aligned}
MTT_{j,i}^{\log(I^1)} = \\
\max_{k \in 1:M} (\|\rho_i^{\log(I^1)} - \log(I_{k,.}^1)\|_m) - \|\rho_i^{\log(I^1)} - \log(I_{j,.}^1))\|_m = \\
\max_{k \in 1:M} (\|\rho_i^{\log(I^2)} - d^{1,2} - (\log(I_{k,.}^2) - d^{1,2})\|_m) - \\
\|\rho_i^{\log(I^2)} - d^{1,2} - (\log(I_{j,.}^2) - d^{1,2})\|_m = \\
\max_{k \in 1:M} (\|\rho_i^{\log(I^2)} - \log(I_{k,.}^2)\|_m) - \|\rho_i^{\log(I^2)} - \log(I_{j,.}^2)\|_m = \\
= MTT_{j,i}^{\log(I^2)}.
\end{aligned} \tag{2.12}$$

Therefore, our representation computed on logRGB space is approximately invariant to the illuminant. An example of this invariance is shown in Fig. 2.4,

where we can see, from left to right, the original RGB image, the results of the MTT representation fixing the number of channels to 3, and a visualization of the MTT channels concatenated as an RGB image. It is clear that the MTT channels are very similar for all the images, making the RGB-like visualization stable under illuminant changes.

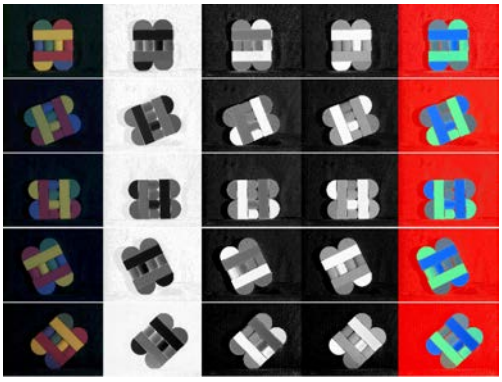


Figure 2.4 – Example of the approximate illuminant invariance of our representation. From left to right: original RGB image, the three MTT channels, and the MTT representation visualized as a RGB image. Illuminants in the original RGB images are (from top to bottom) an approximate D65 illuminant, an illuminant with $CCT = 3500$K plus a blue filter, an illuminant with $CCT = 4700$K, an approximate D65 illuminant plus a blue filter, and a fluorescent light.

2.3 Experiments and Results

As presented in the previous section, MTT provides a new color representation that is based on the specific content of each image. In this section we show its power to build generic color image descriptors. The evaluation is performed in two steps. We first evaluate how MTT overcomes the problems of RGB and opponent spaces to encode spatio-chromatic information of images. We evaluate this improvement in terms of the channels correlation and local contrast and also showing how MTT representation improves the ability of a specific image descriptor; second we evaluate how MTT increases the performance of a descriptor in a scene classification task.

The evaluation is performed in two steps. We first evaluate how MTT overcomes the problems of RGB and opponent spaces to encode spatio-chromatic information of images. We evaluate this improvement in terms of the channels correlation and

local contrast and also show how MTT representation improves the ability of an specific image descriptor.

Considering the problem of generic image description, the comparison between descriptors of different images built on the MTT representation requires adaptation to any number of channels. To overcome this problem we use a the Semi-Joint Texton descriptor (STD) [2] and a variant of it, both are explained on the next subsection. This descriptor gives an intermediate-level representation in terms of image blobs, *i.e.*, color-homogeneous convex regions, that is computed regardless of the color space.

Taking into account previous considerations, we organize this experimental section in four subsections. First, we introduce the image descriptor used in the experiments. Second, we provide the details of the setup used in the experiments, which are fully explained in the remaining two subsections.

2.3.1 Image description: Semi-joint Texton Descriptor

The Semi-joint Texton Descriptor (STD) introduced by Alvarez and Vanrell in [2] describes an image in terms of shape and color attributes of the image blobs. STD can be computed on any color space, and we show that the performance of this descriptor on scene recognition is improved when MTT is used instead of RGB or the opponent representation. An interesting property of this descriptor is that the attributes of the blobs it uses do not depend on the input color space where the blobs are initially detected. Due to this property, the descriptions of two images can be compared independently of the color representation where the blob detection is performed, even if their representations have different number of channels.

The STD algorithm starts detecting the blobs of an image by applying a multi-scale Laplacian in each separate channel of the image representation of choice. From the blobs detected in all the channels, color and shape attributes are extracted. Then the STD is defined as a combination of shape (STD_S) and color (STD_C) descriptors of image blob's attributes (see next two following subsections):

$$STD = [STD_S \; STD_C] \tag{2.13}$$

Shape descriptor

The shape descriptor is a histogram of blobs' shape attributes. For each detected blob, the shape attributes are area, orientation, and aspect ratio, which are obtained independently of the color channel where the blob was detected. Then, all blobs' attributes are quantized in a three-dimensional blob-shape space in order to compute the histogram. In this histogram each bin represents a visual word of the universal shape vocabulary defined by the quantization of the blob-shape space.

Color descriptor

The color descriptor is a histogram of blobs' color attributes. The histogram is computed in the HSI color space, where blobs' color attributes are quantized. In this histogram each bin represents a visual word of the universal color vocabulary defined by the quantization of the color space (see Fig. 9 in [2]).

We also propose to use a variant of the color descriptor defined in [3]. This approach is based on the color-naming model of Benavente *et al.* [9], which categorizes any image pixel p in one of the 11 basic colors defined by Berlin and Kay [12] (*i.e.*, red, green, blue, yellow, orange, brown, pink, purple, white, gray, and black). Such categorization is done by means of an 11-dimensional membership vector $\mu(p)$, where each component $\mu_i(p)$ can be interpreted as the probability of color p to belong to a particular color $\mathscr{C}_i$. Pixels are assigned to the color term with highest membership, which is then backed up with a modifier related to the lightness (*i.e.*, dark, medium, or light). Using this color-naming representation the quantization of the color space is more perceptual than the original quantization [2], where just an equally-spaced division of the space was used.

To avoid confusions, from now on we denote by STD_{OR} the original descriptor defined in [2] (shape descriptor plus color descriptor on HSI), and by STD_{CN} the variant that uses color naming for the color description [3] (*i.e.*, STD_{CN} is formed by the shape descriptor and the color descriptor based on color names). Figure 2.5 shows a graphical representation of the two STD implementations used in this chapter.

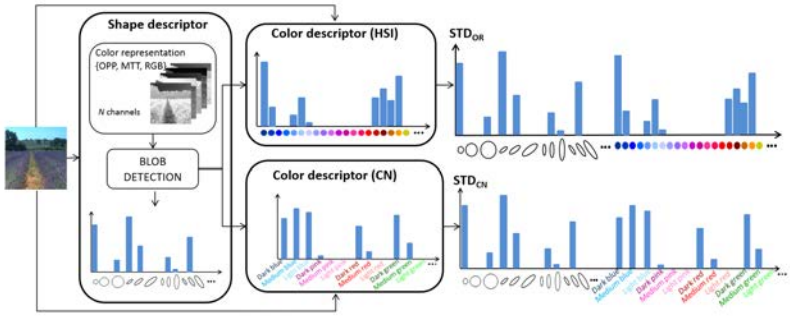


Figure 2.5 – Diagram of the process to obtain STD_{OR} [2] and STD_{CN} [3]. Blobs are detected on each channel of the chosen color representation and shape attributes are computed to generate the shape descriptor STD_S. The color descriptor STD_C is computed either on the HSI color space or using color names.

Adding spatial layout information

The STD descriptor is a global first-order statistic of blob attributes. For scene recognition, the insertion of the spatial layout is a must, since areas with similar colors can represent different things depending on their location in the image. For example, medium and large blue blobs can represent either water (*e.g.*, a lake or the sea) or sky; in this sense, adding their spatial location will help to distinguish whether they represent water (usually located at the bottom images) or sky (usually located at the top).

Hence, we add the spatial component similarly to how it is added in the GIST descriptor [104]. Given an image I we decompose it in a set of non-overlapping sub-images $I_1, \cdots, I_k$, which are obtained by dividing each of the image dimensions by a particular natural number (usually 2, 3, or 4). Then, we compute the descriptor for each of the sub-images and concatenate them, obtaining a final descriptor of the form

$$STD = [STD_{S_1} \cdots STD_{S_k}\ STD_{C_1} \cdots STD_{C_k}], \tag{2.14}$$

where STD_{S_i} and STD_{C_i} represent the shape and color descriptors of sub-image I_i.

2.3.2 Experimental setup

In our experiments, the maximum number of channels for the MTT representation is set to $L = 8$. This value was experimentally found by testing values from $L = 2$ to $L = 11$. Results gradually improve as the value of L increases, but for $L > 8$ the improvement is not significant. If more than eight ridges are extracted from an image (see Section 2.2.1), the eight ridges that represent the largest areas of the image (computed via a watershed in the color histogram of the image) are selected.

To obtain the shape descriptor, we use the following quantization of the shape space: 8 orientations ($0°, 22.5°, 45°, 67.5°, 90°, 112.5°, 135°, 157.5°$), 7 scales (area), and 3 aspect ratio values (isotropic, elliptical, and highly elongated). Isotropic blobs are assigned to orientation $0°$. Thus the shape descriptor has dimension $119 = (8 \text{ orientations} \times 7 \text{ scales} \times 2 \text{ aspect ratios}) + 7$ (one bin per scale for isotropic blobs).

In the case of the color descriptor, we have used the two configurations explained in Section 2.3.1. For STD_{OR}, the HSI color space is quantized in 16 bins for H, 4 for S, and 5 for I, making a size of the color descriptor 320 bins. For STD_{CN}, color is defined in terms of 11 names and three modifiers, which gives a size of 33 bins for the color descriptor. Therefore, the total size of STD_{OR} is $119 + 320 = 439$ bins, whereas STD_{CN} has a the total size of $119 + 33 = 152$ bins. If spatial decomposition is used (see Section 2.3.1), these values should be multiplied by the number

of sub-images considered to obtain the final size of the descriptor.

Finally, the dataset used in all the experiments is the dataset of scenes created by Oliva and Torralba [104], which contains 2688 images of 256×256 pixels from 8 categories: coast, forest, highway, inside city, mountain, open country, street and tall building.

2.3.3 Experiment 1: Analysis of MTT properties

In this first experiment we analyze the properties of the proposed color representation. As we mentioned in the introduction of this chapter, the main problems of usual color spaces to encode the spatio-chromatic image structure are due to the high correlation between channel and the lack of local contrast for specific colors. These two properties are inherent in the channel-based representation derived from the sensor that reduces the capability to represent all the image details. Even when we transform to an opponent representation, the lack of contrast of the new chromaticity channels does not allow representation of all the details of areas with homogeneous chromaticity. Considering these two aspects, in this experiment we have computed the inter-channel correlation and the channel's local contrast for RGB, normalized opponent space [2] (nOPP), and the MTT representation have also considered the space defined by the three eigenvectors obtained by PCA on the RGB space.

For a given image, the inter-channel correlation has been computed as the average of the minimum pairwise-channel correlation[3], and to obtain the local contrast, we use the method defined by Haun and Peli [51].

The results are shown in Table 2.1. We can see that the MTT representation presents a combined result of low inter-channel correlation and high local contrast. If these results are compared to the ones obtained by the opponent space, we see that MTT obtains better results in both measures. PCA presents the lowest correlation at the cost of also obtaining the lowest local contrast. Comparing to the RGB space, the local contrast of RGB channels is slightly higher than in MTT, but in RGB, the correlation between its channels is considerably higher than in MTT. We also looked at the behavior of local contrast when considering only the three MTT channels that have higher local contrast for each image. In this case, the result for MTT is over 10% higher than in RGB, showing that a subset of the MTT channels presents higher local correlation than any other representation of the same dimension.

Let us now show how the better results of MTT in correlation and local contrast

[2] As defined in the C-SIFT descriptor [133].

[3] We use this measure instead of a global correlation average due to the different number of channels in each color representation.

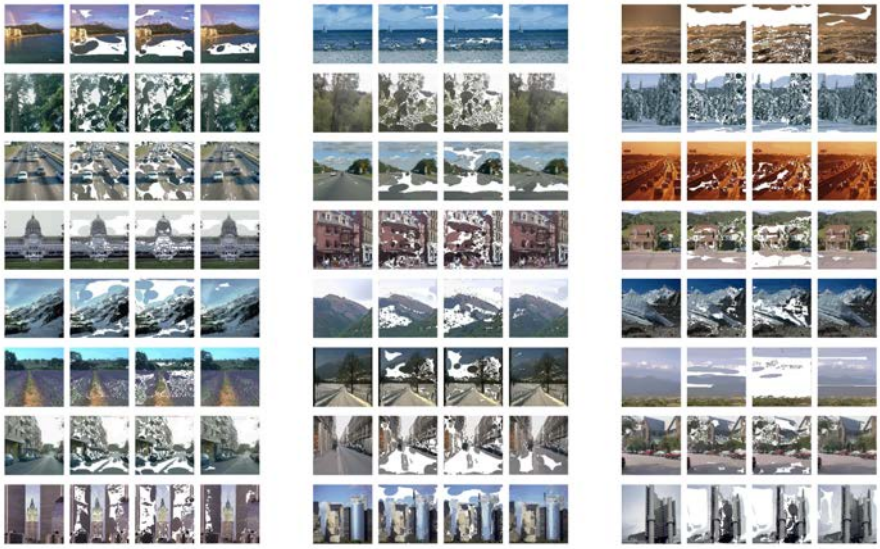
Table 2.1 – Correlation among the different channels and mean local contrast for the different color spaces for all the images on the Oliva and Torralba dataset.

	Correlation	Local contrast
RGB	0.82 (± 0.18)	20.47 (± 10.21)
nOPP	0.30 (± 0.20)	10.75 (± 7.48)
PCA	0.00 (± 0.02)	9.55 (± 8.26)
MTT	0.25 (± 0.18)	19.24 (± 10.76)

enable better image description. To this end, we detect the blobs in each image of the dataset (using the blob descriptor encoded in the STD descriptor) on different color representations to analyze how well these blobs describe the content of the image. We assume that, in general, the more area covered by detected blobs, the better the overall appearance of the image will be described. Thus, an image can be reconstructed by plotting their blobs at the locations where they were detected, and filling them with the corresponding color attribute. Figure 2.6 shows a visual comparison between the blobs detected on the proposed MTT, the normalized opponent color space, and the RGB space. We can appreciate that on MTT, more parts of the image are described, the details are better represented and the overall structure of the original image (*i.e.*, the gist of the image) is more appreciable.

To give a quantitative analysis of the results in the previous figure, in Table 2.2 we show the percentages of covered area by blobs detected on RGB, the opponent space and MTT. As can be seen, the percentage of area covered by blobs detected on MTT is higher than those obtained on the other color representations. This increase can be found in all the categories of the dataset. For example, in the forest category the increase is over 13% with respect to RGB. This could be because images from this category have low contrast and similar hues, which makes that areas of similar color can not be detected as different regions in the opponent or the RGB channels. By contrast, MTT is more able to represent different shades of the same hue in different channels, which facilitates the posterior blob detection.

Finally, let us analyze how our better detection of the gist of the image translates to the shape descriptor part STD_S. To this end, in Fig. 2.7 we compare the distributions of detected blobs from an image using opponent and MTT representations. Distributions are displayed as 3D histograms where one of the axes represents orientation, another jointly represents aspect ratio and area, and the third represents the number of blobs. We can appreciate that each visual word in STD_S clusters blobs with a similar visual appearance (*i.e*, similar area, orientation, and aspect ratio). We note that STD_S on the MTT channels detects more blobs



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k) (l)

Figure 2.6 – Blobs detected on different color representations. Each row corresponds to one of the categories of the dataset. (a), (e) and (i) Original images. (b), (f) and (j) Blobs detected using the RGB color space. (c), (g) and (k) Blobs detected using the normalized opponent space. (d), (h) and (l) Blobs detected using the MTT representation.

Table 2.2 – Percentages of covered area for each category on the Oliva and Torralba dataset using STD_{OR} descriptor on RGB, normalized opponent (nOPP) and MTT channels.

Category	RGB	nOPP	MTT
Coast	85.52%	78.41%	96.10%
Forest	84.61%	83.36%	97.95%
Highway	79.32%	69.03%	91.52%
Inside city	90.01%	85.06%	98.75%
Mountain	85.28%	82.44%	96.24%
Open country	90.93%	87.30%	97.71%
Street	81.74%	73.94%	95.06%
Tall building	87.33%	83.65%	96.43%
All	**85.94%**	**80.99%**	**96.38%**

than on the opponent space, specially on those bins where some blobs are already detected on the opponent space. Moreover, MTT enables the detection of blobs with attributes corresponding to bins where only a few blobs are detected on the opponent channels. These extra blobs detected on the MTT representation are mainly found in large uniform areas, which explains why MTT is more effective representing the overall structures of the image, as we have seen in Fig. 2.6.

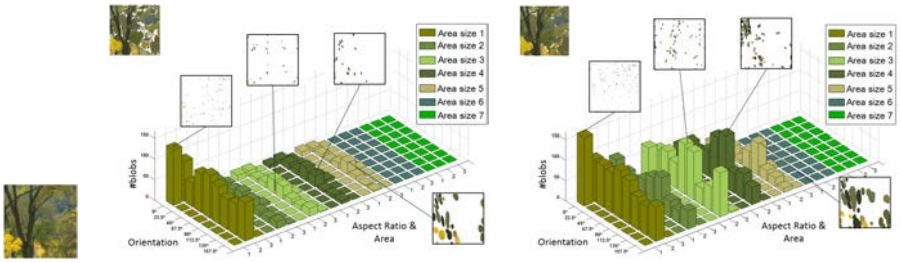


Figure 2.7 – Comparison of the shape descriptors of an image (a) obtained on the opponent space (b) and on the MTT representation (c). Shape descriptors are shown as a 3D histogram where each bin clusters detected blobs with similar area, orientation and aspect ratio. On the aspect ratio and area axis, '1' corresponds to isotropic blobs, '2' to elliptical blobs, and '3' to highly elongated blobs. Bins corresponding to different areas are plotted in different colors. Area increases along the axis.

2.3.4 Experiment 2: Scene recognition

In this experiment, we test the efficiency of the new representation when it is used to compute the STD for scene recognition tasks. We first compare different spatial decompositions to determine the best configuration of STD and then we compare the results to the state of the art on the database of Oliva and Torralba [104]. The experiments are done following the same methodology used in [16]. A linear support vector machine is trained and tested on a randomly selected split of 600 images for training and 120 images for testing. This procedure is repeated 10 times and results are averaged.

Analysis of spatial decomposition

As stated in Section 2.3.1, the inclusion of spatial information on STD can improve the results for general tasks in computer vision. Spatial information is a

building part in some image descriptors, such as GIST [104], but it should not be confused with the idea of spatial pyramids [72], where the descriptor is computed on regions of different sizes and are later combined into a single descriptor.

To analyze the relationship between the number of sub-images used and the accuracy achieved, we have computed the results of the original implementation of STD (STD_{OR}) and STD using color names (STD_{CN}) on different color spaces, considering the whole image (no spatial decomposition) and different number of sub-images (4, 9, and 16). According to these results (see Fig. 2.8), the inclusion of spatial information in the descriptor by dividing the image into four sub-images increases the accuracy by at least 4% in all cases. Considering nine sub-images still increases the accuracy, but the increase is not as remarkable as in the previous case. After that, the increase is not significant or there is even a slight decrease in accuracy in the case of the descriptors computed on RGB.

Comparison to state of the art

Given the results of the previous section, we use four sub-images to compute STD_{OR} and STD_{CN} because this configuration provides us with good performance, and the size of the descriptor does not increase dramatically (1756 for STD_{OR} and 608 for STD_{CN}). Now, these results are compared to the ones reported in [16] for three well-known descriptors: SIFT [85], GIST [104], and HMAX [96] and are presented in Table 2.3. Rows 1 to 3 summarize the results of Brown and Süsstrunk in [16]. We only report the color space where each descriptor achieved the best results. The highest accuracy was obtained with GIST on the opponent space (without normalization). Rows 4 to 6 and 7 to 9 show the performance of STD_{OR} and STD_{CN}, respectively. In both cases, the descriptor is computed on three color representations (RGB, normalized opponent space, and MTT).

Analyzing the results, the use of MTT on both STD descriptors provides with an improvement on the accuracy of about 4% and 4.5% comparing to RGB and the normalized opponent space, respectively. This result can also be observed in Fig. 2.8 where for any number of sub-images, any descriptor computed on MTT overcomes the same descriptor computed on RGB or on the normalized opponent space.

Moreover, we computed the Wilcoxon test with the hypothesis that the results obtained with GIST on the opponent space and with STD_{CN} on MTT in the ten trials of the experiments belonged to the same distribution. We obtained a p-value of 0.0020 with a significance level of 5%. Therefore, we can reject our null hypothesis and conclude that the improvement obtained by STD_{CN} on MTT over GIST on the opponent space is statistically significant. Therefore, we can conclude that the use of MTT improves the results of the STD descriptors with respect to the use of RGB or

Table 2.3 – Accuracy (%) and standard deviation computed over ten trials on the scene recognition experiment. Results for HMAX, GIST, and SIFT were extracted from [16]. In parenthesis we show the color space used.

Descriptor	Accuracy (%)
SIFT(nOPP)	69.6 (± 2.5)
HMAX(RGB)	74.0 (± 4.4)
GIST(OPP)	77.8 (± 3.4)
STD_{OR}(nOPP)	75.7 (± 3.5)
STD_{OR}(RGB)	76.3 (± 3.5)
STD_{OR}(MTT)	**80.1** (± 3.6)
STD_{CN}(nOPP)	78.6 (± 2.5)
STD_{CN}(RGB)	79.1 (± 4.6)
STD_{CN}(MTT)	**83.0** (± 3.0)

the normalized opponent space. Furthermore, both STD_{OR} and STD_{CN} computed on MTT outperform GIST results reported in [16]. Let us remark here that our best result (STD_{CN} on MTT channels with an accuracy of 83.0%) is obtained with a descriptor composed by 608 bins, while the GIST descriptor has a size of 960 bins.

Moving to the analysis by category, Fig. 2.9 shows the confusion matrix of our best result (STD_{CN} on MTT). Each cell of the matrix shows the percentage of images of a class (row) classified as each of the classes (columns). From the matrix we can see that the category with higher accuracy is forest. This could be expected since this category shows a low intraclass variability. By contrast, open country and coast present a high confusion (*e.g.*, 14% of coast images are classified as open country). Similarly, city and tall building are two categories with a certain confusion (8% of images of each class classified in the other class). Both cases can be explained by the fact that images in these pairs of categories show high similarities; for example, open country has many images of lakes and rivers that can be confused with images of coast, and city category contains many images of buildings combined with other elements such as cars and pedestrians that can be confused with images from the tall building category.

2.4 Conclusions

In this chapter we propose a new color representation based on the specific content of the image. With this approach we sought an image color coding that enhances spatio-chromatic information and reduces inter-channel correlation. The

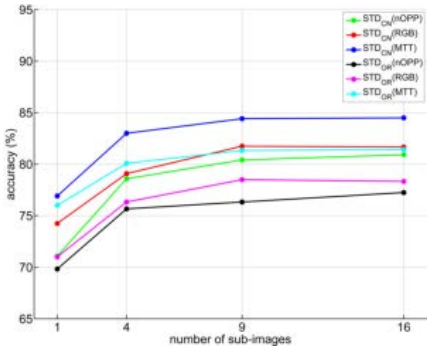


Figure 2.8 – Accuracy on scene recognition in terms of the number of sub-images used to compute STD descriptors on different color spaces.

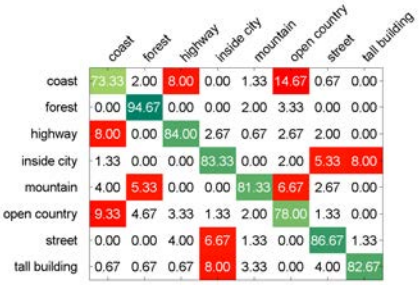


Figure 2.9 – Mean confusion matrix of the scene recognition experiment on Oliva and Torralba dataset using STD_{CN} computed on the MTT representation. Greenish cells correspond to the results with accuracy higher than 70%, while reddish cells correspond to the misclassified results with a percentage above 5%.

goal is achieved in a two-step process. First, we set the number of channels used in MTT with the number of relevant colors the image has, defined as pivots. Second, we build individual channel representation that maximizes contrast differences using a similarity metric with respect to the color pivot related to each channel.

The proposed approach presents some clear advantages:

- Represents images according to its own color complexity, this is with more than three dimensions if required. As each dominant color is mostly represented in one of the dimensions, our approach shows more ability to capture the image details.

- Increases the local contrast and reduces the correlation of the resulting channels, which plays a crucial role in several tasks such as edge and blob detection, segmentation, and recognition.

- Presents illuminant invariance properties if it is built onto a log space. This can be an important benefit, essentially in recognition tasks.

- Increases performance when applied to build color image description for scene classification. This increase is mainly due to the improvement in the blob detection step of the color descriptor.

To demonstrate these advantages we have performed two experiments. First, a qualitative experiment to show the performance of the MTT representation in a blob detection task. We visualize how the proposed approach presents low correlation and high local contrast, and how it improves the area covered by detected features across the full image plane. A second quantitative experiment has been performed for scene recognition. We show how the same descriptor improves its performance when applied on the MTT coding, and we compare the results to current state-of-art descriptors, which are overcome.

With the studies performed in this chapter we have tabled the discussion that is not only an important key point to find which kind of feature is needed to solve a specific problem but also the color space where they are extracted. Therefore,finding the optimal combination of these two properties is a difficult task that has favored the arrival of the new paradigm in computer science of designing architectures capable to automatically learn both color space and visual features to describe the content of an image.

3 Convolutional Neural Networks: Basic definitions

Convolutional Neural Networks (CNNs) are a kind of deep learning techniques defined as feed-forward architectures compound of several stacked layers that operate on their inputs to produce a representation change. In this sense, each layer yields a new level in the encoding process. Mainly, two types of layers are used, the convolutional and the pooling. Nevertheless, the main responsible layers on the encoding process are the convolutional ones, which are compound of several neurons. These neurons are characterized through a set of trainable weights (and biases) that are adapted to solve a specific visual task and act as feature detectors on their inputs. CNNs are biologically inspired methods, so that connectivity between neurons follows the hierarchical scheme found in the visual cortex and increasing complexity through layers.
In this chapter we introduce what exactly are the Convolutional Neural Networks, and how they are designed as well as the functionality of each part.

3.1 Introduction

Convolutional Neural Networks (CNNs) are a kind of artificial Neural Networks that have demonstrated a powerful capability on solving different computer vision problems of diverse nature [74, 75], such as image classification or object recognition. Therefore, they have become a key point in the advances in deep learning. They are designed to be applied on images and to learn visual properties from them. Although they have been designed to solve engineering problems, they take inspiration from the brain and their computations could be implemented by biological neurons, so they consist of several filters distributed along a set of layers that simulate neurons of the brain organized likewise the visual cortex. These filters are, in fact, a set of weights and biases that are learned on a training step pursuing the recognition of patterns in a similar way that our brain learns. As stated by biologists, neuronal learning is the result of achieve strong connections between neurons due to the frequent occurrences on specific patterns [52].

They are based on hierarchical feed-forward architectures concatenating different levels of convolutional and pooling layers, so that each layer operates on

their inputs to produce a representation change. On the one hand, the convolution operations followed by a non-linearity add some translation-invariance of features across the visual field. On the other hand, the max-pooling operations reduce the size of the image representation and add some local tolerance or even introduce scale invariance along the hierarchy. In this sense, they incorporate invariance to some transformations, following principles stated on [109, 140] that also inspired the authors of [118, 119] to develop the HMAX approach.

Main advantages of these CNNs are twofold: *flexibility for easy design* different architectures allowing to solve different kind of vision problems; and *ability to be automatically trained* in order to learn the best weights (for all the network parameters) to achieve the best performance on a specific visual task. As we already mentioned before, these two advantages emerged thanks to the outstanding of technological achievements in three main areas: machine learning, image-specific hardware and software and in the construction of big labeled-image datasets. However, the success on the training has been also benefited by two properties. First, instead of learning fully connected neurons as in Regular Neural Networks approaches, they use *local connectivity*, in the sense that they are just focused on subregions of the input image. This has reduced drastically the huge number of parameters. Second, the *sharing of the parameters* that allows to reuse commonalities existing in different positions in the input image and collaborates to the generalization and in the efficiency of these methods.

Taking advantage of the aforementioned results, CNNs have been proposed by several authors [17, 69, 71] as a suitable framework to model biological vision. Furthermore, several trained CNN architectures were compared in representational performance with the primate IT cortex on the visual recognition task. Cadieu *et al.* [17] proved using a kernel analysis that, contrary to what happened with previous artificial architectures such as HMAX [118, 119], current deep convolutional neural networks are starting to show important representational capabilities. Although this not prove that these computational mechanisms are similar to the primate visual system, we can not exclude these networks as a source of representational inspiration.

In this chapter we report technical details of CNNs in Section 3.2, starting with providing the definition of main concepts involved in these architectures, as well as our notation. Afterwards, in Section 3.2.1, we summarize the principal types of layers that are used in CNNs. Section 3.2.2 provides a guide of the arithmetics implied in these architectures, regarding shape sizes changes through each layer or to determine regions considered in each operation. Finally, the learning process is outlined in Section 3.2.3.

3.2 Convolutional Neural Networks (CNNs). Technical details

The basic architecture of a general CNN is shown in Fig. 3.1, where three types of layers are distinguished. First, an input layer receives an image. Second, hidden layers composed of a set of neurons responsible of mapping its previous input into high dimensional feature space (represented as gray squared in the figure). And finally, the output layer which also consists of a set of filters but, instead of projecting features into a new space, its functionality is to handle the visual task for which the network was trained (*e.g.*, it provides class scores on a classification problem). Although in this thesis we focus on this classical scheme (following the LeNet scheme [74]) where all neurons in a layer are used for the subsequent neurons in a sequential way, let us to mention that some different schemes have been proposed. Authors in [70], with their well-known AlexNet architecture, presented a slightly different scheme by defining two streams as two sub-classical architectures to favor the training process and be computed onto 2 GPUs. Another example is the breaking scheme used in the well-known GoogLeNet [130] which considers the idea of allowing to not always have stacked layers sequentially.

Let us remark here some terminologies and notations regarding CNNs:

- **Input image (I):** Image (usually represented in RGB) of sizes[1] $i_h \times i_w \times i_d$, being i_d the dimension of the color space in which the image is represented (*e.g.*, $i_d = 3$ in case of RGB). It is referred to the original image without any transformation, going beyond the preprocessed changes (*e.g.*, normalization or scaling).

- **Neuron (F):** Set of weights that constitutes a filter[2]. It is a three-dimensional structure of size $n = n_h \times n_w \times n_d$, being n_d equal to the number of neurons that constitutes the previous convolutional layer t, denoted as (k_q^t) or, in the case of a being neuron belonging to the first convolutional layer, to the dimension of the color space of the input image. Each convolutional layer (l) consists of a set of filters $F^l = \{F_j^l\}$ for $j = 1, \cdots, k_q^l$ (k_q^l is the number of neurons in layer l), likewise the j-th neuron of a layer l is denoted $F_j^l = \{F_{j,i}\}$, being $i = 1, .., n_d^k$. In addition, let us to specify that all the neurons belonging to the same layer have the same size.

[1] For consistency, in this thesis the dimensions will be considered as *height* × *width* × *channels*.

[2] Some literature refers neurons to the pixels, either of the input image or feature maps (*e.g.*, in [101]).

