Recognition of Mexican cultural content with Deep learning networks

Abraham Montoya-Obeso

DOCTORAL THESIS

Recognition of Mexican Cultural Content with Deep Learning Networks

A thesis submitted in fulfillment of the requirements
for the double degree of Doctor of Science
in Digital Sistems by the Instituto Politécnico Nacional - CITEDI and
in Informatics by the Université de Bordeaux - LaBRI

By **Abraham Montoya Obeso**

Supervised by
Dr. Mireya Saraí García Vázquez and Dr. Jenny Benois

Dissertation date: July 9th 2020

Committee members:

Dr. Mireya Saraí García Vázquez	Professor, Instituto Politécnico Nacional	Director
Dr. Jenny Benois-Pineau	Professor, Université de Bordeaux	Co-director
Dr. Luis Tupak Aguilar Bustos	Professor, Instituto Politécnico Nacional	President
Dr. Valerie Gouet-Brunet	Research director, Institut Géographique National	Examinator
Dr. Alejandro Álvaro Ramírez Acosta	Research director, MIRAL R&D&I	Invited
Dr. Moisés Sánchez Adame	Professor, Instituto Politécnico Nacional	Invited
Dr. Alexandre Benoît	Professor, Université Savoie Mont Blanc	Reviewer
Dr. Perla Olivia Rodríguez Reséndiz	Professor, Universidad Nacional Autónoma de México	Reviewer

– 2020 –

SIP-14
REP 2017

INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de Tijuana, B.C. siendo las 19:00 horas del día 29 del mes de abril del 2020 se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Posgrado de: CITEDI para examinar la tesis titulada:

Recognition of Mexican Cultural Content with Deep Learning Networks del alumno:

Apellido Paterno:	Montoya	Apellido Materno:	Obeso	Nombre (s):	Abraham

Número de registro: A 1 7 0 9 4 0

Aspirante del Programa Académico de Posgrado: Doctorado en Ciencias en Sistemas Digitales

Una vez que se realizó un análisis de similitud de texto, utilizando el software antiplagio, se encontró que el trabajo de tesis tiene **9%** de similitud. **Se adjunta reporte de software utilizado**.

Después que esta Comisión revisó exhaustivamente el contenido, estructura, intención y ubicación de los textos de la tesis identificados como coincidentes con otros documentos, concluyó que en el presente trabajo **SI** [ ] **NO** [x] **SE CONSTITUYE UN POSIBLE PLAGIO.**

JUSTIFICACIÓN DE LA CONCLUSIÓN: *(Por ejemplo, el % de similitud se localiza en metodologías adecuadamente referidas a fuente original)*

El porcentaje de similitud 9% se localiza principalmente en los encabezados de las tablas, figuras y ecuaciones que forman parte de los artículos que se generaron en este trabajo de tesis, y donde el estudiante forma parte de los autores; y que en el texto se menciona y hace referencia al artículo indicado. Este porcentaje también se localiza en algunos fragmentos cortos de algunos artículos que se generaron en este trabajo de tesis, y de la misma forma en el texto se indica de cual artículo del estudiante se trata. Es importante mencionar que la mayoría de las imágenes, figuras y esquemas son específicamente realizadas para este reporte de tesis; algunas imágenes que no pertenecen al autor del trabajo de tesis, se menciona la fuente y algunas se crearon bajo la inspiración de la original.

****Es responsabilidad del alumno como autor de la tesis la verificación antiplagio, y del Director o Directores de tesis el análisis del % de similitud para establecer el riesgo o la existencia de un posible plagio.**

Finalmente y posterior a la lectura, revisión individual, así como el análisis e intercambio de opiniones, los miembros de la Comisión manifestaron **APROBAR** [x] **SUSPENDER** [ ] **NO APROBAR** [ ] la tesis por **UNANIMIDAD** [x] o **MAYORÍA** [ ] en virtud de los motivos siguientes:

El trabajo de tesis es original y cumple con los requisitos de un trabajo de investigación de nivel de doctorado. Asimismo, cuenta con los productos de investigación que validan la calidad del trabajo en comento.

COMISIÓN REVISORA DE TESIS

Director de Tesis Dra. Mireya Saraí García Vázquez	Dr. Luis Tupak Aguilar Bustos	Dr. Alejandro Álvaro Ramírez Acosta
2° Director de Tesis Dra. Jenny Benois-Pineau	Dr. Moisés Sánchez Adame	Dr. Julio César Rolón Garrido **PRESIDENTE DEL COLEGIO DE PROFESORES**

S.E.P.
INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACIÓN Y DESARROLLO DE TECNOLOGÍA DIGITAL
DIRECCIÓN

INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA CESIÓN DE DERECHOS

En la Ciudad de Tijuana, Baja California. el día 6 del mes de mayo del año 2020, el que suscribe Abraham Montoya Obeso, alumno del Programa de DOCTORADO EN CIENCIAS EN SISTEMAS DIGITALES, con número de registro A170940, adscrito al CENTRO DE INVESTIGACIÓN Y DESARROLLO DE TECNOLOGÍA DIGITAL, manifiesta que es el autor intelectual del presente trabajo de Tesis bajo la dirección de la Dra. Mireya Saraí García Vázquez y de la co-dirección de la Dra. Jenny Benois-Pineau y cede los derechos del trabajo titulado *Recognition of Mexican Cultural Content with Deep Learning Networks*, al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o directores del trabajo. Este puede ser obtenido escribiendo a las siguientes direcciones Av. Instituto Politécnico Nacional No. 1310 Col Nueva Tijuana, Tijuana, Baja California, México, correo electrónico de contacto: posgrado@citedi.mx. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

Abraham Montoya O.

Abraham Montoya Obeso

Abstract

Recognition of Mexican Cultural Content with Deep Learning Networks

by Abraham Montoya Obeso

In Mexico, one of the priority technological problems is the preservation of cultural heritage in its digital form. In this research, the main interest is the ordering, management and identification of intangible cultural heritage in images.

In computer vision, the integration of the Human Visual System (HVS) into automatic learning methods and classifiers has become an intensive research field for object recognition and content mining. The so-called saliency maps, are defined as a topographic representation of visual attention on a scene, modeling attention instantaneously and assigning a degree of interest to each pixel value on the image. Saliency maps proved to be very efficient to point out regions of interest in several tasks of visual content and its understanding. In this context, we focus on the integration of visual attention models in the training pipeline of Deep Neural Networks (DNNs) for the recognition of Mexican architectural structures.

We consider the main contributions of this research are in the following areas of interest:

- Specific purpose dataset: gathering data related to the topic is a key task to solve the problem of architectural classification.
- Data selection: we use saliency prediction methods to select and crop context-relevant regions on images.
- Visual attention modeling: we annotate images through a real task of image observation, we record eye-fixations with an eye-tracker system to build subjective saliency maps.
- Visual attention integration: we integrate visual attention in deep neural networks in two ways; i) to filter out features in a saliency-based pooling layer and ii) in attention mechanisms.

In this research, different essential components for the training of a neural network are tackled down with the aim of recognizing Mexican cultural content and extrapolating these findings to large-scale databases in similar classification tasks, such as in ImageNet. Finally, we show that the integration of visual attention models generated through a psycho-visual experiment allows to reduce training time and improve performances in terms of accuracy.

Keywords: Deep Learning, Visual Attention, Cultural Heritage.

Résumé

Reconnaissance du patrimoine Mexicaine sous forme numérique par des réseaux d'apprentissage profond

par Abraham Montoya Obeso

Au Mexique, l'une des questions technologiques prioritaires est la préservation des contenus culturels immatériels, c'est-à-dire sous leur forme numérique. Dans cette recherche, l'intérêt principal est la génération d'outils numériques pour l'identification et la gestion du patrimoine culturel immatériel en images.

Les œuvres architecturales sont considérées comme faisant partie de l'héritage culturel des générations futures et comme une source d'inspiration pour les architectes, les designers et les ingénieurs. Par conséquent, les systèmes de reconnaissance automatique des contenus culturels sont importants dans les tâches de préservation, de recherche et de diffusion. Toutefois, ces tâches impliquent différents défis découlant d'erreurs dans l'attribution des métadonnées, de la désorganisation de l'information, du stockage dans différents formats et supports, entre autres. En outre, la tâche de reconnaissance elle-même représente un défi supplémentaire lorsque différentes techniques de traitement peuvent être utilisées et que celles-ci doivent pouvoir fonctionner en cas de changement d'éclairage, d'occlusion ou de changement de perspective dans les scènes. Idéalement, les informations devraient être normalisées et contenues de manière à fournir des modèles qui permettent de reconnaître et de différencier les nouvelles informations en vue de leur utilisation.

Dans le domaine de la vision par ordinateur, l'intégration des modèles d'attention (basés sur le système visuel humain) est devenue un domaine de recherche intense, principalement dans les tâches de reconnaissance et d'extraction de contenu visuel. Les cartes dites de proéminence sont définies comme des cartes topographiques qui représentent l'attention visuelle dans une scène, modélisant instantanément l'attention en attribuant un degré d'intérêt à chaque pixel de l'image. Récemment, les cartes de proéminence se sont avérées être une source efficace pour distinguer les régions d'intérêt dans différentes tâches de reconnaissance visuelle de contenu. En outre, les modèles d'apprentissage automatique, dans le domaine de l'apprentissage approfondi, ont montré des résultats exceptionnels dans les tâches de reconnaissance dans les bases de données à grande échelle. Dans ce contexte, l'intérêt principal est d'intégrer des modèles d'attention visuelle dans le schéma d'apprentissage des réseaux neuronaux profonds, spécifiquement pour la reconnaissance des structures architecturales, comme faisant partie du contenu culturel immatériel du Mexique.

Les principales contributions de cette recherche sont concentrées dans : i) la génération de bases de données à usage spécifique, ii) la sélection automatique de contenu pour la formation de réseaux neuronaux, iii) l'annotation d'images par une expérience psycho-visuelle pour la génération de carte de saillance et

iv) l'intégration de l'attention visuelle dans des réseaux d'apprentissage profond (dans des couches spécifiques et des mécanismes d'attention visuelle). Lors de la création de la base de données, une liste des architectures les plus représentatives du pays est proposée. Plus tard, grâce à un programme de collaboration, les images disponibles de chacune des architectures ont été récupérées. Bien que la liste considère 372 structures architecturales, classées en 3 styles architecturaux, la base de données finale comprend 8 155 images dans seulement 142 catégories. Une autre contribution importante de cette recherche est l'annotation semi-automatique d'un groupe d'images à l'aide d'un eye tracker, permettant la génération de cartes d'attention visuelle (cartes de proéminence) pour un groupe relativement petit d'images dans la base de données (284 images) et leur propagation à des images similaires en estimant les changements de perspective entre elles.

En ce qui concerne l'intégration des modèles d'attention dans les réseaux neuronaux profonds, deux aspects sont présentés. Le premier aspect, une couche personnalisée de mise en commun est intégrée qui permet de sélectionner les caractéristiques en fonction des cartes de proéminence fournies avec chacune des images d'entraînement. En utilisant ces couches, il est possible d'augmenter la précision et d'accélérer la convergence des modèles. En deuxième couche, les modèles d'attention visuelle sont intégrés en tant que mécanismes d'attention visuelle basés sur des cartes de proéminence dans des réseaux neuronaux résiduels et la performance est comparée aux mécanismes d'attention présents dans la littérature. Enfin, les modèles équipés de couches et de mécanismes d'attention (équipés de modèles d'attention visuelle) présentent des résultats exceptionnels dans la tâche de reconnaissance des contenus culturels.

Le document est organisé comme décrit ci-dessous. Dans le chapitre 2, les concepts généraux pour la compréhension des réseaux neuronaux artificiels et du système visuel humain sont présentés. Ensuite, dans le chapitre 3, la première contribution, le processus de génération de la base de données, est décrite. Dans le chapitre 4, les expériences pour la formation utilisant les méthodes de sélection de contenu sont introduites. Ensuite, dans les chapitres 5 et 6, l'intégration des modèles d'attention dans la couche de mise en commun et les mécanismes d'attention, respectivement, sont présentés. Enfin, dans le chapitre 7, cette recherche est conclue et les principales contributions sont résumées.

Mots clés: Apprentissage Profond, Attention Visuelle, Patrimoine Culturel.

Unités de recherche

Instituto Politécnico Nacional
Centro de Investigación y Desarrollo de Tecnología Digital (CITEDI)
1310 Ave. Instituto Politécnico Nacional, 22435, Tijuana, Baja California, México

Université de Bordeaux
Laboratoire Bordelais de Recherche en Informatique (LaBRI)
351 Cours de la Libération, F-33405, Talence CEDEX, Bordeaux, France

Resumen

Reconocimiento de contenido cultural Mexicano mediante redes de aprendizaje profundo

por Abraham Montoya Obeso

En México, uno de los problemas prioritarios tecnológicos corresponde a la preservación del contenido cultural intangible, es decir, en su forma digital. En esta investigación, el principal interés es la generación de herramientas digitales para la identificación y el manejo del patrimonio cultural intangible en imágenes.

En México, las obras arquitectónicas son consideradas parte del legado cultural para las futuras generaciones y una fuente de inspiración para arquitectos, diseñadores e ingenieros. Por lo tanto, los sistemas de reconocimiento automático de contenido cultural son importantes en tareas de preservación, búsqueda y diseminación. Sin embargo, estas tareas acarrean diferentes desafíos derivados de errores en la asignación de metadatos, desorganización de la información, almacenamiento en diferentes formatos y soportes, entre otros. Además, la propia tarea de reconocimiento representa un desafío adicional en donde diferentes técnicas de procesamiento pueden ser utilizadas y estas deben ser capaces de funcionar cuando se presentan cambios de iluminación, oclusiones o cambios de perspectiva en las escenas. En un esquema ideal, la información debe estar estandarizada y contenida de forma que ésta proporcione patrones que permitan reconocer y diferenciar nueva información para su uso.

En el área de visión por computador, la integración de modelos de atención (basados en el sistema visual humano) se ha convertido en un área de intensa investigación, principalmente en tareas de reconocimiento y minado de contenido visual. Los denominados mapas de prominencias, se definen como mapas topográficos que representan la atención visual en una escena, modelando instantáneamente la atención, asignando un grado de interés a cada píxel de la imágen. Recientemente, los mapas de prominencias han probado ser una fuente eficiente para distinguir regiones de interés en diferentes tareas de reconocimiento de contenido visual. Adicionalmente, los modelos de aprendizaje automático, en el área del aprendizaje profundo, han presentado resultados sobresalientes en tareas de reconocimiento en bases de datos de alta escala. En este contexto, el interés principal es integrar modelos de atención visual en el esquema de aprendizaje de las redes neuronales profundas, específicamente para el reconocimiento de estructuras arquitectónicas, como parte del contenido cultural intangible de México.