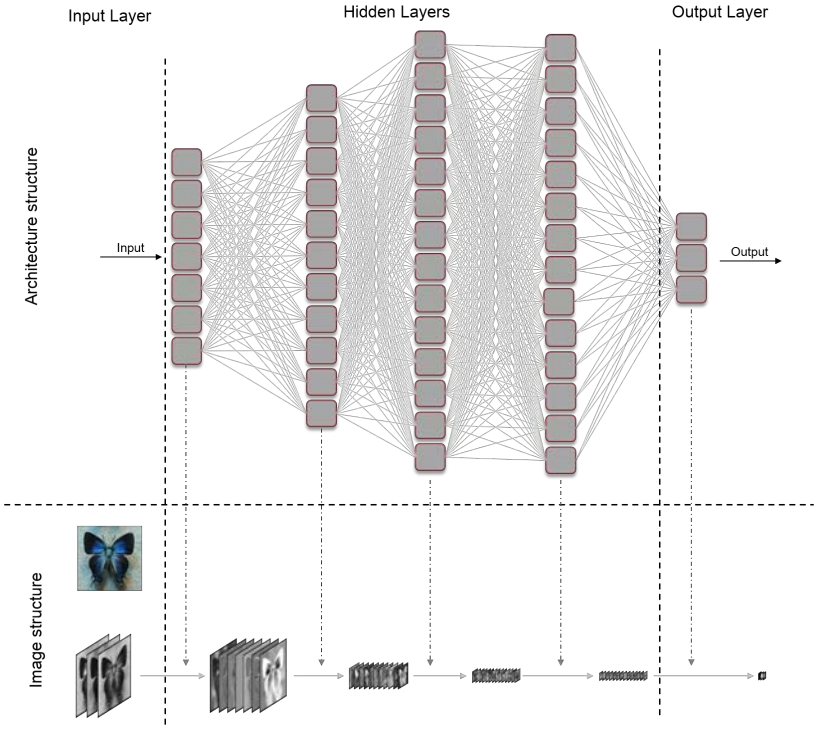


Figure 3.1 – Draft of the structure of a CNN architecture. It consist of an input layer, n hidden layers (here n=4) and an output layer. Top elements simulates the architecture of a network of 5 layers with 7,11, 14, 14 and 3 neurons (represented as gray squares), respectively. In the bottom there is a visualization of the image representation when it is applied into the network. Note that we use the terminology *neuron* to indicate a filter in a convolutional network.

- **Activation (a):** Result of a unique evaluation of an operation, which corresponds to a single value.

- **Feature map (Z):** Image representation after a transformation made by a layer. It is a two-dimensional structure that contains all the activation values obtained from the specific operation:

$$Z = \begin{pmatrix} a_{(1,1)} & a_{(1,2)} & \cdots & a_{(1,w)} \\ a_{(2,1)} & a_{(2,2)} & \cdots & a_{(2,w)} \\ \vdots & \vdots & \ddots & \vdots \\ a_{(h,1)} & a_{(h,2)} & \cdots & a_{(h,w)} \end{pmatrix} \tag{3.1}$$

Following the aforementioned notation, the three-dimensional structure of the set of stacked output feature maps is referred as $Z^l = \{Z_j^l\}$ for $j = 1, \cdots z_d^l$, obtained as a result of applying the layer l on the set of feature maps of the previous layer $\{Z_j^{l-1}\}$. We use the notation of $z^l = z_h^l \times z_w^l \times z_d^l$ to indicate the size[3] of the output feature maps of layer l. Note that first feature maps correspond to the preprocessed input image, $Z^0 = I$.

- **Receptive field (R):** Volume (set of pixels) of the input image that is visible to an operation at a certain time. The receptive field of an activation belonging to the Z^l feature map of a layer l is denoted by $R(a_{(h,w)}^l)$ and, fixing h and w, it is the same $\forall a_{j,(h,w)}^l$, $j = 1, \cdots, z_d^l$. We use r^l to refer to the size of the receptive field of any activation of layer l, which is $r^l = r_h^l \times r_w^l \times r_d^l$, where $r_d^l = i_d$ is the dimension of the color space used for represent the input image[4]. Note that each pixel of a feature map of a given layer has a different receptive field but with the same size.

- **Visual field (V^{l,l_t}):**[5] Volume (set of pixels) of the feature maps Z^{l_t} of a previous layer l_t ($l_t < l$) that has been used for obtaining a specific activation $a_{j,(h,w)}^l$ of the output feature map (Z_j^l) in layer l. Note that $V^{l,0}(a_{(h,w)}^l) = R(a_{(h,w)}^l)$. We will describe this volume from the top-left (h_1, w_1) and the bottom-right (h_2, w_2) coordinates[6], jointly with the indexation of the first (ch_1) and last channel (ch_2) used :

$$V^{l,l_t}(a_{(h,w)}^l) = [(h_1^{l_t}, w_1^{l_t}), (h_2^{l_t}, w_2^{l_t}), (ch_1^{l_t}, ch_2^{l_t})] \tag{3.2}$$

Let us to remark here that coordinates are described from the corresponding

[3] See Section 3.2.2 to know how to calculate the size of the set of stacked feature maps.

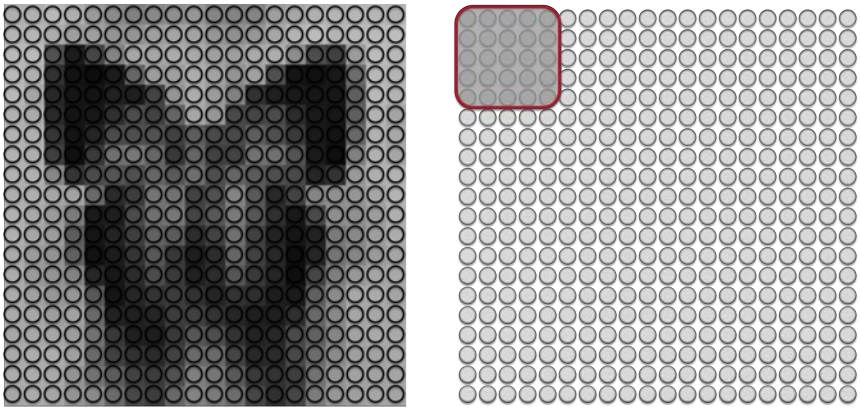
[4] See Section 3.2.2 to know how to calculate the receptive field of a given activation.

[5] Similar to the receptive field, but by *visual field* we mean the volume seen by an operation in each evaluation with respect to a feature map of any previous layer instead of with respect to the input image. Se Section 3.2.2 to know how to calculate the visual field of a given activation.

[6] Coordinates are expressed in the following reference system: the closest pixel to the origin of the system is the top-left (which has coordinates (1,1)), and the farest pixel is the bottom-right (having as coordinates the size of the image)

feature map Z^{l_t} in which the visual field is defined[7]. In typical architectures, $ch_1^{l_t} = 1$ and $ch_2^{l_t} = z_d^{l_t}$. We denote its size as $v^{l,l_t} = v_h^{l,l_t} \times v_w^{l,l_t} \times v_d^{l,l_t}$, being $v^{l,l} = 1$. Likewise in the case of the receptive field, the set of activations of different feature maps of the same layer and with the same coordinates share the same visual field.

Once specified the definition of these terms, see Fig. 3.2 as a graphical explanation. In Fig. 3.2a and Fig. 3.2b we clarify the differences between the *neuron* term used in some literature (*e.g.*, in [101]) and our usage (used also in [62]), to avoid confusions: Fig. 3.2a shows a gray-scale image of 20x20 pixels (black circles), each one considered as neuron (and usually represented by circles) in some literature; while the synthetic representation of an image (or a feature map) of the same size in Fig. 3.2b is operated by a neuron of size 5x5 (plotted in a red framed square), which also corresponds to the receptive field if the synthetic image is the input image. Our usage of neuron term in this artificial architectures fits with the meaning assigned in biological literature: structures in the visual cortex that spikes when a specific stimuli is in their receptive fields.



(a) Input image of 20x20 pixels

(b) Filter of size 5x5 pixels on a synthetic feature map of 20x20 pixels.

Figure 3.2 – Image, feature map, pixels and receptive field concepts.

By the other hand, let us to define three hyper-parameters that directly affect to the output feature map of a layer l (either convolutional or pooling) :

[7] In the case that the layer $l_t + 1$ introduces some padding, then the coordinates are with respect to the image representation that contains the feature map Z^{l_t} with the specific wrapping margin

- **Kernel size (k^l):** Size of the input feature map (Z^{l-1}) used for generate each of the activations of layer l. Note that $k^l = v^{l,l-1}$ and furthermore, in the case of layer l being a convolutional layer, the kernel size is exactly the size of its filters plus the number of neurons considered in this layer (k_q^l), *i.e.*, $k^l = n^l = n_h^l \times n_w^l \times n_d^l \times k_q^l$.

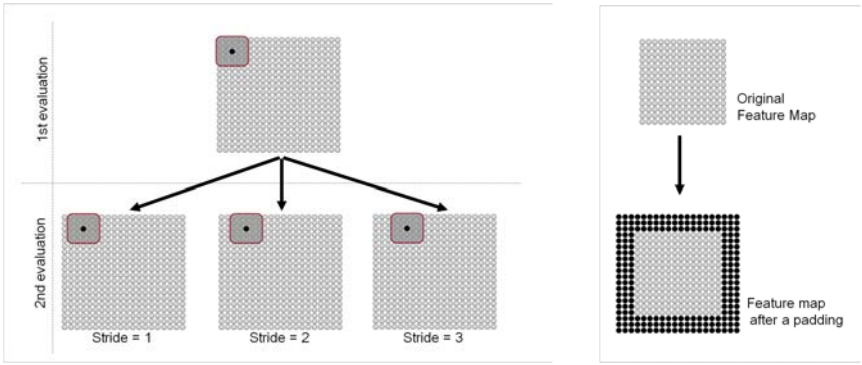
- **Stride parameter (s^l):** Distance between two neighborhood regions in two consecutive evaluations, *i.e.*, the distance to take between the places to allocate two consequent visual fields. If the stride is greater than 1 the operation of the layer is not taken in all the possible subregions and it might discard some pixels from the input feature map. In this case, some information is missed in the output. In Fig. 3.3a we synthesize the effect of three different stride values $(1,2,3)$ between two consecutive evaluations. The center displacement to apply between two subsequent evaluations is, therefore, defined by the stride. In this thesis we consider s as a vector of two elements, to indicate strides in x and y directions. Nevertheless, when both directions contains the same stride, we will express s as a single element vector, for simplicity.

- **Padding parameter (p^l):** Size of the margin to fill (usually, by zeros) the input feature map of a layer before any operation is applied. In Fig. 3.3b we represent the effect of this parameter (black circles) on a given original feature map (gray circles). We consider p^l as a vector of four elements: $[p_1^l, p_2^l, p_3^l, p_4^l]$ =[top, bottom, left, right], although p will be expressed with just a single value when all of them have the same padding. In the example of the figure, p = [3].

As it has been mentioned, these artificial architectures consist of several stacked layers that operates in their input feature maps, providing a new representation of the image (set of feature maps) that is going to be used as input in the subsequent layer. The operation that is carried out defines the type of layer (further details in Section 3.2.1).

3.2.1 Main layers

Most CNNs are defined by stacking Convolutional, Non-linear layers and Pooling-layer although some incorporate other type of layers, such as a normalization layer. The way they are stacked (and the specific hyper-parameters set on each one) defines a specific architecture. The vast majority of the common libraries that are used in the community (*e.g.*, MatConvNet [137], Caffe [60], Theano [132], Torch7 [21], TensorFlow [1]) allow flexibility to easily design custom architectures.



(a) Stride **(b)** Padding

Figure 3.3 – The effect of the stride (left) and padding (right) parameters on feature maps.

Convolutional layers

Convolutional layers are, in fact, the basis of any Convolutional Neural Network. The representational transformation of the images is mainly carried out by the neurons allocated in these layers, which act as feature detectors as a result of an automatic training that learns the best weights to represent their input and also achieve the best performance on a specific visual task. These neuron weights, in fact, control the influence of one neuron of a previous layer with itself and define the visual feature encoded by this neuron (see Fig. 3.4).

Each filter applies the mathematical model of a neuron, which is a dot product between the input and the weights and adds a bias [13, 90], simulating in this way a biological neuron. This operation is done repetitively in different subregions on the input image to find the visual feature (same weights) in any position, allowing to share parameters. This can be computed using the discrete convolution $\star$ operation [59]. Thus, given a set of z_d^l output feature maps $\{Z_i^l\}_{i=1:z_d^l}$ of a layer l, they are transformed by its following convolutional layer $l+1$ as:

$$\{Z_j^{l+1}\}_{j=1:k_q^{l+1}} = \{\sum_i^{z_d^l} Z_i^l \star F_{j,i}^{l+1} + b_j^{l+1}\}_{j=1:k_q^{l+1}} \tag{3.3}$$

where $F_{j,i}^{l+1}$ is the i-th channel of the j-th neuron in layer $l+1$, and b_j^{l+1} is the bias of this j-th neuron. Note that the discrete convolution is, firstly, a channel-wise operation so that the i-th input feature map is convolved by the i-th channel of the

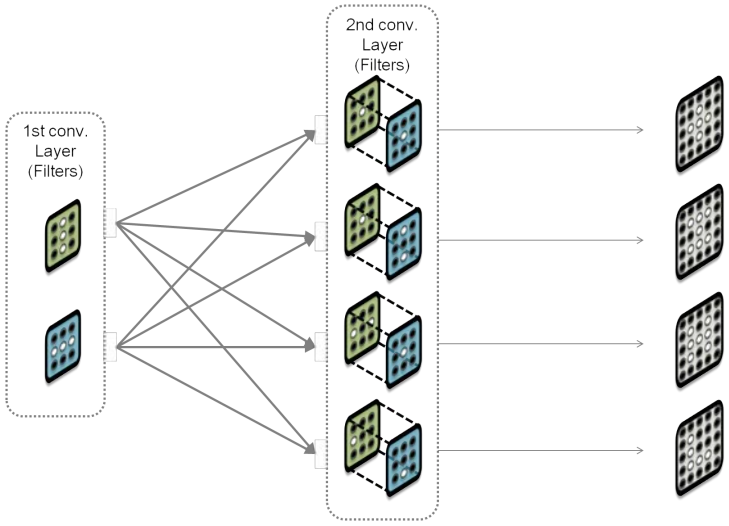


Figure 3.4 – Illustration of the neuron connectivity. First convolutional layer consist of 2 different neurons (or filters) while the second, of four different neurons. Grayish squares plotted in the right graphic how weights control the influence of the neurons. Note how the vertical and horizontal lines represented by the weights in the first convolutional layer are used to build the visual features +, *I, H, L* in the second layer.

neuron F of the subsequent layer and afterwards sums results from all channels. The output of each filter generates a feature map that is stacked by the rest of the output filter maps of the layer (generated by the remaining neurons). In this way, pixels belonging to the same feature map share the same neuron (*sharing of the parameters*). A graphical illustration of this multidimensional operation is shown in Fig. 3.5.

In this way, these layers computes a template matching searching the set of features encoded by each neuron on the input image (see Fig. 3.6).

Fully-connected layers

Some literature distinguishes between a fully-connected layer and a convolutional layer, although they can be considered the same type since the operation performed is quite similar. The main difference is that neurons in full-connected layers act as neurons in Regular Neural Networks in the sense that filter's size gather all the previous activations, *i.e.*, kernel size of these filters overlay the entire input feature maps. Then, Eq. 3.3 can be simplified as a dot product:

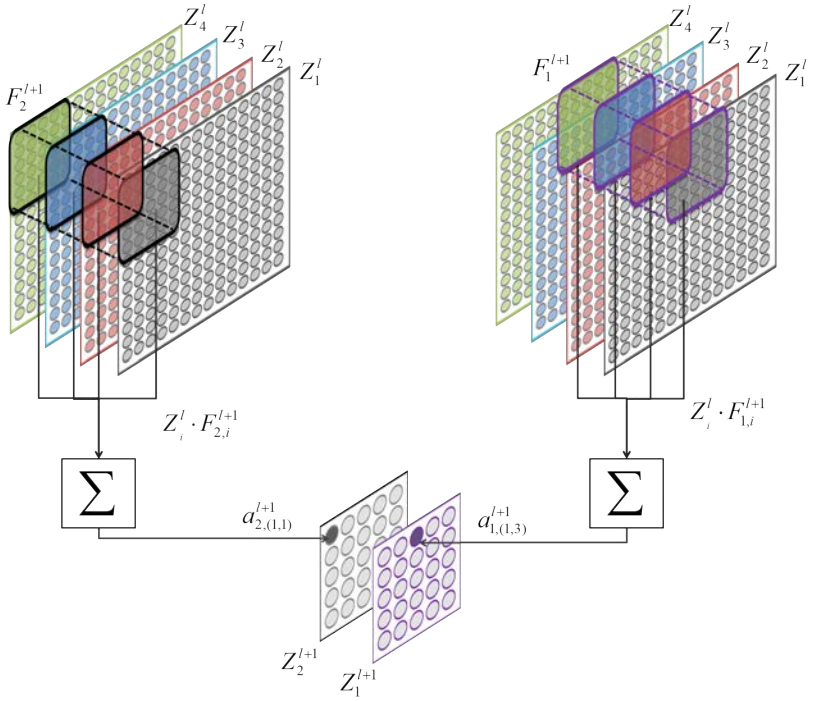


Figure 3.5 – Illustration of a convolutional layer of two neurons (represented in black and purple volumes). Input feature maps are channel-wise operated with the corresponding channel of the filter and afterwards are combined by summing them. Each neuron in this layer is a three-dimensional structure of size 5x5x4 (note that depth is constrained by the number of input feature maps, which is equal to four). At each evaluation, each filter (neuron) operates with the set of pixels of the input feature map that are covered by the area of the neuron and provides a single value, so that it builds the corresponding feature map that will be afterwards stacked with the rest of generated feature maps by the same layer.

$$\{Z_j^{l+1}\}_{j=1:k_q^{l+1}} = \{\sum_{i}^{z_d^l} Z_i^l \cdot F_{j,i}^{l+1} + b_j^{l+1}\}_{j=1:k_q^{l+1}} \tag{3.4}$$

From the point of view of their functionality, there is another reason to distinguish between both types of layers. As well as convolutional layers are devoted to the feature detection task, fully-connected layers are more related to the classification from the high-dimensional representation of the image gotten from the set of

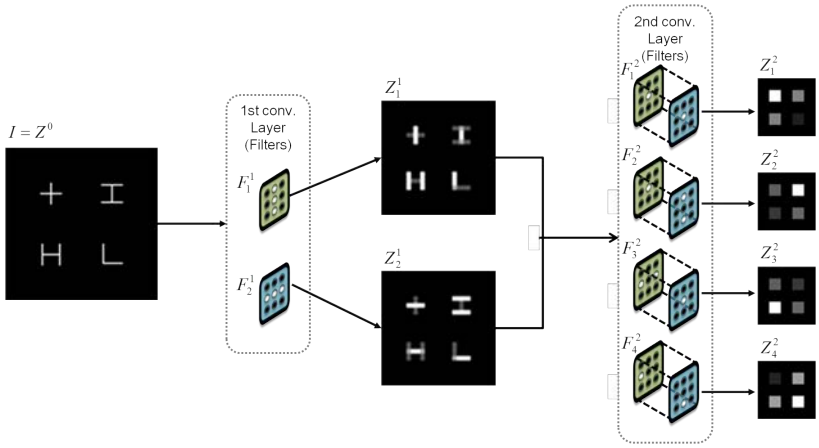


Figure 3.6 – Toy example of an image representation through a CNN with two convolutional layers. Each neuron is more activated (more white in the illustration) when its encoded feature is found on the image (template matching).

convolutional layers.

Pooling layers

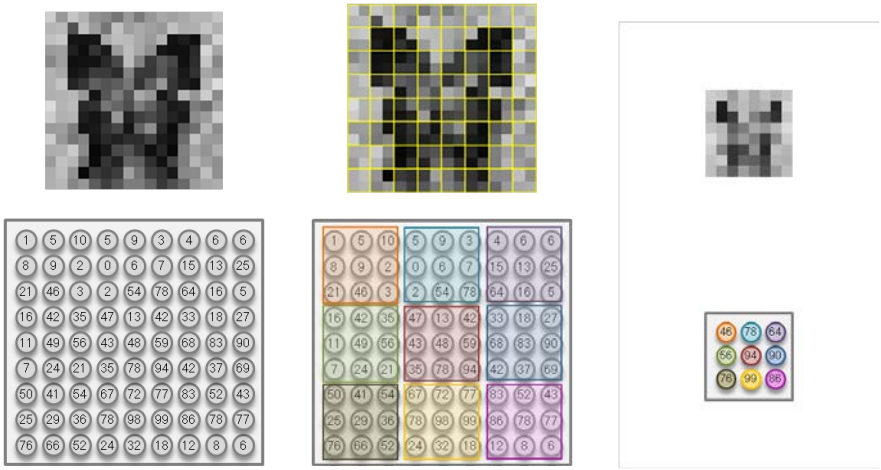
These layers introduce some local translation invariance and reduce the size of the image representation (feature maps) by gathering several features over a small neighborhood. In consequence, the use of these layers contribute to the reduction of the amount of parameters as well as the overfitting and provide some generalization to the architecture. Nevertheless, it computes a channel-wise operation, so that each slice (channel) is operated independently. Typical operation used for compute the pooling is the MAX operation, which do not require any parameter learning. However, Springenberg *et al.* in [127] concluded that these layers can be substituted by a convolutional layer with a greater stride achieving the same performance.

Each activation of the output feature map of a max-pooling layer l is computed as:

$$a_{j,(h,w)}^{l+1} = max(V^{l+1,l}(a_{j,(h,w)}^{l+1})) \tag{3.5}$$

Therefore, the max-pooling subsamples the image representation by keeping the maximum value within a region defined with the kernel size (k) of the layer (see Fig. 3.7).

Other operations that can be computed as a pooling are average, stochastic or



(a) Pooling input **(b)** Pooling grid **(c)** Pooling output

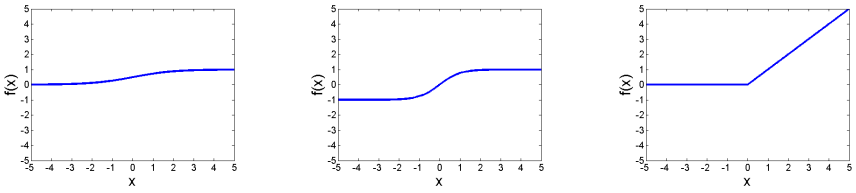
Figure 3.7 – Illustration of a max-pooling operation. In the first row there is shown an example of an image, while in the second row a toy example. For both cases we have assumed a set of hyper-parameters that provide non overlaping regions $k = [2,2]$, $s = [2]$ for the first row and $K = [3,3]$, $s = [3]$ for the second. Max-pooling reduces the image representation by keeping the maximum within a small region. In this figure are shown the (a) input representations, (b) the subregions where the operation is carried out, and (c) the output result.

L2-norm poolings, among others [14, 15, 76, 148].

Non-linear layers

Although convolutional layers constitute the basis of any CNN, they apply a linear operation. Even thought several several convolutional layers are stacked, the result is still a linear mapping of the input images. Nevertheless, typical classification problems where CNNs are applied to solve them consist on complex data which their representation is not good enough through several linear mappings and be clustered by a linear classifier. In this context is where non-linear layers gain importance: they are nonlinear functions that introduce to the network a way to learn nonlinear transformations of the input data and achieve feature representations into a new space where they are linearly separable. These are element-wise functions that are generally applied after each convolution and do not vary the size of the input feature maps. Common non-linear functions are sigmoid, hyperbolic tangent and rectified linear units (ReLU) (see Fig. 3.8), although the last function

has become more popular due to its benefits [45, 97, 125]: outperforms sigmoid results, facilitates a faster and effective training due to its constant gradient (which also prevents the vanishing problem[8]) and provides more sparse representations, among others.



(a) Sigmoid, $f(x) = \frac{1}{1+e^{-x}}$ **(b)** Tanh, $f(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$ **(c)** Relu, $f(x) = max(0,x)$

Figure 3.8 – Common non-linear functions applied in CNNs.

In classification problems it is common to use the softmax operation as a non-linear layer in the output layer. This operation causes the output to be interpreted as a probability vector:

$$a^{l+1}_{j,(h,w)} = \frac{\exp\left(a^{l}_{j,(h,w)}\right)}{\sum_{t=1,\cdots,z^{l}_{d}} \exp\left(a^{l}_{t,(h,w)}\right)} \tag{3.6}$$

so that the predicted class becomes the one with highest estimated probability.

Normalization layers

In order to normalize the responses achieved in the network between different neurons there are specific layers called Local Response Normalization (LRN), which incorporates some kind of inhibition scheme in the network based on the lateral inhibition found in biological brains: capacity of a neuron of being highly activated by attenuating the activation of neighbor neurons and, on the contrary, reducing its activation when similar responses are obtained in the neighborhood neurons. This element-wise operation gathers features from η different feature maps (channels) but the resulting set of feature maps keeps the input size. Krizhevsky *et al.* in [70] defined this inhibition as:

[8] The vanishing problem appears on gradient-based methods when gradients are within the range (-1,1) so that a successive multiplication of them decreases considerably, tending to zero quickly.

$$a^{l+1}_{j,(h,w)} = a^{l}_{j,(h,w)}\Big(\kappa + \alpha \sum_{t=max(0,-\eta/2)}^{t=min(z^l_d,-\eta/2)} (a^l_{t,(h,w)})^2\Big)^{-\beta} \tag{3.7}$$

where α, β and κ are hyper-parameters. Although in [70] LRNs have demonstrated to improve accuracy results, in [124] concluded that its utility does not benefit beyond memory and computation efficiency and LRNs ave been overcome by other techniques such as batch normalization[9] or dropout[10].

3.2.2 CNN arithmetics

In previous subsection the operations computed by each kind of layer are described but we have not specified how they modify input representation sizes or which is the region used for obtaining a specific activation. Here we expose how to calculate these properties.

Output feature maps size

Depending on the parameter settings and the type of the layer l, the size of its output feature maps will be three-dimensional structure of size $z^l = z^l_h \times z^l_w \times z^l_d$, with height and width (*i.e.*, $z^l(i)\ \forall i = 1,2$) :

$$z^l(i) = \begin{cases} floor\Big(\frac{z^{l-1}(i)+p^l(2(i-1)+1)+p^l(2(i-1)+2)-k^l(i)}{s^l(i)}\Big)+1, & \text{if } l \text{ is conv. or pool.} \\ z^{l-1}(i), & \text{otherwise} \end{cases} \tag{3.8}$$

and depth:

$$z^l_d = z^l(3) = \begin{cases} k^l_q, & \text{if } l \text{ is convolutional} \\ z^{l-1}(3), & \text{otherwise} \end{cases} \tag{3.9}$$

Note that only convolutional and pooling layers can modify input sizes and, from the first fully-connected layer, feature maps should have $z^l_h = z^l_w = 1$, due to $k^l(i) = z^{l-1}(i) + p^l(2(i-1)+1) + p^l(2(i-1)+2)$, $\forall i = 1,2$. As a general norm, feature maps sizes are decreased through layers.

[9] Technique performed across different images used in the same training batch that minimizes the covariance shift (mean activations close to 0 and standard deviation close to 1), proposed in [58] showing a benefit on the training computational time.

[10] Technique to overcome the overfitting problem by disabling some random neurons during the training process [129].

As a particular comment, let us to remark that when division in Eq. 3.8 is not an integer value could be considered an invalid configuration of the architecture, since neurons do not symmetrically analyze the entire image.

Receptive field size (or intermediate visual field size) of an activation

Going back to Fig. 3.6, there is somehow represented another concept (and explicitly shown in Fig. 3.9) which has not yet been commented in detail: the receptive field size (r) is (usually) increased through layers. Assuming that in this figure hyper-parameters are set to $k^1 = [3,3,1,2]$, $k^2 = [3,3,2,4]$, $p^{1,2} = [0]$ and $s^{1,2} = [1]$ the receptive field size of a neuron in the second convolutional layer is $r^2 = 5 \times 5$ while $r^1 = 3 \times 3$; although both layers consist of filters of sizes 3x3 pixels. It means that an activation obtained from a neuron in the second layer responds to 5×5 pixels in the input image.

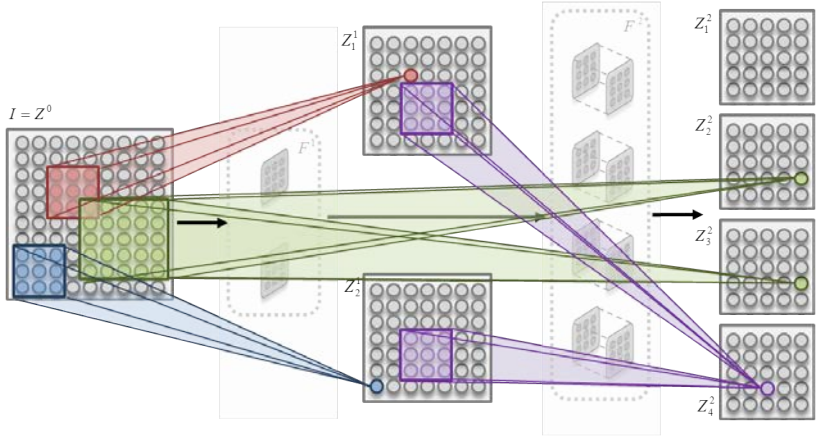


Figure 3.9 – Receptive Field (and visual field) illustration of a CNN with two convolutional layers with 3×3 filters. Receptive field sizes are (usually) increased through layers.

Isolating from Eq. 3.8 and Eq. 3.9, and taking into account that padding parameter has no effect on computing the visual field of an activation, the receptive field size of any activation obtained from a layer l can be computed with iterative process shown in Algorithm 1.

Corresponding receptive field (or intermediate visual field) of an activation

Following with the connection between an activation and its receptive field, here we address to specify the set of pixels in the input image that are described

Algorithm 1 Receptive field size of any activation of layer l

$l_t \leftarrow l$
while $l_t > 0$ **do**
  if l_t is convolutional or pooling **then**
    $v_h^{l,l_t-1} \leftarrow v_h^{l,l_t} \times s^{l_t}(1) - s^{l_t}(1) + k^{l_t}(1)$
    $v_w^{l,l_t-1} \leftarrow v_w^{l,l_t} \times s^{l_t}(2) - s^{l_t}(2) + k^{l_t}(2)$
  else
    $v_h^{l,l_t-1} \leftarrow v_h^{l,l_t}$
    $v_w^{l,l_t-1} \leftarrow v_w^{l,l_t}$
  end if
  if l_t is convolutional **then**
    $v_d^{l,l_t-1} \leftarrow k_d^{l_t}$
  else
    $v_d^{l,l_t-1} \leftarrow v_d^{l,l_t-1}$
  end if
  $l_t = l_t - 1$
end while
return $v^{l,0}$

in a specific layer l with an activation $a^l_{j,(h,w)}$, *i.e*, to define $R(a^l_{j,(h,w)})$. This can be obtained from an iterative process as shown in Algorithm 2. Note that $\forall\, j$ all the set of activations $\{a^l_{j,(h,w)}\}$ are obtained from the same receptive field (and the same intermediate visual fields).

Note that if some padding is added in some layer, activations allocated in the borders of a feature map have a visual field with respect to a previous layer that contains part of the margin added by the padding.

Total padding with respect the input image

We finished last section with the observation that some receptive fields (or intermediate visual fields) can fall into a padding region although we have not specified the amount of padding pixels to be added on the input image, which is necessary to correctly allocate the region obtained through Algorithm 2 on the input image. Using Algorithm 3 we can obtain the amount of pixels to wrap input image in the top, bottom, left and right directions (add).

Algorithm 2 Receptive field of an activation of layer l

$h_1^l \leftarrow h$ ▷ (h, w) are the coordinates of the activation
$w_1^l \leftarrow w$
$h_2^l \leftarrow h$
$w_2^l \leftarrow w$
$l_t \leftarrow l$
while $l_t > 0$ **do**
  if l_t is convolutional or pooling **then**
    $h_1^{l_t-1} \leftarrow s^{l_t}(1) \times (h_1^{l_t} - 1) + 1$ ▷ If $h_1^{l_t-1}$ <= $p^{l_t}(1)$, V^{l,l_t-1} contains padding pixels.
    $w_1^{l_t-1} \leftarrow s^{l_t}(2) \times (w_1^{l_t} - 1) + 1$ ▷ If $w_1^{l_t-1}$ <= $p^{l_t}(3)$, V^{l,l_t-1} contains padding pixels.
    $h_2^{l_t-1} \leftarrow s^{l_t}(1) \times h_2^{l_t} + k^{l_t}(1) - 1$ ▷ If $h_2^{l_t-1} > z_h^{l_t-1}$, V^{l,l_t-1} contains padding pixels.
    $w_2^{l_t-1} \leftarrow s^{l_t}(2) \times w_2^{l_t} + k^{l_t}(2) - 1$ ▷ If $w_2^{l_t-1} > z_w^{l_t-1}$, V^{l,l_t-1} contains padding pixels.
  else
    $h_1^{l_t-1} \leftarrow h_1^{l_t}$
    $w_1^{l_t-1} \leftarrow w_1^{l_t}$
    $h_2^{l_t-1} \leftarrow h_2^{l_t}$
    $w_2^{l_t-1} \leftarrow w_2^{l_t}$
  end if
  $ch_1^{l_t-1} \leftarrow 1$ ▷ Common architectures use all the feature maps
  $ch_2^{l_t-1} \leftarrow z^{l_t-1}$
  $V^{l,l_t-1} \leftarrow [(h_1^{l_t-1}, w_1^{l_t-1}), (h_2^{l_t-1}, w_2^{l_t-1}), (ch_1^{l_t-1}, ch_2^{l_t-1})]$
  $l_t = l_t - 1$
end while
return $V^{l,0}$

3.2.3 Learning weights: backpropagation algorithm

In previous sections we have described technical details of Convolutional Neural Networks likewise their architecture and their different types of layers that operate with their input in a specific way. Nevertheless if these techniques have become popular and one of the most successful models on solving computer visual problems is, apart from their intrinsic hierarchical architecture, due to their ability on learning the optimal visual features for solving the visual task. Therefore, in this section we summarize how each neuron is adjusted to search for a specific visual task (*e.g.*, an oriented edge) and to be connected in a specific way with previous neurons.

In a preliminary stage, neuron weights are initialized with some method (usually,

Algorithm 3 Total padding of a layer l

$tmp \leftarrow [1,1,1,1]$
$add \leftarrow [0,0,0,0]$
$l_t \leftarrow 1$
while $l_t < l$ **do**
  if l_t is convolutional or pooling **then**
    $add(1) \leftarrow add(1) + p^{l_t}(1) \times tmp(1)$ ▷ Top margin
    $add(2) \leftarrow add(2) + p^{l_t}(2) - \text{mod}\big(z_h^{l_t-1} + p^{l_t}(1) + p^{l_t}(2) - k_h^{l_t}, s_h^{l_t})\big)$ ▷ Bottom
    $add(3) \leftarrow add(3) + p^{l_t}(3) \times tmp(2)$ ▷ Left margin
    $add(4) \leftarrow add(4) + p^{l_t}(4) - \text{mod}\big(z_w^{l_t-1} + p^{l_t}(3) + p^{l_t}(4) - k_w^{l_t}, s_w^{l_t})\big)$ ▷ Right
    $tmp(1) \leftarrow s^{l_t}(1) \times tmp(1)$
    $tmp(2) \leftarrow s^{l_t}(2) \times tmp(2)$
  end if
  $l_t \leftarrow l_t + 1$
end while
return add

using a random initialization) as well as their biases (generally initialized by 0), so that they do not contain any visual feature encoded. These parameters are updated during the learning stage. This is habitually performed with a gradient-based procedure known as stochastic gradient descend and using backpropagation, which follows the idea of computing the gradient with respect to the input by propagating gradients through all layers starting from the output of the network and successively propagate gradients until the first layer [73] thanks to the chain rule.

The training starts with a *forward pass* which analyses an input image I through the entire network and provides an output (Z^l). Note that the network is a function $g^l(I) = Z^l$, as a result of a composition of several functions:

$$g^l(I) = (g^l \circ g^{l-1} \circ \cdots \circ g^1)(I), \tag{3.10}$$

where g^i is the function applied by the layer i.

Since network parameters (weights and biases (θ)) are initially randomized (or set null for the biases), the output mapping of the first iteration will be distant from the desired, so that this error should be measured. Therefore, during the training and whatever the type of training (supervised or unsupervised), is necessary to define a *loss function* ($\mathscr{L}$) be optimized[11]. This loss function measures how well

[11] Often, loss functions are combined with some regularizations (a penalty term) in order to deal with the overfitting problem and the optimization algorithm is performed considering both concepts.

the method fits the data to solve a specific problem by grading some property and should be defined accordingly with the activation function used at the output layer and also with the task to solve. For example, within the group of supervised learning, a CNN that is applied to solve image classification, a useful loss function to minimize is the softmax loss or the cross-entropy loss (such as in [70, 149]) or the hinge loss function (*e.g.*, in [20]) to compare the predicted class with the expected one. Contrary, L2 squared norm is commonly used for learning image representation (likewise in [64, 151]) or a log-likelihood for generative models (*e.g.*, in [47]).

It is somehow expected to train the network in order to accomplish its goal of solving the task without errors and this idea is carried out by upating the parameters θ along the negative direction (gradient) of the loss function:

$$\theta_{i,j}^{l} = \theta_{i,j}^{l} - \eta \frac{\partial \mathscr{L}}{\partial \theta_{i,j}^{l}} \tag{3.11}$$

being η the learning rate[12]. Each partial derivative is indicating how each parameter influences to the loss function, so that the error calculated at the output is distributed back through the network layers. The backpropagation procedure gives an efficient way to compute these partial derivatives by using the chain rule:

$$\frac{\partial \mathscr{L}}{\partial \theta_{i,j}^{l_t}} = \frac{\partial \mathscr{L}}{\partial g^{l}} \frac{\partial g^{l}}{\partial g^{l-1}} \cdots \frac{\partial g^{l_t+1}}{\partial g^{l_t}} \frac{\partial g^{l_t}}{\partial \theta_{i,j}^{l_t}} \tag{3.12}$$

By doing these updates successively for several iterations the architecture is adjusting its parameters and, therefore, defining the optimal visual features to solve the specific problem.

[12] The *learning rate* is a value to specify how fast the set of weights have to be adjusted.

4 CNN visualization and Neuron Feature

In parallel with the success of CNNs to solve vision problems, there is a growing interest in developing methodologies to understand and visualize the internal representations of these networks. How the responses of a trained CNN encode the visual information is a fundamental question for computer and eventually for human vision. Image representations provided by the first convolutional layer as well as the resolution change provided by the max-polling operation are easy to understand, however, as soon as a second and further convolutional layers are added in the representation, any intuition is lost.
This chapter reviews the state-of-the-art approaches for understanding the intermediate features and introduces the *Neuron Feature* as a new proposal for visualizing the main intrinsic properties encoded by each neuron.

4.1 Introduction

The evolution of going from flat to hierarchical descriptions (specifically, Convolutional Neural Networks) has been accompanied with the lack of understanding visual representations. Therefore, within the new paradigm of computer vision where advances have moved away from designing handcraft features using deep learning architectures and achieving promising performance, there is a growing interest in developing methodologies to understand and visualize the internal representations of these techniques. However, this is a complex task due to the generality achieved with these kind of approaches, their inherent depth and the non-invertible operations that they apply.
In this chapter we introduce the problem of visualizing intermediate features by reviewing the state-of-the-art approaches in Section 4.2, which is generally addressed by proposals that are image-dependent or methods that synthesize an image using an optimization algorithm that finishes with artificial artifacts. Trying to overcome these problems, in Section 4.3 we propose a new methodology to get an image showing the intrinsic features of each neuron (called *Neuron Feature*), which seeks to get an image visualization by keeping natural image properties and by not beeing dependent of a single input image. Some visualizations are shown for

neurons of the VGG-M network of Chatefield *et al.* [20] (see Section 4.3.2) trained in the well-known ImageNet dataset [114] (see Section 4.3.1). We emphasize the potential of our visualization by decomposing images in their most relevant and non-overlapped neuron features through each convolutional layer in Section 4.4 and improving the understanding of deeper neurons through their hierarchical composition of shallower neurons in Section 4.5. We enclose this chapter with a discussion of the conclusions in Section 4.6.

4.2 State-of-the-art: Visualizing features

The remarkable increase in performance of Convolutional Neural Networks (CNNs) to solve computer vision problem is somehow diminished by the lack of understanding of the internal representations that capture the intrinsic image features, as a natural consequence of an automatic feature learning to achieve the goal task. Each of the stacked layers of the architecture is operating on their inputs to produce a representation change. Taking into account that convolutional layers are the main responsible of detecting visual features encoded through their set of neurons, these representation changes are done in terms of the features encoded in each layer, likewise each neuron is codifying some feature based on the previous convolutional layer feature space. This relationship is what can be derived from the weights of the neuron. As well as the effects of the first convolutional layer can be easily understood, the understanding of the learned features becomes more difficult through deeper layers. This unawareness has been promoting the interest on understanding and analyzing the learned features and several works have proposed different methodologies to address this understanding problem, going beyond proposing different CNN architectures or learning techniques.

Recently, in [80] two main groups of works are mentioned. On one side those works that deal with the problem from a *theoretical* point of view. These are works such as [94] where kernel sequences are used to conclude that deep networks create increasingly better representations as the number of layer increases, [108] which explains why a deep learning network learns simple features first and that the representation complexity increases as the layers get deeper, [48] where an explanation for why an adversarial example created for one network is still valid in many others and they usually assign it the same (wrong) class, or [4] that presents algorithms for training certain deep generative models with provable polynomial running time. On the other side, an *empirical* point of view, which comprises approaches that pursuit methodologies to visualize intermediate features in the image space, or approaches that analyze the effect of modifying a given feature map in a neuron activation. Our work is framed in the first subset of empirical

approaches.

Visualizing intermediate features seeks to describe the activity of individual neurons. This description is the basis of this thesis hypothesis that is based on the idea that a proper understanding of the activity of the individual neurons allow us to draw a map of the CNN behavior. This behavior can be understood either in terms of relevant image features or in terms of the discriminative power of the neurons across the full architecture.

The first and most obvious way to describe the activity of a single neuron is given by the inherent set of weights of the learned filters. These weights can be used to compare neurons between them, either within the same layer or versus neurons in similar CNNs which have been trained under different initialization conditions, as it is proposed by [80]. A direct visualization of these weights is intuitive when they belong to neurons of a first convolutional layer. However, when layers are stacked, this intuition disappears and the capability to understand the neuron activity is lost. Firstly, due to the high dimensionality of the neurons structure, and secondly, because of the fact that the deeper the neuron, the bigger receptive field has, so that the understanding of its intrinsic feature directly by the filter weights is too compressed for its comprehension.

A second method to describe neuron activity is projecting the filter weights into the image space, trying to get the inherent feature that maximally activates the filter. The projection can be computed by composing the inversion of the layer operators under a specific neuron towards the image space. In this line we, developed an approach to obtain what we named Decoded Filter (DF), the projection of the neuron into the image space. The resulting image represents an estimation of the feature that should highly activate such neuron. The decoding algorithm inverts filter weights independently of any image. To this end, the effect of a convolution layer is inverted by computing the deconvolution operation[1] [150] between a neuron with the set of neurons in the previous layer. On the other hand, the non-invertible operation of pooling layers is estimated through a simple upsampling of the representation. This approach would give a good estimation of the feature image if most of the layer operators were invertible. However, when the number of non-invertible operators increases, the estimation becomes unintelligible. The projection of the filter itself have also been explored in [128] for architectures with no pooling layers since pooling is the less invertible operator. They point out the interest of obtaining such a representation, since it would allow the understanding of neuron activity independently of the input image. However, the majority of proficient CNNs contain pooling layers.

A third way to describe neuron activity is by exploring the images that maximally

[1] The deconvolution operation is a convolution with the filter transposed.

activate the neuron. In this sense, authors in [44] sorted images from the highest to the lowest activation and display the set of receptive fields of the top-scoring images. From the observation of each set of top-scoring receptive fields, they can give an intuition about to which kind of features is each neuron selective to, extracting common features from them. They observe that some neurons were related to concepts (such as faces or text) and others to material properties (such as specular reflections). Nevertheless, this conceptualization is a human post-process that might be biased: is the neuron really selective to an entire face shape or its intrinsic feature is just the composition of two-sided blobs (eyes) with an horizontal edge (mouth) in the bottom defining a shape structure that is more commonly found in faces? Following this idea, appeared one of the most relevant works pursuing the visualization of intermediate features, the one proposed by Zeiler and Fergus in [149]. They also use the top-score images to understand the behavior of each neuron but, instead of visualizing their receptive fields, they use the neuron representation for each top-score image and project them into the image space. For this purpose, they isolate the strongest activation inside the feature map obtained by the neuron of interest. Through the deconvolution approach [150], they invert convolutional layers and, for reversing max-pooling layers, they use the switches[2] to replace each representation in the correct place. With this, they are able to obtain a visualization in the images space of the feature of the receptive field that provoke a high activation to the studied neuron. By observing different projections that maximally activate a certain neuron they get the intuition about the main features learned on the network. Later on, in [128] the guided backpropagation improved Zeiler and Fergus approach with a new way of inverting rectified linear (ReLU) nonlinearities, achieving better visualizations of the activations. The guided back-propagation is also based on the idea of keeping localizations (switches), but focused on those pixels where ReLU layers discard the activations during the forward pass. In this way, they apply the ReLU operation (likewise in the forward pass) and also refuse (set to 0) any pixel value belonging to the switches when inverting the representations through this non-linear layer. These approaches present a main drawback, their feature visualization is image-specific, since the maximum activation of a neuron not always generalize the intrinsic feature of the neuron, or should be extrapolated as a post-process by observing commonalities of the set of top-score images (or representations). To solve this problem, in some works instead of using the image that provokes the maximum activation, they use optimization techniques to generate an image that maximizes the activation. The key point of these works is to use an appropriate regularization in the generation

[2] In [151] the term of *switch* refers to the localization (coordinates) where each maximum came from during the forward pass.

process, otherwise, the resulting image appearance is unrealistic and difficult to understand. This idea was firstly proposed for authors of [35] but focused on two deep learning models different from CNNs (the Deep Belief Networks [53] and the Stacked Denoising Auto-Encoder [138]). Their research was still focused on looking for the response of each individual unit but, using a gradient ascent algorithm, they synthesize an image which provokes a high activation of a specific neuron. This was afterwards extended to CNNs by authors of [123], who propose a method to generate an image which is representative of a certain final class by maximizing the score of this image to be classified in a certain class (or highly activates the specified neuron) with an L_2-regularization. A similar work was performed in [147], but taking advantage of combining three different regularizations to achieve more recognizable images. They show results for neurons in any layer (not just the ones in the last convolutional layer). Nevertheless, Nguyen *et al.* showed in [100] that generated synthetic images that maximally activate a specific neuron may have unrecognizable structures or simply with unnatural patterns far away from the inherent nature of any object. The work [48] concluded that the local linear behavior of CNNs might cause these high activations from unrealistic images. Recently, these generative approaches have been extended in [99] to contemplate the hypothesis that a neuron may be highly activated by different intrinsic features or different patterns (named *multifaceted*). Under this criterion, they give several representational images that may highly activate a specific neuron, capturing, in this way, the intrinsic feature variability linked to each neuron. Although this set of works have explored different regularizations to achieve more realistic intrinsic feature representations, their visualizations present important artifacts that complicate the understanding of the intrinsic property.

Finally, other works focus on proposing approaches able to reconstruct the input image given a feature map, going further of analyzing the individual neuron activity. In [89], authors make use of optimization algorithms to search for an image whose feature map best matches a given feature map by incorporating natural image priors. Contrary, in [34], the authors propose to reconstruct the input image from its feature maps of a given convolutional network by training a new deconvolutional network to learn filter weights that minimize the image reconstruction error when these filters are applied to the image feature maps. With this approach they are also able to get an image reconstruction with natural priors.

In the second subset of empirical approaches, [33] train a generative deconvolutional network to create images from neuron activations. With this methodology, the variation of the activations enables the visualization of the differences in the generated images. A similar analysis is done by [5], but instead of forward-propagate different activations to the image space and comparing them, they observe the changes on neuron activations when similar computer-generated images with dif-

ferent scene factors are introduced into a CNN. These works contribute in giving a deeper understanding on the internal CNN behavior. Both works conclude that there are specific neurons which are sensitive to color changes, point of views, scale or lighting configurations.

Likewise, in [149] in this chapter we pursuit visualizing the intrinsic feature of a neuron by analyzing the images that maximally activates a specific neuron. However, to avoid the lack of generality of this approach, we define the *Neuron Feature* (see Section 4.3) which is not based on a single maximum activation. The Neuron Feature is a weighted average version of a set of maximum activation images that capture the essential properties shared by the most important activations and makes it not to be image-specific. Additionally, our Neuron Feature overcomes the problem of unrealistic representation we mentioned earlier, by directly averaging on the image space. In this way we achieve two main advantages: (a) keeping the properties of the natural images, and (b) providing a very straightforward approach to compute it.

4.3 Neuron Feature visualization

In this section we finally present our approach for visualizing the intrinsic features of each individual neuron, which will allow to describe the neuron activity for each unit. As we already mentioned, we propose to visualize the image feature that activates a neuron, whenever is possible, by directly computing a weighted average of the N-th first receptive field images that maximally activate this neuron (also labeled as cropped images), likewise sorted in [44]. We will refer to this visualization as the *Neuron Feature* (NF).

In order to build the NF we first need to calculate the activations associated to each individual neuron. For each neuron we select the set of images that achieve a minimum normalized activation value but constrained to a maximum number of images for practical reasons[3]. By normalized activation ($\hat{a}$) we mean the value of the maximum activation of a neuron for a specific input image (I_i), which is normalized by the maximum of these values achieved by the same neuron over all the images in the dataset:

$$\hat{a}^l_j(I_i) = \frac{\max\left(a^l_{j,(h,w)}(I_i)\right)}{\max\left(a^l_{j,(h,w)}(I_t)\right)}, \forall (h,w) \in [(1,1),\cdots,(z^l_h, z^l_w)], \forall I_t \in \text{dataset} \quad (4.1)$$

In Fig. 4.1 we can see the behavior of the ranked normalized responses of a

[3] In this thesis we use a maximum number of images equal to $N_{max} = 100$ and a minimum activation value over a 70% of the maximum activation for computing the NF. We plot these values on Fig. 4.1

subset of neurons for every convolution layer of the VGG-M CNN (See Sec. 4.3.2) trained on ImageNet by [20] (See Sec. 4.3.1). The y-axis represents the normalized activation value of a single neuron to an image of the dataset. Images are ranked on the x-axis according with their activation value, from highest to lowest activation (we just plot the first 400 images for each neuron). Therefore, the first normalized activation value is always 1 for all neurons and then its values decrease monotonically. This normalization allows to compare different neuron behaviors, from neurons which are activated by most of the images (flatter behavior), to neurons that are highly activated only for a subset of images and have very little activation for the rest (steeper behavior). In this figure we also provide the percentage of area for each plotted curve. This percentage is computed over the area of the neuron that presents the maximum AUC in the entire architecture. We can observe different behaviors in all layers. In general, we can state that in deeper layers the behavior of the neurons is steeper (lower AUC), *i.e.*, neurons highly spike for a small number of images. However, in shallower layers the behavior is flatter, *i.e.*, neurons highly spike for a lot of images. This is an expected behavior, since the image features spiking neurons in first layers (*e.g.*, oriented edges) are shared by almost all the images, while the features spiking shallow neurons are more selective features (*e.g.*, faces) that only spike for specific images. The observation of the responses confirms the adequacy of our assumption to fix a minimum value for the activation (here, we use $\hat{a} \geq 0.70$) and a maximum number of images to capture the most important activations for all the neurons (we use $N_{max} = 100$).

Thus, the NF is computed as:

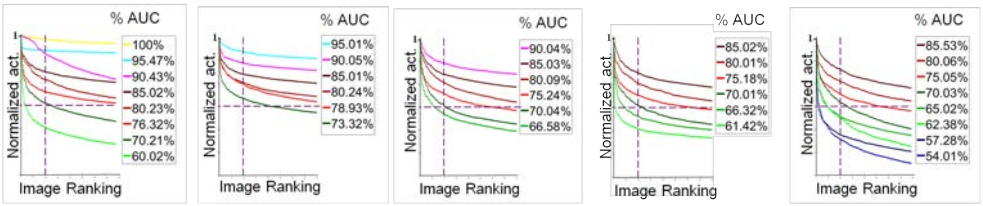
$$NF(F_j^l) = \frac{1}{N_{max}} \sum_{t=1}^{N_{max}} \hat{a}_j^l(I_t) \times I_t \tag{4.2}$$

where $\hat{a}_j^l(I_t)$ is the normalized activation of the t-th cropped image, denoted as I_t, of the j-th neuron F_j^l) of a layer l.

With this computation we pursuit on visualizing shared features between the top-scored images preserving as much as possible a realistic and natural appearance of the visualization.

4.3.1 The ImageNet Dataset

The success obtained with Convolutional Neural Networks has been achieved with the existence of large datasets needed for training the huge space of parameters insight them. Computer vision field was used to work in small datasets (such as PASCAL Visual Object Challenge dataset [36], which consists of 20 object classes)



(a) Conv1 **(b)** Conv2 **(c)** Conv3 **(d)** Conv4 **(e)** Conv5

Figure 4.1 – Normalized activations of a subset of neurons for the first 400 ranked images through all convolutional layers. For each layer we plot the normalized activation for the neurons with highest and smallest AUC (Area Under Curve), and some other examples in between these extremes. For all neurons the highest normalized activations is 1, and the percentage of AUC is computed with respect to the neuron AUC achieving the biggest area in the entire network. Doted lines indicate the constrains we use for computing the NF. The vertical corresponds to the maximum number of images to take into account for its building (N_{max} = 100), while the horizontal one is the minimum activation to be achieved to be considered (Th=70%).

during the first decade of the 21st century, achieving a good progress on its projects. The ambition of having enough data for representing the real world (which obviously contains more than 20 objects) encouraged researchers to collect millions of real world images of thousands of classes and to label them. This effort gave rise to the well-knwon ImageNet dataset [114], which is a large visual dataset where images are classified according to the lexical WordNet hierarchy [92]. Experiments performed in this thesis are done on the ILSRVC12[4] version of this dataset. It consists of around 1.2M images labeled in 1.000 different object categories. Images are mostly given in uncalibrated RGB color and some of them are gray-level and they can present different sizes within the dataset. Sensors used in acquisition are unknown for each image, thus there is no chance to work on any RGB calibrated color space. Moreover, it is built from a huge variety of scenes, including objects belonging to 1000 different categories (such as dog classes, clocks, flower classes or type of buildings amongst others). This diversity of objects can moreover appear with large variations in size, points of view, poses, backgrounds and a large range of lighting conditions (indoor or outdoor).

[4]Images can be browsed on http://image-net.org/challenges/LSVRC/2012/browse-synsets, as well the concepts (*synsets*) that describe each image.

4.3.2 Case of study: the trained network VGG-M

In this thesis we analyze the neurons of a CNN architecture trained on the ILSRVC12 version of the ImageNet, although our proposals can be applied in most of current networks that follows a sequential scheme. We report the results for the VGG-M CNN[5] that was trained by Chatfield *et al.* [20] for a generic visual task of object recognition. The supervised training step of this network was performed through the minimization (with a gradient descent technique) of a loss function representing the classification error over 1000 object categories. The training process provides us a large set of fitted filter weights associated to each neuron and the parameters of the functions involved in this architecture, achieving a 36.9 and 15.5 top-1 and top-5 errors, respectively. Details of the CNN architecture are given in table 4.1[6]. We selected this network since it has a similar structure to those which have been reported as having a representational performance that competes with human performance (as was proved in [17]).

The network expects inputs of $224 \times 224 \times 3$ pixels, with images represented in RGB. Due to the fact that the dataset contains images with diverse sizes, they are scaled to fit the constraints of the network, and gray-level images are channel-wise replicated. From Table 4.1 note that the receptive field from the 16th layer to the last is bigger ($331 \times 331 \times 3$) than the input size ($224 \times 224 \times 3$), as a consequence of have added some paddings along the architecture.

Regarding the convolutional layers, the network presents eight different layers. Nevertheless the fifth firsts are devoted to the representational task, while the remaining three are used for the classification tasks (note that these are fully connected layers and are named as *FC*). Most of the studies done in this thesis are focused in the convolutional layers that are trained to act as feature detectors (Conv1, Conv2, Conv3, Conv4 and Conv5).

4.3.3 Examples of Neuron Features

Once seen how to compute our visualization, in this section we plot some neurons through their corresponding Neuron Features.

In Fig. 4.2 we can see some NFs and their corresponding set of first 100 maximum activations, and in Fig. 4.3 (*left*) we can see a selected subset of 24 NF per layer. In this image we can identify specific shapes that display the intrinsic property that

[5] For the MatConvNet library [137], the trained VGG-M can be downloaded from http://www.vlfeat.org/matconvnet/models/imagenet-vgg-m.mat

[6] In Conv2 we have detected 37 dead neurons and they are discarded on this plot. By dead neurons we mean neurons whose activation for any input image is not enough to drive the subsequent ReLU (*i.e.*, negative activation). Therefore, results reported in these thesis considers that COnv2 has 219 neurons instead of 256.

(a) Conv1 **(b)** Conv2 **(c)** Conv3 **(d)** Conv4 **(e)** Conv5

Figure 4.2 – Neuron Feature (NF) visualizations (top) for 5 neurons of the different convolutional layers of VGG-M with their corresponding 100 cropped images (bottom). Note that NFs are scaled bigger than the set of 100 top-scored for a better understanding, but each one of the top-scored images has, originally, the same size as the corresponding NF. See in Table 4.1 the correct receptive field size (r) for each layer to know the real size of these figures.

fires a single neuron. At first glance, we can see how in this particular network the first two layers are devoted to basic properties. Oriented edges of different frequencies and in different colors in the first layer; textures, blobs, bars and more specific curves in the second layer. The rest of the layers seem to be devoted to more complex objects. We can see that dog and human faces, cars and flowers are detected at different scales in different layers, since the size of the NF and their corresponding receptive fields increase with depth. This visualization of the neuron activity can be seen as a way to visualize a trained vocabulary of the CNN that opens multiple ways to analyze the global behavior of the network from its single units. However, not all neurons present such a clear tuning to an identifiable shape. Some neurons present a blurred version of NF, such as, those in the right of Fig. 4.3. The level of blurring is directly related to a high variability between the maximally activated images for a neuron.

At this point, we want to make a short parenthesis to relate the previous representational observations with the scientific problem about neural coding that is focus of attention in visual brain research ([68]). We are referring to the hypothesis about distributed representations that encode object information in neuron population codes, that co-exist with strong evidences of neurons which are only activated by a very specific object. In line with this idea, we invite to speculate about neurons presenting a highly structured NF could be closer to localist code neurons while neurons with a blurred NF as closer to a distributed code. This neuron variability has been recently explained in terms of neuron multifacets by [99] and they have

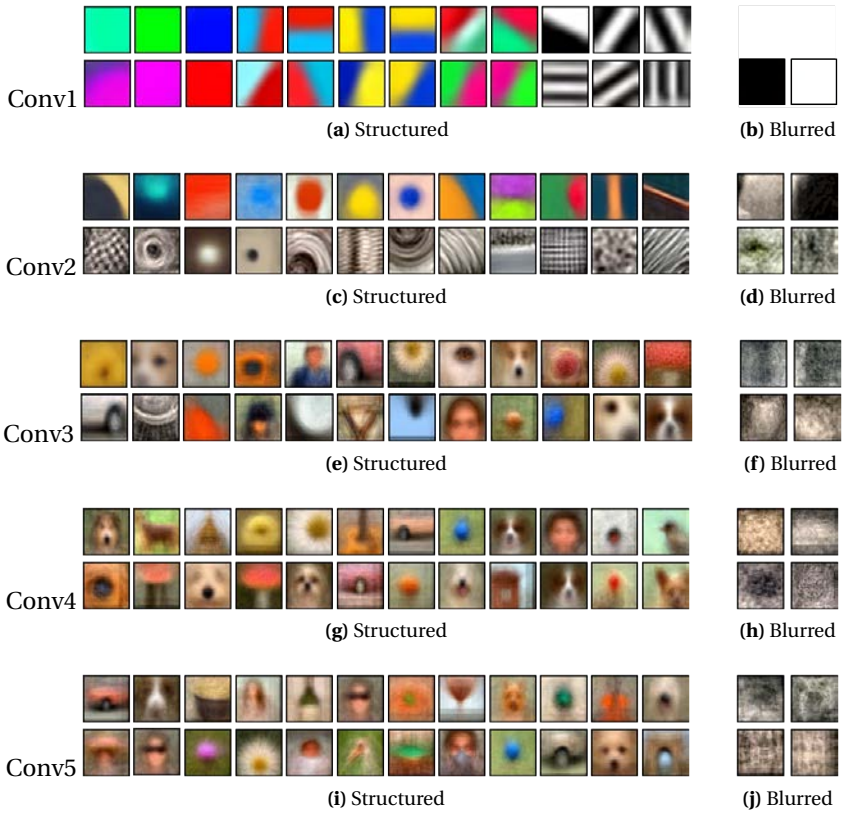
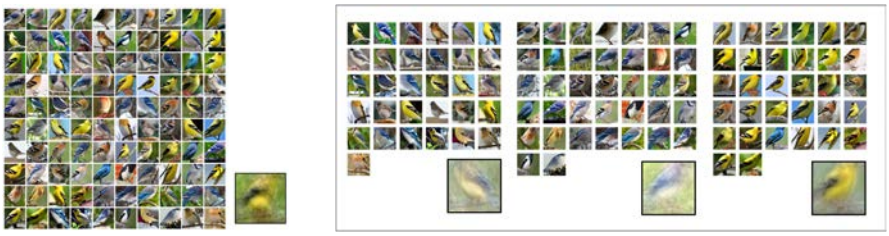


Figure 4.3 – Examples of NFs for each convolutional layer of the network VGG-M (see section 4.3.2. (*Left*) 24 examples of structured NF, (*Right*), blurred NFs. Although sizes of NF increments through layers, we scale them into the same size. Original sizes are: $7 \times 7 \times 3$, $27 \times 27 \times 3$, $75 \times 75 \times 3$, $107 \times 107 \times 3$ and $139 \times 139 \times 3$ for Conv1, Conv2, Conv3, Conv4 and Conv5, respectively (see Table 4.1).

introduced a class of algorithms for neuron multifaceted feature visualization. In this line, our NF visualization can be adapted to the multifacets of each neuron by clustering top-scored images from their output feature map of the neuron layer (like in [99]), *i.e.*, applying a k-means approach to the output maps and computing the NF for each cluster (see Fig. 4.4). Each multifaceted representation is visualizing spatial feature similarity and can describe the variability allowed by the neuron.



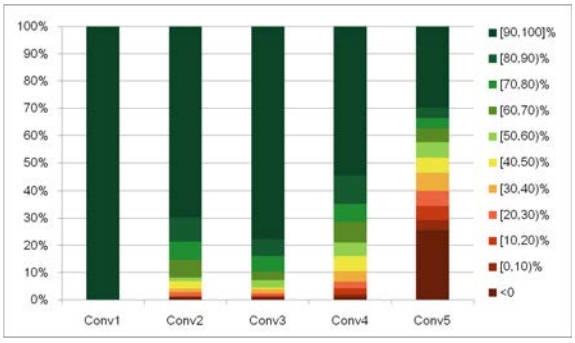
(a) Original NF **(b)** Multifaceted representation

Figure 4.4 – NF visualization (a) without considering multifacets, and (b) for each cluster on the representation. Birds are classified by visual features, splitting between left and right profiles, or more bluish and yellowish birds in different groups.

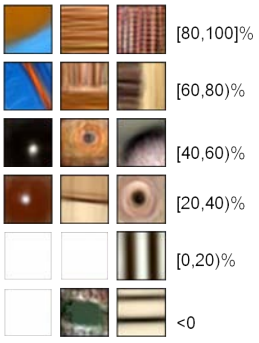
Finally, we want to add a further analysis about how neuron feature is related to the neuron activity is representing. In Fig. 4.5a we plot the level of the neuron responses when the input image is its own NF. This level is determined by comparing the maximum activation achieved by the neuron from overall Neuron Features. We can observe a high degree of activation (in greenish) between the NF and the response of the net to this feature. However we have some disagreements between the NF and the neuron activations, which increase through Conv3, Conv4 and Conv5. That seems to be explained by an increase in invariance that is obvious when the size of the image increases. In fact, from Figs. 4.5b- 4.5e blurred NFs tends to have obtain lower activations on their corresponding neurons, while NFs with a clear structure are able to highly spike the neuron they represents. Note that in Fig. 4.5b all the NF have, apparently, a structured shape but some of them produces a negative activation. Both NFs are build form receptive fields that contain text and this level of detail is diminished on the weighted average. This is another situation showing that shallower layers are focused to image details.