Las principales contribuciones de esta investigación se derivan de:

- la generación de bases de datos de propósito específico,
- la selección automática de contenido para el entrenamiento de las redes neuronales,

- la anotación de imágenes mediante un experimento psico-visual para le generación de mapas de prominencias y,

- la integración de la atención visual en las redes de aprendizaje profundo en capas específicas y como mecanismos de atención visual.

Durante el proceso de generación de la base de datos, se propone una lista de las arquitecturas más representativas a lo largo del territorio Mexicano. Posteriormente, mediante un esquema colaborativo se recuperaron las imágenes disponibles de cada una de las arquitecturas. Aunque en la lista se consideran 372 estructuras arquitectónicas, clasificadas en 3 estilos arquitectónicos, la base de datos final consta de 8,155 imágenes en solo 142 categorías. Otra de las aportaciones importantes en esta investigación, es la anotación semiautomática de un grupo de imágenes utilizando un seguidor ocular, permitiendo generar mapas de atención visual (mapas de prominencias) para un grupo relativamente pequeño de imágenes de la base de datos (284 imágenes) y su propagación a imágenes similares mediante la estimación de cambios de perspectiva entre ellas.

Con respecto a la integración de los modelos de atención en redes neuronales profundas, se presentan dos vertientes. La primera vertiente, se integra una capa personalizada de pooling que permite seleccionar características en función de los mapas de prominencias suministrados con cada una de las imágenes de entrenamiento. Al utilizar estas capas, es posible incrementar la precisión y acelerar la convergencia de los modelos. Como segunda vertiente, se integran los modelos de atención visual como mecanismos de atención visual basados en mapas de prominencias en redes neuronales residuales y se compara el desempeño con respecto a los mecanismos de atención presentes en la literatura. Finalmente, los modelos equipados con capas y mecanismos de atención (equipados con modelos de atención visual) presentan resultados sobresalientes en la tarea de reconocimiento de contenido cultural.

El documento se organiza como se describe a continuación. En el capítulo 2, se presentan los conceptos generales para el entendimiento de las redes neuronales artificiales y el sistema visual humano. Después, en el capítulo 3, se describe la primera contribución, el proceso para la generación de la base de datos. En el capítulo 4, se introducen los experimentos para el entrenamiento utilizando los métodos de selección de contenido. Posteriormente, en el capítulo 5 y 6, se presenta la integración de los modelos de atención en la capa de pooling y los mecanismos de atención, respectivamente. Finalmente, en el capítulo 7, se concluye esta investigación y se resumen las principales contribuciones.

Palabras clave: Aprendizaje Profundo, Atención Visual, Herencia Cultural.

Unidades de investigación

Instituto Politécnico Nacional
Centro de Investigación y Desarrollo de Tecnología Digital (CITEDI)
1310 Ave. Instituto Politécnico Nacional, 22435, Tijuana, Baja California, México

Université de Bordeaux
Laboratoire Bordelais de Recherche en Informatique (LaBRI)
351 Cours de la Libération, F-33405, Talence CEDEX, Bordeaux, France

Acknowledgements

First, I would like to thank my supervisors Dr. Mireya Saraí García Vázquez and Dr. Jenny Benois-Pineau for allowing me to work on this interesting subject, but mostly for guiding me throughout with their scientific rigor and numerous advice. I appreciate the freedom given to me during these last years to explore different research topics and their high availability in spite of the distance.

I am truly grateful for the supervision that Dr. Alejandro Ramírez has given me over the years. For all the observations on the different works that we developed but especially for always being available and attentive to help. Thanks Doc!

Thanks to the members of my committee, who were always available and attentive. Also, thanks to my external reviewers, Dr. Alexandre Benoît and Dr. Perla Olivia Rodríguez Reséndiz for reviewing this manuscript in such complicated times.

Thanks to the Instituto Politécnico Nacional and to the Université de Bordeaux for hosting me, and to the Consejo Nacional de Ciencia y Tecnología for the doctoral grant and to the Eiffel Scholarship of Excellence program for financing part of my research mobility.

To my friends, Enrique, Octavio and Yoshio, members of the prestigious Underwater & Co. group, for the support, laughs and recommendations on any topic.

To my friends, co-workers and people who participated in dataset annotation, document reviews and data gathering: Laura, Mijes, Mario, Maldonado, Manuel, Pepe, Osvaldo, Beth, Espejo, P-e, Karim, Tina and Ricardo.

Finally, I would specially thank to Fernanda Lencioni, my most faithful adventure partner who has driven me without hesitation during these last years. For always pushing me to be better, to give everything from myself and for always trusting me.

–Abraham

For my parents ***Blanca Julia*** *and* ***Humberto****,*
for my brother ***Hernán****,*
for my little brother ***Adrián****.*

Contents

List of Figures

List of Tables

List of Abbreviations

AI Artificial Intelligence. 6

ANN Artificial Neural Network. 8–10

BoVW Bag of Visual Words. 60

CBIR Content-Based Image Retrieval. 60

CNN Convolutional Neural Network. 4, 15, 19, 21, 45–47, 51, 57, 60, 61, 66, 67, 76, 81, 84, 85, 93, 97, 101–103

COSAL Cosaliency Maps. 61, 67, 73, 76, 77, 80, 101

CWDF Cummulative Weibull Distribution Function. 62, 63

DL Deep Learning. 1–3, 11, 21, 35, 36, 46, 60, 64

DNN Deep Neural Network. 1, 4, 5, 10, 22, 45, 47, 59, 60, 93

FCN Fully Convolutional Network. 60

FIT Feature Integration Theory. 25, 31, 35, 60

FoA Focus of Attention. 32, 34

GBVS Graph-Based Visual Saliency. 34, 49, 60, 61, 67–69, 75, 76, 102, 103

GC Geometrical Cropping. 48, 50, 54–58

GPU Graphics Processing Unit. 97

GT Ground Truth. 13, 29, 48, 49, 58, 117

HSV Hue, Saturation and Value. 48

HVS Human Visual System. 4, 5, 25, 28, 32, 49

ILSVRC ImageNet Large Scale Visual Recognition Competition. 21–23, 93

LReLU Leaky Rectified Linear Unit. 12

LSD Line Segment Detector. 69

MCBD MexCulture Buildings Dataset. 41, 45, 91

MCG Multiscale Combinatorial Grouping. 45, 46, 48–50, 53, 54, 58

MD Multi Directions. 69, 70

ML Machine Learning. 7, 15, 46

MLP Multi-Layer Perceptron. 7–9, 12, 13, 15, 16

MSE Mean Square Error. 12, 13

NAG Nesterov's Accelerated Gradient. 52

ODD One Dominant Direction. 69, 70

PCA Principal Component Analysis. 93, 95–97

R-CNN Regions with Convolutional Neural Networks. 46

ReLU Rectified Linear Unit. 11, 12, 14, 21, 87

RGB Red, Green and Blue color model. 48, 51, 75

ROI Region of Interest. 48, 49, 51, 53, 54, 57

SA Saliency Attention. 89

SB Saliency-Based. 49, 50, 54–58

SGD Stochastic Gradient Descent. 52, 79, 93

SIFT Scale Invariant Features Transform. 71, 72

SMCBD Saliency MexCulure Buildings Dataset. 70, 91, 96

SMUIC Saliency Maps for Urban Image Contents. 61, 67, 68, 70, 75, 76, 80, 103

SS Selective Search. 45, 46, 48–50, 53, 54, 58

SVM Support Vector Machine. 35, 46

TDD Two Dominant Directions. 69, 70

TRECVID TREC Video Retrieval Evaluation. 40

VGG Visual Geometry Group. 22

WTA Winner-Take-All. 25, 32, 34

Chapter 1

Introduction

Preservation of cultural heritage in digital has become one of the priority problems of Mexican scientific policy. The automatic identification, management and access to the heritage in form of digital cultural archives is an actual social problem and a large governmental support is granted to this axis. From France, the National Agence of Research (in french Agence Nationale de Recherche (ANR)) and the Mexican National Council of Science and Technology (in spanish, Consejo Nacional de Ciencia y Tecnología (CONACYT)) financed the Blanc International Mex-Culture project, whose objectives were the development of methodological digital tools for the preservation, access and extraction of information on cultural archives in digital form; by focusing research on new methods for digital media such as sound, image and videos as source information.

Cultural heritage content projects have been proposed in recent years around the world by the research community, with the aim of preserving intangible heritage and facilitating its dissemination (e.g., Europeana, 2008; Tagcloud, 2013). The Europeana project is still in development, the strategy for the period 2015-2020 is to provide services for Cultural Heritage Institutions to share high-quality content with partners and end-users, as well to people in general. The platform is in constant evolution in order to change as the technology changes, constantly improving services, breaking linguistic barriers, finding better browsing solutions and dealing with content and media, such as newspapers and video-streaming. Presented models can potentially bring benefits in different technological and social areas. Most of these systems are made up of different stages where it is required to optimize or propose effective processing schemes to perform a recognition task or a virtual reconstruction task (Presious, 2013; I-TREASURES, 2013). In the social context, these models could benefit different institutions that possess Mexican cultural information by increasing the possibility of retrieving similar information given a reference, in a digital collection. Deep Neural Networks (DNNs), considered within the area of Deep Learning (DL), are able to automatically find which general characteristics of

the input data are relevant for subsequent recognition in new unseen data.

In the Mexican context, architecture is considered as a legacy for future generations and a source of inspiration for architects, designers and engineers. Then, automatic recognition tools of cultural content are very important for the preservation, search and dissemination of cultural content. However, the automatic recognition of this content brings challenges that are derived in different ways, the reference information is sparse in different collections, stored as non-standardized formats, incorrect labeling or they are stored in different media supports. Besides, architectural recognition, as in object recognition or image classification tasks also involves managing several image processing techniques, dealing with changes in the scene context such as illumination, perspective changes and occlusions. Then, in an ideal scheme we need to standardize and concentrate the information that automatically allows recognizing in new collections these cultural elements for everyday use in different institutions around the country.

In the DL community, different classification schemes have been proposed. One of the most significant proposals is the AlexNet architecture (Krizhevsky et al., 2012). Later, different architectures were adopted, e.g., VGGNet (Simonyan and Zisserman, 2014), GoogLeNet (Szegedy et al., 2015), ResNet (He et al., 2016), among other variants. Each of these architectures presented better results in the well-known ImageNet database (Russakovsky et al., 2015). However, in recent years, the community is focused on integrating bioinspired methods, mainly attention mechanisms, often called self-attention. Self-attention in neural networks, plays a very important role in the analysis of temporary extracted features and in the determination of global relevance, considering a specific classification task. In this research, we are motivated by previous proposals and we are interested on how the integration of real attention can improve neural networks performance. Hence, we consider the main contributions of this work are classified into different areas of interest:

- Specific purpose dataset: gathering data related to the topic is a key task to solve the problem of architectural classification.
- Data selection: we use saliency prediction methods to select and crop context-relevant regions in images.
- Visual attention modeling: we annotate images through a real task of image observation, we record eye-fixations with an eye-tracker system to build subjective saliency maps.
- Visual attention integration: we integrate visual attention in deep neural networks to filter out features during the training process.

This research topic had not been studied in depth at the beginning of this research. However, during the last three years, multiple works related to attention mechanisms in multiple areas –to solve different problems– have been presented; such as in natural image processing (Vaswani et al., 2017), image classification (Chen et al., 2018), video captioning (Zhu and Jiang, 2019), action recognition (Wang et al., 2018), among others. This indicates that the topic is relevant to the community and that derived applications can impact many sectors of interest.

1.1 Thesis work

This thesis work is the result of a cooperation between two research laboratories, CITEDI (Center for Research and Development of Digital Technology) of the National Polytechnic Institute in Mexico and LaBRI (Bordeaux Laboratory for Research in Computer Science) of the University of Bordeaux in France. The purpose is to develop and implement new digital techniques for the cataloging of Mexican cultural content trough deep learning models.

The approaches covered in this research are: i) specific purpose dataset creation, ii) training data selection methods, iii) visual attention modeling and iv) visual attention integration in deep neural networks. The creation of specific purpose databases has been mandatory for problem solving in computer vision, specifically for machine learning models, where learning takes place through the analysis of patterns from training examples. Therefore, the MexCulture buildings database (Montoya Obeso et al., 2016a) was conceived with the objective of providing a reference of the main characteristics that better describe Mexican architecture. In this database, the most relevant architectures of the country are considered, each architectural structure was strategically chosen to cover the Mexican territory.

In this scenario, dealing with size limited datasets for models training, we found that taking the most relevant information about the object of interest and taking a part of the context where the object is contained, we can train slightly better deep models. Traditionally, object identification has been addressed in DL by taking the whole image as input or various crops, normalizing thus the size of input data. It is our belief that the optimization process can be improved by the quality of relevant data we provide for model training.

The integration of visual attention into deep learning models is the last contribution of this work. First, we model visual attention through record data during a psycho-visual experiment using an eye-tracking system, where the main task of participants was to observe images while we register the coordinates where they are looking at in an experimental screen: i.e., the most attractive regions in the image. In this way, based on D. S. Wooding method (Wooding, 2002), we generate subjective saliency maps upon gaze fixations. Subsequently, this information is

injected as input to the DNN (as support information for raw image data) with the aim of filtering out relevant characteristics for deeper layers in the network.

Visual attention integration was tackled in two steps. First, we consider feeding the Convolutional Neural Networks (CNNs) with three types of pre-computed saliency maps, Harel's maps (Harel et al., 2007), Guissous model (Guissous and Gouet-Brunet, 2017) and our subjective saliency maps. Saliency maps are explicitly integrated into CNNs in a saliency-based processing layer. Then, we compare our saliency-based layer against attention mechanisms with the aim of validating that the modeled visual attention partially represents human behavior and that it improves trained neural models performances when compared to automatic attention models.

1.2 Thesis outline

The document is organized as follows. In chapter 2, we review the main concepts of artificial neural networks and we briefly describe the state-of-the-art on deep neural networks for image classification, then, we provide a review of the main concepts about the Human Visual System (HVS) to understand the following chapters on visual attention. Then, chapter 3 brings our first contribution regarding the cultural dataset; for architectural style and specific buildings identification. Introducing thus, a very brief review of predominant architectural styles in Mexico; prehispanic, colonial and modern architectural styles. After, in chapter 4, we describe the experiments results of training CNNs given different methods of data selection. We review the integration of visual saliency models into DNNs in chapter 5 and we compare a saliency-based attention mechanism against other automatic attention mechanisms in chapter 6. Finally, chapter 7 summarizes the conclusions and the main contributions of this thesis.

Chapter 2

Background

The research on automatic classification brought many benefits to the industry, in health applications and to everyday life of people. Automatic learning algorithms help with the detection of similar patterns in data on the basis of seen data. On one hand, DNNs, which are one of the research interests in this thesis, have the ability of identifying abstract patterns contained in a training dataset by optimizing internal parameters in order to minimize an error given a metric. Then, these models can predict categories with some level of uncertainty of unseen data. They are now able to identify from human faces to dog breeds. On the other hand, the HVS, has been also widely explored in the last years and its integration in solving real problems of computer vision is still a topic of research.

In this chapter, in the first two sections, we present the main concepts to understand artificial neural networks and deep neural networks on image classification tasks with the main goal of clarifying the most common architectures. Then, we analyze the main concepts regarding the HVS, how it is conformed and the key concepts to understand the current attention models in computer vision.

2.1 Artificial neural networks

When the concept of a computer was conceived and the first computer algorithm was published, the scientific community thought that machines would quickly become intelligent (Menabrea and Lovelace, 1842). However, it has taken more than a hundred years for them to embark on the field of Artificial Intelligence (AI) as we know it today. It is clear that the basic concepts that brought to life the artificial neuron date back to the late 1940's. Accordingly to Hebb's theory of cell assembly (Hebb, 1949), neurons are interconnected and the synapses are weighted when the output of nodes are connected to the input of others. A few years later, Rosenblatt (Rosenblatt, 1958) inspired by Hebb's research, specifically for the mathematical modeling of the synaptic connections of neurons, concluded on the definition of what is now known as the first formal model of an artificial neuron.

The basic neuron is known as the Perceptron. When this unit receives an input signal, it can respond or not, the response depends on the activation function, i.e., its threshold. The input of the neuron is considered as $\mathbf{x} = (x_1, x_2, ..., x_n)$ and a function $f(\cdot)$ is applied as a linear combination with a set of weights $\mathbf{w} = (w_1, w_2, ..., w_n)$ and a bias term b_0. Then, formally, the weighted sum of each element of the input x_i and its corresponding weight w_i is,

$$z = \sum_{i=1}^{n} (w_i \cdot x_i) + b_0. \tag{2.1}$$

Note that this is a simply dot operation between the input $\mathbf{x}$ and the weights $\mathbf{w}$. Then, z, as the result of the affine transformation, is passed through a threshold function $f(z)$,

$$y = \begin{cases} 1, & \text{if } z \geqslant 0, \\ 0, & \text{otherwise}, \end{cases} \tag{2.2}$$

where this condition determines the binary behavior of the Perceptron. This first model allowed to identify if the input vector corresponds to a class or not, a binary classification. However, it is possible to expand its ability to handle more categories, by simply adding more units and associating each of them to a different category. An example of the artificial neuron is presented in Fig. 2.1

Rosenblatt, undoubtedly provided a first model which was capable of classifying patterns given a configuration of weights that exhibited the desired behavior. However, the algorithm to update the weights could be one of his most important contributions. This algorithm, given a pair $(\mathbf{x}, y)$, is capable of estimating an adjustment of the weights – often called parameters of the network – according to the following rule,

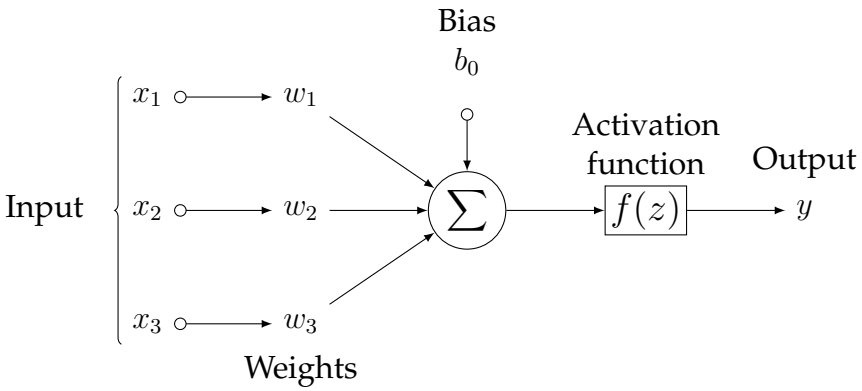


Figure 2.1: The formal neuron.

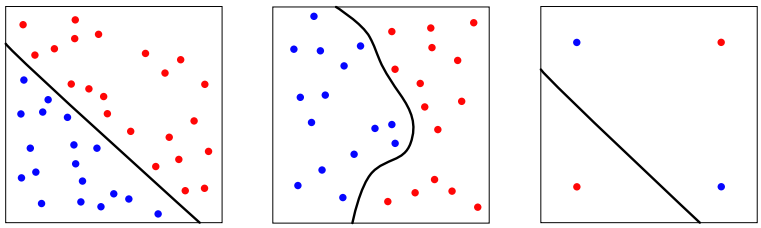


Figure 2.2: (a) Linearly separable problem, (b) nonlinearly separable problem and (c) the XOR problem with a failed solution to separate points.

$$w_i^n = w_i^o - \eta \cdot (\hat{y} - y) \cdot x_i \tag{2.3}$$

where w_i^o denotes the old weights of the network, w_i^n the new updated weights, $\hat{y}$ the output of the Perceptron, y the target value and finally η is the magnitude of how weights are modified, i.e., a learning factor.

The introduction of the Perceptron in the scientific community to solve tasks automatically by obtaining experience from data, attracted attention and was welcomed with excitement as the beginning of a new era on neural networks approaches. This lasted until Minsky and Papert published a detailed analysis of the limitations of the Perceptron in solving simple tasks, such as the XOR operation (see Fig. 2.2 c). The main limitation of Perceptrons is that they are only able to solve linearly separable problems, such as the one illustrated in Fig. 2.2 a. This study gave birth to the Multi-Layer Perceptron (MLP) we review in next section in this chapter.

When we compare the early models in the early days of Machine Learning (ML) with the current models, it is obvious that there is an incredibly large gap in performances. Given that, in the late 50's, when Rosenblatt introduced the MLP, the available computational power was very limited, so the number of problems to be solved was limited as well and not to mention the insufficient amount of data that

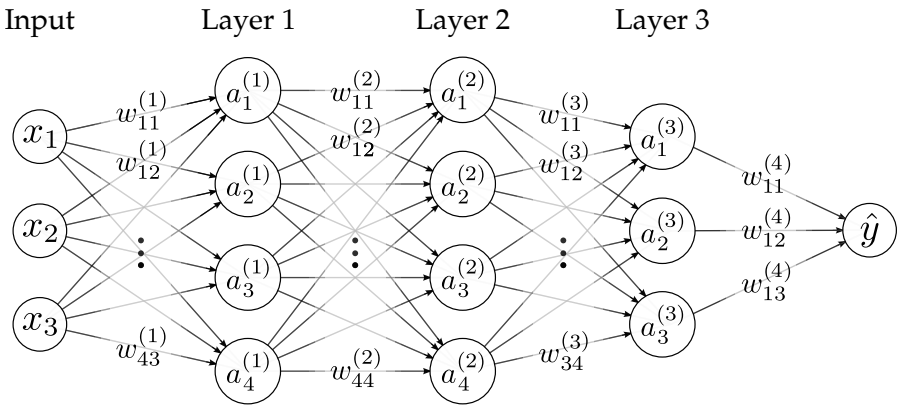


Figure 2.3: The Multilayer Perceptron, a Artificial Neural Network (ANN). Input data is only fed into the fist layer. Each node in the hidden layer reacts based on its activation function. The output y is given the linear combination of hidden neurons response and the weights of the output layer. The bias term have been omitted for clarity.

had been collected at that time aiming to solve more complex tasks.

Nowadays, the picture is completely different and new challenges had been to be faced. With greater computing capacity, we have chosen to explore different architectures, each time requiring more trainable parameters and more layers, i.e., deeper architectures. In addition, different architectures have been proposed in recent years to solve complex tasks, where the arrangement of artificial neurons in the form of a MLP with several hidden layers is not enough to effectively solve different types of tasks, such as segmentation, natural language processing or detection of multiple objects in images.

2.1.1 The Multi-Layer Perceptron

Consider the Fig. 2.3, the MLP is equipped with multiple hidden layers, where each hidden unit is connected to all other units in both; next (l) and previous layer $(l-1)$. Each connection at layer l receives as input previous activations as $\mathbf{a}^{(l-1)}$ and are associated to a weight $w_{ij}^{(l)}$.

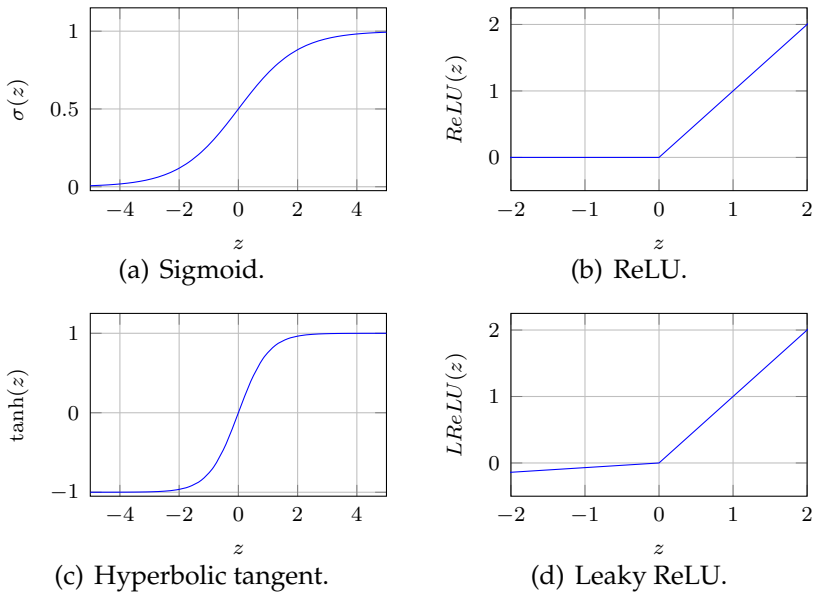
Then, such as in the Perceptron, each layer computes an affine transformation,

$$\mathbf{z}^{(l)} = \mathbf{W}^{(l)} \cdot \mathbf{a}^{(l-1)}, \tag{2.4}$$

followed by a nonlinear activation function, which is normally smooth, compared with threshold function in the original Perceptron,

$$\mathbf{a}^{(l)} = \sigma(\mathbf{z}^{(l)}). \tag{2.5}$$

Then we can consider a hidden layer as a function $g^{(l)}$ that depends on some



(a) Sigmoid. (b) ReLU.

(c) Hyperbolic tangent. (d) Leaky ReLU.

Figure 2.4: Most common activation functions in neural networks include the logistic sigmoid $\sigma(z)$, the hyperbolic tangent $tanh(z)$, the rectified linear unit $RelU(z)$ and its leaky variant $LReLU(z)$.

parameters such as a set of weights, a bias term, the number of units and the activation function. Commonly, the trainable parameters that characterizes a layer are denoted as $\boldsymbol{\theta}^{(l)}$. The output $\mathbf{y}$ of a MLP results in a composition of multiple functions,

$$\hat{\mathbf{y}} = h_{\boldsymbol{\theta}}^{(L)} \circ h_{\boldsymbol{\theta}}^{(L-1)} \circ \cdots \circ h_{\boldsymbol{\theta}}^{(1)}. \tag{2.6}$$

Generally, each layer of a ANN generates a response based on its input and a nonlinear activation function. Choosing the correct activation function can impact considerably on the performance of the ANN. Most used activation functions are illustrated in Fig. 2.4.

2.1.2 Activation functions

The activation functions have been one of the fundamental parts in the construction of a ANN. As mentioned above, to satisfactorily solve problems that are characterized by being nonlinear separable it is necessary to map the input data to a dimension in which they are linearly separable, normally to a higher dimension. This is the main characteristic of the activation function when computing a non-linear element-wise transformation of pre-activations that comes from the affine transformation. Then, the activation function and the affine transformation work together: the first, remains fixed while the second one is determined by the trainable

weights, which change during the training.

Many activation functions have been proposed during the last years. However, despite the understanding we have about this component in ANNs, the topic is an object of active and in constant change research field in the community of computer science. Some of the activations functions we consider the most useful are described in the following sections.

Sigmoid

The logistic or sigmoid function takes a real-valued number and maps it directly to [0,1]. This function, illustrated in Fig. 2.4 a is smooth and differentiable. For large negative numbers the function is asymptotic to 0 and for large positive numbers is asymptotic to 1.

The sigmoid function is defined as,

$$sigmoid(\mathbf{z}) = \frac{1}{1 + e^{(-\mathbf{z})}}. \tag{2.7}$$

The Sigmoid function has probably been the most used function in the literature. The main interpretation is related to that its shape is similar to the firing (threshold) function of the original neuron. Besides, the derivative of sigmoid function is very cheap to compute, once it is solved analytically,

$$\begin{aligned}
\nabla sigmoid(\mathbf{z}) &= \frac{e^{(-\mathbf{z})}}{\left(1 + e^{(-\mathbf{z})}\right)^2}, \\
&= \frac{1}{1 + e^{(-\mathbf{z})}} \cdot \frac{e^{(-\mathbf{z})}}{1 + e^{(-\mathbf{z})}}, \\
&= sigmoid(\mathbf{z}) \cdot \frac{e^{(-\mathbf{z})}}{1 + e^{(-\mathbf{z})}}, \\
&= sigmoid(\mathbf{z}) \cdot \frac{1 + e^{(-\mathbf{z})} - 1}{1 + e^{(-\mathbf{z})}}, \\
&= sigmoid(\mathbf{z}) \cdot \left(1 - \frac{1}{1 + e^{(-\mathbf{z})}}\right), \\
&= sigmoid(\mathbf{z}) \cdot (1 - sigmoid(\mathbf{z})).
\end{aligned} \tag{2.8}$$

Although the sigmoid function has been used extensively in the literature and in practice, it is currently not as widely used in DNNs as it has two drawbacks. First, the saturation. The backpropagation of errors relies on the gradient to determine parameters update. The sigmoid function saturates at both tails, resulting in a very small gradient. This problem, usually referred to as vanishing gradient, makes the training very slow and gradients tend to disappear in early layers when the network is very deep. The second drawback is that the output of the sigmoid function is not

zero-centered. It is widely believed that normalizing outputs (zero-center mean and unit variance) helps training (Szegedy et al., 2015).