4.4 Relevant NFs of an image

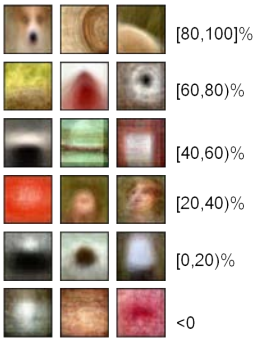
The Neuron Feature visualization can be also useful to visualize how an image is mainly represented through the network. As principal representation we mean the feature encoded by the neuron which achieves the highest normalized activation in a certain location of the image, compared with all neurons of the same layer. Here we provide some examples in Figs. 4.6, 4.7, 4.8, 4.9, where an image of a daisy, cardoon, red wine and school bus categories are shown through these non-overlapped main representations along the five convolutional layers. Note that here



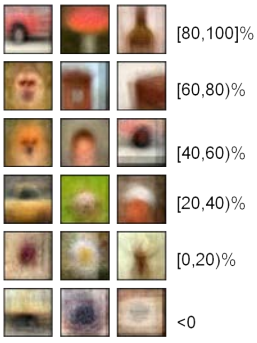
(a) Distribution of the normalized activations of each neuron to their corresponding NF.



(b) Conv2



(c) Conv3



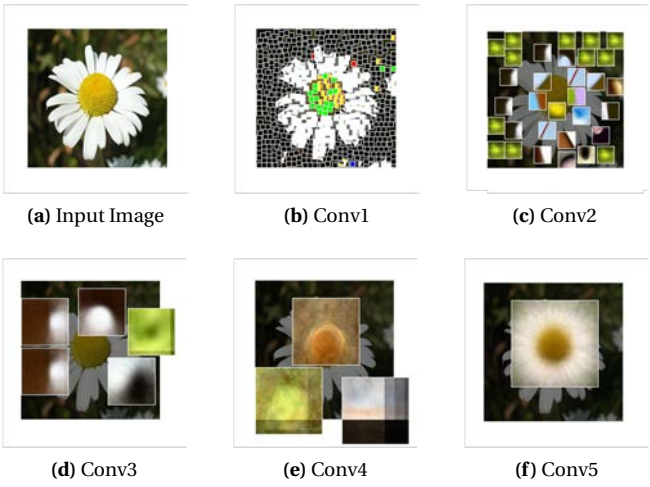
(d) Conv4

(e) Conv5

Figure 4.5 – Normalized activations for each NF on their corresponding neuron. (a) Distribution of the normalized activations along the five convolutional layers. (b)-(e) Examples of NF organized depending on their normalized activation

are plotted a subset of Neuron Features of shallower layers, selecting them from the one with higher activation to the lower such that no overlapping occurs. For each layer, the background is an attenuated version of the original image and it is overlapped with the Neuron Feature of the neuron that is maximally activated. Note how visualized features in the NF describe adequate visual properties for each image. However, this visualization is only showing the principal activation and it is usually accompanied with other activations, so that, their combination allows to describe the complete image. For exmaple, in Fig. 4.9e there is a NF with a wheel of a red vehicle, although the input image corresponds to a yellow bus. Therefore, this

region is mainly described as "there is a wheel" but have also high activations on neurons with a yellowish appearance. A similar situation appears in Fig. 4.6c, where the edges between the stamens region and petals of the daisy flower are described with edges correctly oriented, but with different color appearance (although it is quite similar). On the other hand, this visualization is somehow enhancing the most discriminative features of the image and how they are represented through the layer. For example, in Fig. 4.7 the cardoon representation starts in Conv1 with several homogeneous magenta neurons defining the flower of the plant. The subsequent layers define this region through greenish and magenta oriented edges that finally ends with a neuron feature with a center (and magenta) surround (and greenish) shape.



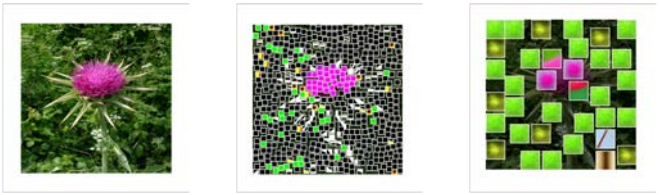
(a) Input Image **(b)** Conv1 **(c)** Conv2

(d) Conv3 **(e)** Conv4 **(f)** Conv5

Figure 4.6 – A daisy image seen through the NFs corresponding to the neuron that is maximally activated on each position.

4.5 Hierarchical feature composition

In the same line, a similar visualization can provide an insight of how shallower neurons are composed to provide a more complex feature in a deeper layer. The set of receptive fields used for building the NF of a specific neuron can be used for obtaining such hierarchy, so that the main representation of each location is the neuron who achieves a maximum weighted mean normalized activation in these



(a) Input Image **(b)** Conv1 **(c)** Conv2

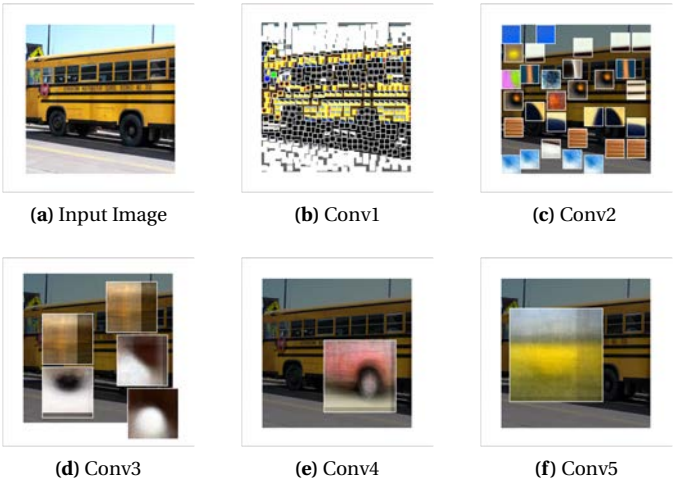
(d) Conv3 **(e)** Conv4 **(f)** Conv5

Figure 4.7 – A cardoon image seen through the NFs corresponding to the neuron that is maximally activated on each position.

(a) Input Image **(b)** Conv1 **(c)** Conv2

(d) Conv3 **(e)** Conv4 **(f)** Conv5

Figure 4.8 – A red wine image seen through the NFs corresponding to the neuron that is maximally activated on each position.



(a) Input Image **(b)** Conv1 **(c)** Conv2

(d) Conv3 **(e)** Conv4 **(f)** Conv5

Figure 4.9 – A school bus image seen through the NFs corresponding to the neuron that is maximally activated on each position.

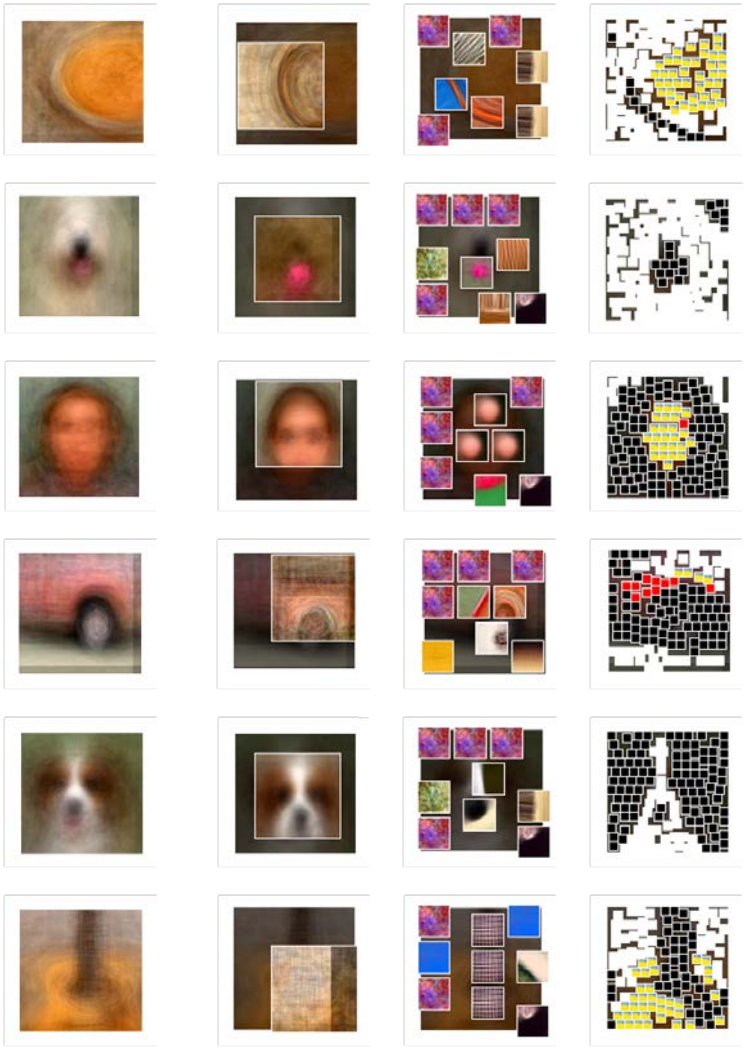
coordinates. This weighted mean normalized activation is the average of the set of normalized activations obtained for each top-score image ($\{I_j\}_{j=1:N_{max}}$) weighted by the corresponding contribution to the building of the NF ($\hat{a}_j(I_j)$).

In Fig. 4.10 and 4.11 we show these composition for six neurons of Conv4 and Conv5 through their previous convolutional layers, respectively. This visualization allows us to better understand the activity of the studied Neuron. In the following lines we report some conclusions that arise from these visualizations. First, the location variability allowed by the neuron to some edges or curves are represented through textured and oriented edges neurons in Conv2 (see bowl neurons on Fig. 4.10 first row and Fig. 4.11 second row). Second, discriminant colors of the Neuron Feature are also highly represented through almost all layers (see the pinkish tongue of the dog or the skin-tone in Fig. 4.10 or the magenta-green and the green-pinkish center surrounds neurons in Fig. 4.11). Third, textures may be relevant in some neurons and these features gain interest on this composition (see the representations of the guitar strings in Fig. 4.10). Other arised observation is that the increasing of the feature complexity is achieved from the consecutive composition of simple features (see how the white and dark edges are relevant to define a dog face neuron in Fig. 4.10, the wine glass and the triumphal arch neurons in Fig. 4.11. Finally, note that some neurons are highly activated by most of the

posterior neurons, such as the white or black neurons in Conv1 or the magenta and grainy texture in Conv5. This kind of neurons have a flat behavior (recover Fig. 4.1) and they provably are not too much informative by their own and they to be understood jointly with another neuron with a high activation.

4.6 Conclusions

Convolutional Neural Networks are deep architectures that sequentially stack layers to increase, in this way, the complexity of the encoded visual features. Each convoloutional layer is somehow defining a new representational space characterized by the encoded features in its set of neurons. Due to the non-invertible operations involved in these techniques, as well as the high dimensionality of these feature spaces and the subsequent compression of information through layers make visualization of intrinsic features a difficult task.
Nevertheless, several works that have been proposed to understand and visualize these intermediate representations are reviewed in this chapter, concluding that they are giving non-realistic images or too much image-dependent. Our Neuron Feature (NF) visualization tries to overcome these problems (although in some cases, the variability allowed by the network provokes a blurred visualization without any recognizable structure) and visualize the intrinsic features for each neuron. Most of the NFs have shown to be good neuron feature representative, considering that they generally provoke a high activation on the neuron they are representing when they are analyzed through the network. Moreover, NFs can be used to understand which features are more representatives in intermediate levels of the network or even to visualize a composition neurons of previous levels adding more useful information to the comprehension of the neuron activity.

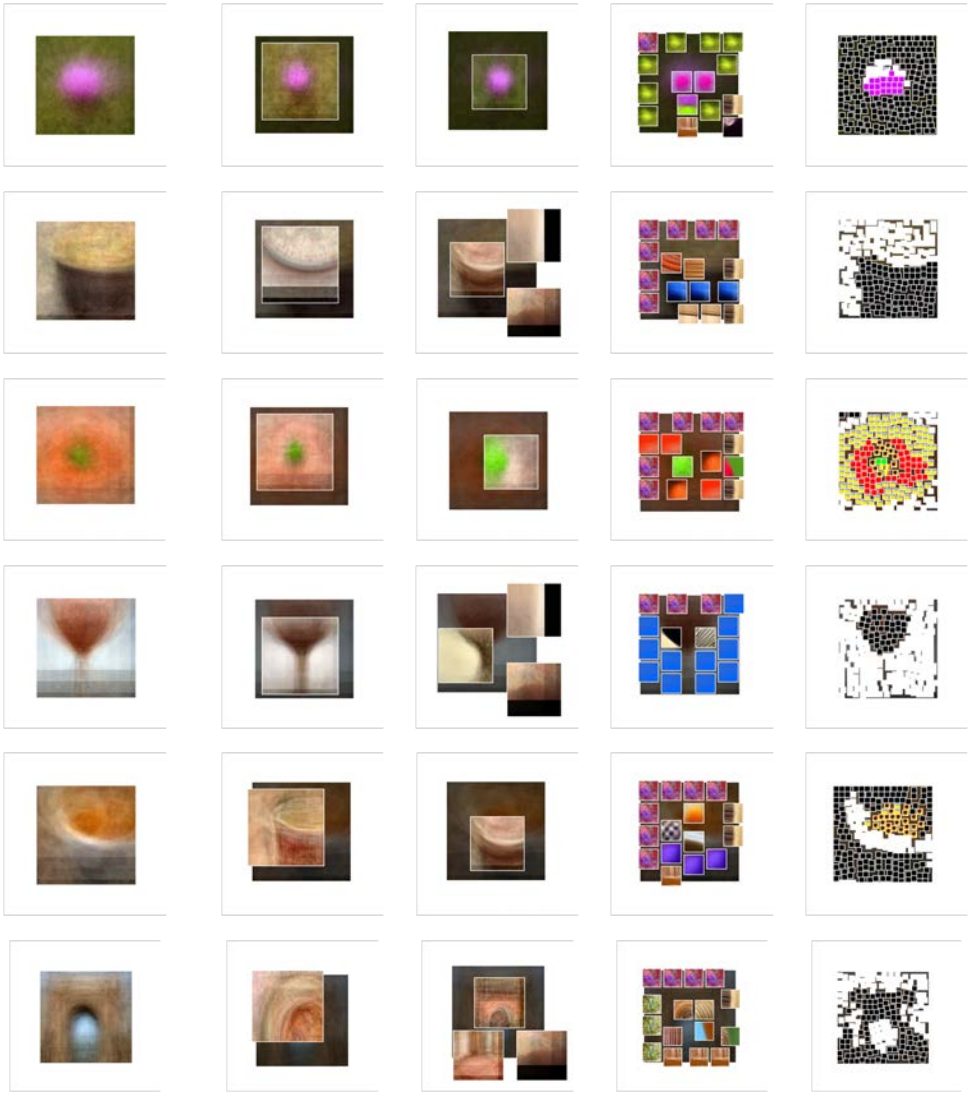


(a) Neuron Feature (b) Conv3 (c) Conv2 (d) Conv1

Figure 4.10 – Neuron composition for six neurons of Conv4. For each convolutional layer, non-overlapped main represenatations are shown, so that it shows stronger connections between the Neuron Feature (a) and Neurons in previous layers ((b) - (d))

l	Type	Name	Parameters	Input size	Output Size	r
1	Convolutional	Conv1	$k = 7 \times 7 \times 3 \times 96$ $p = 0$ $s = 2$	$224 \times 224 \times 3$	$109 \times 109 \times 96$	$7 \times 7 \times 3$
2	Non-linear	Relu1	ReLU	$109 \times 109 \times 96$	$109 \times 109 \times 96$	$7 \times 7 \times 3$
3	Normalization	Norm1	$\eta = 5$ $\alpha = 2$ $\beta = 1 \times 10^{-4}$ $\kappa = 0.75$	$109 \times 109 \times 96$	$109 \times 109 \times 96$	$7 \times 7 \times 3$
4	Pooling	Pool1	$k = 3 \times 3 \times 1$ $p = [0, 1, 0, 1]$ $s = 2$	$109 \times 109 \times 96$	$54 \times 54 \times 96$	$11 \times 11 \times 3$
5	Convolutional	Conv2	$k = 5 \times 5 \times 96 \times 256$ $p = 1$ $s = 2$	$54 \times 54 \times 96$	$26 \times 26 \times 256$	$27 \times 27 \times 3$
6	Non-linear	Relu2	ReLU	$54 \times 54 \times 256$	$54 \times 54 \times 256$	$27 \times 27 \times 3$
7	Normalization	Norm2	$\eta = 5$ $\alpha = 2$ $\beta = 1 \times 10^{-4}$ $\kappa = 0.75$	$54 \times 54 \times 256$	$54 \times 54 \times 256$	$27 \times 27 \times 3$
8	Pooling	Pool2	$k = 3 \times 3 \times 1 \times 1$ $p = [0, 1, 0, 1]$ $s = 2$	$54 \times 54 \times 256$	$13 \times 13 \times 256$	$43 \times 43 \times 3$
9	Convolutional	Conv3	$k = 3 \times 3 \times 256 \times 512$ $p = 1$ $s = 1$	$13 \times 13 \times 256$	$13 \times 13 \times 512$	$75 \times 75 \times 3$
10	Non-linear	Relu3	ReLU	$13 \times 13 \times 512$	$13 \times 13 \times 512$	$75 \times 75 \times 3$
11	Convolutional	Conv4	$k = 3 \times 3 \times 512 \times 512$ $p = 1$ $s = 1$	$13 \times 13 \times 512$	$13 \times 13 \times 512$	$107 \times 107 \times 3$
12	Non-linear	Relu4	ReLU	$13 \times 13 \times 512$	$13 \times 13 \times 512$	$107 \times 107 \times 3$
13	Convolutional	Conv5	$k = 3 \times 3 \times 512 \times 512$ $p = 1$ $s = 1$	$13 \times 13 \times 512$	$13 \times 13 \times 512$	$139 \times 139 \times 3$
14	Non-linear	Relu5	ReLU	$13 \times 13 \times 512$	$13 \times 13 \times 512$	$139 \times 139 \times 3$
15	Pooling	Pool5	$k = 3 \times 3 \times 1 \times 1$ $p = [0, 1, 0, 1]$ $s = 2$	$13 \times 13 \times 512$	$6 \times 6 \times 512$	$171 \times 171 \times 3$
16	Convolutional	FC6	$k = 6 \times 6 \times 512 \times 4096$ $p = 0$ $s = 1$	$6 \times 6 \times 512$	$1 \times 1 \times 4096$	$331 \times 331 \times 3$
17	Non-linear	Relu6	ReLU	$1 \times 1 \times 4096$	$1 \times 1 \times 4096$	$331 \times 331 \times 3$
18	Convolutional	FC7	$k = 1 \times 1 \times 4096 \times 4096$ $p = 0$ $s = 1$	$6 \times 6 \times 512$	$1 \times 1 \times 4096$	$331 \times 331 \times 3$
19	Non-linear	Relu7	ReLU	$1 \times 1 \times 4096$	$1 \times 1 \times 4096$	$331 \times 331 \times 3$
20	Convolutional	FC8	$k = 1 \times 1 \times 4096 \times 1000$ $p = 0$ $s = 1$	$1 \times 1 \times 4096$	$1 \times 1 \times 1000$	$331 \times 331 \times 3$
21	Non-linear	Prob	Softmax	$1 \times 1 \times 4096$	$1 \times 1 \times 4096$	$331 \times 331 \times 3$

Table 4.1 – VGG-M architecture designed by [20], where l is the layer ID, r the receptive field size, and k, p and s are the hyper-parameters kernel size, padding and stride, respectively. $\alpha, \beta, \kappa, \eta$ are hyper-parameters of non-linear layers.



(a) Neuron Feature (b) Conv4 (c) Conv3 (d) Conv2 (e) Conv1

Figure 4.11 – Neuron composition for six neurons of Conv5. For each convolutional layer, non-overlapped main represenatations are shown, so that it shows stronger connections between the Neuron Feature (a) and Neurons in previous layers ((b) - (e))

5 Color selectivity index in trained CNNs

Convolutional Neural Networks have been proposed as suitable frameworks to model biological vision. Some of these artificial networks have shown representational properties that rival primate performances in object recognition tasks as proved by Cadieu *et al.* [17]. In this chapter we explore how color is encoded in one of these artificial networks by estimating a selectivity index for each individual neuron, we use the one trained by Chatfield *et al.* [20] on the ImageNet dataset [114]. The proposed index allows to classify the full neuron population of the network, whether they are color selective or not and if they are single or double color. We find out that all five convolutional layers of the network present a large number of color selective neurons. Color opponency clearly emerges in the first layer, presenting 4 main axes *(Black-White, Red-Cyan, Blue-Yellow and Magenta-Green),* but it is reduced and rotated as we go deeper in the network. In the second convolutional layer we find a more dense hue sampling of color neurons and opponency is almost reduced to one new main axis, which is *Bluish-Orangish* coinciding with the dataset bias. The three remaining layers have color neurons are similar between them, presenting different type of neurons detecting specific colored objects (e.g. orangish faces), specific surrounds (e.g. blue sky or green grass) or specific colored or contrasted object-surround configurations (e.g. blue blob in a green surround). Overall, our work concludes that color and shape representation are successively entangled through all the layers of the studied network, showing up some parallelisms with reported evidences in primate brain that can provide and providing some inspiration about intermediate hierarchical spatio-chromatic representations.

5.1 Introduction

The understanding of the intermediate representations achieved by Convolutional Neural Networks has been achieved through different methods that try to visualize them in the input image space (see Sections 4.2 and 4.3). However, the interpretation of the neuron activity emerged from these visualizations has been done by their mere observation and characterizing them through significant concepts. This opens a new line of research for developing methodologies seeking how

to define the roles played for each neuron which is the basis of this chapter. Here we pursuit on assessing the affinity of any neuron to be activated when a specific color appears in the input sample.

Moreover, in this chapter we hypothesize that, considering we can find some parallelisms between layers of a trained artificial network with known evidences in the human visual cortex [69], we can pursuit some inspiration about how color could be encoded in beyond-opponent human visual pathway by understanding how color is encoded in layers of artificial networks. To this end, we propose a method to explore how this artificial network is encoding color information based on the estimation of color selectivity indexes over the whole neuron population of the network. Proposed method is based on two basic ideas. First, to compile the set of image patches that maximally activates a neuron. Second, to estimate a color-selectivity index on each neuron based on this set. Once we measured color selectivity indexes we can discriminate different groups of neurons, accordingly with their ability to be color selective or not, or being selective to a single-color or to a double-color pair[1]. The classification of neurons at each layer allows to extract interesting representational properties, such as the amount of color tunned neurons appearing at each layer, or how color and shape are entangled through network layers, or opponency properties emerging from double-color neurons. Once we obtain the map of the network color selectivity we show a clear correlation with the color distribution of the image dataset used to train the network. Reported results provide a compelling hypothesis about color representation beyond cone-opponency.

5.2 Method

In this section we propose to describe neurons by their inherent response to a specific property, using an index. The index has to allow to rank them in a proportional order between their response and the existence of the property in the input image. Therefore, we translate the problem of describing neuron activity to the problem of proposing methods which are able to quantify specific image facets that correlate with the degree of activation of the neuron holding such a property. A selectivity index of a single unit is a flexible an independent method for discriminating or clustering between neurons inside the same network. This analysis can

[1] In this thesis we will use this terminology *single* and *double* to refer to color neurons which are either selective to one single color or to a pair of colors appearing on a specific shape configuration. Note that it differs from the terminology used in [121] where the terms single-opponent and double-opponent are used to refer cells responding to large areas of homogeneous color, or responding to color patterns, textures, and color boundaries, respectively. Although they could broadly present some similarities, they should not be directly equated.

be done by collecting the set of images that maximally activates each neuron and deriving specific measurements on these image patches or receptive fields and on their activation values. Additionally, selectivity indexes can be defined either for image features or for image labels. In what follows, we propose a selectivity index belonging to the first group related to the color property.

5.2.1 Selectivity index

The concept of neuron selectivity has long been established in neuroscience. Authors in [13] provide a general definition of the selectivity, which is a property related to a neuron that measures how much the neuronal response (*i.e*, the neuron activity) is affected by changes on the stimulus. In this sense, a high selectivity index characterizes a neuron to be highly dependent to a specific property, so that when this property is slightly changed, the neuron activity is considerably decreased. In this thesis, then, we propose to adopt this neuroscientist concept to model the activity of neurons in a trained CNN related to a specific concept, such as color (see next section 5.2.2).

5.2.2 Color selectivity index

Color selectivity is a property that can be proved in specific neurons of the human brain. The level of activation of the neuron when the observer is exposed to a stimulus with a strong color bias, and its corresponding low activation when the color is not present, is the object of attention in vision research that pursuits the understanding of how color is coded in the human visual system ([121],[25]). Following this idea, we propose a computational algorithm to measure color selectivity of an artificial neuron, understanding color selectivity as the property of a neuron that highly activates when a particular color appears in the input image and, contrary, gives a low activation when this color is not present.

Our color selectivity index of a neuron is computed by estimating the variation between its global activation to color patches with respect to its global activation to their corresponding gray-level patches. Following this idea, color selectivity index of a neuron can be measured as the ratio of the *area under the* activation *curve* (AUC) to the gray-scale version of the N-top image patches that maximally activate it, divided by its AUC obtained from the original images in RGB. For this, we use the set of top-scored images used to compute the Neuron Feature for each neuron (see Section 4.3). In order to maximally preserve the shape pattern of an image, we propose to use a gray-level transformation based on the image color distribution in the OPP color space, hich allows to isolate intensity from chromaticity. We use the first eigenvector using the Principal Component Analysis (PCA) on this

distribution as the axis where to project each color pixel. In this way, we obtain a gray-scale image that maximizes the color image variance. Selected space is as a linear transform on the RGB color space in order not to introduce more non-linearities besides those already included by each different sensor. This RGB to OPP (opponent space) transform is given by:

$$O_1 = (R+G+B-1.5)/1.5, \tag{5.1}$$

$$O_2 = (R-G), \tag{5.2}$$

$$O_3 = (R+G-2B)/2 \tag{5.3}$$

which is based on the one proposed by Platanoits *et al.* in [110] but normalizing and shifting the three axes within the range $\left[-1,1\right]$. This space was conceived to achieve some physiological inspiration on uncalibrated RGB, and it has provided interesting results in computer vision[2].

Thus, given the set of N-scored images $\{I_t\}_{t=1:N}$ and their corresponding gray-versions $\{I'_t\}_{t=1:N}$ of the j-th neuron at layer l, we define the color selectivity index as follows:

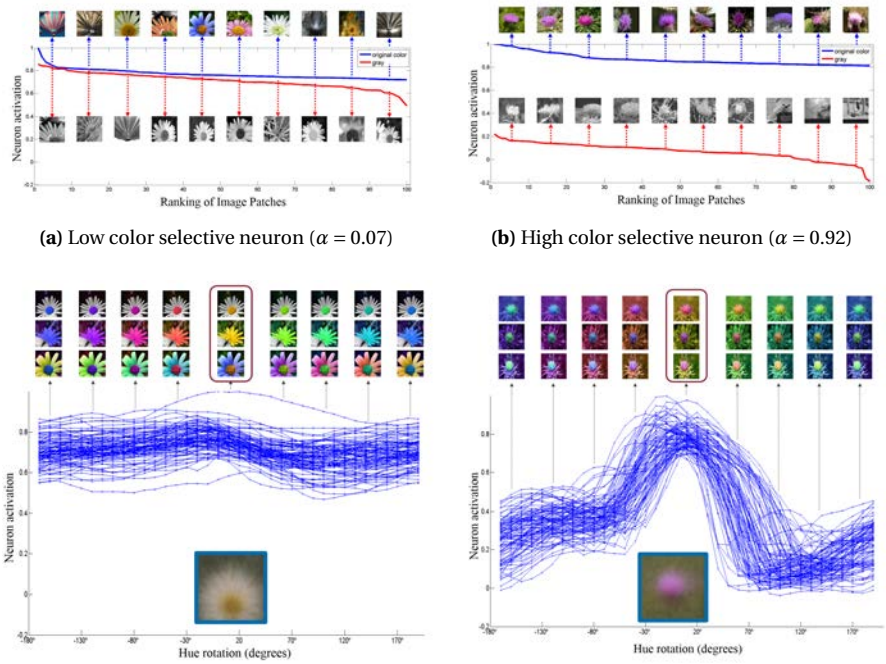
$$\alpha_j^l = 1 - \frac{\sum_{t=1}^{N} \hat{a}_j^l(I'_t)}{\sum_{j=1}^{N} \hat{a}_j^l(I_t)} \tag{5.4}$$

where $\{\hat{a}_j^l(I_t)\}_{j=1:N}$ are the neuron normalized activation values to the original N-top ranked image patches, and $\{\hat{a}_j^l(I'_t)\}_{j=1:N}$ are the normalized activation values obtained by the same neurons to the gray-level versions of the N-top images.

In Figs. 5.1a and 5.1b we show the neuron activity curves of two different neurons in Conv5 of the VGG-M network (see Section 4.3.2), activation values (Y axis) are ranked in a decreasing order from left to right (X axis). We plot neuron activation curves to the N-top original image patches (in blue) and to the corresponding gray-level image versions of these N-top image patches (in red). Note that in the same X axis we are representing two different image rankings, one for color images and another for gray-level images. Neuron in (a) shows equivalent activations for both image sets (color and gray-level), while neuron in (b) plots a clear decrease in activation for gray-level images. Proposed index gives $\alpha = 0.07$ for the first neuron which is a non-color selective neuron, and gives $\alpha = 0.92$, considered color selective.

To confirm the adequacy of the proposed index, in the same figure we explore

[2]Regarding this opponent transform we also want to mention that it was proved to show a large discriminant power for image segmentation using a Principal Component Analysis in the experiments reported by Ohta *et al.* in [102]. And it has also been proved to give the best results in color-shape descriptors for object recognition in Van de Sande *et al.* in [133].



(a) Low color selective neuron ($\alpha = 0.07$)

(b) High color selective neuron ($\alpha = 0.92$)

(c) Flat neuron activation ($\alpha = 0.07$)

(d) Gaussian-like neuron activation ($\alpha = 0.92$)

Figure 5.1 – Activation values to different image versions. (a) and (b) Ranked activation values (Y axis) to the first 100-top activated image patches (X axis), for two different neurons. Blue lines link ranked activations to the 100-top patches of a neuron. Red lines link ranked activations to the gray-level versions of the 100-top patches. (c) and (d) Activation values of the same neurons to different color transformation (rotated along hue axis) of the 100-top image patches. Blue lines are linking the activation values to all color-rotated versions of the same image patch.

the neuron activation to the same N-top image patches when they are transformed into different hue distributions. This transformation is achieved by a pixel-wise operation which computes chromatic rotation on the chromaticity plane of the OPP color space and keeping constant intensity. In Figs. 5.1c and 5.1d we plot the neuron activation (Y axis) for each color rotation (X axis) for the same neurons shown in Figs. 5.1a and 5.1b, respectively. In these plots we show three of the N-top image patches and some of their color transformations along the X axis. Framed

with a red rectangle are the original RGB image patches. At the bottom and framed in blue is the corresponding NF of the neuron. Each blue line links all the neuron activation values to the corresponding images along chromaticity transformations. Again, two different behaviors emerge: a neuron that its activation is not changed along the color transformations is a non-color selective neuron (flat behavior), while a neuron that clearly changes its activation in front of a color transformation is a color-selective neuron (Gaussian behavior). Mean variance computed on these activation curves for each neuron present a clear correlation with our proposed color selectivity index. Pearson's correlation (r) indexes through layers are 0.95, 0.85, 0.85, 0.88 and 0.88, from shallower to deeper convolutional layers.

We want to note that this index is just measuring selectivity to color but not to what color, we work on this point in subsequent sections.

5.2.3 Classifying neuron population

In order to analyze color coding though all network layers, we propose to use color selectivity index to classify neurons in several classes, we do this classification in three stages. A first classification discriminates the entire neuron population in three groups: *color selective*, *low color selective* and *non color selective*. A second stage classifies color selective neurons in two groups: *single* or *double*. And a third stage classifies double color neurons in *opponent* or *non opponent*.

First classification is directly done applying some threshold over the color selectivity index (α). Non color selective neurons are those with $\alpha < 0.10$, and when $\alpha > 0.25$ we label them as color selective neurons. Which means, if the AUC of a neuron activity in front of a gray scale version of the N-top patches decreases more than 25% with respect to their original RGB patches, it is considered high color selective neuron. While neurons are non color selective when this AUC variation is less than 10%. Between these two groups, neurons are considered low color selective neurons. See Fig. 5.1a and 5.1b as examples of low and high selective neurons, respectively. Although the thresholds we applied can seem arbitrary, they were coherently set on the observed selectivity over the set of top ranked image patches activating the neurons, and from the behavior of the neuron activity through a chromatic transformation of the same image patch (see Figs. 5.1c and 5.1d). From our experience, different variations of these thresholds would bring to similar conclusions. Shallower layers have neurons with extreme color index values (either very high or very low) while deeper neurons are mainly described by intermediate index values. A neuron with $\alpha > 0.25$ presents a clear Gaussian behavior of its neuron activity, while a neuron with $\alpha < 0.10$ presents a flat behavior.

Within the group of color selective neurons we distinguish two main groups: single color neurons, presenting selectivity to one single color; and double color

neurons, presenting selectivity to a pair of colors. The definition of these two types of neurons was already introduced in section 5.1. Classification is performed by fitting a Gaussian mixture model on their NF hue distribution using a Expectation-Maximization (*EM*) algorithm. Each fitted univariate Gaussian is defined by its mean and covariance. *EM* is applied for different numbers of Gaussians (from one to four), and it is finally set to the minimum number that differs less than a 10% over the global mean square error with the distribution. This step allows to get one (a pair of) representative hue (hues) for each single (double) color neuron. Selectivity to three or four colors was never found.

Once we have the color-map of all double color neurons through the network layers, we will analyze whether specific chromatic axes emerge representing spatial color opponency which is a central property in early stages of the primate visual systems [30, 77]. In Fig. 5.6 we plot the axis related to each double color selective neuron. To measure how close it is to be in an opponent axis we compute the angular distance to be at 180° from the actual angle formed by the pair. These angles are considered with respect to the center of the O_2-O_3 chromatic plane.

5.3 Results and Discussion

General purpose CNN architectures are usually trained on RGB color images. However there is a strong belief in the computer vision community that color is a dispensable property. The results we obtain by indexing color selective neurons make us conclude that there is no basis for such a belief. Our analysis shows that color is strongly entangled at all levels of the CNN representation. In a preliminary experiment we have tested a subset of ImageNet images with VGG-M (trained on ImageNet dataset [114] and designed by Chatefield *et al.* [20], see Sections 4.3.1 and 4.3.2) in their original color and the same subset in a gray scale representation. Classification performance show a considerable decrease: while original RGB images are classified with a 27.50% top-1 error and 10.14% top-5 error, gray scale image versions present 51.12% and 26.37% errors, top-1 and top-5 errors, respectively.

Results of computing color selectivity index on all neurons of the VGG-M are summarized in Table 5.1[3]. Neurons are classified in seven groups following the criteria defined in section 5.2.3.

To visualize how this color index correlates with the neuron activity, we plot

[3]Originally, VGG-M CNN has 256 neurons in the second convolutional layer. However, as a result of the training process, 37 *dead neurons* were detected, as we mentioned in previous chapter. Due to *Dead neuron* is a neuron which do not contributes to the internal representation of the network and generally have weights are almost random and it is equally activated for any image patch, they are discarded. This explains why presented results on Conv2 corresponds to just 219 neurons.

Selectivity #Neurons	Conv1 96	Conv2 219	Conv3 512	Conv4 512	Conv5 512
Non Color	56 (58.33%)	118 (53.88%)	225 (43.95%)	113 (22.07%)	52 (10.16%)
Low Color Sel	2 (2.08%)	28 (12.79%)	69 (33.01%)	255 (49.80%)	250 (48.83%)
Color Sel	38 (39.58%)	73 (33.33 %)	118 (23.05%)	144 (28.13%)	210 (41.02%)
Single Color	12 (12.50%)	49 (22.37%)	102 (19.92%)	134 (26.16%)	198 (38.67%)
Double Color	26 (27.08%)	24 (10.96%)	16 (3.13%)	10 (1.95%)	12 (2.34%)
Opponent	19 (19.79%)	14 (6.39%)	8 (1.56%)	1 (0.20%)	1 (0.20%)
Non opponent	7 (7.29%)	10 (4.57%)	8 (1.56%)	9 (1.76%)	11 (2.15%)

Table 5.1 – Distribution of color and non color selective neurons through layers. Within the color selective neurons two subgroups: single color and double color, referring to the number of color the neuron is selective to. Within the double color neurons two subgroups: opponent and non opponent, depending how close are colors to present a hue-angle close to 180° or not, respectively. In parenthesis (%) percentage of neurons of the group within the layer.

some examples in Fig.5.2. In this figure each neuron is described twofold: through its corresponding NF and also with their corresponding set of 100 receptive fields used to build this NF. These patches are sorted decreasingly by their activation on the neuron (from left to right and from top to bottom). We can see that the appearance of the NF describes features that are mostly shared by the 100 image patches. Note how neurons sided on the right have a clear color dependence, compared with the ones shown in the left of the figure.

First conclusion derived from Table 5.1 is that there are color selective neurons in all the layers. This correlates with the idea that color is encoded all the way from V1 to IT cortex as concluded by Shapley *et al.* in [121]. A graphical representation of selectivity index values across layers and corresponding percentage of neurons is plotted in Fig. 5.3. Average color selectivity index per layer is 0.35, 0.24, 0.22, 0.24, 0.22 and 0.28 for Conv1 to Conv5, respectively, which is quite constant. But, while neurons with higher color selectivity indexes are more concentrated in shallow layers, it compensates with a higher number of lower color selectivity neurons in deeper layers (percentage of color selective neurons in Conv4 and Conv5 surpass shallow layers).

In the following sections we deeply analyze these results from two points of view. First, directly from the general derived color properties (Sections, 5.3.1, 5.3.2 and 5.3.3), and second, from the specialization of three main groups: Conv1, Conv2 and Conv3-Conv5 together considering the similarity of their neuron population (Sections 5.3.4, 5.3.5, 5.3.6). Here we take some risk hypothesizing some parallelism of these three groups with the V1, V2 and V4/PIT/TE, suggested in the hierarchical

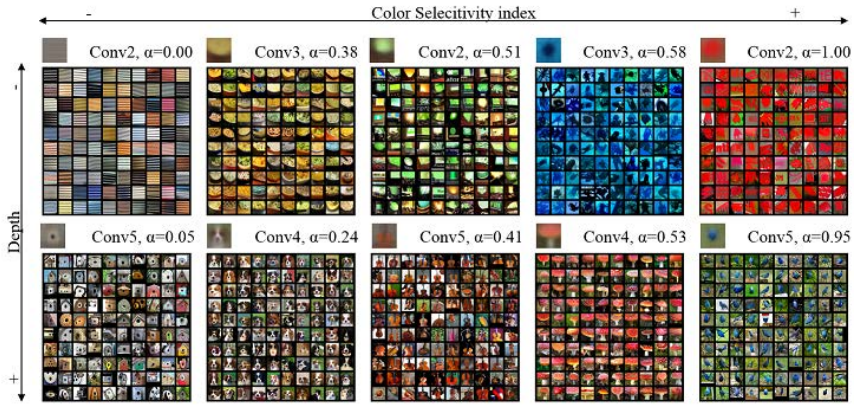


Figure 5.2 – NFs and their 100-top image patches corresponding to 10 neurons of different layers. Neurons in shallow layers are described in two first rows, while for the ones in deeper layers, through the last two rows. 1st and 3rd row correspond to NFs showing the intrinsic properties that may describe the neuron activity of the corresponding neuron. 2nd and 4th rows plot the set of 100-top image patches that maximally activate the corresponding neuron. Note that these patches are sorted from left to right and from top to bottom according to activation value in a decreasing order. Color selectivity index, α is given for each neuron. Note that NFs are shown with a bigger size than each image patch in order to help on the intrinsic visualization.

model of color processing in macaque cerebral cortex summarized by Conway *et al.* in [25].

5.3.1 Single and double color neurons

Regarding single and double selective neurons, the number of single neurons increases with depth while the number of double decreases. In Fig. 5.4a we plot all single neurons along different convolutional layers, from Conv1 (inner ring) to Conv5 (outer ring). Each single color neuron is plotted at its representative hue (estimated Gaussian mean). Observe that the distribution falls in two main hue regions on deeper layers (orangish and bluish), while representatives are more distributed over hue in shallow layers. The rest of plots (Figs. 5.4b - 5.4f) show single (outer ring) and double color neurons (inner ring) per layer. Double color neurons are plotted by their two representative hues (two estimated Gaussian means), same NF is reproduced on both hues and they are linked by a line to visualize their

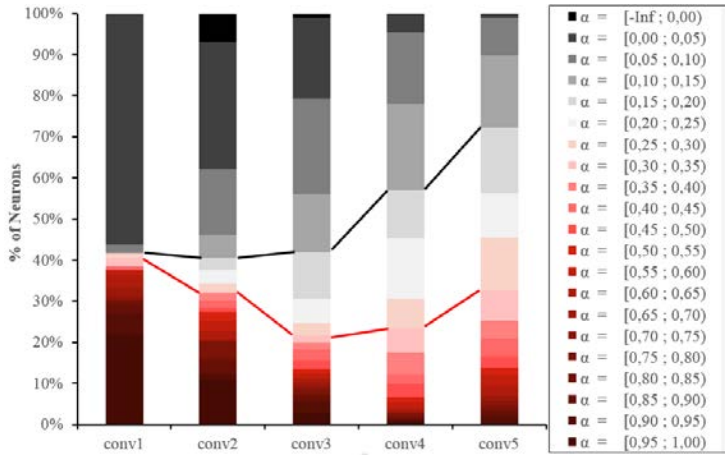
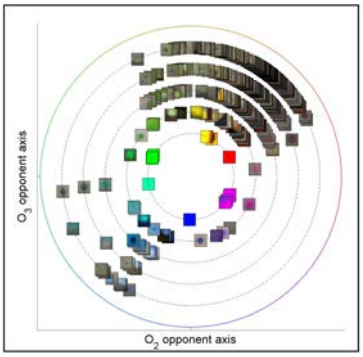


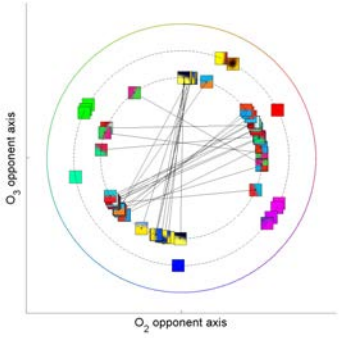
Figure 5.3 – Map of network color selectivity. Percentage of neurons within different ranges of color selectivity indexes for each layer. Thresholds applied in this thesis $\alpha = 0.1$ and $\alpha = 0.25$ are marked with the black and red lines, respectively.

connection. Intersection of lines in similar directions outlines the emergence of color axes. A *Bluish-Orangish/Brownish* axis clearly arises in deeper layers, but this is studied in subsequent lines.

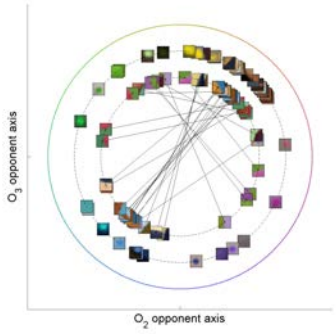
Although ImageNet is built from millions of different real images, its color distribution is not homogeneous. We estimate the color distribution of the complete ImageNet dataset and it is shown in Fig. 5.5. Colored bars plotted represent its color distribution (Y axis) based on hue-angle (X axis) computed on O_2-O_3 plane of the opponent color space given in equation 5.2.2. The hue-angle, γ, of a given a pixel, $\rho = (o_1, o_2, o_3)$, is computed as $\gamma(\rho) = \arctan(\frac{o_3}{o_2})$. Distribution presents a clear bias (a bimodal Gaussian distribution), peaking at orangish hues and at bluish hues. They could be due to a rich presence of brownish animals and people skin likewise sky backgrounds respectively. We will not analyze if this dataset bias correlates with natural scene statistics, but it is normal that calibrated natural scene statistics present some kind of bias, which can vary depending on season, area or latitude [142]. Therefore, the emergence of a more dense sampling in brownish and bluish regions observed in Fig. 5.4a with these color peaks on ImageNet suggest a similar distribution. To this aim, both distributions are compared. The one obtained from color selective neurons according to their selective hue is also shown in Fig. 5.5 as a black line on colored bars. We combine single and double neurons (for double neurons we consider both hues and duplicate the single neuron hues). We can



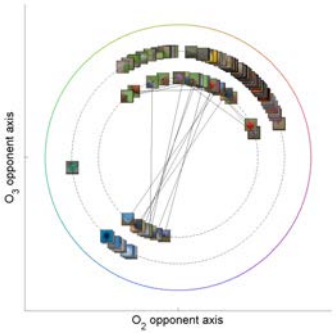
(a) All layers (Only single color neurons)



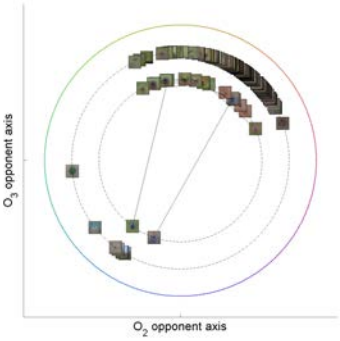
(b) Conv1



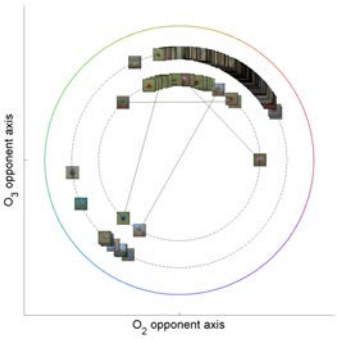
(c) Conv2



(d) Conv3



(e) Conv4



(f) Conv5

Figure 5.4 – Chromaticity of color selective neurons across layers. (a) Single color neurons for all layers. (b), (c), (d), (e) and (f) Single color neurons (outer ring) and double color neurons (inner ring) for layers Conv1, Conv2, Conv3, Conv4 and Conv5 respectively. Double color neurons are plotted twice (to represent double chromaticity) and linked with a line.

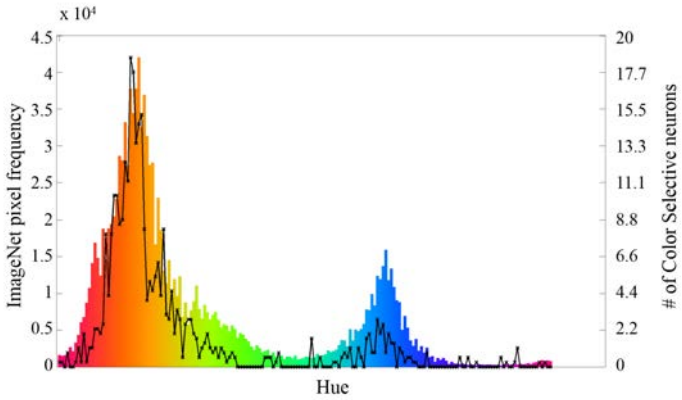


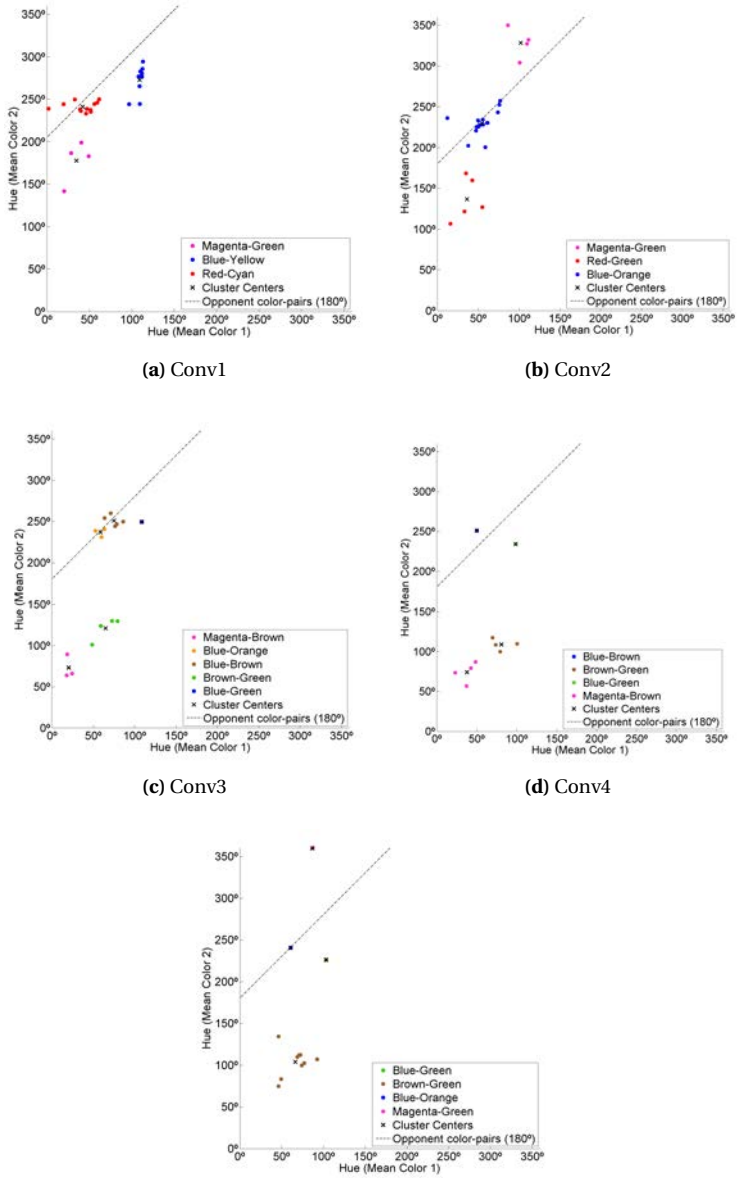
Figure 5.5 – ImageNet hue distribution (colored bars) and Number of color selective neuron per hue (black line).

observe a clear similarity between both distributions, presenting two significant peaks on orangish and bluish hue regions. Concretelly, both distributions are highly correlated, with a Pearson correlation coefficient (r) of 0.89. This result confirms that color selective neurons learned by the CNN are adapted to the dataset bias, in a similar way to what happens with color bias in natural scenes that has been proved to have implications in higher color sensitivity in the human visual system [91].

5.3.2 Opponency property

To analyze opponency property of double color neurons we represent their color pairs (Mean Color1 and Mean Color2) as an (X,Y) point in a two dimensional hue space (see Fig. 5.6). Being Color1 the one with smallest hue angle and Color 2 with largest. In this way, perfect opponency (or 180° of angular distance between them) is represented by the location of the black dashed line (top left corner). The closer a neuron is to this dashed line, the more opponent it is. Overall, we can see from these plots that opponency decreases with depth. The maximum is found at first convolutional layer. We performed a cluster analysis to find main emerging axes, defined as groups of neurons sharing the same axes direction. For this purpose, we use k-means technique and test from $K = 3$ to 7 to detect the best number of clusters using Elbow method, which is based on a ratio of the between-group variance with respect to the total variance.

In Table 5.2 we list obtained clusters. We assigned an axis name to each cluster



(a) Conv1 **(b)** Conv2

(c) Conv3 **(d)** Conv4

(e) Conv5

Figure 5.6 – Emergent axes from Cluster analysis on double color neurons, from (a) Conv1 to (e) Conv5. Neurons are given by their pair of colors: mean hue of Color 1 in X axis (smallest hue) and mean hue of Color2 in Y axis (biggest hue). Top left dashed line represents location of perfect opponent pairs (180°)

Double neurons	Conv1	Conv2	Conv3	Conv4	Conv5
Blue-Yellow	**11.69**	-	-	-	-
Red-Cyan	**13.26**	-	-	-	-
Magenta-Green	**20.05**	32.64	89.86	101.66	65.72
Red-Green	-	56.32	-	-	-
Blue-Orange	-	**3.27**	**1.46**	-	-
Blue-Brown	-	-	**3.03**	**14.81**	**0.31**
Blue-Green	-	-	27.71	31.85	40.38
Brown-Green	-	-	87.56	107.54	100.97

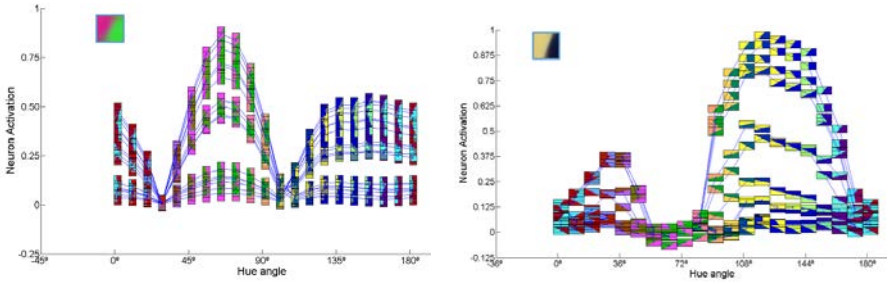
Table 5.2 – Deviation from opponency for clustered double color neurons through all layers. Neuron cluster with small deviation $< 21°$ (in bold) are proposed as opponent emergent axes.

and compute its angular distance to perfect opponency. Color names used to label each axis were approximately assigned by the observation of the NFs in the cluster. From this table we can conclude two main observations: (a) in layer Conv1 all double color neurons present a remarkable opponency property (rows 1 to 3), (b) a special *Bluish-Orangish* (or a similar *Bluish-Brownish*) axis emerges from layer Conv2 up to the deepest layers (Rows 5-6). The emergency of these axes is supported by the small angular distance (in bold) to perfect opponency. Although we will refer to this axis as *Blue-Orange* (or *Blue-Brown*) the *Blue* hue presents a clear clockwise rotation with respect to *Blue-Yellow* axis found in Conv1 (see Fig. 5.4).

5.3.3 Color and shape entanglement

As seen in previous sections, color is an important property that may characterize the activity of a neuron. Nevertheless, this property always appears strongly linked with the intrinsic shape that also activates the neuron. In this section we analyze the entanglement of both properties, shape and color, which can be understood as a template matching scheme linked to the activity of any neuron along the artificial network. With color-shape entanglement we mean that the activation of a single neuron requires the appearance of a specific color in an appropriate configuration or shape.

This color-shape entanglement appears through all the layers. Regarding shallower layers we plot in Fig. 5.7 a set of neuron activation curves for two color neurons in 5.7a Conv1 and 5.7b Conv2, both representing colored edges oriented in $135°$. Each neuron activation curve is obtained from a set of oriented edge images, sharing the same orientation but with different opponent color pairs. Different oriented edges are tested for both neurons and maximum activations are achieved



(a) Conv1 color neuron ($\alpha = 0.99$) **(b)** Conv2 color neuron ($\alpha = 0.44$)

Figure 5.7 – Neuron activation of two edge-oriented double color selective neurons, (a) Conv1 ($\alpha = 0.99$) and (b) Conv2 ($\alpha = 0.44$). Activation curve for different synthetic oriented color edges, varying in hue-pairs and orientation. Blue line connects activations corresponding to images with the same orientation but different color-pair. Neuron NFs are framed in blue on the top-left corners.

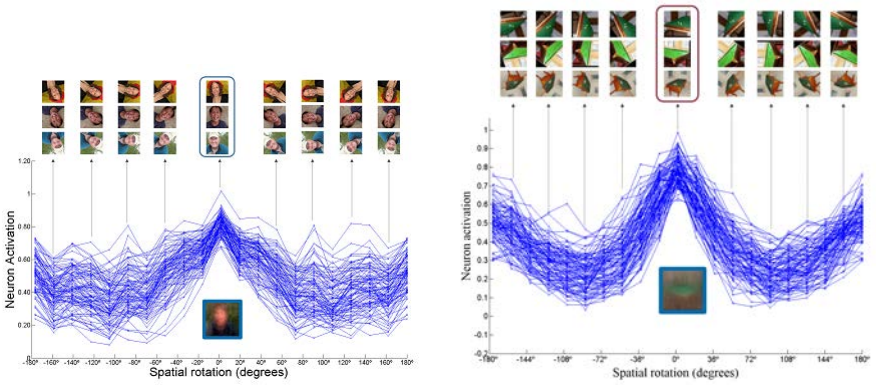
when color pair and orientation match with the input image (see the neuron NF visualization on the top left corner).

In the same line, neurons in deeper layers which are responsible to detect more complex shapes also present this color-shape dependence. For these neurons we show two examples of activation curves where shape structure is changed through a spatial rotation. In Fig. 5.8 we plot several activation curves of a face selective neuron found in Conv4 (Fig. 5.8a) and pf a pooltable neuron in Conv5 (Fig. 5.8b), both with a high color selective index ($\alpha = 0.59$ and $\alpha = 0.65$, respectively). These curves show the variation of the activation curves when same face or pool table images that highly activate each neuron are spatially rotated. Note that the same image with spatial rotations modifies the activity of the neuron, although sharing same color appearance.

The strong entanglement between color and shape in the human visual system has been proved in several works, revised by Shapley *et al.* [122], concluding that this high relation between shape and color is found in HVS neurons along different areas.

5.3.4 Layer Conv1

Trained neurons in layer Conv1 are compiled in Fig. 5.9. Our classification of the neuron population in this layer can be resumed in two main groups: selective and non selective neurons around 40% and 60%, respectively. Only two neurons



(a) Conv4 color neuron ($\alpha = 0.59$)

(b) Conv5 color neuron ($\alpha = 0.65$)

Figure 5.8 – Neuron activation of two edge-oriented double color selective neurons, (a) Conv4 ($\alpha = 0.99$) and (b) Conv2 ($\alpha = 0.44$). Activation curve for different synthetic oriented color edges, varying in hue-pairs and orientation. Blue line connects activations corresponding to images with the same orientation but different color-pair. Neuron NFs are framed in blue on the top-left corners.

are found as low color selective, and as a particularity of this layer, there are more double color neurons than single. The low spatial frequency selectivity observed in color selective neurons jointly with the diversity of spatial frequencies of non color selective neurons correlates with reported evidences in human vision in [78, 117]. Moreover, we want to state that CNN training does not hold any constraint about similarities between neurons of the same layer. Then, boundaries between layers are fuzzy and we can find neurons in second layer that could be grouped with these Conv1 neurons, and we find neurons in first layer that for its spatio-chromatic properties would fit better in Conv2 (framed in red). Thus, for clarity reasons we keep our conclusions at the layer level, not trying to give a global and exhaustive classification of the color neurons.

First convolutional layer presents a key role in color-pairs representation due to the strong opponency shown by all clusters of double color neurons. They are representing a three dimensional opponent space based on three chromatic channels: two with a higher number of neurons *Red-Cyan* and *Blue-Yellow*, that could correlate with the findings of Derrington *et al.* in [30] (extensely reviewed in [77]). And a third one, *Green-Magenta*, with less neurons, but which could be related

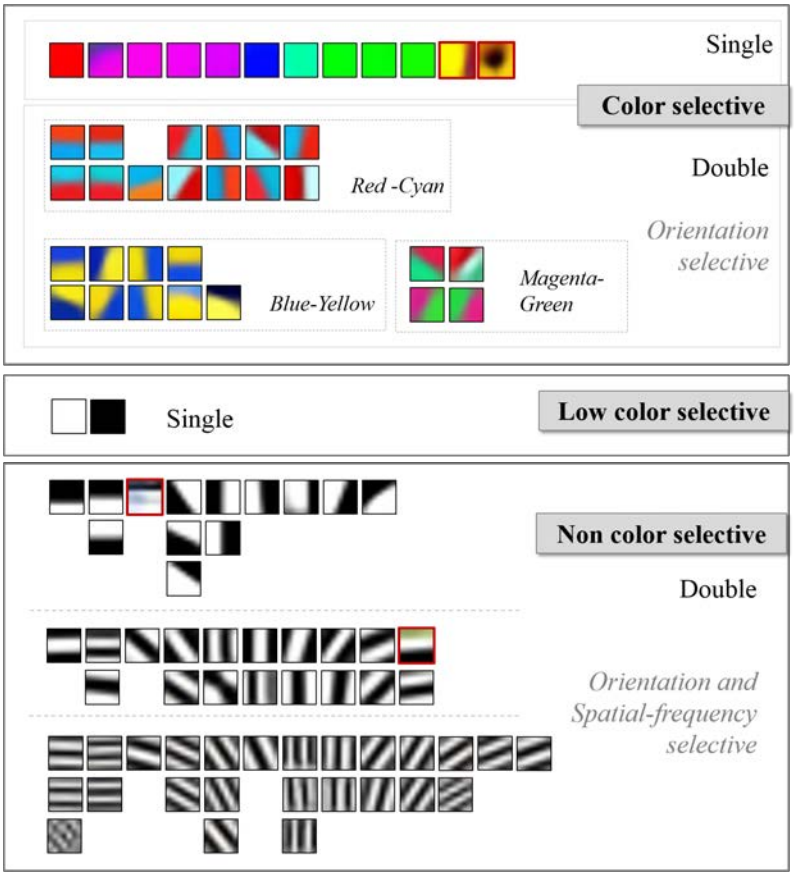


Figure 5.9 – 96 Neuron Features of the set of Conv1 neurons classified in terms of color selectivity. 38 Color selective neurons (39.58%). 12 single Color (12.50%) and 26 double opponent neurons (27.08%). 13 Red-Cyan, 10 Blue-Yellow and 3 Magenta-Green cells.

with the fourth oponnent channel reported by Conway *et al.* in [22] (*Black-White* is counted). Each pair of colors has an angular distance (opponency property) of 166.74°, 168.31° and 160°, respectively. We want to remark here that our results are just an approximation on an uncalibrated space where labels are assigned by mere NF observation. This opponency is highlighted by Fig. 5.10, where activation curves of three double color neurons in Conv1 oriented in 45° (one for each color axis) to different synthetic color edges in the same orientation are plotted. We compute

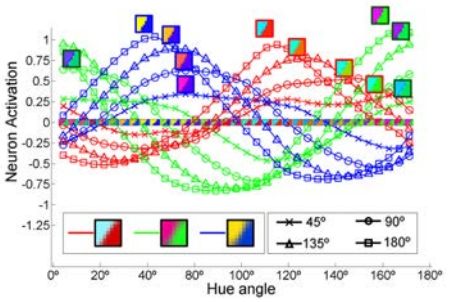


Figure 5.10 – Representation of color edges. Activation curves of three different double opponent color neurons at Conv1 to 4 different types of edges regarding the opponency property of the pair of colors on the tested edges: opponent edges (180), 135, 90 and 45 color edges.

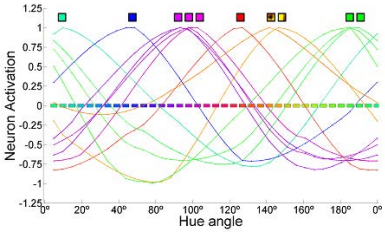


Figure 5.11 – Activation curves of single neurons in Conv1 over series of images varying along the hue dimension. Stimuli are rotated hue versions of the maximum activating image for each neuron.

the activation on 4 different type of edges composed by pairs of colors holding different angular distances between them: 45°, 90°, 135° and 180°, each type of edge is generated along a sampling of the entire hue. In this figure, for each type of edges we plot the color edge achieving the maximum activation for each neuron. For the case of opponent edges (180°, maximum color pair contrast difference), the maximal neuron activations are achieved when the edge coincides with the NF pattern (see the legend for a visualization of NFs of the studied neurons). The rest of color edges are represented by different triplets of weights on the neuron basis. In this way we show how any color-edge is represented by the three main group of opponent axes emerging in this layer Conv1.

Similarly, we study how regions of a single color are represented by the network. We observe that they are based on a composition of basic primary colors represented by single color selective neurons single color neurons in this layer are specialized on the following basic hues: Red, Green, Blue, Magenta, Cyan and Yellow, which is a quite standard and balanced sampling of the hue circle, whose combination allows to represent all the color hues. In Fig. 5.11 we can see the activation curves of these basic neurons to homogeneous regions along the hue dimension. Some of them present more than a neuron for a similar primary hue.