Hyperbolic tangent

The hyperbolic tangent function can be used as an activation function as the sigmoid function as well. This function, shortened as tanh, is also a differentiable function in [-1,1]. Furthermore, this function as the sigmoid function suffers of the vanishing gradient but its output is zero-centered. The tanh function is defined as,

$$tanh(\mathbf{z}) = \frac{1 - e^{(-2\mathbf{z})}}{1 + e^{(-2\mathbf{z})}}. \tag{2.9}$$

Rectified Linear Unit

The Rectified Linear Unit (ReLU), since its introduction in DL applications, has become the reference of choice in many approaches (Krizhevsky et al., 2012; Szegedy et al., 2015; He et al., 2016). The activation function, illustrated in 2.4 b is defined as,

$$ReLU(\mathbf{z}) = max(0, \mathbf{z}). \tag{2.10}$$

This function, is very simple but has some interesting properties and a very few known drawbacks. First, there is no positive saturation. This function ensures a flow of the gradient whenever the input is positive speeding up the convergence. Then, as the sigmoid function, it is very cheap to compute in both ways, during forward propagation, a simple thresholding at zero is needed while its derivative is trivial:

$$\nabla(ReLU(\mathbf{z}^{(l)})) = \begin{cases} 0, & \text{if } \mathbf{z}^{(l)} < 0, \\ undefined, & \text{if } \mathbf{z}^{(l)} = 0, \\ \mathbf{a}^{(l-1)}, & \text{if } \mathbf{z}^{(l)} > 0. \end{cases} \tag{2.11}$$

Other desirable property of activation functions is to induce sparsity. This property allows producing representation that are easier to separate in a higher dimension, besides, only a few entries change when some small variations of the input are present (Glorot et al., 2011). Finally, ReLUs can die. When large gradients flow through a ReLU, it can change its weights in such a way that the unit will never be active again. Once the unit is by default "dead", any gradient will flow and the unit will never leave this state. Fortunately, there are two ways to solve this, one is to reduce the learning rate and the other is to modify the flat zero side to correct the sensitivity to this problem, e.g., such as in the following activation function, the Leaky ReLU.

Leaky Rectified Linear Unit

Recently, the most adopted form of the ReLU is the Leaky Rectified Linear Unit (LReLU). They have been proposet to alleviate the problem of dead ReLUs, by preventing saturation of the unit and allowing always to flow of a gradient through the unit. This, potentially can recover extreme values of the weights that can produce the unit shutdown. The LReLU is defined as,

$$LReLU(\mathbf{z}) = max(\alpha\mathbf{z}, \mathbf{z}), \tag{2.12}$$

where α is small, close to zero.

Such as the case of the ReLU, the derivative of LReLU is also trivial,

$$\nabla LReLU(\mathbf{z}) = \begin{cases} \alpha, & \text{if } \mathbf{z}^{(l)} < 0, \\ 1, & \text{if } \mathbf{z}^{(l)} > 0. \end{cases} \tag{2.13}$$

2.1.3 Backward propagation

The original learning rule to train a Perceptron does not allow to train a multi-layer artificial neural network such the MLP drawn in Fig. 2.3. Since to compute the weights update with for an intermediate layer is necessary to know its output and with equation 2.3 only the output of the last layer is given.

To solve this problem we can realize that the output of each unit is a non-linear function and differentiable of the inputs. Then, it is possible to compute the partial derivatives of the error, e.g., the mean square error. In other words, through the chain rule, it is possible to measure how much each of the weights has contributed to the current error of the model. In this way, accordingly to Rumelhart et al. (Rumelhart et al., 1986), the error can be propagated backwards so that each of the weights receives an update that through an optimization algorithm can iteratively adjust the weights to minimize the error.

Then, formally, during learning procedure is desired to minimize an error through learning the weights,

$$\mathbf{W}|_t = \mathbf{W}|_{t-1} - \eta \left.\frac{\partial E}{\partial \mathbf{W}}\right|_{t-1}, \tag{2.14}$$

where η is the learning rate parameter, E the error, and $\mathbf{W}|_{t-1}$ the old weights.

For a practical example, let us consider the MLP in Fig. 2.3. The network receives an input $\mathbf{x}$, then it returns an output $\hat{\mathbf{y}}$. In the case of the MLP illustrated in Fig. 2.3, the output is a scalar but it will be denoted here as a 1×1 vector to generalize when the output of the network has more output units, that is $\hat{\mathbf{y}}$.

Then, we consider a given metric, in this case the Mean Square Error (MSE),

$$E = MSE = \frac{1}{M} \sum_{\mathcal{D}} \frac{1}{2} \left\| \mathbf{y} - \hat{\mathbf{y}} \right\|_2 . \tag{2.15}$$

where the sum is performed over M samples of the dataset $\mathcal{D}$ and $\mathbf{y}$ represents the Ground Truth (GT), the desired output.

Then, the error contribution of each weight can be computed by the derivative of the error with respect to it,

$$\begin{aligned} \frac{\partial E}{\partial w_{ij}^{(l)}} &= \frac{\partial}{\partial w_{ij}^{(l)}} \left(\frac{1}{m} \sum_{\mathcal{D}} \left[\frac{1}{2} \left\| \mathbf{y} - \hat{\mathbf{y}} \right\|_2 \right] \right), \\ &= \frac{1}{m} \sum_{\mathcal{D}} \left[\frac{1}{2} \frac{\partial}{\partial w_{ij}^{(l)}} \left[(\mathbf{y} - \hat{\mathbf{y}})^T (\mathbf{y} - \hat{\mathbf{y}}) \right] \right], \\ &= \frac{1}{m} \sum_{\mathcal{D}} \left[(\mathbf{y} - \hat{\mathbf{y}})^T \frac{\partial (-\hat{\mathbf{y}})}{\partial w_{ij}^{(l)}} \right]. \end{aligned} \tag{2.16}$$

Considering again the MLP in the Fig. 2.3, the backpropagation method allows to compute the partial derivative of the output with respect each weight in the network. If we consider one weight of the first layer $w_{ij}^{(1)}$,

$$\begin{aligned} -\frac{\partial \hat{\mathbf{y}}}{\partial w_{ij}^{(1)}} &= -\frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{z}^{(4)}} \cdot \frac{\partial \mathbf{z}^{(4)}}{\partial w_{ij}^{(1)}}, \\ &= -\frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{z}^{(4)}} \cdot \frac{\partial \mathbf{z}^{(4)}}{\partial \mathbf{a}^{(3)}} \cdot \frac{\partial \mathbf{a}^{(3)}}{\partial w_{ij}^{(1)}}, \\ &= -\frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{z}^{(4)}} \cdot \frac{\partial \mathbf{z}^{(4)}}{\partial \mathbf{a}^{(3)}} \cdot \frac{\partial \mathbf{a}^{(3)}}{\partial \mathbf{z}^{(3)}} \cdot \frac{\partial \mathbf{z}^{(3)}}{\partial w_{ij}^{(1)}}, \\ &= -\frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{z}^{(4)}} \cdot \frac{\partial \mathbf{z}^{(4)}}{\partial \mathbf{a}^{(3)}} \cdot \frac{\partial \mathbf{a}^{(3)}}{\partial \mathbf{z}^{(3)}} \cdot \frac{\partial \mathbf{z}^{(3)}}{\partial \mathbf{a}^{(2)}} \cdot \frac{\partial \mathbf{a}^{(2)}}{\partial w_{ij}^{(1)}}, \\ &= -\frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{z}^{(4)}} \cdot \frac{\partial \mathbf{z}^{(4)}}{\partial \mathbf{a}^{(3)}} \cdot \frac{\partial \mathbf{a}^{(3)}}{\partial \mathbf{z}^{(3)}} \cdot \frac{\partial \mathbf{z}^{(3)}}{\partial \mathbf{a}^{(2)}} \cdot \frac{\partial \mathbf{a}^{(2)}}{\partial \mathbf{z}^{(2)}} \cdot \frac{\partial \mathbf{z}^{(2)}}{\partial w_{ij}^{(1)}}, \\ &= -\frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{z}^{(4)}} \cdot \frac{\partial \mathbf{z}^{(4)}}{\partial \mathbf{a}^{(3)}} \cdot \frac{\partial \mathbf{a}^{(3)}}{\partial \mathbf{z}^{(3)}} \cdot \frac{\partial \mathbf{z}^{(3)}}{\partial \mathbf{a}^{(2)}} \cdot \frac{\partial \mathbf{a}^{(2)}}{\partial \mathbf{z}^{(2)}} \cdot \frac{\partial \mathbf{z}^{(2)}}{\partial \mathbf{a}^{(1)}} \cdot \frac{\partial \mathbf{a}^{(1)}}{\partial w_{ij}^{(1)}}, \end{aligned} \tag{2.17}$$

and we replace $\hat{\mathbf{y}}$ with its corresponding activation $\mathbf{a}^{(4)}$, such as in equation 2.4 and equation 2.5,

$$-\frac{\partial \hat{\mathbf{y}}}{\partial w_{ij}^{(1)}} = -\frac{\partial \mathbf{a}^{(4)}}{\partial \mathbf{z}^{(4)}} \cdot \mathbf{W}^{(4)} \cdot \frac{\partial \mathbf{a}^{(3)}}{\partial \mathbf{z}^{(3)}} \cdot \mathbf{W}^{(3)} \cdot \frac{\partial \mathbf{a}^{(2)}}{\partial \mathbf{z}^{(2)}} \cdot \mathbf{W}^{(2)} \cdot \frac{\partial \mathbf{a}^{(1)}}{\partial \mathbf{z}^{(1)}} \cdot [x_j]_i, \tag{2.18}$$

where the vectorized input $[x_j]_i$ with the same shape of $\mathbf{z}^{(1)}$ whose *i*-th element is equal to x_j and all others are 0. Each derivative corresponds a $n \times n$ matrix,

$$\frac{\partial \mathbf{a}^{(l)}}{\partial \mathbf{z}^{(l)}} = \begin{bmatrix} \frac{\partial \sigma_1(\mathbf{z}^{(l)})}{\partial z_1^{(l)}} & \cdots & \frac{\partial \sigma_1(\mathbf{z}^{(l)})}{\partial z_n^{(l)}} \\ \vdots & \ddots & \vdots \\ \frac{\partial \sigma_n(\mathbf{z}^{(l)})}{\partial z_1^{(l)}} & \cdots & \frac{\partial \sigma_n(\mathbf{z}^{(l)})}{\partial z_n^{(l)}} \end{bmatrix}, \tag{2.19}$$

where n denotes the cardinality of both, $\mathbf{z}^{(l)}$ and $\mathbf{a}^{(l)}$.

Finally, when σ is element-wise applied, e.g., the sigmoid, tanh or ReLU, then $\sigma_i(\mathbf{z}^{(l)}) = \sigma(z_i^{(l)})$ due to that $\sigma(z_i^{(l)})$ depends on the *i*-th element of z. Then, the previous matrix is a diagonal matrix,

$$\frac{\partial \mathbf{a}^{(l)}}{\partial \mathbf{z}^{(l)}} = \begin{bmatrix} \sigma'(z_0^{(l)}) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma'(z_n^{(l)}) \end{bmatrix}. \tag{2.20}$$

here, this property can be exploited to speed up computation.

2.2 Deep neural networks

In recent years, artificial neural networks have dominated and surpassed classical computer vision methods in different fields. These advances have undoubtedly brought great benefits to different sectors of society, the research community and industry. The area of computer vision is no exception. The greatest advances have been achieved with a combination of three parts: i) computing capacity, ii) large-scale datasets and iii) deep neural networks. In the previous section, the MLP was presented as part of the history in ML and we precised the limitations of these kind of neural networks. Their topology and the way input data is taken and transformed is not suitable for new perspectives where structured data is the input, such as images, video or audio clips. Here, Convolutional Neural Networks (CNNs) come into play.

In CNNs the main operation for features extraction is convolution. Hence the name. The first CNN, the Neocognitron, was proposed by Fukushima in 1980 (Fukushima, 1980). However, in the literature the first application of convolutional networks is attributed to LeCun et al. in 1998 for handwritten digits recognition (LeCun et al., 1998). The limited computing capacity of the time when those approaches where released turned off the attention of a big portion of research community, but not all of them. In 2012, the new era of deep learning started with AlexNet (Krizhevsky et al., 2012), a CNN used to beat the state-of-the-art in the ImageNet challenge reducing the classification error drastically in a large-scale dataset with more than one million images and a thousand categories[1].

In contrast to MLPs, a CNN can exploit the structure of data. The intuition behind is that input data –such as images, video or sound clips– contains a local topological structure which does not depend on the specific location given a reference system, i.e., the image coordinates. That means that objects can appear at any position in the image. Then, a CNN uses this understanding to apply the same detector at any location in the image. The pattern detector is in this case applied by convolution layers.

Additionally, in CNNs also feature more operations than convolution, such as pooling layers, activation layers and fully-connected layers. The conjunction of all these operations in a chain can generate useful features that allow patterns identification. However, despite all the layers needed to build up a CNN, convolution and pooling layers are considered as the most important to extract multiple features and reduce computational complexity, respectively.

The convolution operation is formally introduced in section 2.2.1, focusing on the main parameters and its functionality in CNNs. After, in section 2.2.2, the pooling operation is introduced as it is a fundamental concept of interest in this research.

[1] In LeCun's work, the dataset is composed of 10,000 training images and 10 categories

Finally, we briefly review the most common deep architectures from the literature in section 2.2.3.

2.2.1 Convolution

Previously, in section 2.1.1, the Multi-Layer Perceptron was introduced. We have stated that the nonlinear responses of affine transformations are the master key that allow these artificial neural networks to learn from data (of course, applying backpropagation of errors). The input vector is multiplied with a set of weights to produce an output which then passes through a nonlinear activation function. This is applicable to any type of data: images, sound clips or a collection of descriptors. Even when these data are spatially arranged and related, such as a grayscale image is a 2D matrix, they can always be flattened before being fed into the neural network.

Any type of data is stored into a multidimensional array. Each axis of these arrays is key to access to different intrinsic properties of the data, i.e., columns, rows and channels in images or time axis in sound clips. This arrangement is not exploited when we feed flattened versions of the data into the so-called fully connected layers in MLPs. Preserving this order in input data, we can take advantage to solve very specific tasks, such as object recognition, where pixels in images are related with their neighbors.

A discrete convolution is a linear operation that takes advantage and preserves this notion of ordering. When the convolution is applied to the *input feature map*, a *kernel* is superimposed to the *input feature map* sparsely. That means that only a few inputs contribute to generate a single value on the output map. Besides, parameters are reused, i.e., the same weights are used over all the regions of the *input feature map*. An example of a simple discrete convolution is presented in Fig. 2.5. In this example only a feature map is shown for simplicity. However, in practice, multiple feature maps are treated along channel axis with the same kernel or with a collection of different kernels.

As shown in Fig. 2.5, the kernel is superimposed over elements to produce a single value in the output feature map. For each location, the product of each element of the input and each element of the kernel is computed and summed up to produce a single value in the current location. This process in repeated in convolution layers using K kernels. The product of convolutional layers are the so-called feature maps[2]. In the case of multiple input feature maps the kernel would be 3-dimensional and each feature map has to be summed up element-wise to produce a master feature map. In addition, the 3-dimensional kernel slides only across width and height, finally for the case of K kernels, master feature maps

[2]For the sake of clarity, the term convolution is used in this manuscript to denote the operation of extracting feature maps through the network in convolution layers. The kernels are learned during training.