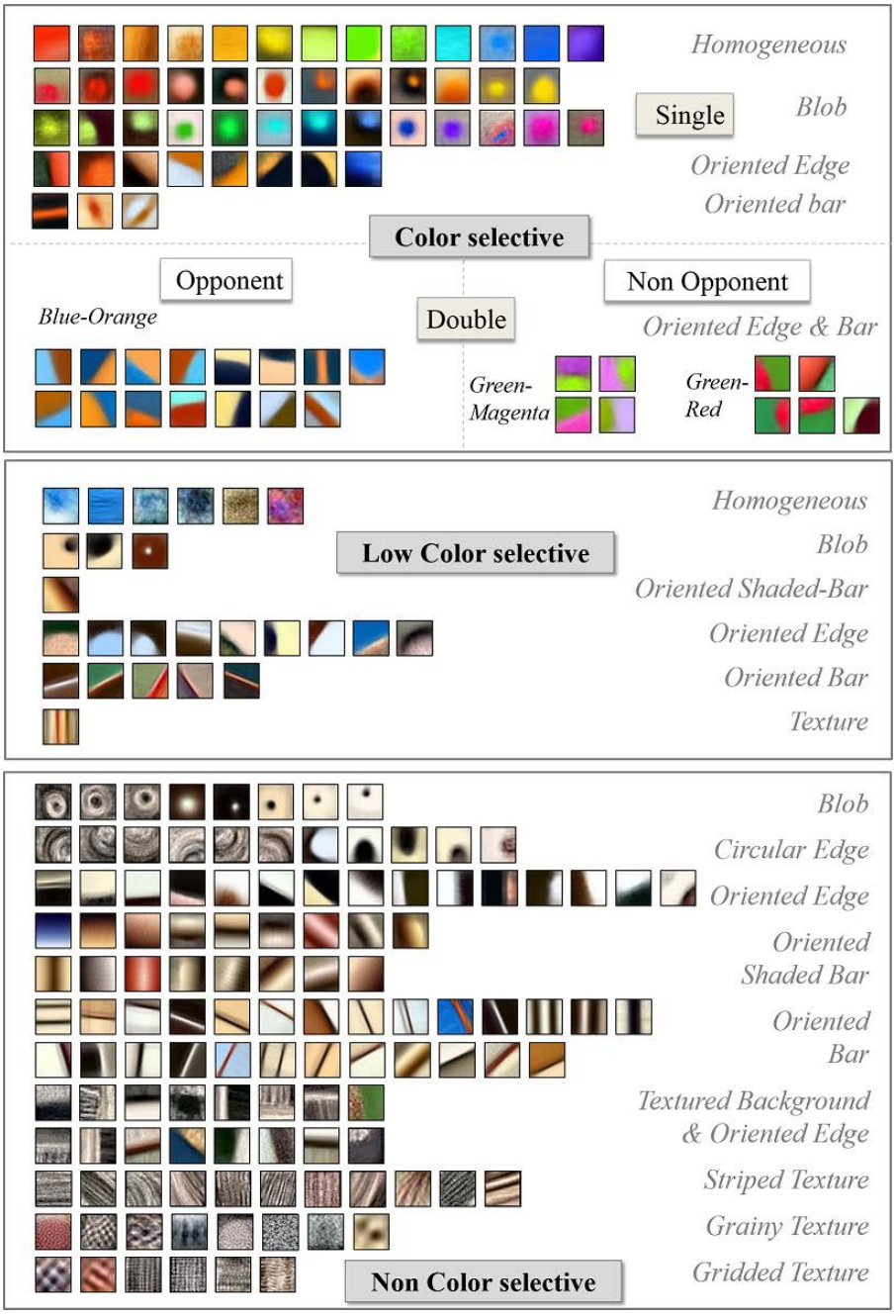


Figure 5.12 – 219 Neuron Features of Conv2: 75 being color selective neurons (34.25%) with 54 single neurons and 21 double neurons (13 for the Blue-Orange, 4 for the Magenta green and other 4 for the Red-Green

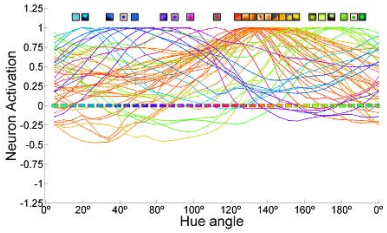


Figure 5.13 – Activation curves of single neurons in Conv2 over series of images varying along the hue dimension. Stimuli are rotated hue versions of the maximum activating image for each neuron.

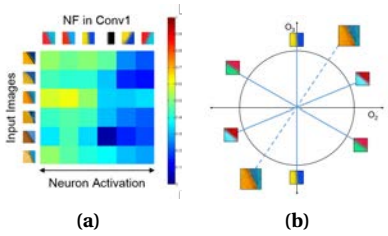


Figure 5.14 – Orange-blue edge representation in Conv1. (a) Activation values of some neurons in Conv1 (NFs are in first row) to a set of images (first column) in a color scale from blue (minimum) to red (maximum). (b) Double neurons in Conv1 over the hue space an a Orange-Blue image (dashed line), which is highly represented in Conv2 through the Blue-Orange axis, is represented between two main neurons in Conv1 (Blue-Yelow and Red-Cyan) in this hue space.

5.3.5 Layer Conv2

Neurons in Conv2 were classified as shown in Fig. 5.12[4]. At a first glance, we can see that non color selective neurons present an increase in shape complexity with respect to previous layer: more complex edges as circular edges in diverse directions, oriented bars, shading effects, centered and shifted blobs or homogeneous textures and edges between textures. They are more complex features than oriented edges and basic gratings of the previous Conv1. However, they can not be identified as object shapes like those we will find in subsequent Conv3-Conv5. This layer seems to represent surface details beyond its boundaries.

Regarding color selective neurons we find colored edges and homogeneous neurons as in layer Conv1. The main novelty regarding their shape is that there are also colored blobs and oriented bars. Likewise in Conv1, single color neurons are responsible to detect single color regions based in a combination of basic hues but in this level of representation, it is more detailed due to a more dense sampling of primary hues is covered by the set of single neurons, as can be perceived from Fig. 5.13. Thus the color of a region can be more precisely described in this more

[4] Organization and label assignment in this figure were visually performed from NF shape and color, with the exception of the opponent axes that come from a cluster analysis

extensive basis.

This density of representation becomes a peculiarity of this layer for both, single and double color neurons, as can be observed in Fig. 5.4c, where color selectivity turns into a more dense sampling on the hue circle, in comparison to the rest of layers. To quantize this observation, we computed a sparsity measure l^0, studied in [57][5], on the hue distribution of colors that neurons are selective to. We performed this measure for different hue sampling sizes. Results are shown in Fig. 5.3(b), where a clear minimum in sparsity emerges at Conv2. This continuum in hue selectivity could be related with measurements reported by [49, 141, 146] about the existence of hue maps in V2 cortical areas revised in [23].

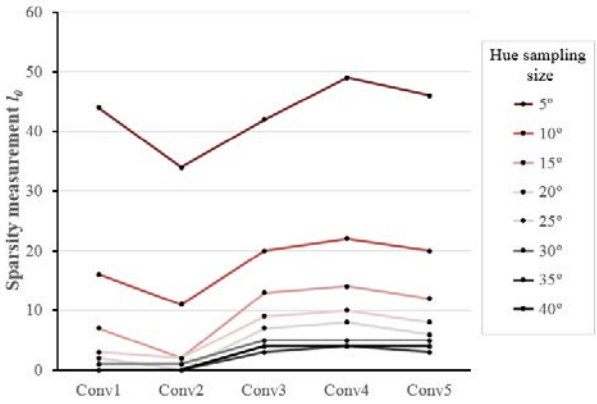


Figure 5.15 – Sparsity of network color selectivity: how many color selective neurons and how many different colors are selective to. Sampling of the hue space by neuron selectivity. The lower sparsity is the more dense is the sampling, i.e. the higher the number of different colors neurons are selective to.

Double color neurons were clustered in three main axes (see Fig. 5.6b: *Magenta-Green*, *Red-Green* and *Blue-Orange*. Two first axes (*Red-Green* and *Magenta-Cyan*) seem an extension of axes in Conv1, but loosing the opponency property. However, the emergent *Blue-Orange* axis with a small deviation from opponency of 3.27° (see Table 5.2) is a novelty of this layer, that anticipates edges of object shapes that will be represented in posterior layers laying on the hue bias of the dataset. This axis is a rotation of the Blue-Yellow axis found in Conv1 that matches the bias of the training dataset. In Fig. 5.14a we show how several image patches that highly

[5]Measure l^0, defined as $l^0 = \{j : C_j = 0\}$, counts the number of zero bins in a sampled distribution, denoted as $\{c_j\}$.

activate a Bluish-Orangish neuron of Conv2 are represented in Conv1. NFs of the first convolutional layer that are most activated by these kind of Orange-Blue edge stimuli (oriented in 135°) are plotted in the first row. Tested images are shown in the first column. Activation values are shown in a scale from blue (minimum) to red (maximum). This composition can be easily understood from their location in the chromaticity plane of the opponent color space shown in Fig. 5.14b, where double color neurons are allocated twice in corresponding coordinates of their pair of colors in this chromaticity space, likewise the representation of a Blue-Orange image (shown bigger since its size corresponds to the receptive field size of a neuron in Conv2). This example brings us to speculate about the hierarchy of the Bluish-Orangish neuron population code in Conv2: it is activated when a Red-Cyan neuron of Conv1 with an edge of 135° is highly activated jointly with a high activation of a Yellow-Blue neuron (also in Conv1) with a 90° oriented edge[6].

Neuron peculiarities of this layer can be summarized as: more complex surface features (essentially non color), a more dense hue sampling, reminiscent neurons from earlier opponent axes and neurons defining a new *Blue-Orange* axis anticipating the edges of the dataset bias where color neuron activity will be concentrated in subsequent layers. Previous conclusions could correlate with the singular intermediate role attributed to V2 in biological systems reported in several works [24, 95, 121, 126].

5.3.6 Deeper layers: Conv3, Conv4 and Conv5

The last three convolutional layers of the architecture consist of color selective neurons that seem to be highly linked to object shapes. Note that NFs present more blurred edges of averaged images, since increase of size affects pixel-wise spatial variability.

In Figs. 5.16, 5.17 and 5.18 we can overview all color selective neurons in these layers, from Conv3 to Conv5 respectively. Note that even in deeper layers we found high color selective neurons. Nevertheless, further research is required for a full understanding of color and shape representation in these layers, although we can glimpse like 4 main groups of neurons that should be more carefully explored. First, neurons devoted to specific object shapes with a characteristic color (e.g. red and brown mushrooms, skin in faces and human bodies, dog faces among others). Second, neurons activated by homogeneous image areas of specific shapes simulating surround areas (e.g. sky and grass backgrounds). Third, double color neurons that can represent colored objects or objects parts on specific colored

[6]Note that in Conv1 a Blue-Yellow neuron having an oriented edge of 135° (with yellow in the bottom-left and blue in the top-right) was not found. In its absence, the vertical edge neuron is the most similar.

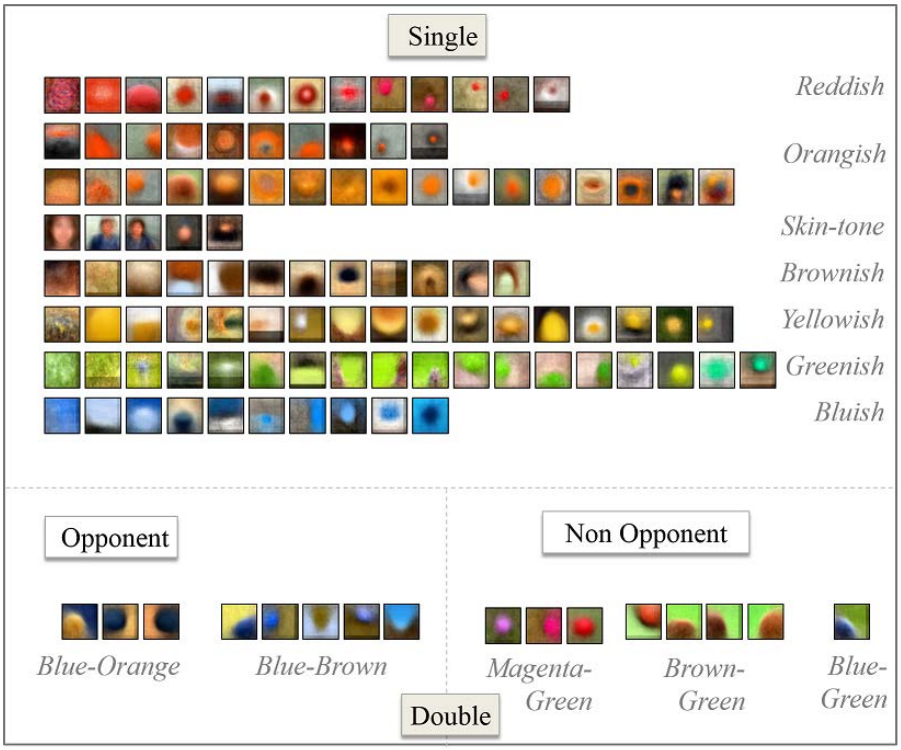


Figure 5.16 – Color selective neurons in Conv3 classified regarding to their color properties.

backgrounds (e.g. blue bird in a green surround or ladybugs in green leaves). Fourth, single color selective neurons whose NF does not identify a specific shape, but presenting colored regions either as a central blob or as a surround, and with strong intensity variations. From the observation of these NFs we can also conclude that scale invariance is represented by multiples neurons representing similar shapes at different layers (different resolution) or small and large versions of the same shape within the same layer. In Fig. 5.19 we show the effect of a neuron of each of these groups when the same shape stimuli has different colored versions. These modifications are computed by rotating color image pixels on the chromaticity plane (along hue dimension). Note that these neurons are uniquely selective to a specific color appearance.

Double color neurons only represent a 2.5% of the three layers. Some of them present clear opponency in the *Blue-Orange* or *Blue-Brown*, but some others (non-

Figure 5.17 – Color selective neurons in Conv4 classified regarding to their color properties.

opponent) present different combinations devoted to represent green or brown surrounds jointly with different colored objects (brown, orange, blue or magenta).

Parallelism with biological systems is difficult to be established at this point, since higher-order visual areas are not as known as V1 and our analysis requires further research. However, in what follows we report some conclusions in primate visual systems that can show some similar ideas with previous conclusions, like linked color-shape or object-surround selectivity.

Multiple areas have been reported to present color selectivity (reviewed by Conway *et al.* in [24]). Although deeper visual ares of the human visual system are not known like in previous areas, some areas seem to combine shape and color selectivity like V4 and PIT (posterior inferior temporal cortex) while others present narrowly color and saturation tuning and weak shape selectivity in TE (anterior IT). Additionally, in an earlier work, Schein *et al.* in [116] stated that neurons in V4

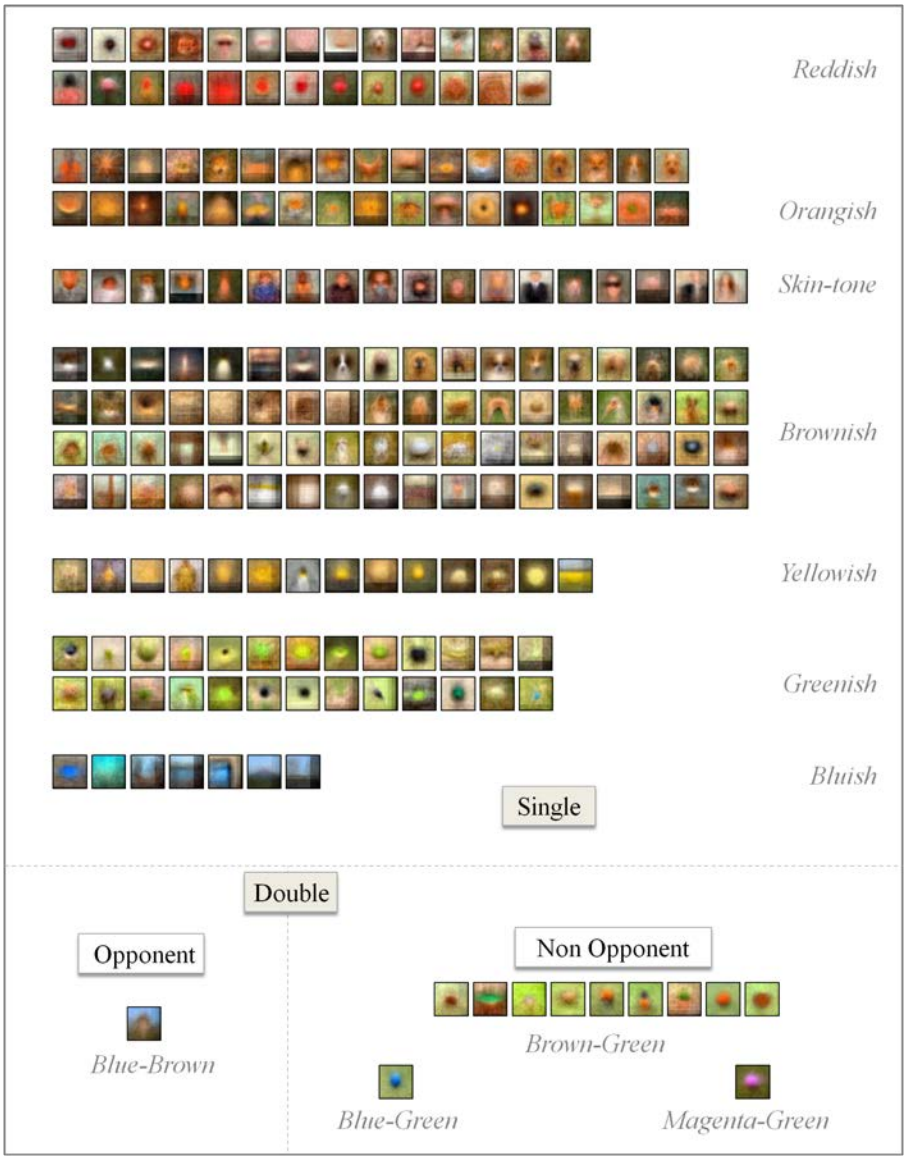
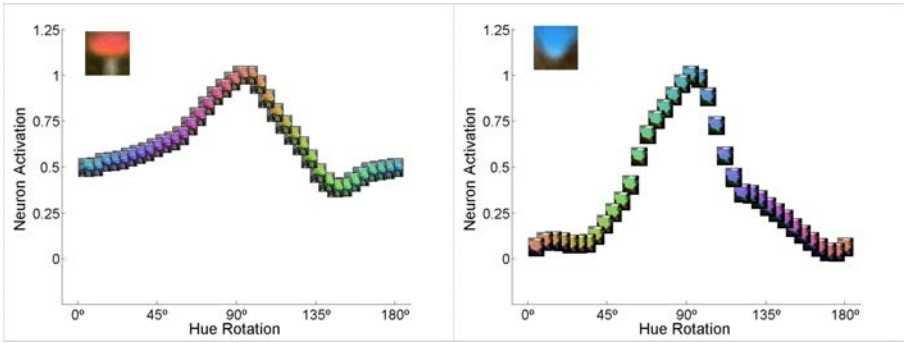
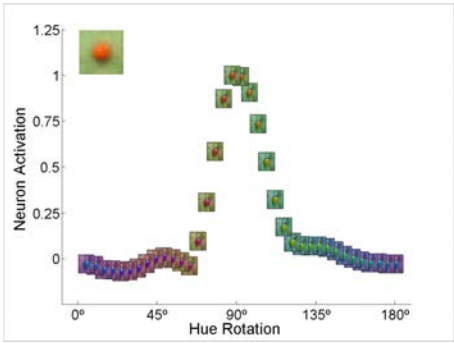


Figure 5.18 – Color selective neurons in Conv4 classified regarding to their color properties.

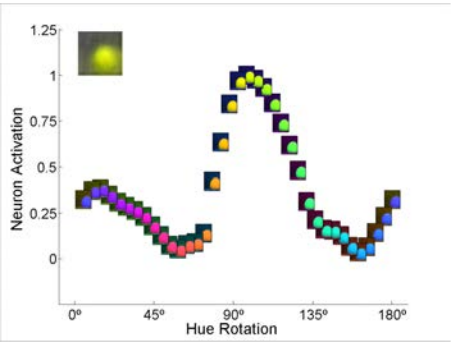


(a) Red Mushroom neuron in Conv4 ($\alpha = 0.53$)

(b) Sky neuron in Conv3 ($\alpha = 0.70$)



(c) Ladybug in a green surround neuron in Conv5 ($\alpha = 0.91$)



(d) Central yellowish blob in Conv3 ($\alpha = 0.96$)

Figure 5.19 – Activation curve of color selective neurons in deep layers for different versions of the same image by rotating hues. Maximum activation corresponds to a perfect matching with the original color of the image patch that maximally activates this neuron. Neuron NF is framed in blue (top-left).

present high probability to be color selective to a large range of colors and white surfaces, as well as an unusual spectral property sensitive to surrounds that may play a role in figure/ground separation. In any case, a detailed study of spatial selectivity on color artificial neurons could help in putting some more light on spatio-chromatic behaviors at these higher levels.

Although deeper visual ares of the human visual system are not known like in previous areas, in V4 and PIT (posterior inferior temporal cortex) color and complex

shape selectivity has been found [24]. Moreover, [116] stated that neurons in V4 are selective to large range of colors and white surfaces and also sensitive to surrounds that may participate in the separation between object and background.

5.4 Conclusions

In this chapter we study how color is encoded by a trained convolutional neural network. We propose a color selectivity index to characterize the neuron activation to the presence of a specific color in the input images. This is done by computing a color selectivity index on all the neurons and, concretelly, we report results on the VGG-M with five convolutional layers trained for object an object classification task on the ImageNet dataset by Chatefield *et al.* After analyzing indexes and observing the NFs across all the network neurons, we arrived to the following conclusions:

First, a large number of color selective neurons are found through all five layers of the neuronal architecture, although index is higher in shallow layers and lower in deeper layers. Color and shape are entangled together at each color selective neuron in all layers. Moreover, single color neurons represents single color regions while double color neurons are devoted to color edge detection.

Second, layer Conv1 shows a strong opponent property with three axes and a clear distinction between color and non color selective neurons. These two groups also show different spatial properties: low spatial frequency selectivity in color neurons, and high spatial frequency selectivity in non color neurons. Both conclusions show a clear correlation with human vision. Single neurons encode all colors in a primary basis of six basic colors, while double color neurons provoke the emergence of a color space holding a strong opponency property. This space is mainly characterized by the *Red-Cyan* and the *Blue-Yellow* axes jointly with a slightly representation of the emerged *Green-Magenta* axis.

Third, layer Conv2 presents two main particularities: (a) emergence of a new opponent axis in the same direction of the image dataset bias; and (b) a more dense sampling of hue of color selective neurons (suggesting some correlation with hue maps in V2). Additionally, non color selective neurons present more complex features than the oriented edges and basic gratings of the first layer. But they can not be seen as object shapes like those in subsequent layers. This layer seems to be representing surface details beyond its boundaries.

Fourth, layers Conv3, Conv4 and Conv5, all of them present color selective neurons with similar properties. Neurons are selective to colors mostly within the dataset bias, that lies on the *Blue-Orange* (or *-Brown*) axis plus some extensions towards *Green*. Regarding the spatial activation of color neurons, we identified four main groups of color selective neurons presenting different types of color

shape interactions: specific object shapes (as dog-faces, mushrooms, human body), homogeneous surround areas (as sky or green-grass), specific object-surrounds (blue-bird in grass or ladybug on leave), or generic colored shaped-blobs with strong intensity contrast. Finally, to mention that scale invariance is represented by using multiple neurons selective to different scales.

6 Extending neuron selectivities and other indexes

The neuron activity is characterized by several factors, such as color or shape of the intrinsic feature of the neuron. In consequence, the understanding of how these architectures are encoding features through layers requires to describe the neuron activity by a large set of properties. In this chapter we deal with this characterization by extending previous color selectivity index to shape or class selectivity and going opening a new research line focused beyond the individual neuron description through starting on jointly analyzing neuron activities of more than one property.
The grouping of all the selectivities give rise to our *NeFesi package*, which gather the information of each neuron and becomes a tool to improve on the understanding of the representational capabilities involved in CNNs.

6.1 Introduction

Understanding the neuron activity can be described through the responses of each neuron to a given specific property. Last chapter was focused on describing a methodology to quantify the correspondence between the activation of a neuron with the color property. However, to completely understand the neuron activity there are more properties that can be analyzed and, therefore, increasing the comprehension of what kind of properties are learned with these Convolutional Neural Network techniques. Following with the aim of proposing more selectivity indexes to describe each neuron activity in this chapter we propose other selectivity indexes. First, likewise in color, related to the image property but here are focused to shape selectivities. We define the shape involved in a neuron by its symmetry selectivity index and its orientation selectivity index. Second, a selectivity index that corresponds to characterize each neuron in a higher level as it is its relation with a specific class of the dataset. Let us to remark that this last selectivity index make sense in those kind of networks that are train to solve the problem of object recognition in a supervised way.

The extension of selectivity indexes is used to analyze the VGG-M network designed and trained by Chatefield *et al.* (see Sec. 4.3.2 for details in the architecture)

on the ImageNet dataset (introduced before in Sec. 4.3.1). However, like in previous chapters, our indexes could be applied to other networks following a sequential scheme. Let us to remark that any of the studied selectivities related to image properties are introduced as a constraint during the training, so that results emerge from the nature of the dataset. By the other hand, the training was performed in order to solve the problem of object recognition and therefore, the CNN tries to provide a prediction which coincides with the assigned label in the dataset. Nevertheless, the interest on analyzing the class selectivity falls on figure out if some of the neurons are specialized to encode features mainly related to a specific class.

Note that most of the approaches dealing with the understanding problem describes each neuron independently. However, as stated in [68], a fully understanding of these architectures requires an analysis of the distributed codes. Bringing our work closer to this idea, we propose to compare neurons regarding their neuron activity, so that a similarity between neurons can be identified. Moreover, the class selectivity index can be also used to describe the population code involved to a definition of a specific class.

This chapter presents shape selectivities in section 6.2 and the class selectivity index in section 6.3. Our proposal to compute the similarity between neurons is introduced in section 6.4. Finally, we summarize our proposals in a package called *NeFeSi* which group all the selectivities to describe some neuron activities in a section 6.5. We enclose this chapter with some discussion in section 6.6.

6.2 Shape selectivities

Assuming that each neuron acts as a feature detector by searching its own encoded visual property on the input image like a template matching, it is obvious that any neuron is selective to specific shape, so that we can assume that all the neurons have a direct relation with a particular shape structure. From the mere observation of the Neuron Features (see Section 4.3) we can conclude that shallower layers are devoted to simple features such as homogeneous regions, blobs, bars, edges or textures. However, deeper layers present a higher complexity in terms of their shape structure. Nevertheless, we pursuit on quantifying two properties related to shape, the mirror symmetry (Section 6.2.1) and rotation (Section 6.2.2) using a methodology similar to our proposal for defining color selectivity index (see Section 5.2.2).

6.2.1 Symmetry selectivity index

Taking into account that there are a large types of symmetry, let us to specify that by symmetry selectivity we refer to the capacity of a neuron to encode a visual property which has a half of its structure as a mirror of the other part. Therefore, a neuron with a high symmetry selectivity index does not modify its neuron activation to an image an its mirrored transformation through a symmetry axis. Thus, given the set of N-scored images $\{I_t\}_{t=1:N}$ of the j-th neuron at layer l and the angle θ of the symmetry axis, we define the θ-symmetry selectivity index as follows:

$$\sigma^l_{\theta,j} = \frac{\sum_{t=1}^{N} \hat{a}^l_j(\phi(I_t,\theta))}{\sum_{j=1}^{N} \hat{a}^l_j(I_t)} \tag{6.1}$$

where ϕ is the function that reflects each of the top-scored images along the symmetry axis of angle θ. In our experiments we consider four types of θ-symmetry depending on the angle of this axis : 0°, 45°, 90° and 0°. These, are averaged to obtain an index of a global symmetry selectivity:

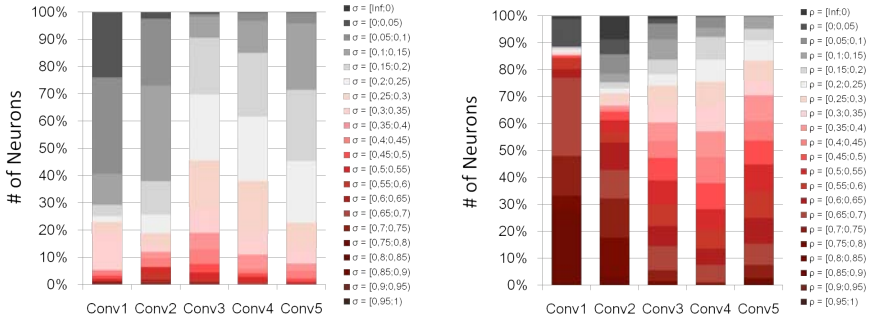
$$\sigma^l_j = \frac{1}{N_\theta} \sum_{\theta} \sigma^l_{\theta,j} \tag{6.2}$$

where N_θ is the number of different symmetry axis defined.

This global symmetry seeks to identify neurons encoding visual features that are invariant to any symmetry transformation, such as homogeneous, blob or radial features.

6.2.2 Orientation selectivity index

The generalization achieved with these techniques is partially done through: (a) using the convolution operation that allows translation invariance of a specific feature inside the overall input image; and (b) using pooling operations which introduces some local translation invariance. However, we believe that some neurons allow some degree of invariance to small rotations and we propose to quantify this degree using an orientation selectivity index. In fact, these degree was shown in Chapter 5 in Fig. 5.8, where variations in rotation affect to the neuron curve in a Gaussian shape. The more narrow Gaussian, the more orientation selectivity has the neuron. In this sense, given the set of N-scored images $\{I_t\}_{t=1:N}$ of the j-th neuron at layer l and the angle θ of the rotation, we define the θ-orientation selectivity index as:



(a) Global symmetry selectivity

(b) Global orientation selectivity

Figure 6.1 – Percentage of network shape selectivity indexes within different ranges along the set of convolutional layers. (a) Global symmetry selectivity, and (b) Global orientation selectivity.

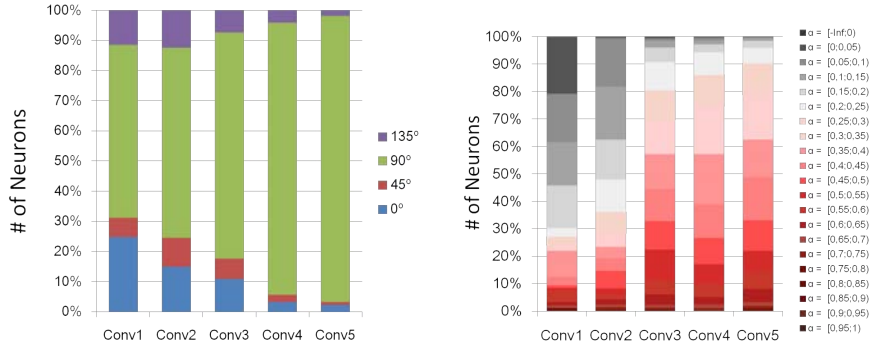
$$\rho^l_{\Theta,j} = 1 - \frac{\sum_{t=1}^{N} \hat{a}^l_j(\Phi(I_t,\Theta))}{\sum_{j=1}^{N} \hat{a}^l_j(I_t)} \tag{6.3}$$

where Φ is the function that rotates each of the top-scored images in Θ degrees. Thus, a high Θ-orientation selectivity index indicates that the visual feature encoded by a specific neuron presents a strong alignment with a specific orientation, while a low index characterize the neuron as allowing some degree of invariance through a rotation of Θ degrees. Likewise in symmetry selectivity, we also define a global orientation selectivity by averaging the set of Θ-orientation selectivities:

$$\rho^l_j = \frac{1}{N_\Theta} \sum_{\theta} \rho^l_{\Theta,j} \tag{6.4}$$

where N_Θ is the number of different rotations defined. Note that lower global orientation selectivity indexes will be achieved when the neuron is insensitive to most of rotation changes, and therefore, it is another procedure to identify visual features related to homogeneous regions, centered blobs or centered and radial structures.

We compute our shape selectivity indexes for all the neurons in the VGG-M. The distribution of them along the different convolutional layers can be seen in Fig. 6.1, where the distribution of global symmetry indexes (Fig. 6.1a) and for the global rotation (Fig. 6.1b) are shown. From this plot we see that few of neurons are encoding



(a) Maximum θ-symmetry selectivity neuron distribution comparing the selectivity index for $\theta = 0°, \theta = 45°, \theta = 90°, \theta = 135°$

(b) Vertical symmetry ($\theta = 90°$) distribution along layers

Figure 6.2 – Partial symetry distribution. (a) shows the distribution of neurons regarding which kind of partial symmetry is higher compared to the other partial symmetry indexes. (b) shows the distribution of the vertical symmetry along layers.

features with a high global symmetry property (most found in Conv3), while a large number of neurons present a high correlation with a specific orientation. However, within the different symmetries tested ($\theta = 0, 45, 90, 135$) we found that there are more neurons that present a higher vertical symmetry (*i.e.*, $\theta = 90°$) compared with the remaining, as can be seen in Fig. 6.2a, where we plot the distribution of neurons regarding their maximum θ-symmetry. Although few neurons present a high global symmetry index, there exist a high amount of neurons presenting selectivity to vertical symmetry, as shown in Fig. 6.2b which increase through layers. This can be correlated with several studies that confirm that vertical symmetry has a clear advantage in visual perception, compared to the horizontal (reviewed in [43, 139]) and also to differentiate a specific object from the background [87]. In Fig. 6.3 we show some Neuron Features classified regarding their horizontal and vertical symmetries.

Regarding the orientation, note from Fig. 6.1b shallower layer present higher global orientation selectivity indexes, which also confirms that deeper neurons admit higher variability in their features. In Fig. 6.3 we plot several Neuron Features (top left) with their corresponding set of top-scored images (bottom) for each neuron example, sorted from less global orientation selectivity (left) to higher global orientation selectivity (right). Four examples are shown for each convolutional layer (from Conv1 in the top to Conv5 in the bottom). We can see how neurons with lower

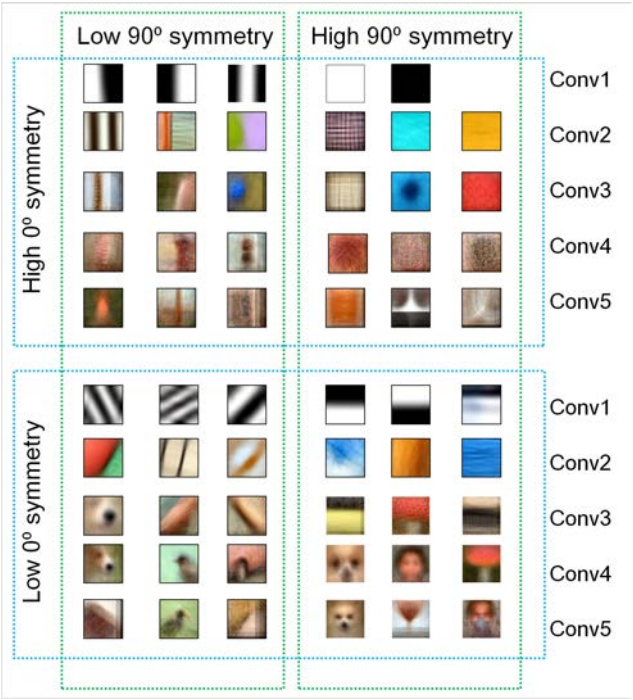


Figure 6.3 – Examples of Neuron Features classified regrading their horizontal and vertical symmetry selectivities for each convolutional layer

global orientation selectivity indexes are related to (a) homogeneous regions (see left example of Conv1), (b) blob regions (left examples of Conv2 or Conv3) or (c) radial structures (left examples of Conv4 and Conv5). Moreover, neurons presenting a higher global orientation selectivity index present less variations in their structure, as can be seen from the set of N-top-scored images.

6.3 Class selectivity index

Previous work has been focused on describing selectivity indexes related to image properties. Nevertheless, this section translate the selectivity property to a higher level of abstraction like image labels. The idea has been also recently proposed in [7], where authors have introduced an approach to quantify the relationship between a neuron and a visual concept based on broadly and dense

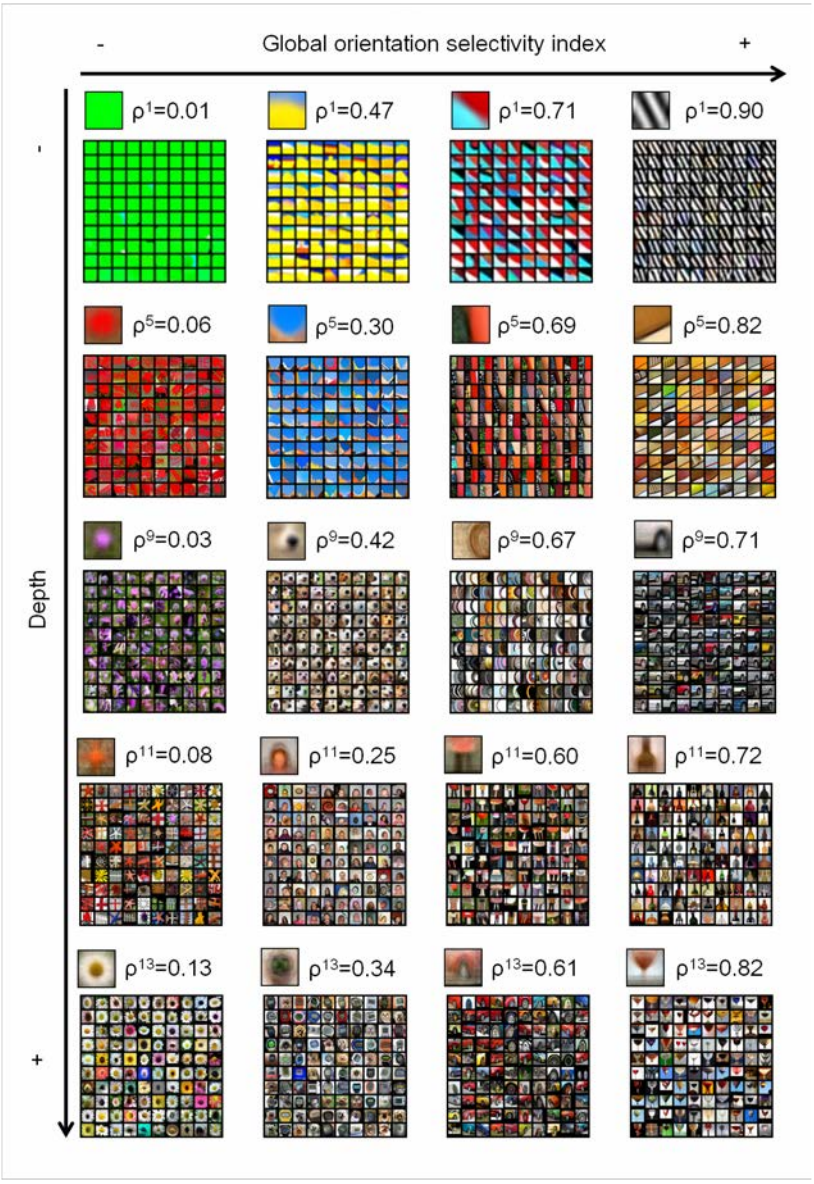


Figure 6.4 – Examples of Neurons, sorted from low to high global orientation selectivity index from left to right and from early to deeper layers from top to bottom.

labeled dataset. Therefore, the goal is to define the class selectivity as a property of a neuron that can help to establish its discriminative power for one specific class or can allow to cluster neurons accordingly with the ontological properties of their class labels.

We propose a method to compute a class selectivity index for individual neurons by compiling the class labels of the images that maximally activate this neuron in a single descriptor. We define class selectivity from the set of class labels of the N-top scored images used to build the NF of a specific neuron. To quantify this index we build the class label distribution of the full set of images. As in the color selectivity index, we weight the significance of a class label by the relative activation of its image. Thus, the relative frequency of each class c for a certain neuron is defined as:

$$\mathscr{H}^l_{c,j} = \frac{\sum_u^{N_c} \hat{a}^l_j(I_u)}{\sum_t^N \hat{a}^l_j(I_t)} \tag{6.5}$$

where N_c refers to the number of images, among the N top scored images activating this neuron, that belong to class c ($\{I_u\}_{u=1:N_u} \subseteq \{I_t\}_{t=1:N}$). From now on, we will refer to vector $\mathscr{H}_c$ of a given neuron as the c-selectivity index.

A set of c-selectivity indexes involved in a neuron can be used for measuring the class intersection index between two different neurons, *i.e.*, a metric to compare how many classes are shared by these two neurons. We propose to compute this metric as:

$$\cap^l_{j_1,j_2} = \frac{|c_t|}{\max(|c_{j_1}|,|c_{j_2}|)}, \quad \text{s.t.} \quad \mathscr{H}^l_{c_t,j_i} \geq th \quad \forall i = 1,2 \tag{6.6}$$

where c_{j_1} are the classes involved in the j_1-th neuron, *i.e*, the set of c_t such that $\mathscr{H}^l_{c_t,j_i} \geq th$ (and analogous for the j_2-th neuron); and th is a threshold to avoid classes with a low c-selectivity index.

Given the densities for all the classes, our class selectivity index is defined as follows:

$$\gamma^l_j = \frac{N-M}{N-1} \tag{6.7}$$

where M is the minimum number of classes that covers a pre-fixed ratio, th, of the neuron activation. This can be denoted as $\sum_c^M \mathscr{H}_c \geq th$. This threshold allows to avoid considering class labels with very small activation weights. Jointly with the index value, the selectivity provides the set of M classes that describe the neuron selectivity and their corresponding relative frequency values.

Thus, a low class selectivity index indicates a poor contribution of this neuron

to a single class (minimum is 0 when $M = N$), while a high value (maximum is 1) indicates a strong contribution of this neuron to a single class. In between we can have different degrees of selectivity to different number of classes. Obviously, this index is irrelevant for the last fully connected layers in a CNN, but it allows to group related neurons across different convolutional layers.

Here we want to point out, that this index can also contribute to give some insights about the problem of how information is coded through layers, in the debate of localist and distributed neural codes we mentioned before in Sec. 4.3.3 ([68]). Neurons with high class selectivity index should be in line with a localist code, while neurons with low class selectivity index should be part of a distributed code. The way the index is defined allows a large range of interpretations in between these two kinds of coding as it has been outlined in the visual coding literature.

Following with the analysis of ranking neurons by their response to a certain property, here we focus on the proposed selectivity index that relates to image labels instead of to an image property, is the class selectivity index, which only applies for classification networks. We report the results of different experiments where we have fixed $th = 1$, which means we consider all the class labels for the $N = 100$ images that maximally activates the neuron. As we mentioned before, this index can enlighten how classes are encoded through the net layers, that again it can be related to the scientific problem of how general object recognition is encoded in the human brain.

Here we hypothesize that the difference between localist or distributed codes [68] could correlate with the idea of neurons highly selective to a single class and neurons highly selective to several classes, respectively. We already discussed about these two concepts in Sec. 4.3.3 where we speculate by relating the shape variability of each neuron with these terms, *i.e.*, different visual features highly spike a specific neuron. However, as stated in [68], the interpretation of this terms can be confused. From another point of view, localist codes can be referred to the relation of the neuron activation for a given specific object, *i.e.*, a specific class, while distributed codes are achieved when several neurons are jointly activated, describing, in this way, the object and differing from another object when, for example, these neuron configuration presents small changes in some of the neurons (*e.g.*, some of the neurons highly activated in the first case are disabled and substituted by others for codifying the second object). In this sense, the debate falls to relate each neuron activation with the presence (or absence) of a specific object. Taking into account this new interpretation, let us to make a short parenthesis on the class selectivity index and go back to the Neuron Feature visualization. In Sec. 4.3.3 we present a multifaceted NF visualization (and shown in Fig. 4.4) that tries to include the variability of the neuron taking into account the representation of visual features involved to the activation of a specific neuron, likewise is pursued in [99]. Here,

and incorporating the new interpretation of the localist and distributed codes, we present a new way to visualize the multifaceted NF visualizations by splitting the N-top scored images regarding the class. Therefore, each facet visualization is obtained by computing the NF over the subset of images belonging to a specific class. Following with the same example of the bird-neuron in Conv4, this neuron presents a high class selectivity index, $\gamma = 0.95$ being, therefore, highly related with different but few classes. Concretely, it is mostly related to the goldfinch (with a c-selectivity index of $\mathscr{H} = 0.33$) , jay ($\mathscr{H} = 0.30$) and brambling ($\mathscr{H} = 0.17$) classes. These visualizations are shown in Fig. 6.5 where the discriminative power of the color to distinguish between these three classes of birds emerged in each NF.

(a) original NF **(b)** Class Multifaceted representation

Figure 6.5 – NF visualization (a) without considering multifacets, and (b) for each class. Birds are classified by their type of birds (*brambling, jay* and *goldfish*, from left to right). Note that these classes are distinguished by their inherent color.

Regarding the class selectivity index, in a first experiment we analyze how many neurons present different degrees of class selectivity through layers. The bars in Fig. 6.6 plot the relative quantity of neurons that are class selective compared to those that are not. Grey represents the ratio of neurons that are not activated by a single class and reddish represent neurons that are highly activated by a single class. Opposite to what we showed about color selectivity, we found most of class selective neurons in deeper layers, and no class selectivity in shallow layers, as expected. We have moved from a very basic image property, color, to a very high level property, class label. This fact corroborates the idea that CNNs start by defining basic feature detectors that are shared by most of the classes, and the neurons become more specialized when they belong to deeper layers representing larger areas in the image space and therefore more complex shapes. We start to have neurons with relevant class selectivity in layer Conv3, where 5% of neurons is quite class selective and we found some neurons with a degree of selective close to 1. These ratios progressively increase up to layer conv5 where we have more than 50% of neurons with a class

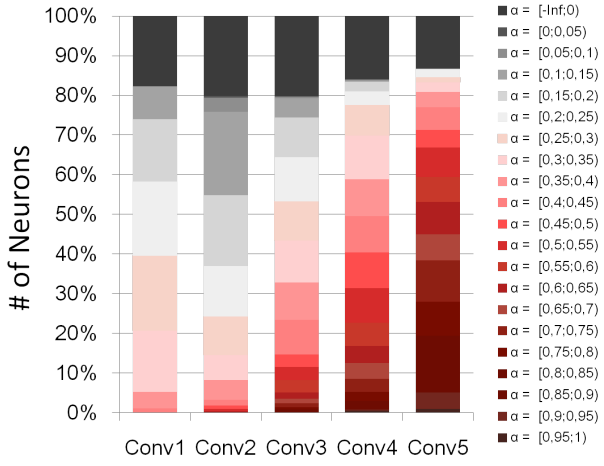


Figure 6.6 – Number of neurons and degree of class selectivity through layers. Grayish bars are for low index values and reddish for high index values.

selectivity index greater than 0.6, that means that we have less than 40 different classes activating this neuron, which is a very selective ratio considering the number of classes of the ImageNet dataset. In the same layer, 20% of neurons present a high class selectivity index, than means less than 20 different classes.

Secondly, we have visualized the properties of a set of images presenting different degrees of class selectivity in Fig. 6.7 for different levels of depth. We visualize each neuron with their NF visualization and the corresponding cropped images. We also show two *tag clouds* of each neuron. They visualize the importance of each class label. With an orange frame we plot the leave classes of the ImageNet ontology, while in the green frame we plot generic classes. This second analysis could help finding neurons that are specialized to a general semantic concept that different final classes share. Note that neurons with high class selectivity index have a set of cropped images that we can identify as belonging to the same class.

Combining our NF visualization with the NF having a high class selectivity index of a specific class, we can infer main visual structures that allow an optimal representation. In Fig. 6.8 we show the NFs of neurons with high bell pepper selectivity distribution and a sample of the cropped images that highly activates each of the neurons. This figure shows how bell peppers are codified or seen through the network describing the variety of elements of the class that allow its discrimination. We can see the importance of color for this class, there are

Figure 6.7 – Neurons with different class selectivity indexes. For each neuron two images (top: NF, bottom: cropped images) and two tag clouds (top: leave classes, bottom: all classes in the ontology).

neurons selective to a variety of pepper colors (yellowish, reddish or greenish), additionally we can see the importance of the pepper stem as another discriminative property, and finally the global shape of the pepper as an elongated colored blob ending. While in the first two layers we find edges combining different pairs of colors characterizing boundaries of different colored peppers, in the deepest layers we find the higher level properties (stems, pepper-shapes and color of peppers) similarly encoded in all the layers at different scales. Finally we stress the utility of ranking images by selectivity indexes in Fig. 6.9, where we show interesting neurons in different convolutional layers that present high values for both selectivity indexes, neurons which are both, color and class selective.

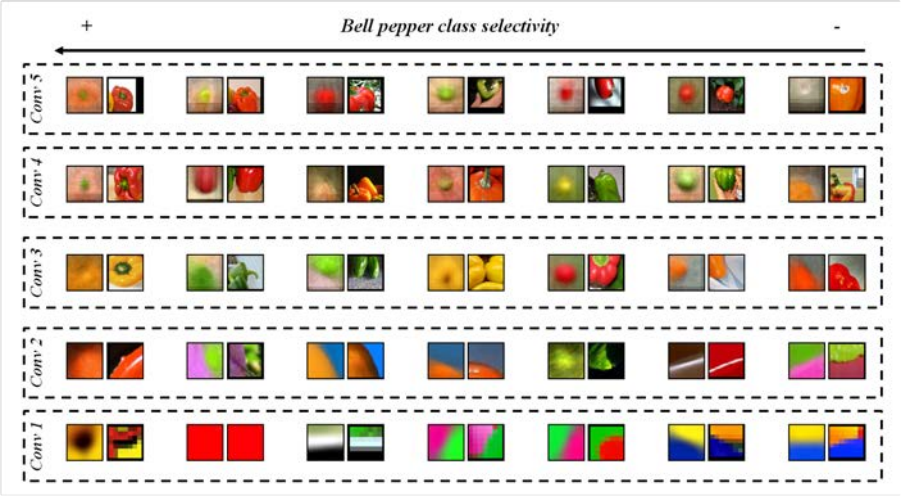


Figure 6.8 – Neurons with a high class selectivity to bell pepper class. For each pair we show the corresponding NF (left) and one of the top cropped images categorized as a bell pepper that highly activates the neuron (right). Neurons are shown from high class selectivity to less from left to right.

6.4 Neuron-pair similarity index

The methodology used for describing these set of color, symmetry and rotation selectivity indexes can be also extended for study neuron features beyond individual activities and identifying neurons that are learned to encode similar features. Similarly to the work reported in [80] this metric pursuits to compare different neurons and give a tool to seek for similar or different neurons in the same layer. However, it can also be extended to compare neurons in two different architectures. For this purpose, we define the neuron-pair similarity index by comparing the crossed area under the activation curve between two neurons (j_1-th and j_2-th neurons) of a layer l as follows:

$$\xi^l_{j_1,j_2} = \frac{1}{2}\left(\frac{\sum_{t=1}^{N} \hat{a}^l_{j_1}(I_t^{j_2})}{\sum_{j=1}^{N} \hat{a}^l_{j_1}(I_t^{j_1})} + \frac{\sum_{t=1}^{N} \hat{a}^l_{j_2}(I_t^{j_1})}{\sum_{j=1}^{N} \hat{a}^l_{j_2}(I_t^{j_2})}\right) \tag{6.8}$$

where $\{I_t^{j_1}\}_{t=1:N}$ and $\{I_t^{j_2}\}_{t=1:N}$ are the set of N-top scored images the j_1-th and j_2-th neurons of a layer l, respectively.

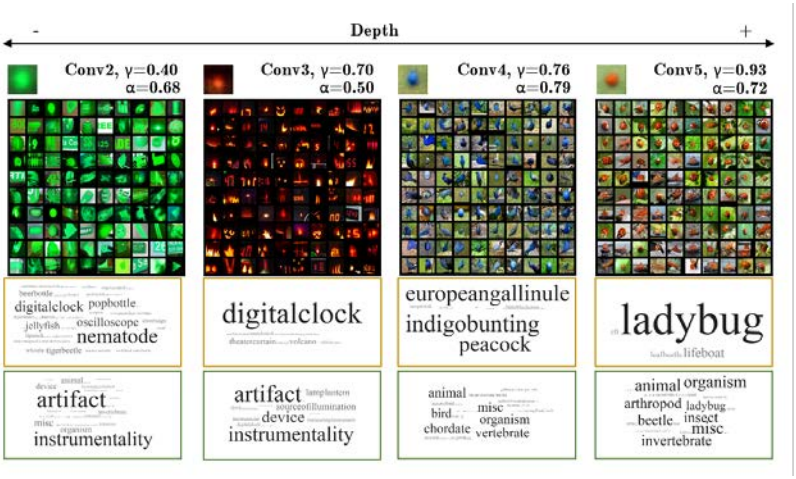


Figure 6.9 – Examples of neurons with high color and class selectivity indexes.

Our neuron-pair similarity index allows to compare neurons regarding their neuron activity to same stimuli. Figs. 6.10, 6.11 and 6.12 visualize this similarity from a neuron in Conv3, Conv4 and Conv5, respectively. The studied neurons are plotted in the centers of the figures, and the rest of neurons are plotted in circles depending on their similarity index with the center neuron. Inner circle contains closest neurons while outer circle contains neurons with less similarity. Note that closest neurons share concepts with the central neuron: rounded edge or corner for the neuron shown in Fig. 6.10, blackish-blob mostly related to a car wheel in Fig. 6.11 or the face shape in Fig. 6.12. The density of each circle also informs about the discriminative power of the neuron compared with the remaining neurons in the same layer: the less density in inner circles the more discriminative the neuron is. By discriminative power we refer to the specialization of its visual feature. In this sense, the car wheel (Fig. 6.11) or the face (Fig. 6.12) neurons represent specific features that are not shared with most of neurons, while a rounded edge (Fig. 6.10) is highly represented with several neurons in Conv3.

Moreover, the set of similarity index computer by all possible neuron pairs within the same layer can be combined with the t-SNE visualization [134] to plot a set of neurons in a two-dimensional map where similarities between neurons are preserved. In this case, we use the opposite similarity index, *i.e.*, we consider $1-\xi$ to compute the distance between neurons. In Fig. 6.13 we show this mapping

Figure 6.10 – Similarity between neurons of a neuron in Conv3 representing a rounded edge. Each neuron is plotted in a circle depending on the similarity index with respect to the central neuron. Outer circles gather NFs with low similarity index, while inner circles contain the visualization of similar neurons.

Figure 6.11 – Similarity between neurons of a neuron in Conv4 representing a car wheel. Each neuron is plotted in a circle depending on the similarity index with respect to the central neuron. Outer circles gather NFs with low similarity index, while inner circles contain the visualization of similar neurons.

Figure 6.12 – Similarity between neurons of a neuron in Conv5 representing a human face. Each neuron is plotted in a circle depending on the similarity index with respect to the central neuron. Outer circles gather NFs with low similarity index, while inner circles contain the visualization of similar neurons.

for the set of neurons in Conv1. Neurons are gathered depending on their color appearance, orientation and frequency.

6.5 A visualization tool

This thesis has been mainly focused to propose a set of selectivity indexes to favor on the neuron understanding problem. Each selectivity index is devoted to describe a property by isolating it from other factors. However, neurons are spiked by a combination of several properties and gathering all the information the understanding of each neuron activity is improved. Thus, we group all of our proposals in a visualization tool called *NeFeSi: Neuron Feature and Selectivity index for CNN visualization*, which helps on the understanding of the intermediate learned features for CNNs that follows a sequential scheme.

In this section, therefore, we include some visualizations that include more than one selectivity index and tries to highlight the potential of visualizing intrinsic features encoded by each neuron, as well as the description of the neuron activity through individual properties for the understanding problem.

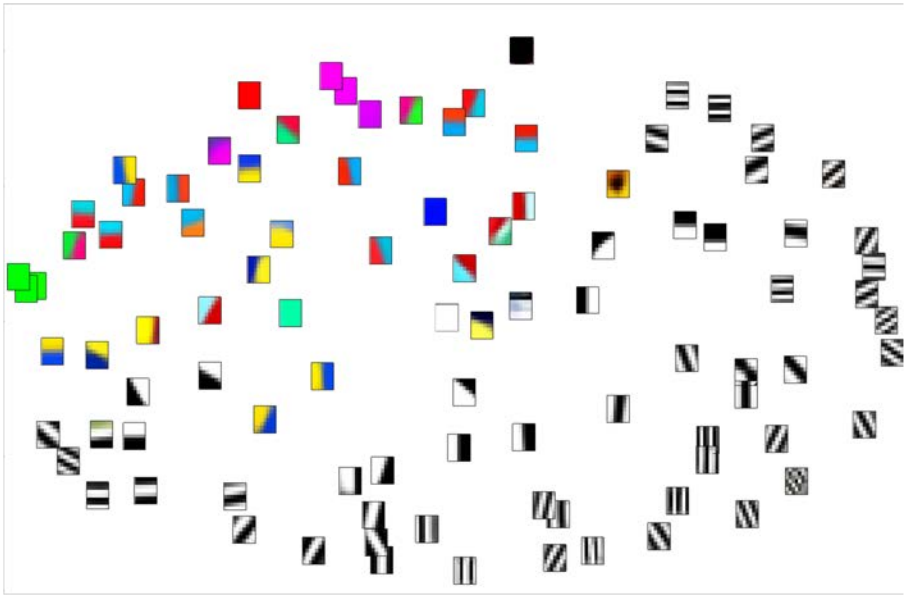


Figure 6.13 – t-SNE visualization of the neurons in Conv1 from the set of similarity indexes.

In Fig. 6.14 we show the highest similar neurons to a specific neuron (query) to and the most distant neurons. Each studied neuron is shown in the left, while the other NF plotted in the same row are sorted by their similarity from high to low similarity index (from left to right). All presented selectivity indexes are described under each visualization[1]. Let us to deeply discuss about the examples shown in this figure in the following lines.

Studied neuron for Conv1 is cleary color-selective ($\alpha = 0.99$) as well as this property is discriminant from its similar neurons. Although query neuron is selective to red color closest neurons are selective to magenta, since any other neuron in this layer is devoted to reddish homogeneous regions. Other interesting property of this neuron (and of the set of similar neurons) is that it is not selective to any specific orientation. The most distant neuron also accomplishes these properties (high color selectivity and low orientation selectivity index) but it is representing a distinct color. Note that red and green color tones have a perfect opponent property in the Opponent color space.

[1] In Fig. 6.14 the intersection index is calculate by setting $th = 0.95$.

For the case plotted in Conv2, color is contemptible but orientation selectivity gains relevance. This is inferred not only for the high orientation selectivity shown by the query neuron ($\rho = 0.71$) but also for the index of most similar neurons. Due to the importance of this property for the studied neuron, neurons of the tail present also a high orientation selectivity but, from their visualization, their orientation is almost perpendicular to the studied neuron.

The example of the face neuron shown for Conv3 does not present a property that covers up the rest of the factors, but as much as color, class, orientation and symmetry selectivities are defining its neuron activity. Closest neurons shares a color selectivity to a similar color tone and are also related to few of the classes in the dataset. Concretely, same classes are represented from both studied neuron and closer neurons, since the class intersection index is high enough.

The remaining examples (neurons of Conv4 and Conv5) present similar properties. These neurons are mostly characterized by having high class selectivity index (although the set of classes in which they are related is poorly shared by their closest neurons, since their class intersection indexes are low) and holding some symmetry property. Note, specially for the neuron in Conv5, the high global symmetry indexes jointly with low orientation selectivities for the most similar neurons and the studied one (*i.e.*, it is encoding a feature which have a radial symmetry). Most distant neurons have lost this symmetry property.

The t-SNE visualization shown in Fig. 6.13 provides a mapping where similar neurons are grouped spatially. However, due to the amount of neurons belonging to deeper layers, the analysis from this plot can be difficult, as can be seen in Fig. 6.15a for the case of the Conv4. However, some clusters emerges from this plot: dog faces in the bottom right, close to human faces; homogeneous regions in the bottom left; circular edges in the top left; or car wheels in the middle top. Nevertheless, this visualization can be used to dissect from the neuron population under some criteria. For example, taking into account the c-class selectivity this visualization allows to (a) visualize those NF highly involved to codify a feature of the class c, (b) infer about the population code that uses the network to characterize a specific class c, and (c) visualize how similar are the neurons involved in this class. The remaining plots in Fig. 6.15 use this dissection to show the NFs of the neurons involved for three classes of birds, the *brambling* (Fig. 6.15b), the *jay* (Fig. 6.15c) and the *goldfinch*(Fig. 6.15d). In the top left corner of each plot we show an image of the ImageNet dataset belonging to the corresponding class. We have already these classes in Fig. 6.5 where, from the multifaceted visualization obtained from the type of classes where each of the N-top scored images belongs, it emerged that color was a discriminative property to differentiate between these types of birds. In this figure, common Neuron Features are framed in red. Note that orangish Neuron Features appear at top right in Fig. 6.15b, while yellowish neurons emerged at the middle

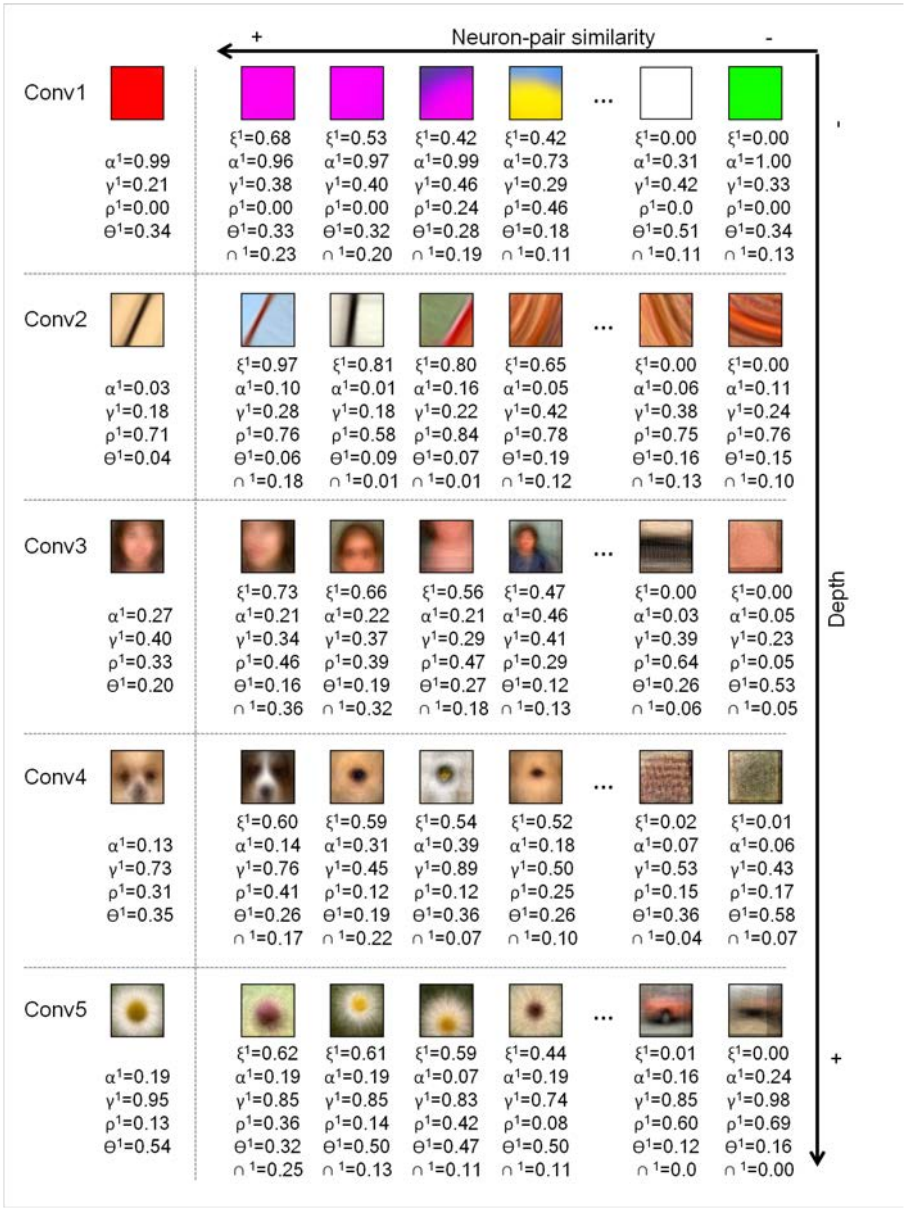
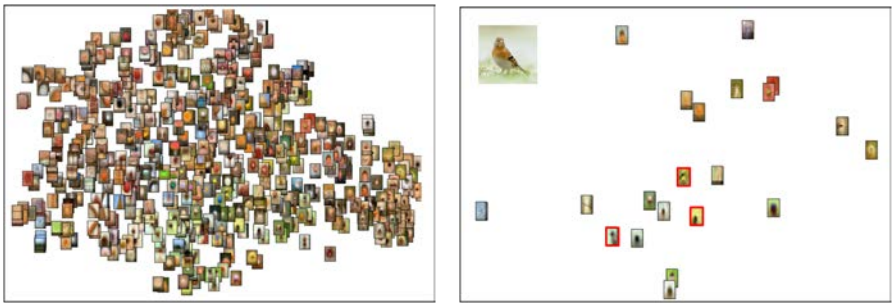


Figure 6.14 – Description of five neurons of the VGG-M network. The analysis is performed through the observation of the most similar neurons and from the set of indexes analyzed in this thesis.First row shows NFs of neurons in Conv1, while last row is devoted to the Conv5. First column shows the set of neurons we are trying to describe (queries), and the set of NFs plotted in the right part shows the head and the tail of the sorted neurons regarding their similarity with the query neuron.