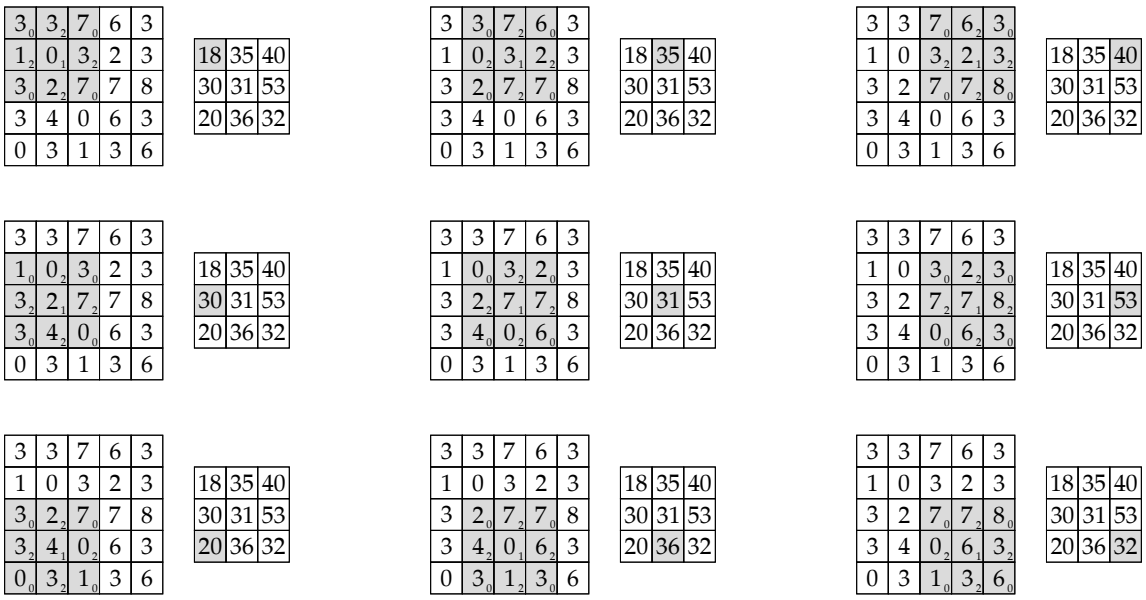


Figure 2.5: An example of the discrete convolution. The *input feature map* is a matrix of 5×5 and a 3×3 kernel. The shaded are superimposed on the *input feature map* denotes the neighborhood being processed and the shaded area over the *output feature map* denotes the output element at the current step.

extracted per kernel are concatenated to produce a multi-channel output feature map. This procedure is illustrated in Fig. 2.6.

The output of the convolution layer takes its shape based on the following parameters,

- n: input shape,
- k: kernel size,
- s: stride, and
- p: zero padding.

Note that stride parameter can be considered as the measure of how much the kernel is shifted over the input feature maps. For instance, in Fig. 2.7, a convolution is performed between a 5×5 input and a 3×3 kernel. The input is padded with a 1×1 border of zeros and a stride of 2×2. Padding feature maps is useful when we want to identify partial objects in images, with no padding, the kernel does not explore the hole image. In the illustrative examples is not that clear but with images as inputs and bigger kernels could be clear to see. For more information, we encourage the reader to review the textbook "Deep Learning" by Ian Goodfellow (Goodfellow et al., 2016).

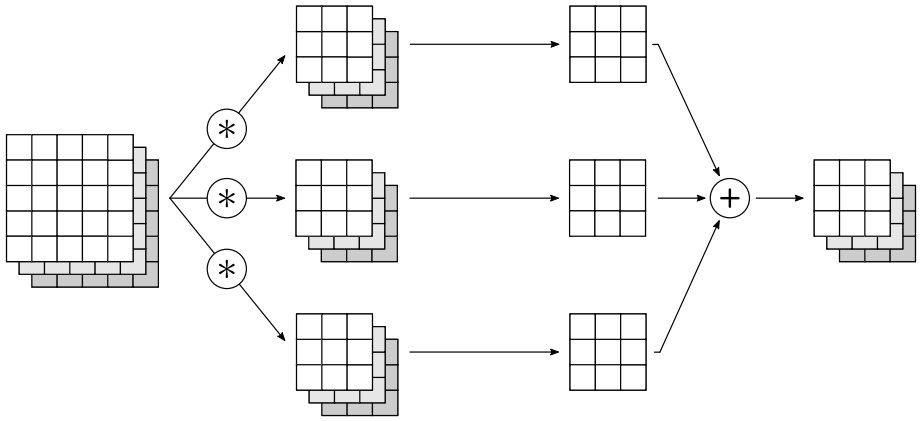


Figure 2.6: Output feature maps given a collection of 3-dimensional kernels. Kernels depth is equal to the depth of input feature maps. However, the resulting feature maps depth is equal to the number of filters used. Convolution is denoted as "$*$" and concatenation as "$+$".

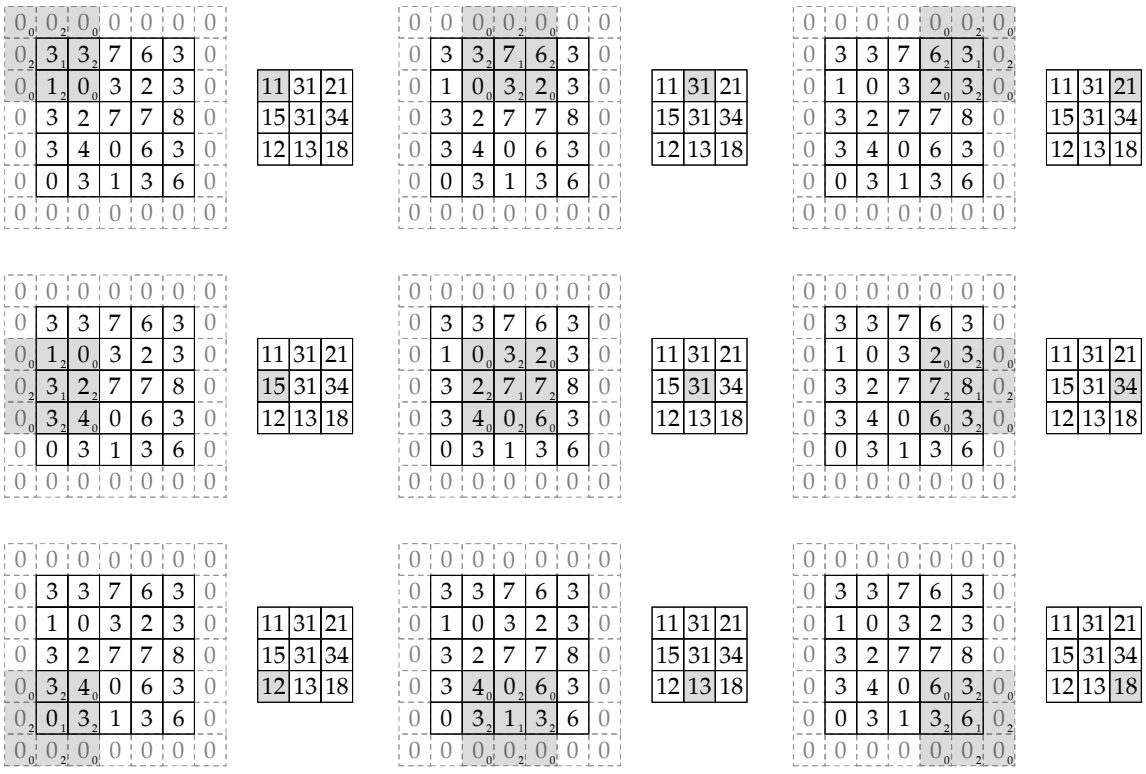


Figure 2.7: Output values of a discrete convolution with 1×1 zero padding, a 2×2 stride and a kernel size of 3×3.

2.2.2 Pooling

Pooling operation constitute another important block in CNNs. This operation reduces the dimension of the input feature maps while applying a summarizing operation over regions, in a very much similar way that convolution does but not using a kernel.

Although pooling operation seems to be very similar to convolution, the linear combination described by the kernel is not present, in his place a pooling function is used, such as max pooling or average pooling. Such as in convolution, a sliding window is used to pool out features from input feature maps. Two examples are given in Fig. 2.8 and in Fig. 2.9, for max pooling and average pooling, respectively.

The shape of output feature maps is affected by the following parameters,

- n: input shape,
- k: sliding window size,
- s: stride, and
- p: padding.

More formally, e.g., the max pooling outputs the maximum value of each neighborhood to summarize regions as,

$$o(x, y) = \underset{(k,s,p)}{maxpool} \left(I(\bar{x}, \bar{y}) \right), \tag{2.21}$$

where $(\bar{x}, \bar{y})$ denotes the neighborhood around coordinates (x, y). The max pooling operation is parameterized by k, s and p, the kernel size, stride parameter and the padding, respectively.

Pooling strategies are widely used in modern deep neural networks. This operation has two primary benefits: i) reduces computational complexity by reducing the dimensionality of the data for subsequent layers and ii) helps the network to achieve a degree of spatial invariance. In pooling layers is also possible to add zero padding, however this approach is more adopted into convolutional layers in order to maintain a volume in deep architectures, otherwise, desirable size/volume of feature maps could rapidly collapse into very small feature maps in deeper layers. More details about this operation and some modifications we propose, are presented in next chapters.

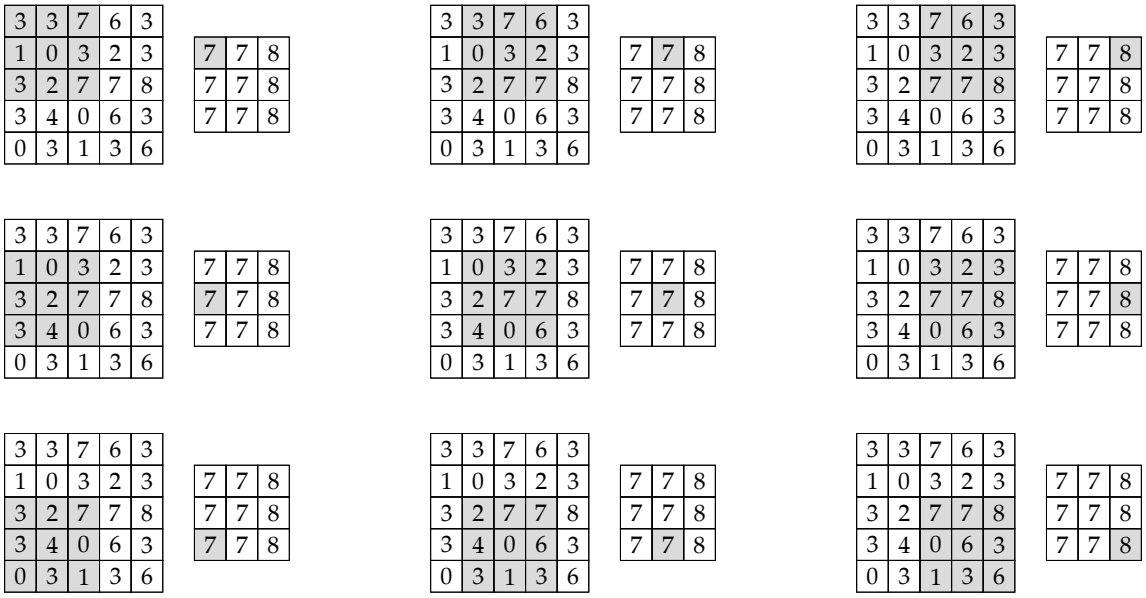


Figure 2.8: Compute max pooling output values.

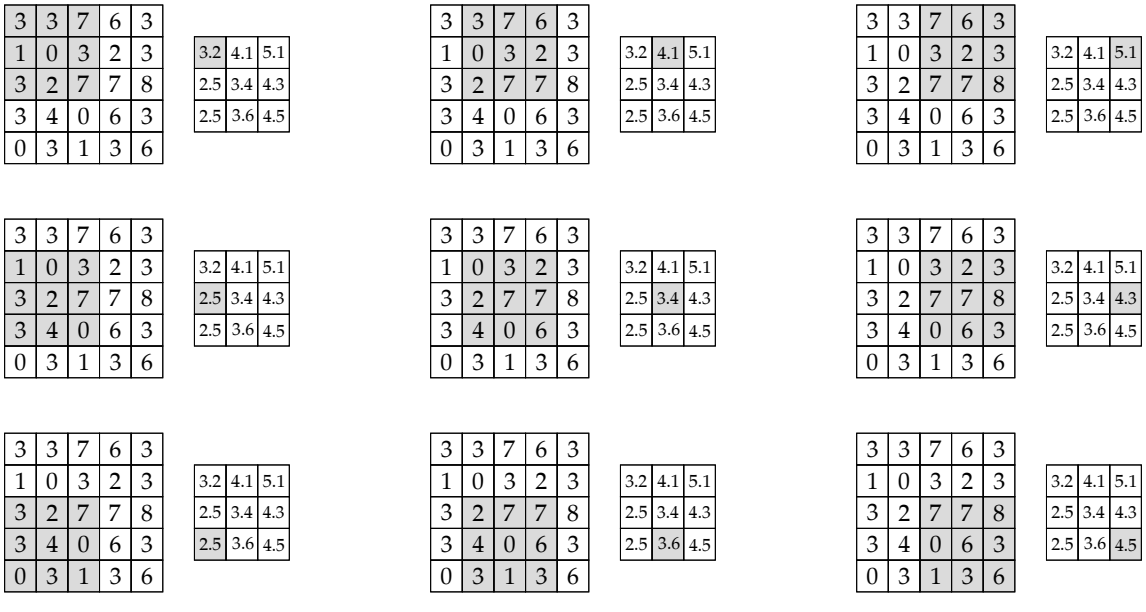


Figure 2.9: Compute average pooling output values.

2.2.3 Common architectures

In the collection of methods that have been proposed in the area of DL for automatic classification, there are mostly variants of CNNs. These variants have been proposed from different perspectives and pursuing also some other tasks. Hence, some of the models we briefly review in this section are multi-task architectures, e.g., for image classification, object recognition and even for semantic segmentation.

We consider as common architectures those that have been set a new record, primarily in the ImageNet Large Scale Visual Recognition Competition (ILSVRC), the models that are part of most of all the frameworks and taught in classes. However, other variants and more complex models have been proposed in the last years. An illustrative example of the architectures is presented in Fig. 2.10, the reader is invited to have a look while reading the following sections.

Krizhevsky et al., 2012: AlexNet

The AlexNet architecture is proposed by Alex Krizhevsky et al. in 2012 (Krizhevsky et al., 2012). This is the first deep model that outperformed other classic approaches, winning the ILSVRC (Russakovsky et al., 2015). This achievement, gave rise to the new era of learning from data and revived the interest on artificial neural networks for large-scale classification problems.

The proposed architecture is a 8-layer CNN which consists of convolutional layers, max pooling layers, activation layers and fully-connected layers. The architecture is composed of 5 convolutional layers parameterized with $k = \{[11 \times 11], [5 \times 5], [3 \times 3], [3 \times 3], [3 \times 3]\}$ and $s = \{4, 1, 1, 1, 1\}$, 3 max pooling layers with $k = \{[3 \times 3], [3 \times 3], [3 \times 3]\}$ and $s = \{2, 2, 2\}$. Finally, 3 fully-connected layers with $u = \{4096, 4096, 1000\}$, where u denotes the number of output units and 1000 the categories in ImageNet dataset. All activation layers in this architecture are configured with ReLUs. In the original paper, the authors include drop-out layers in between fully-connected layers to avoid over-fitting. The best model, reached a top-5 error rate of 15.04%.

Zeiler and Fergus, 2014: ZFNet

After the success of AlexNet in the ILSVRC 2012, Zeiler and Fergus proposed an AlexNet-like architecture, winning the ILSVRC 2013 (Zeiler and Fergus, 2014). The work of Zeiler and Fergus was not to propose a new architecture but to present a visualization that was very useful for the selection of new parameters through a hard ablation study. The process of visualization revealed some special features related to different layers in the network; such as filters in layer 1 are a mix of high-low frequencies, in layer 2, they found aliasing artifacts, in layer 3, the network identifies some general patterns, then, in layer 4, class-specific patterns were identified, such

as dog's faces, and finally in layer 5 objects with significant pose variation are completely identified. This architecture is omitted in Fig. 2.10 due to the similarity with AlexNet.

Based on preliminary analysis from visualization results, the AlexNet architecture was mainly modified by changing the first convolution layer kernel size from 11×11 to 7×7 and the stride parameter to 2, instead 4. This model achieved a top-5 error rate of 11.2%.

Szegedy et al., 2014: GoogLeNet

The GoogLeNet model is presented by Szegedy et al. in 2014. This model is the winner in the ILSVRC'14 with a significant improvement over ZFNet (ILSVRC'13) and AlexNet (ILSVRC'12).

The topology of this network allows stacking inception blocks to very deep convolutional networks. As shown in Fig. 2.10, the inception block is composed of convolution layers with 1×1, 3×3 and 5×5 with 1 pixel stride and 3×3 max pooling. The input, a set of feature maps coming from a previous layer is processed by these convolutions, the feature maps generated then, are concatenated as output. The key idea is to maintain the computational complexity as low as possible. Besides, the authors claim that by reducing the computation bottleneck, then, the width and depth of networks can be increased.

Finally, GoogLeNet outperformed previous approaches in the same context (ILSVRC). The top-5 error achieved is 6.67%.

Simonyan and Andrew Zisserman, 2014: VGGNet

The VGGNet was invented by the Visual Geometry Group (VGG) from University of Oxford. This network improved results over ZFNet (ILSVRC'13) and AlexNet (ILSVRC'12) in the classification task of ImageNet dataset. During ILSVRC'14, this model finished in second place.

The backbone architecture is based on the idea of using only 3×3 filters in convolution layers. Using only 3 convolution layers with 3×3 filters (27 weights) is possible to cover the same area of a single layer of 7×7 (49 parameters). Using this approach is possible to replace any layer with a bigger receptive field using only back-to-back layers with 3×3 filters. The top-5 error rate achieved with this proposal in image classification is 7.5%. However, combining models the authors present a better performance of a top-5 error of 6.8%.

He et al., 2016: ResNet

After the success of DNNs to solve classification and recognition tasks, Microsoft Asia came up with a very deep residual neural network, the one called ResNet He

et al., 2016.

This model set new records in performances on image classification, object recognition and object location, all of them were set with same architecture. It is believed by the authors that residual connection in a residual unit carries original information, allowing a easier optimization process. In the original paper, many versions of the network are analyzed being ResNet-152 the deepest. However versions of 1000-layers where trained. Using an ensemble of models, this approach resulted in the winner of ILSVRC'2015 with a top-5 error of 3.57%.

Other variants

During the development of all these reference architectures many other models where proposed, such as the Xception network where inception modules are replaced with depth-wise separable convolutions (Chollet, 2017). Then, the new version of the GoogLeNet architecture (Inception-V3), the Inception-V4 (Szegedy et al., 2017) where the main difference is that the authors integrated a new stem module and added more inception modules using the same number of filters for each module. In the same paper, in 2017, Szegedy et al. proposed the Inception-ResNet-V2 architecture, converting inception modules into residual inception blocks and proposed a new version of the original inception block they used only once after the stem module. Finally in 2017, Xie et al. (Xie et al., 2017), presented the ResNeXt architecture. It is similar to what we know as ResNet-50. The main difference is that in convolution modules and identity modules, 32 parallel branches/towers are added.

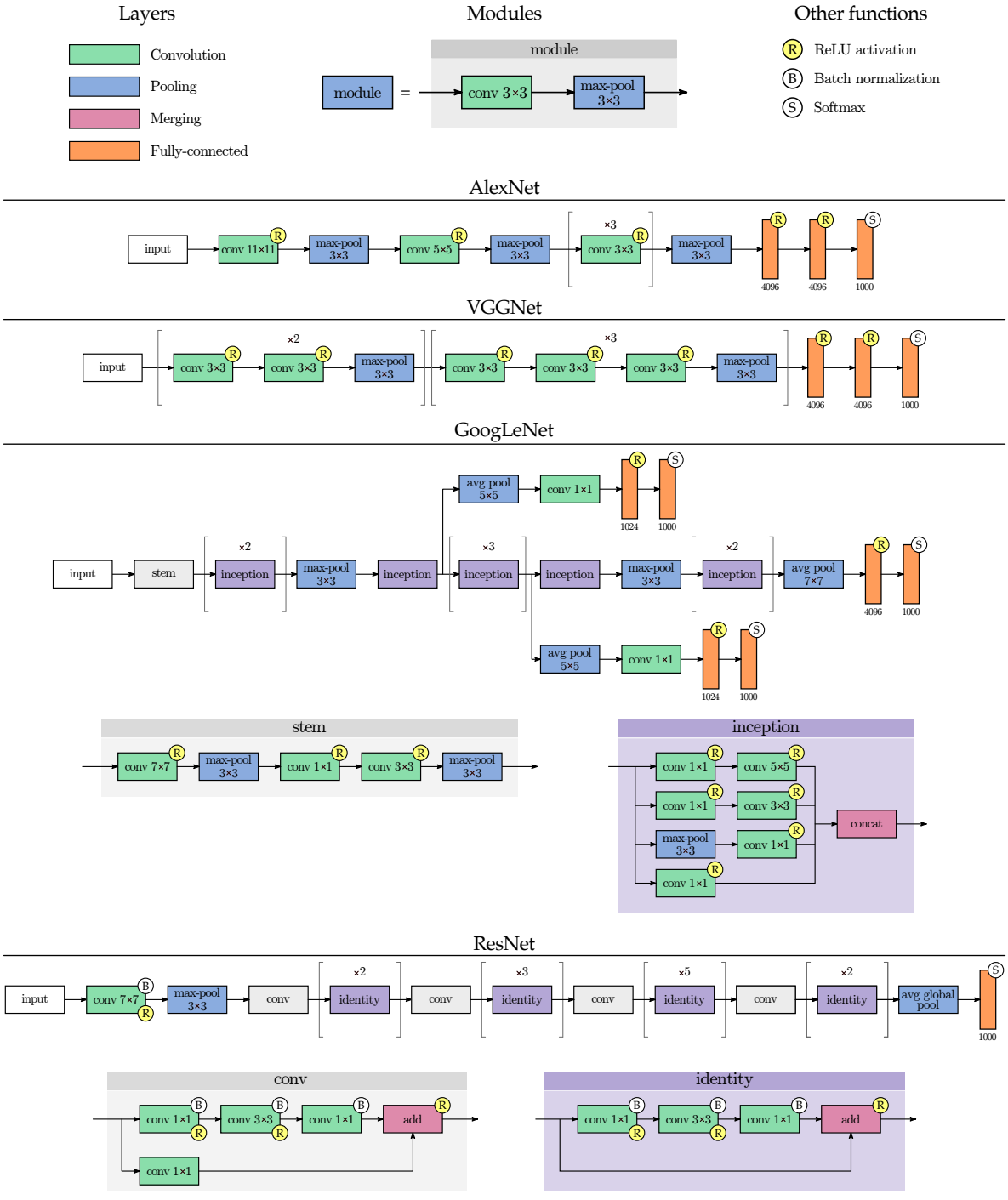


Figure 2.10: Common deep neural networks architectures.

2.3 The human visual system

Even when visual perception can represent a very natural and simple task for humans, the brain is not capable of processing all the information that is captured by the eyes every millisecond. The task is not only to capture information, but also to treat it and extract characteristics associated with objects, semantically associate them and finally understand the scene.

The scene, generally speaking, is composed of multiple objects, in different positions, in movement or not, at different depths, from different points of view, occlusions, among other characteristics. In addition, each object in the scene has different characteristics, such as shape, color or texture. In order to understand a scene, the human brain developed very advanced processes to minimize the quantity of relevant incoming visual information to process. Then, our perception is built according to what we find attractive in the scene. This process is only possible made by the HVS and is called visual attention.

The basis of the visual attention modeling dates back to the 80's when Treisman and Gelade's proposed the Feature Integration Theory (FIT) (Treisman and Gelade, 1980). According to FIT, any scene can be decomposed into simple features such as intensity, color, size, movement and orientation, just as the brain does. These features may be most or may be less salient and which are the reason why attention is attracted when they are combined by the HVS.

A few years later, the first approach was proposed by Koch and Ullman to generate visual attention maps combining features using a feed-forward network (Koch and Ullman, 1987). These maps are generated by a Winner-Take-All (WTA) neural network to filter out and select high salient regions of the image. Then model employs a mechanism that allows attention to be directed to next most prominent position. These authors are known for introducing the concept of saliency map, as a topographically arranged map that represents the visual attention on a given scene.

Many models were proposed then to predict saliency maps from digital images (Baluja and Pomerleau, 1995; Tsotsos et al., 1995). However, one of the first validated implementation of Koch and Ullman's method was published by Itti et al. (Itti et al., 1998) a few years later. This method was applied to natural as well as synthetic scenes. Since then, the field gained a lot of interest and several applications emerged based on it; object recognition, video compression and quality assessment, to mention a few.

While many other approaches have been proposed, it is obvious that a general scheme that accurately represents the functioning of the human visual system is not easy to find. Even considering studies such as FIT, which helps to understand how the brain perceives low-level characteristics and how these characteristics should be filtered out while being more relevant than others to the task of predicting eye movements and their derivatives (fixations), it is an open problem, given that, the

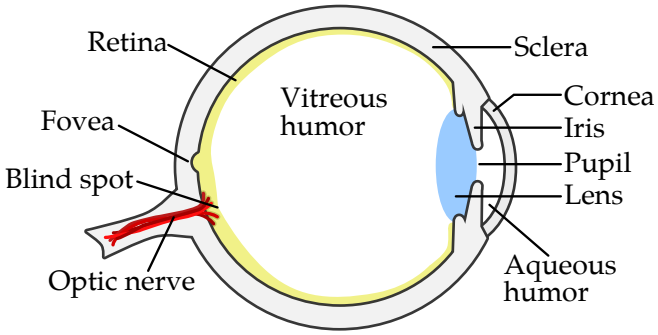


Figure 2.11: Parts of the human eye.

mechanisms of visual attention fall into both directions, into the combination of high-level and low-level characteristics.

2.3.1 The human eye

The human eye can be compared to a digital camera. On a camera, the image is projected on the sensor through the lens. In order to capture a good picture, the projected image should be in focus on the sensor. The focus, on a digital camera is controlled by adjusting the lens distance from the sensor. A good image, should not be too dark or bright; the amount of light is controlled by the diaphragm. On the human eye, it is very similar (see Fig. 2.11). The focus is controlled by the lens. The aperture, performed by the diaphragm in the digital camera, is adjusted by the iris. Finally, the image is projected on the photosensitive region on the back of the eye called the retina.

When light comes into the human eye, it passes through the exposed part, the cornea. Then, light rays cross the acqueos humor to reach the iris, where the eye takes its color. This muscle controls the size of the pupil, the hole at the iris center. Finally, the light crosses the lens, the vitreous humor and reaches the retina. The retina us supported by the sclera and contains photo-receptive cells; rods and cones. The dispersion of rods and cones on the retina is not uniform and it is depicted on Fig. 2.12. Cones are dominant on the retina center region, called the fovea and rods are located on the peripheral retina. The optical disc is where the optical nerve is connected is also known as the blind spot on the retina. The protective layer of the eye is called the sclera.

The aqueous and the vitreous humor are important parts of the optical system of the eye. The light refraction phenomenon that takes place in these liquids allows the projection of images on the retina. Under nine meters, the lens plays a very important role to focus on relatively near objects and the bend of the cornea refracts parallel light rays from objects in far vision conditions.

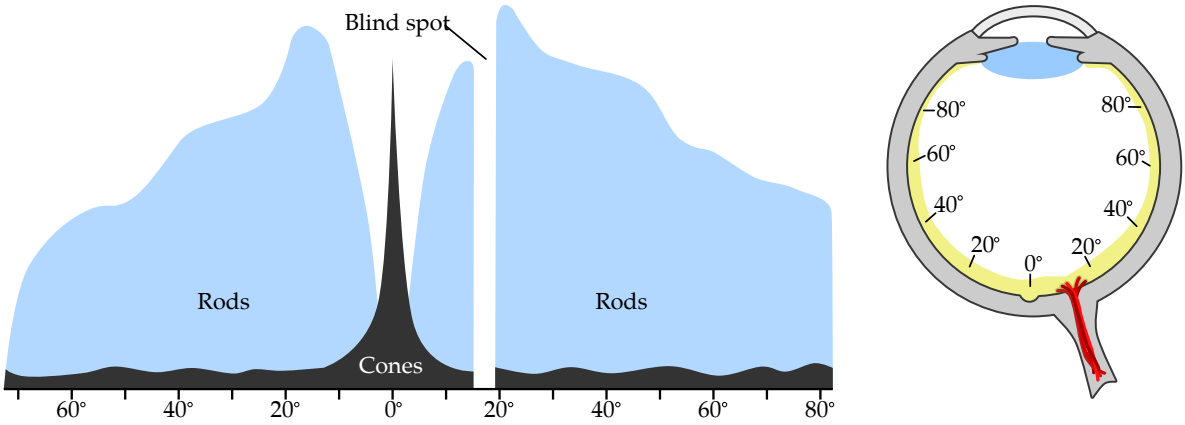


Figure 2.12: Distribution of rods and cones in the retina.