(a) L11 t-SNE visualization from similarity indexes

(b) Neurons with $\mathscr{H} > 0$ for *c= brambling*

(c) Neurons with $\mathscr{H} > 0$ for *c= jay*

(d) Neurons with $\mathscr{H} > 0$ for *c= goldfinch*

Figure 6.15 – (a) t-SNE visualization from similarity index for Conv4. (b),(c) and (d) shows the subset of neurons in Conv4 involved in the *brambling, jay* and *goldfinch* birds, respectively.

right in Fig. 6.15d and a white blob surrounded by a blue background comes out at the bottom left in Fig. 6.15c.

6.6 Conclusions

In this chapter we have extended the description of the neuron activity through other properties with the aim of providing a general tool improving the understanding of each neuron role.

First, two selectivities are defined to extract properties related to the shape of the visual feature: the symmetry and the orientation. These global selectivities are obtained from a set of partial selectivities that are devoted to characterize

changes on the neuron activity when the set of images that highly spike a specific neuron suffer a specific symmetry or specific rotation transformation. The analysis performed in these selectivities concludes that a few of the neurons present a high global symmetry (*i.e.*, few neurons encodes visual features that are symmetric in different symmetry axes simultaneously), but within different angles of symmetry axes, the vertical symmetry is the most preserved in general. In fact, this property is mostly kept in deeper layers. On the other hand, there are neurons sensitive on rotation changes in all the layers. However, deeper layers allow some degree of tolerance in this transformation that favors to the generalization achieved by the CNN.

Second, and in a higher level of abstraction, we propose to describe the neuron activity from the relation of a neuron of having a high activation to a stimuli of a specific class. This property is related to deeper layers, since the number of class selective neuron increases through layers. Nevertheless, the results show that CNNs present neurons specialized to some class (or classes) much before the fully connected layers, even few neurons in early layers present a high class selectivity, such as the digital clock or the window screen neurons found in Conv3 and Conv2, respectively. Using this abstracted selectivity, we also speculate about the debate of localist and distributed codes. Here we suggest that neurons with high class selectivity index may be closer to a localist code.

This chapter opens a new research line in the understanding problem, motivating to extend the analysis of the neuron activities beyond an individual neuron. In this sense, it proposes a similarity index that compares the neuron activities of two neurons. With this description, the discriminative power within a layer of each neuron can be inferred: the more amount of similar neurons has a specific neuron, the less discriminative power.

Finally, we enhance the potential of providing selectivities to describe each neuron activity by isolating several factors and, afterwards, gathering them to improve and generalize the understanding of a neuron. We enclose the visualization and understanding proposals of this thesis in a package called *NeFeSi*, which can be useful to analyze and provide a general map of the learned features of a trained CNN.

7 Conclusions and Further work

7.1 Conclusions

The problem of visual recognition is one of the main problems in computer vision. It consists on understanding, identifying and recognizing visual properties from an image. However, it is a difficult task. First, because there exist an infinity of different objects int the real world, and each one presents a large variability of appearance regarding the shape, point of view, color and texture, among others. Second, because it is unknown how to represent the image content in a optimal way. With the aim of providing an approach able to deal with this problem, the feature extraction plays a key role.

Taking inspiration from known evidences in early stages of ventral stream of the primate visual system, the progress on this science is achieved by directly defining handcraft specific features to decompose image in a high-dimensional space, good enough for solving a determinate visual task. Most of these processes consisted on a simple hierarchy based on a single step devoted to feature extraction and combined with a learning algorithm (named as flat schemes in [71]). However, the feature extraction is performed in a color space and it is so important to propose a set of characteristics that is capable to deal with the maximum possible generalization as well as the color space where these characteristics are represented.

In order to enhance color differences within a channel and favors feature extraction, in the second chapter we propose a new color representation called *more-than-three color coding* (MTT) which minimizes the inter-channel correlation of the color representation and maximizes the local contrast to favor the feature detection in each channel. However, this representation is adapted to the image content, so that it allocates a channel to each distinctive color in the image. In general, the more diversity the image has, the greater number of color channels our representation has. Other properties of this representation beyond a maximization of the intra-channel local contrast and a minimization of inter-channel correlation are the ability to capture image details, the illuminant invariance and the improvement on the feature detection. These advantages demonstrate to favor the scene recognition problem, overcoming state-of-the-art results.

In spite of the simplicity of flat schemes, computer vision achieved a promising

progress on solving tasks such as object recognition or object detector. However, they presented a lack of generality far away from the capabilities of the visual cortex. Advances in neuroscience, jointly with the appearance of large datasets and the computational resources give arise to a new way of developing computer vision algorithms which follow a deep hierarchy (likewise the type of scheme followed by the visual cortex) combined with learning algorithms capable to automatically learn optimal visual features. This new paradigm has been overcoming results on different visual tasks, specifically with the Convolutional Neural Networks (CNNs) techniques. Moreover, CNNs have shown capabilities that rival the primate performances [17] and a suitable framwork to model biological vision [17, 69, 71].

Convolutional Neural Networks are deep architectures that sequentially stack layers to increase, in this way, the complexity of the encoded visual features. They are usually built from a composition of convolutional layers, pooling layers and non-linear layers, which successively produce several image representation changes. Convolutional layers are composed of a set of neurons, each of them acts as a feature detector. Therefore, this kind of layers are the main responsible of the representational changes and the codification the input image is based on the visual features encoded by each one of the neurons. However, this weights are successively adapted using an automatic learning process that gives rise to the lack of understanding of their intermediate representations.

In this thesis we deal to face the problem of the understanding the intermediate representations. First, a visualization of the intrinsic features encoded by each neuron is proposed in chapter 6. We label it as *Neuron Feature* (NF) and it is built by computing weighted average of the images of the dataset that activate a specific neuron the most. With this computation we achieve a visualization keeping more natural properties compared to most of the state-of-the-art approaches. Although neurons with a high variability of visual features entail blurred NF, most of the neurons present a meaningful NF visualization. We focused our studies on the VGG-M trained and designed by Chatefield *et al.* on the ImageNet dataset [114]. The visualization of its neurons shows that early layers are devoted to simple and basic features, such as oriented edges, oriented bars, textures or homogeneous regions. However, deeper layers are specialized to more complex features (*e.g*, a human face) or object surround neurons. In addition, a hierarchical composition of NFs improves the understanding of the neuron activity.

In parallel to the visualization and trying to provide a visualizatin tool (called *NeFeSi: Neuron Feature and Selectivity index)* able to automatically describe a neuron. For this propose we propose a methodology inspired in physiological techniques to quantify and describe each of the neurons in terms of specific features through several selectivity indexes and, therefore, allowing to classify neurons of a trained CNN. These indexes reflect the ability of a neuron to response when a

specific visual property is introduced as a stimulus. Several selectivity indexes are proposed related to color, symmetry, orientation and class. The use of these indexes jointly with the visualization of the NF of the neurons simplifies the description of the neuron activity and improves the understanding of its activity. In general, we find that there are color selective neurons in all the layers, while few of the neurons contains visual features presenting a symmetry simultaneously to different directions. However, the vertical symmetry is the most preserved and neurons in deeper layers present higher indexes for this property. On the other hand, oriented selective neurons are found in all the layers. However, while early layers present higher indexes of orientation selectivities, class selectivity is more devoted to deeper neurons.

Finally, in this thesis, color selectivities are deeply studied and it allows to find some parallelisms with primate visuals system. In Conv1 color edges are represented through opponent pair of colors with a low frequency, while brightness neurons present a larger sampling on the frequency that correlates with findings in V1. Regarding color neurons in Conv2, they cover a higher sampling of the color hue that can be equated with the hue maps in V2. However, the overall color neurons present a strong correlation with the *Orange-Blue* bias of the dataset. Although color is studied independently from other properties, neurons present a strong color and shape entanglement in all the layers, moving from simple and basic features in early layers to object surround neurons in deeper layers.

7.2 Further Perspective

The impressive performance achieved by the new trend of solving computer vision problems with deep learning techniques such as Convolutional Neural Networks awakes the curiosity of how they encode visual information to achieve such results. As presented in this thesis, describing neurons from their inherent properties can be an useful approach to understand the representation capabilities of these architectures. However, this study opens new research lines that we summarize in the following paragraphs.

A direct research continuation should be focused on defining other neuron properties, such as to determine the orientation (vertical, horizontal or oblique, among others) of the presented feature. We develop a global orientation selectivity index which only characterize the neuron in terms of how sensitive it is to small changes on rotation of a given stimulus. Therefore, we suggest to evolve our index to concrete the orientation of the intrinsic feature. In addition, we encourage to define other factors that may be involved in the encoded features like frequency selectivity or selectivities to simple features like bars, edges, circles or corners.

From a engineer computer vision point of view, the presented approach can be extended to improve trained architectures. Since the computational cost of training a CNN from scratch is expensive, there is a new trend of fine-tuning trained CNN to be adapted into a new vision problem. A practical alternative may be to figure out which properties directly affect to the performance on the visual task and address the fine tuning just to the subset of neurons that do not present these factors; or even include some restrictions to guide on the learning step on codifying neurons holding such interesting properties.

From a scientific point of view, this thesis recover the open debate of the existence of localist and distributed codes[68] in visual perception. In this sense, we rise the need of adapting understanding techniques to analyze interactions between neurons and expand the analysis beyond individual neuron descriptions. In this line, our similarity index can be used to detect when two similar neurons present different activations on a given stimulus and figure out which feature are they codifying in this situation.

Finally, assuming that these neurons have shown to model biological vision[17, 69, 71] and that they show important representational capabilities compared to the primates vision [17], in further studies could emerge other parallelisms with the HVS that can be interesting for both computer vision and biological areas.

7.3 Scientific Articles

7.3.1 Abstracts

- Ivet Rafegas and Maria Vanrell. **Colour Visual Coding in trained Deep Neural Networks**. *European Conference on Visual Perception (ECVP)*. Barcelona. 2016.

7.3.2 Journals

- Ivet Rafegas, Javier Vazquez-Corral, Robert Benavente, Maria Vanrell and Susana Alvarez. **Enhancing spatio-chromatic representation with more-than-three color coding for image description**. *Journal of the Optical Society of America A (JOSA)*. 34 (5). 827–837. 2017.

- Ivet Rafegas and Maria Vanrell. **Color encoding on biologically-inspired convolutional neural networks**. *Special Issue on Color: cone Opponency and Beyond. Vision Research*. 2017. ***Under review***.

7.3.3 International Conferences and Workshops

- Ivet Rafegas and Maria Vanrell. **Color spaces emerging from deep convolutional networks**. *Color and Imaging Conference (CIC)*. (1). 225-230. San Diego. 2016. ***Award for the best interactive session.***
- Ivet Rafegas and Maria Vanrell. **Color representation in CNNs: parallelisms with biological vision**. *ICCV Workshop on Mutual Benefits of Cognitive and Computer Vision (MBCC)*. Venice. 2017

Bibliography

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] S. Alvarez and M. Vanrell. Texton theory revisited: a bag-of-words approach to combine textons. *Pattern Recognition*, 45(12):4312–4325, 2012.

[3] Susana Alvarez. *Revisión de la teoría de los Textons. Enfoque computacional en color*. PhD thesis, Universitat Autònoma de Barcelona, July 2010.

[4] Sanjeev Arora, Aditya Bhaskara, Rong Ge, and Tengyu Ma. Provable bounds for learning some deep representations. In *ICML*, pages 584–592, 2014.

[5] Mathieu Aubry and Bryan C. Russell. Understanding deep features with computer-generated imagery. In *ICCV*, 2015.

[6] Jonathan T. Barron. Convolutional color constancy. In *IEEE International Conference on Computer Vision*, pages 379–387, 2015.

[7] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Computer Vision and Pattern Recognition*, 2017.

[8] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Comput. Vis. Image Underst.*, 110(3):346–359, June 2008.

[9] Robert Benavente, Maria Vanrell, and Ramon Baldrich. Parametric fuzzy sets for automatic color naming. *JOSA*, 25(10):2582–2593, Oct 2008.

[10] Yoshua Bengio. Learning deep architectures for ai. *Found. Trends Mach. Learn.*, 2(1):1–127, January 2009.

[11] Yoshua Bengio and Yann LeCun. Scaling learning algorithms towards AI. In *Large Scale Kernel Machines*. MIT Press, 2007.

[12] Brent Berlin and Paul Kay. *Basic Color Terms: their Universality and Evolution*. University of California Press, 1969.

[13] E.L. Bienenstock, L.N. Cooper, and P.W. Munro. Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. *Journal of Neuroscience*, 2(1):32–48, 1982.

[14] Y. L. Boureau, N. Le Roux, F. Bach, J. Ponce, and Y. LeCun. Ask the locals: Multi-way local pooling for image recognition. In *2011 International Conference on Computer Vision*, pages 2651–2658, Nov 2011.

[15] Y-Lan Boureau, Jean Ponce, and Yann Lecun. A theoretical analysis of feature pooling in visual recognition. In *27TH INTERNATIONAL CONFERENCE ON MACHINE LEARNING, HAIFA, ISRAEL*, 2010.

[16] M. Brown and S. Süsstrunk. Multispectral SIFT for scene category recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 177–184, 2011.

[17] Charles F Cadieu, Ha Hong, Daniel L K Yamins, Nicolas Pinto, Diego Ardila, Ethan A Solomon, Najib J Majaj, and James J DiCarlo. Deep neural networks rival the representation of primate it cortex for core visual object recognition. *PLoS computational biology*, 10, 2014 Dec 2014.

[18] John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1986.

[19] E. Cernadas, M. Fernández-Delgado, E. González-Rufino, and P. Carrión. Influence of normalization and color space to color texture classification. *Pattern Recognition*, 61:120–138, 2017.

[20] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *BMVC*, 2014.

[21] R. Collobert, K. Kavukcuoglu, and C. Farabet. Torch7: A matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*, 2011.

[22] Bevil R. Conway. Spatial structure of cone inputs to color cells in alert macaque primary visual cortex (v-1). *Journal of Neuroscience*, 21(8):2768–2783, 2001.

[23] Bevil R. Conway. Colour vision: A clue to hue in v2. *Current Biology*, 13(8):R308 – R310, 2003.

[24] Bevil R. Conway, Soumya Chatterjee, Greg D. Field, Gregory D. Horwitz, Elizabeth N. Johnson, Kowa Koida, , and Katherine Mancuso. Advances in color science: from retina to behavior. *The Journal of Neuroscience*, 30(45):14955–14963, 2010.

[25] Bevil R. Conway and Doris Y. Tsao. Color-tuned neurons are spatially clustered according to color preference within alert macaque posterior inferior temporal cortex. *Proc Natl Acad Sci U S A.*, 42(106):18034—18039, 2009.

[26] B.R. Conway, S. Moeller, and D.Y. Tsao. Specialized color modules in macaque extrastriate cortex. *Neuron*, 56(3):560–573, 2007.

[27] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, Sep 1995.

[28] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, CVPR '05, pages 886–893, Washington, DC, USA, 2005. IEEE Computer Society.

[29] Rina Dechter. Learning while searching in constraint-satisfaction-problems. In Tom Kehler, editor, *AAAI*, pages 178–185. Morgan Kaufmann, 1986.

[30] A. M. Derrington, J. Krauskopf, and P. Lennie. Chromatic mechanisms in lateral geniculate nucleus of macaque. *J. Physiol.*, pages 241–265, 1984.

[31] S. Di Zenzo. A note on the gradient of a multi-image. *Computer Vision, Graphics, and Image Processing*, 33(1):116–125, 1986.

[32] James J DiCarlo, Davide Zoccolan, and Nicole C Rust. How does the brain solve visual object recognition? *Neuron*, 73:415–34, 2012.

[33] A. Dosovitskiy, J. T. Springenberg, M. Tatarchenko, and T. Brox. Learning to generate chairs, tables and cars with convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.

[34] Alexey Dosovitskiy and Thomas Brox. Inverting visual representations with convolutional networks. *CoRR*, abs/1506.02753, 2015.

[35] Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network. *Dept. IRO, Université de Montréal, Tech. Rep*, 2009.

[36] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, January 2015.

[37] Giovanni Maria Farinella, Sebastiano Battiato, and Roberto Cipolla. *Advanced Topics in Computer Vision*. Springer Publishing Company, Incorporated, 2013.

[38] Daniel J. Felleman and David C. Van Essen. Distributed hierarchical processing in the primate cerebral cortex. *Cereb Cortex*, pages 1–47, 1991.

[39] G.D. Finlayson, M.S. Drew, and B.V. Funt. Color constancy: Generalized diagonal transforms suffice. *Journal of the Optical Society of America A*, 11:3011–3020, 1994.

[40] G.D. Finlayson, M.S. Drew, and B.V. Funt. Spectral sharpening: sensor transformations for improved color constancy. *Journal of the Optical Society of America A*, 11(5):1553–1563, 1994.

[41] Graham Finlayson and Ruixia Xu. Illuminant and gamma comprehensive normalisation in logRGB space. *Pattern Recognition Letters*, 24(11):1679 – 1690, 2003.

[42] Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193–202, Apr 1980.

[43] Vaitsa Giannouli. Visual symmetry perception. *Encephalos*, 50:31–42, 2013.

[44] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, pages 580–587, Washington, DC, USA, 2014. IEEE Computer Society.

[45] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In Geoffrey Gordon, David Dunson, and Miroslav Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial*

Intelligence and Statistics, volume 15 of *Proceedings of Machine Learning Research*, pages 315–323, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR.

[46] E. González-Rufino, P. Carrión, E. Cernadas, M. Fernández-Delgado, and R. Domínguez-Petit. Exhaustive comparison of colour texture features and classification methods to discriminate cells categories in histological images of fish ovary. *Pattern Recognition*, 46(9):2391–2407, 2013.

[47] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[48] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. 2015.

[49] Lim H., Wang Y., Xiao Y., Hu M., and Felleman DJ. Organization of hue selectivity in macaque v2 thin stripes. *Journal of Neurophysiology*, 102(5):2603–2615, 2008.

[50] Chris Harris and Mike Stephens. A combined corner and edge detector. In *In Proc. of Fourth Alvey Vision Conference*, pages 147–151, 1988.

[51] A.M. Haun and E. Peli. Perceived contrast in complex images. *Journal of Vision*, 13(13):3,1–21, 2013.

[52] Donald O. Hebb. *The organization of behavior: A neuropsychological theory*. Wiley, New York, June 1949.

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

[54] T.S Huang. Computer vision: Evolution and promise. *19th CERN School of Computing. Geneva*, 1996.

[55] D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160:106–154, January 1962.

[56] David H. Hubel and Torsten N. Wiesel. Receptive fields and functional architecture in two nonstriate visual areas (18 and 19) of the cat. *Journal of Neurophysiology*, 28(2):229–289, 1965.

[57] Niall Hurley and Scott Rickard. Comparing measures of sparsity. *IEEE Trans. Inf. Theor.*, 55(10):4723–4741, October 2009.

[58] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In David Blei and Francis Bach, editors, *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 448–456. JMLR Workshop and Conference Proceedings, 2015.

[59] Kevin Jarrett, Koray Kavukcuoglu, Marc'Aurelio Ranzato, and Yann LeCun. What is the best multi-stage architecture for object recognition? In *ICCV*, pages 2146–2153. IEEE, 2009.

[60] 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.

[61] B. Julesz and J.R. Bergen. Textons, the fundamental elements in preattentive vision and perception of textures. *Bell System Technical Journal*, 62(6):1619–1645, 1983.

[62] Andrej Karpathy. Cs231n convolutional neural networks for visual recognition, 2015.

[63] M. Kass and A. Witkin. Analyzing oriented patterns. *Computer Vision, Graphics, and Image Processing*, 37(3):362–385, 1987.

[64] Koray Kavukcuoglu, Pierre Sermanet, Y-Lan Boureau, Karol Gregor, Michaël Mathieu, and Yann LeCun. Learning convolutional feature hierachies for visual recognition. In *Advances in Neural Information Processing Systems (NIPS 2010)*, volume 23, 2010.

[65] N.A. Khanina, E.V. Semeikina, and D.V. Yurin. Color blob and line detection in scale-space. *Pattern Recognition and Image Analysis*, 21(2):267–269, 2011.

[66] N.A. Khanina, E.V. Semeikina, and D.V. Yurin. Scale-space color blob and ridge detection. *Pattern Recognition and Image Analysis*, 22(1):221–227, 2012.

[67] Roozbeh Kiani, Hossein Esteky, Koorosh Mirpour, and Keiji Tanaka. Object category structure in response patterns of neuronal population in monkey inferior temporal cortex. *Journal of Neurophysiology*, 97(6):4296–4309, 2007.

[68] Nicolaus Kriegeskorte and Gabriel Kreiman. *Visual Population Codes - Toward a Common Multivariate Framework for Cell Recording and Functional Imaging*. MIT Press, 2011.

[69] Nikolaus Kriegeskorte. Deep neural networks: A new framework for modeling biological vision and brain information processing. *Annual Review of Vision Science.*, 1:417–446, 2015.

[70] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[71] Norbert Kruger, Peter Janssen, Sinan Kalkan, Markus Lappe, Ales Leonardis, Justus Piater, Antonio J. Rodriguez-Sanchez, and Laurenz Wiskott. Deep hierarchies in the primate visual cortex: What can we learn for computer vision? *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8), August 2013.

[72] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for rcognizing natural scene categories. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2169–2178, 2006.

[73] Yann Lecun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 5 2015.

[74] Yann Lecun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998.

[75] Yann LeCun, Koray Kavukcuoglu, and Clément Farabet. Convolutional networks and applications in vision. In *ISCAS*, pages 253–256. IEEE, 2010.

[76] Chen-Yu Lee, Patrick W. Gallagher, and Zhuowen Tu. Generalizing pooling functions in convolutional neural networks: Mixed, gated, and tree. *CoRR*, abs/1509.08985, 2015.

[77] P. Lennie and M. D'Zmura. Mechanisms of color vision. In *CRC Crit. Rev. Neurobiol.*, chapter 3, pages 333–400. 1988.

[78] P. Lennie, J. Krauskopf, and Sclar G. Chromatic mechanisms in striate cortex of macaque. *The Journal of Neuroscience*, 10:649—-669, 1990.

[79] Ming Li, Fang Liu, Mikko Juusola, and Shiming Tang. Perceptual color map in macaque visual area v4. *The Journal of Neuroscience*, 34(1):202–217, 2014.

[80] Yixuan Li, Jason Yosinski, Jeff Clune, Hod Lipson, and John E. Hopcroft. Convergent learning: Do different neural networks learn the same representations? In *ICLR*, 2016.

[81] T. Lindeberg and Jan-Olof Eklundh. On the computation of a scale-space primal sketch. *Journal of Visual Communication and Image Representation*, 2(1):55–78, 1991.

[82] A. M. Lopez, D. Lloret, J. Serrat, and J. J. Villanueva. Multilocal creaseness on the level-set extrinsic curvature. *Computer Vision and Image Understanding*, 77:111–144, 2000.

[83] A.M. Lopez, F. Lumbreras, J. Serrat, and J.J. Villanueva. Evaluation of methods for ridge and valley detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(4):327–335, 1999.

[84] David G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, ICCV '99, pages 1150–, Washington, DC, USA, 1999. IEEE Computer Society.

[85] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[86] M. R. Luo, G. Cui, and B. Rigg. The development of the cie 2000 colour-difference formula: Ciede2000. *Color Research & Application*, 26(5):340–350, 2001.

[87] Bart Machilsen, Maarten Pauwels, and Johan Wagemans. The role of vertical mirror symmetry in visual shape detection. *Journal of Vision*, 9(12):11, 2009.

[88] T. Mäenpää and M. Pietikäinen. Classification with color and texture: jointly or separately? *Pattern Recognition*, 37(8):1629–1640, 2004.

[89] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. *CVPR*, 2015.

[90] Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, Dec 1943.

[91] Kyle C. McDermott and Michael A. Webster. Uniform color spaces and natural image statistics. *J. Opt. Soc. Am. A*, 29(2):A182–A187, Feb 2012.

[92] George A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, November 1995.

[93] A. Ming and H. Ma. A blob detector in color images. In *6th ACM International Conference on Image and Video Retrieval*, CIVR'07, pages 364–370, 2007.

[94] Grégoire Montavon, Mikio Braun, and Klaus-Robert Müller. Kernel analysis of deep networks. *JMLR*, 12:2563–2581, 2011.

[95] K. Moutoussis and S. Zeki. Responses of spectrally selective cells in macaque area v2 to wavelengths and colors. *Journal of Neurophysiology*, 87:2104–2112, 2002.

[96] J. Mutch and D.G. Lowe. Object class recognition and localization using sparse features with limited receptive fields. *International Journal of Computer Vision*, 80(1):45–57, 2008.

[97] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In Johannes Fürnkranz and Thorsten Joachims, editors, *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814. Omnipress, 2010.

[98] Loris Nanni, Stefano Ghidoni, and Sheryl Brahnam. Handcrafted vs non-handcrafted features for computer vision classification. *Pattern Recognition*, 2017. (Link to article from abstract).

[99] A. Nguyen, J. Yosinski, and J. Clune. Multifaceted Feature Visualization: Uncovering the Different Types of Features Learned By Each Neuron in Deep Neural Networks. In *Workshop on Visualization for Deep Learning, International Conference on Machine Learning (ICML)*, June 2016.

[100] Anh Mai Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 427–436, 2015.

[101] Michael Nielsen. *Neural Networks and Deep Learning*. Determination Press, 2015.

[102] Y. Ohta, Takeo Kanade, and T. Sakai. Color information for region segmentation. 13(3):222 – 241, July 1980.

[103] T. Ojala, M. Pietikainen, and D. Harwood. Performance evaluation of texture measures with classification based on Kullback discrimination of distributions. In *Pattern Recognition, 1994. Vol. 1 - Conference A: Computer Vision amp; Image Processing., Proceedings of the 12th IAPR International Conference on*, volume 1, October 1994.

[104] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001.

[105] I. Omer and M. Werman. Color lines: Image specific color representation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 946–953, 2004.

[106] Seymour Papert. The summer vision project. *MIT AI Memo 100*, 1966.

[107] Laura M. Parkes, Jan-Bernard C. Marsman, David C. Oxley, John Y. Goulermas, and Sophie M. Wuerger. Multivoxel fmri analysis of color tuning in human primary visual cortex. *Journal of Vision*, 9(1):1, 2009.

[108] Arnab Paul and Suresh Venkatasubramanian. Why does deep learning work? - A perspective from group theory. *CoRR*, abs/1412.6621, 2014.

[109] D.I. Perrett and M.W. Oram. Neurophysiology of shape processing. *Image and Vision Computing*, 11(6):317 – 333, 1993.

[110] K.N. Plataniotis and A.N. Venetsanopoulos. *Color Image Processing and Applications*. Springer, 2000.

[111] Maximilian Riesenhuber and Tomaso Poggio. Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 2:1019–1025, 1999.

[112] Anna W. Roe, Leonardo Chelazzi, Charles E. Connor, Bevil R. Conway, Ichiro Fujita, Jack L. Gallant, Haidong Lu, and Wim Vanduffel. Toward a unified theory of visual area V4. *Neuron*, 74(1):12 – 29, 2012.

[113] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Neurocomputing: Foundations of research. chapter Learning Representations by Back-propagating Errors, pages 696–699. MIT Press, Cambridge, MA, USA, 1988.

[114] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition

Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[115] Allah Bux Sargano, Plamen Angelov, and Zulfiqar Habib. A comprehensive review on handcrafted and learning-based action representation approaches for human activity recognition. *Applied Sciences*, 7(1), 2017.

[116] S. J. Schein and R. Desimone. Spectral properties of v4 neurons in the macaque. *VR*, 51(7):701–717, 4 2011.

[117] Denis Schluppeck and Stephen A. Engel. Color opponent neurons in v1: A review and model reconciling results from imaging and single-unit recording. *Journal of Vision*, 2(6):5, 2002.

[118] Thomas Serre, Aude Oliva, and Tomaso Poggio. A feedforward architecture accounts for rapid categorization. *PNAS Proceedings of the National Academy of Sciences*, 104(15):6424–6429, 2007.

[119] Thomas Serre, Lior Wolf, Stanley M. Bileschi, Maximilian Riesenhuber, and Tomaso A. Poggio. Robust object recognition with cortex-like mechanisms. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 29(3):411–426, 2007.

[120] R. M. Shapley and D. J. Tolhurst. Edge detectors in human vision. *The Journal of Physiology*, 229(1):165–183, 1973.

[121] Robert Shapley and Michael J. Hawken. Color in the cortex: Single- and double-opponent cells. *VR*, 51(7):701–717, 4 2011.

[122] Robert Shapley and Michael J. Hawken. Color in the cortex: Single- and double-opponent cells. *Vision Research*, 51(7):701–717, 4 2011.

[123] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *In ICLR Workshop 2014*, 2014.

[124] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. 2015.

[125] Andrew J. R. Simpson. Abstract learning via demodulation in a deep neural network. *CoRR*, abs/1502.04042, 2015.

[126] S.G. Solomon and P. Lennie. The machinery of colour vision. *Nature Review Neuroscience*, 8(4):276–286, 2007.

[127] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin A. Riedmiller. Striving for simplicity: The all convolutional net. 2015.

[128] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin A. Riedmiller. Striving for simplicity: The all convolutional net. *ICLR*, 2015.

[129] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[130] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[131] H. Tanigawa, H.D. Lu, and A.W. Roe. Functional organization for color and orientation in macaque V4. *Nature neuroscience*, 13(12):1542–1548, 2010.

[132] Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, May 2016.

[133] K. van de Sande, T. Gevers, and C. Snoek. Evaluating color descriptors for object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1582–1596, Sept 2010.

[134] Laurens van der Maaten and Geoffrey E. Hinton. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[135] E. Vazquez, R. Baldrich, J. van de Weijer, and M. Vanrell. Describing reflectances for colour segmentation robust to shadows, highlights and textures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):917–930, 2011.

[136] J. Vazquez-Corral and M. Bertalmío. Spectral sharpening of color sensors: Diagonal color constancy and beyond. *Sensors*, 14(3):3965–3985, 2014.

[137] A. Vedaldi and K. Lenc. Matconvnet – convolutional neural networks for matlab. 2015.

[138] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.*, 11:3371–3408, December 2010.

[139] J. Wagemans. Detection of visual symmetries. *Spatial Vision*, 9:9–32, 1995.

[140] G. Wallis and E. Rolls. A model of invariant object recognition in the visual system. *Progress in Neurobiology*, 51:167–194, 1997.

[141] Michael A. Webster, Yoko Mizokami, and Shernaaz M. Webster. Hue maps in primate striate cortex. *NeuroImage*, 35(2):771–786, 2007.

[142] Michael A. Webster, Yoko Mizokami, and Shernaaz M. Webster. Seasonal variations in the color statistics of natural images. *Network: Computation in Neural Systems*, 18 (3):213–233, 2007.

[143] J. Weickert. Coherence-enhancing diffusion of colour images. *Image and Vision Computing*, 17(3-4):201–212, 1999.

[144] P. J. Werbos. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. PhD thesis, Harvard University, 1974.

[145] G. Wyszecki and W.S. Stiles. *Color science: concepts and methods, quantitative data and formulae*. John Wiley & Sons, 2nd edition, 1982.

[146] Y. Xiao and Y. Wangand D.J. Felleman. A spatially organized representation of colour in macaque cortical area v2. *Nature*, 421:535–539, 2003.

[147] Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. Understanding neural networks through deep visualization. In *Deep Learning Workshop, (ICML)*, 2015.

[148] Matthew Zeiler and Robert Fergus. *Stochastic pooling for regularization of deep convolutional neural networks*. 2013.

[149] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014.

[150] Matthew D. Zeiler, Dilip Krishnan, Graham W. Taylor, and Rob Fergus. Deconvolutional networks. In *CVPR*, 2010.

[151] Matthew D. Zeiler, Graham W. Taylor, and Rob Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In *ICCV*, 2011.

[152] Jun Zhang, Youssef Barhomi, and Thomas Serre. A new biologically inspired color image descriptor. In *12th European Conference on Computer Vision (ECCV'12)*, pages 312–324, 2012.