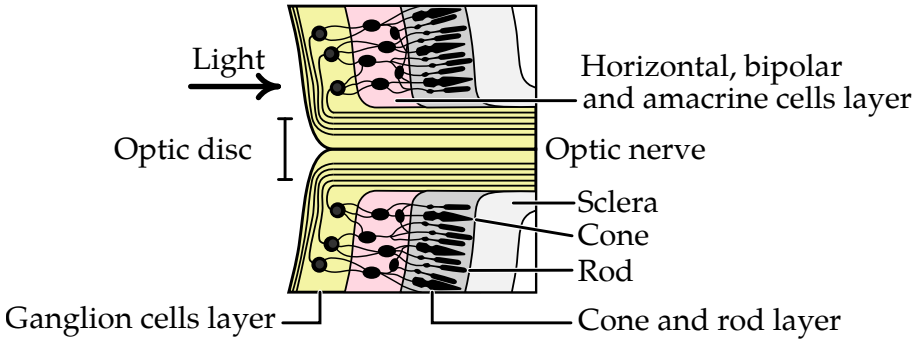


Figure 2.13: Cell layers in the retina.

2.3.2 The retina

The retina, as the main receptor of our visual system, is composed of five layers; photo-receptors, horizontal cells, bipolar cells, amacrine cells and ganglion cells (see Fig. 2.13). The photo-receptors, cones and rods, constitute the deepest layer of the retina. Cones are color sensitive and useful during daytime vision while rods are sensitive to low light intensities and night vision. Both, cones and rods are interconnected to horizontal cells and bipolar cells. Then, horizontal cells are connected from photo-receptors to smooth and transport information of average of luminance to bipolar cells. Bipolar cells that connects as well photo-receptive cells to ganglion cells are sensitive to luminance contrast through the center-surround mechanism. After, amacrine cells modulate the response gain of bipolar and ganglion cells, they are sensitive to temporal contrast, playing a very important role on motion detection. Finally, ganglion cells constitute the last layer of the retina and transmit signals to the brain, their axons meet the optic nerve.

2.3.3 Eye movements

There are three main eye movements, saccades, the smooth pursuit and micro-saccades. The saccades are very fast eye movements to foveate objects we are interested in. This, ensures that the object is directly projected on the center of the fovea. These movements are extremely fast, between 30 and 80 ms. Between two saccades, the eye stops moving over a new region of interest to fix the gaze from 250 to 500 ms. This period is called fixation and during this period input signals are treated. Fixations are very important on the attention modeling process and pixel-based methods of attention representation take advantage of this particular characteristic of the human visual system, e.g., Wooding's method (Wooding, 2002). In order to track moving objects in scene, smooth pursuit movements of the eye are performed in around 100 ms. If the moving object remains in the foveal vision, the HVS is able to extract more information while the object is on movement. Finally, the micro-saccades are a kind of a fixational eye movement, small and involuntary, it is believed that micro-saccades are very important to prevent image fading on the retina.

2.3.4 Recording gaze fixations

Human visual attention can be recorded by the eye-movements the observer performs during a given recognition task. As mentioned in the previous section, eye movements are classified into saccades, fixations and smooth pursuits. During the visual scene exploration or the execution of the recognition task, all these movements occur sequentially. Here, for the creation of subjective saliency maps, the fixations are of interest to us because it is when the observer positions the object of interest on the fovea and features are processed by the HVS.

Initially, the recording of eye movements was done using intrusive devices, derived from contact lenses, called suction caps by A. L. Yarbus in 1967 (Yarbus, 2013). Nowadays, the eye movements are measured with non-intrusive devices called eye-trackers such as the one presented in Fig. 2.14. This device is composed of three main parts: i) two lamps, ii) a mirror and iii) a camera, all of them works only in the infrared segment of the electromagnetic spectrum. The lamps illuminates the eye while the camera captures the image of a single eye. With the infrared light, the pupil appears dark in the image which facilitates its identification. In addition, two reflections corresponding to the two infrared lights appear within the region of the pupil. To identify eye movements and record the movements, image processing is needed to identify both reflections and the center of the pupil. Finally, comparing measurements with the reference obtained during calibration, the coordinates of gaze tracking can be obtained on the coordinates of the experimental monitor.

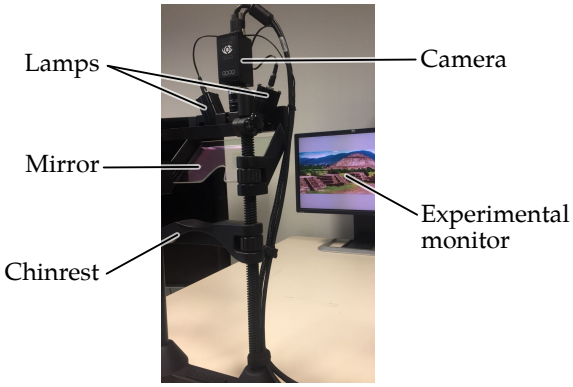


Figure 2.14: The main parts of the Cambridge eye-tracker system.

2.3.5 Gaze fixations density maps

When a visual task is performed, raw gaze data can be recorded using an eye-tracker system. This system provides useful information to build a GT reference dataset about visual attention. The raw data is normally recorded in the coordinates of the recorder –the eye-tracker default coordinates– where commonly the origin is the center of the experimental monitor (see Fig. 2.14). Then, the measurements can be translated to screen coordinates; i.e., top-left origin based coordinates. Recorded raw data is just a list of coordinates gathered with a given recording rate, e.g., 250Hz. Then, fixations are estimated from raw data. Saliency maps are then computed from fixations. In order to describe visual attention, gaze fixations are gathered from several participants due to that a single participant is not representative.

Accordingly to David S. Wooding (Wooding, 2002), given a set of fixations, a saliency map can be generated in three main steps: i) a partial saliency map is computed by depicting a Gaussian at each fixation coordinates, ii) all partial subjective maps are accumulated in a single matrix (element-wise sum) and iii) saliency maps are normalized.

The Gaussian is spread σ to a fixed angle $\alpha = 2^\circ$ to compute σ_{mm} based on the distance between the observer and the experimental screen $D = 3H$, with H being the monitor height (see Fig. 2.15),

$$\sigma_{mm} = D \times tan(\alpha). \tag{2.22}$$

Measurements in millimeters are converted to pixels based on the monitor resolution R, in pixels per millimeter,

$$\sigma = R \times \sigma_{mm}. \tag{2.23}$$

Then, given a collection of n measurements (fixations) $\mathbf{M} = [m_1, m_2, ..., m_n]$,

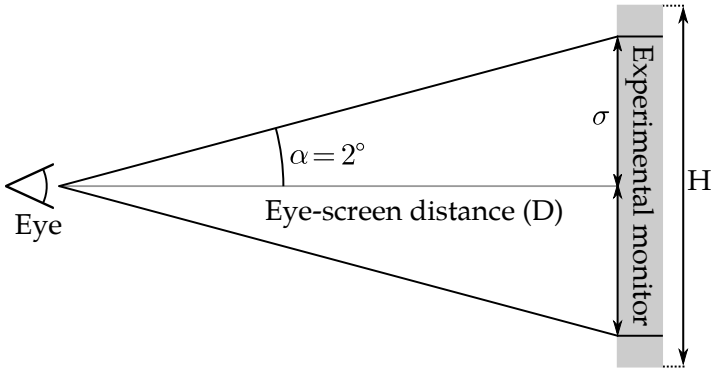


Figure 2.15: Fovea projection on the experimental monitor. The angle $\alpha = 2°$ denotes the half of the aperture of the eye regarding to the experimental monitor.

(a) (b)

Figure 2.16: (a) An illustrative example of recorded fixations over the source image and (b) the resulting saliency map.

where each m is a pair (x_m, y_m) denoting the center coordinates to draw the Gaussian. Thus, a partial saliency map is computed as,

$$P(x_m, y_m) = A \exp - \left(\frac{(x - x_m)^2}{2\sigma_x^2} + \frac{(y - y_m)^2}{2\sigma_y^2} \right). \tag{2.24}$$

Afterwards, we compute the sum of partial saliency maps as,

$$S(\mathbf{M}) = f\left(\sum_{i=1}^{n} P(m_i) \right), \tag{2.25}$$

where, $f(\cdot)$ is the min-max normalization function to map values in [0,1].

An illustrative example of recorded fixations over an image and the resulting saliency map is shown in Fig. 2.16.

2.4 Visual saliency modeling

The FIT describes how a scene or visual stimulus can be broken down into different elementary attributes for its understanding. These attributes are the key to attract bottom-up attention. Intensity, orientation and color are the main features considered in the FIT and are combined together in computer vision methods as elementary cues to build a saliency map.

2.4.1 Types of attention models

Focusing attention on specific regions of the scene is influenced and characterized into two main branches: bottom-up and top-down.

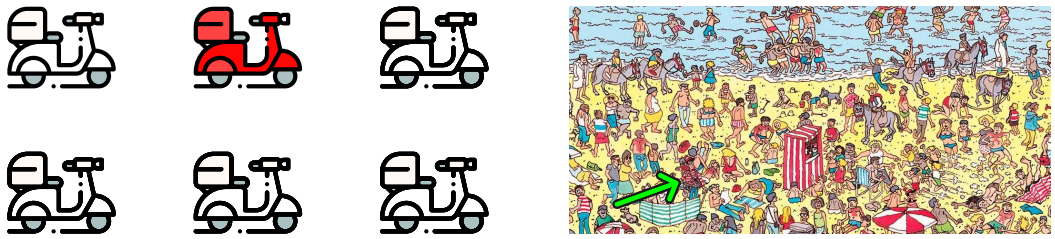
The first one, bottom-up attention, is driven by external stimuli and their inherent properties regarding to the background (Katsuki and Constantinidis, 2014), it is a exogenous process. This induced process in which information to be processed is automatically selected because noticeable features related to the task on stimuli, is involuntary and instinctive (see Fig. 2.17 a). Early approaches to modeling attention in images are based on psychological studies (Treisman and Gelade, 1980; Koch and Ullman, 1987). During the bottom-up attention process, target stimuli or objects pop-out from the background in terms of specific features, such as texture, color and shape. More recent methods, based on deep learning, generate saliency maps in a bottom-up manner (Pan et al., 2017; Chaabouni et al., 2019; Xia et al., 2016; Chi et al., 2019).

Top-down process is associated with a set of high-level cognitive operations and referred as endogenous process. This process is typically driven by a task, is voluntary and driven also by external influences, emotions, semantic relations, knowledge and contextually constrained (see Fig. 2.17 b). It is now widely known in the community that top-down effects are an important and inherent component of attention and that these effects cannot be overcome when no explicit task is assigned to observers (Benois-Pineau and Le Callet, 2017).

Different studies have been presented on the characterization of visual attention, either bottom-up (Koch and Ullman, 1987; Itti et al., 1998; Harel et al., 2007) or top-down mechanisms (Torralba et al., 2006; Yarbus, 2013; Carrasco, 2011). These approaches have proved that the bottom-up mechanisms precede the top-down influences and are faster, that takes more time for the brain to identify them but that they last longer afterwards (Tatler et al., 2005).

2.4.2 Bottom-up models

The FIT has introduced how elementary features and cues can compose a visual scene and why these features are important to attract human's attention. Then,



(a) (b)

Figure 2.17: Basic representation of exogenous bottom-up process in (a)[1], which is fast and unconscious, driven by its properties (mainly color) and an endogenous top-down process in (b)[2] which requires a cognitive effort to find Waldo, it is slower, semantically driven and task dependent.

1: Source: www.flaticon.com, 2: Source: Wikimedia Commons.

in this section we present the main contributions on predictive models of visual attention that are biologically-inspired from the HVS and from this theory as well.

Koch and Ullman: in 1987, Koch and Ullman presented their biologically inspired model of visual attention (Koch and Ullman, 1987). This model was the reference for further research, such as (Itti et al., 1998). According to the original authors, Koch and Ullman, it is first required to extract a collection of feature maps from the original image. These maps must be combined into a single attention map to encode salient elements in scene (stimuli). Then, they introduced a WTA network to filter out and select the most salient region of the map, this single region is called Focus of Attention (FoA). After the first and most salient region is selected, attention is shifted to a new FoA using once more time the WTA network but discarding the previous FoAs. This process is done iteratively until several zones of interest are found.

Itti, Koch and Niebur: There is no doubt that the most famous and cited model is the one proposed by Itti et al. in 1998 (Itti et al., 1998). This model, based on Koch and Ullman's architecture (Koch and Ullman, 1987) is illustrated in Fig. 2.18. The first step is to extract different attribute maps from the input image. The maps are organized into three families described below:

– **Color maps**: four color maps are extracted, $RGBY$, for red, green, blue and yellow, respectively:

$$R = r - (g + b)/2 \tag{2.26}$$

$$G = g - (r + b)/2 \tag{2.27}$$

$$B = b - (r + g)/2 \tag{2.28}$$

$$Y = \frac{r + g}{2} - \frac{|r - g|}{2} - b \tag{2.29}$$

– **Intensity maps:** an intensity map defined as:

$$I = \frac{(r + g + b)}{3} \tag{2.30}$$

– **Orientation maps:** using Gabor filters, orientation maps are obtained (oriented pass-band filters) using as input the intensity map. Only four maps are extracted using 0°, 45°, 90°, 135° filter orientation.

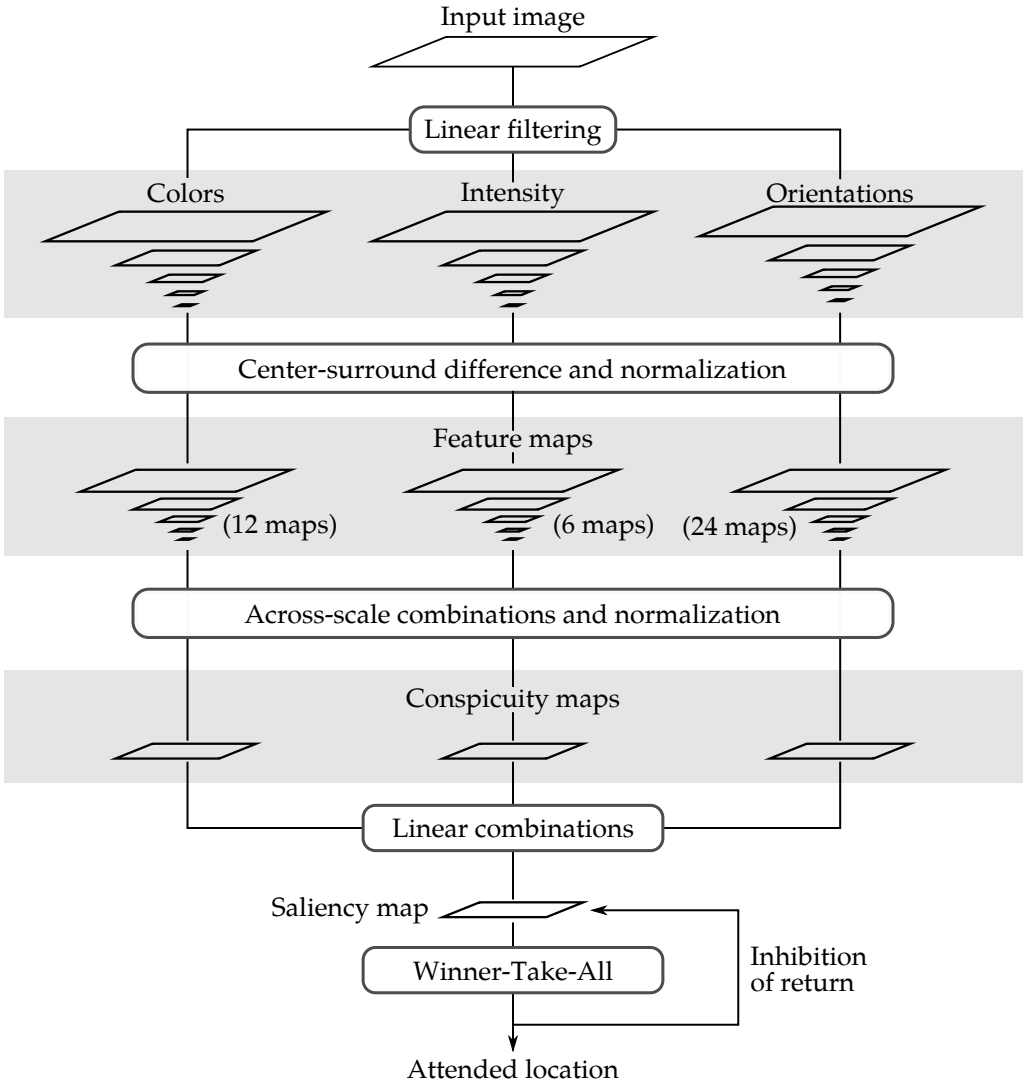


Figure 2.18: Operation of the model presented by Itti, Koch and Niebur. Redrawn from (Itti et al., 1998).

In a second step of this method, the attribute maps are taken to turn them into conspicuity maps to give more relevance to the regions which are different from their close neighbors. Then, in order to obtain specific maps of color, intensity and

orientations, they combine conspicuity maps from the same attributes family using the operator $\mathcal{N}$ which is in charge of maps normalization. This map normalization process is given by the multiplication between $(M - \bar{m})^2$ and each single map. Where M the maximum global value on the map and $\bar{m}$ an average of other local peaks/maximums of the map. This results in the conspicuity maps: $\bar{C}$, $\bar{I}$ and $\bar{O}$, for color, intensity and orientations, respectively. The main objective of this normalization is to give relevance to the maps that have a few intensity peaks but corresponding to real regions out of the image, and, besides, to penalize the maps that have a high amount of intensity peaks, which corresponds to a moderate but uniform amount of peaks in the input map.

Finally, these normalized conspicuity maps, $\bar{C}$, $\bar{I}$ and $\bar{O}$ are summed up in a conspicuity map to build the saliency map,

$$S = \frac{\mathcal{N}(\bar{C}) + \mathcal{N}(\bar{I}) + \mathcal{N}(\bar{O})}{3} \tag{2.31}$$

Finally, salient zones selection follows the architecture proposed by Koch and Ullman, which uses a WTA network to select the most relevant regions and there the FoA is found. Then, discarding previous FoA, subsequent FoA regions are obtained with the same WTA network.

Harel, Koch and Perona: the Graph-Based Visual Saliency (GBVS) model proposed by Harel, Koch and Perona in 2007 (Harel et al., 2007), presents a biologically inspired model consisting in two steps.

First, inspired by previous works, the activation maps are extracted by linear filtering, such as in Itti et al. (Itti et al., 1998). Let us suppose a given feature map $M : [n]^2 \to \mathbb{R}$, the main goal is to obtain an activation map $A : [n]^2 \to \mathbb{R}$. In this map, locations $(i, j) \in [n]^2$ where $M(i, j)$ will correspond to high activation values in A. The proposal is based on the definition of dissimilarity of $M(i, j)$ and $M(p, q)$,

$$d((i, j)||(p, q)) \triangleq \left| \log \frac{M(i, j)}{M(p, q)} \right|, \tag{2.32}$$

where $p(i, j) = Pr\{M(i, j)|neighborhood\}$.

The authors consider a fully-connected directed graph by connecting every node of feature map M labeled with indices $(i, j) \in [n]^2$ with all other $n - 1$ nodes. The directed edge from the node (i, j) to (p, q) is assigned to a weight as,

$$w_1((i, j), (p, q)) \triangleq d((i, j)||(p, q)) \cdot F(i - p, j - q), \tag{2.33}$$

where $F(a, b)$,

$$F(a, b) \triangleq \exp\left(-\frac{a^2 + b^2}{2\sigma^2}\right), \tag{2.34}$$

here, σ is a free parameter of this method. However, the authors recommend to set it to one tenth to one fifth of the map width.

Subsequently, normalization of activation maps begin with $A : [n]^2 \to \mathbb{R}$ which is aimed to normalize. A second graph is constructed with n^2 nodes labeled with indices from $[n]^2$. Then, for each node (i, j) and every node (p, q), including (i, j), an edge is introduced from (i, j) to (p, q) with weight,

$$w_2((i, j), (p, q)) \triangleq A(p, q) \cdot F(i - p, j - q), \tag{2.35}$$

Normalizing the edges of each node gives the opportunity to compute the equilibrium distribution over the nodes. Mass flows preferentially to nodes which concentrates a high activation, it is an algorithm by construction and parallelizable.

Other models: after the FIT and the architecture of Koch and Ullman were released, many other models of bottom-up attention prediction have emerged. In 2009, Seo and Milanfar (Seo and Milanfar, 2009), proposed a method to predict saliency based on how similar pixels are to their surroundings based on extracted local features. Another model was proposed by Judd et al. in 2009 (Judd et al., 2009), the core idea was to incorporate visual data, collected using an eye-tracker system, into the saliency prediction process. They propose as well to integrate a collection of low, medium and high level features of the salient regions of the image to train a Support Vector Machine (SVM) (Cortes and Vapnik, 1995) as a prediction model. For more details, we refer the reader to the well-known benchmark of several saliency models presented by Judd et al. (Judd et al., 2012).

2.4.3 Top-down models

Although the literature on bottom-up attention models is much more extensive than the literature on top-down attention models, it has expanded in recent years with impressive results. The inclusion of factors such as faces, human figures, objects, music, voice, among others, in the attention modeling process has allowed to improve the results in the prediction of top-down attention maps (Ma et al., 2005; Cerf et al., 2008; Coutrot and Guyader, 2014). Besides, some approaches in the literature try to integrate both models of attention in a generic way so that the factors that allow for top-down attention can be explained with the help of the detection of different objects, considering that the photographer also takes consider the objects that appear in the picture (Borji, 2012; Tian et al., 2014; Zhu et al., 2014).

In recent years, some proposals in the field of machine learning have been based on externally-generated data, such as manually annotated attachments or fixations. This has proven to be very effective for predicting saliency in static images (Torralba et al., 2006; Kanan et al., 2009). However, with recent advances in the field of DL,

some proposals are able to generate salient patches on the image and subsequently classify them in a salient-object classification process approach (Shen and Zhao, 2014). Other approaches, such as the ChaboNet, proposed by Chaabouni et al. (Chaabouni et al., 2019), are specialized on predicting saliency maps on videos in a top-down scheme combining different inputs as additional channels with images. The proposed network gives competitive prediction results with a very shallow network. In the field of DL, many other schemes have been proposed, e.g., for visual saliency prediction with adversarial networks (Pan et al., 2017), integrating low and high-lever features for saliency prediction (Chi et al., 2019), to detect gaze and saliency on images (Chaabouni and Precioso, 2019) and for image quality assessment (Yang et al., 2019).

The proposal of a universal method to predict top-down attention saliency maps is impossible. The biggest challenge is that top-down attention is normally based on a particular task, is voluntarily directed to characteristics that allow to solve the task and is very subjective. However, when modeling this type of attention, it is possible to use some elements that are easily recognizable and semantically associated to identify the regions of interest in images we want to, i.e., salient regions.

Chapter 3

Mexican architecture dataset

In this chapter, we are interested the Mexican architecture and its relevance as part of country's identity. Mexican architecture, developed since the prehispanic era is considered a primordial piece of the history of ancient civilizations. In addition, with different influences on Mexican architecture in the following eras, during the time of the conquest and the revolution, each of these elements found in Mexican territory is important and tells part ot the history. Nevertheless, for this research, it is crucial to generate a reference database that allows us gather all these characteristics that make each of the architectures special and that allow their recognition.

Figure 3.1: Examples of Mexican architecture.

3.1 Introduction

Many architectural structures in Mexico including prehispanic and colonial cities have been designated World Heritage sites by the UNESCO, given their historical and artistic importance. The architecture in Mexico is mainly related to three periods, the prehispanic, colonial and modern.

In the prehispanic period, humanity has left its mark on the Mexican territory. Findings of utmost importance have contributed to the explanation of the natural habitat of ancient civilizations. From the materials used for the construction of households, to the evolution of the architecture itself, derived in aesthetic changes, from simple to complex forms. Nowadays, archaeologists in Mexico continue excavations, such as in Teotihuacan, resulting in an increase in quality and quantity of the information related to the site and its ancient civilization.

With the arrival and establishment of Spanish government in Mexico, the first churches and monasteries were built based on principles of classical and Arabic architecture. In 1521, when New Spain was established, the first cathedrals in Mexico were built. The Merida Cathedral is the oldest church in the continent and was completed on November 4, 1598. During the colonial period, the dominant form of art was Baroque style in many forms, the trend was to follow rectilinear patterns, squares, columns, pilasters, orbs, arcs and ornaments.

During the last period, after the end of the Mexican revolution in 1917, the endorsements for housing, education and health care building programs began. In Mexico, modern architecture has notable parallels with North America and European counterparts, its evolution highlights several unique characteristics, which represents a challenging in modern architecture around the world. A fine example is the studio designed for Diego Rivera and Frida Kahlo in Mexico City and the Soumaya museum. Many other architectural structures were built around the country between 1950 and 1982, for example a complex of buildings of the National Autonomous University of Mexico, the National Auditorium, The Vasconcelos Library, the Latin-American Tower, among others. Illustrations of all this architectures are shown in Fig. 3.1.

Given all this changes in the Mexican architecture history, we recover images from big data collections to build a dataset of Mexican most representative architectural structures. This is fundamental for the development of the main objective of this research. In the remaining part of this section, we present the main

characteristics of the dataset based on the architectural styles. Then, a dataset labeled for specific architectural recognition in 142 categories is introduced.

3.2 Mexican architectural styles dataset

Given the relevance of cultural content in its digital form and the importance of technologies in a globalized world, allowing users around the world to have immediate access is one of the challenges and priorities of any technological project, from a cultural development perspective. The development of this technologies should enable users to make use of digital content, browse, discover and share knowledge about any topic. In the cultural context, recognition tasks are more focused, which means that the current recognition systems specialize in recognizing specific content, of which there is a compilation by experts in the field and a content curation process. This is a critical task, expensive but useful for more advanced developments.

A dataset, is the result of a process of gathering relevant information to a particular topic of interest. In our case, Mexican architecture. We partially gathered images for this dataset during the development of the bi-national MEX-CULTURE project (Ramírez et al., 2015), a multimedia platform developed for the preservation and dissemination of the Mexican Culture, interested on content recognition in two main branches:

- Image and video: for the recognition of natural landscapes, population and architecture.
- Speech and audio: for the recognition of indigenous languages, sound classes for the National Record Library and important Mexican speakers.

Each category to recognize in this dataset was populated using publicly available content in YouTube channels of Mexican cultural institutions, such as Once IPN, TV UNAM, Mexico Today and O2 Mexico. This first dataset, contains 781 MP4-encoded documents involving all the categories, where just a few videos were related to architecture. Hence, our dataset building methodology is based on these roots.

3.2.1 Main architectural styles

Mexican architectural structures are classified in three different periods. According to the main characteristics representing each of them, one can easily distinguish styles describing each period as follows,

- **Prehispanic:** this style is composed of buildings with slopes and large steps. They are usually built out of adobe or rock blocks and generally have irregular forms.

- **Colonial:** most of these architectural structures are conformed with arches, domes and towers.
- **Modern:** this category is populated mainly by skyscrapers with rectangular geometries, glass, rectangular windows and straight lines extended along the structure.

The samples in this database must represent each architecture as good as possible given captured images by different users, from different perspectives throughout the day. This makes the classification task more complex but allows for more robust predictions, where these conditions might appear, varying mainly the context on the scene. Here, Mexican architecture is understood as any work located in the Mexican territory. The databases presented in this work exclude abroad works made by Mexican architects.

3.2.2 Gathering related videos

In 2016, we presented our first effort on architectural recognition. The main task was to identify these structures that are more relevant in the country based only on the architectural styles; prehispanic, colonial and modern. Thus, we started gathering copyright-free videos from any YouTube channel using a set of keywords we prepared based on architectural styles and an internally-developed video downloader (Montoya Obeso et al., 2016b). As expected, using just a few keywords the amount of non-related videos to architecture was tremendous. Then, we manually choose related videos, for two reasons; i) it was easy to identify each architectural style, usually a single architectural style was present in each footage and ii) given this first set, we could be able to have a perspective of most relevant architectures in the country and prepare a first list of specific Mexican buildings.

3.2.3 Mexculture buildings dataset

During gathering and annotation process, each video footage is manually segmented and annotated using ELAN software (Wittenburg et al., 2006). Here, each video shot stands for an annotated video segment where the object of interest appears. For each shot, the main task of the annotator was to identify the style of the architectural structure. Using non-annotated sections we are able to extract discriminative samples[1] (often called negative samples). During this annotation task, we considered TREC Video Retrieval Evaluation (TRECVID) concepts to describe better each scene, such as Building, Sky, Person, Car, Nighttime, Daytime, among others, and specific concepts to identify the style of architectural structures

[1]For more information the reader is referred to Montoya et al. (Montoya Obeso et al., 2016b).

Figure 3.2: Samples in MexCulture Buildings Dataset. Three architectural styles are considered from top to bottom row; prehispanic, colonial and modern.

in Mexico. This list, is part of the thesaurus presented in the Mex-Culture project (Ramírez et al., 2015).

Using the annotation methodology based on shots, we gathered positive samples for architectural style recognition where each category in this set stands out for a specific style, such as prehispanic, colonial and modern, and a discriminative category, called *other* with images extracted from non-annotated segments, totaling 16,700 images. Here, each category for style recognition is balanced, i.e., each category contains the same number of samples. This dataset is called MexCulture Buildings Dataset (MCBD) in the remaining of this document.

Most of the extracted images from videos in the MCBD are in range of 640×480 and 1200×720 pixels. Some examples of considered structures are illustrated in Fig. 3.2. Although the shots-based annotation methodology is very efficient and well structured to deal with main categories (styles) and content related concepts, we found that many other architectures were not present in this first set for style recognition. Hence, we continued our research on specific architectural structures recognition.

3.3 Specific Mexican buildings database

In order to identify each architectural structure, during the construction of the MCBD, we consider now intra-class categories for its identification, i.e., specific structures for each style and distributed around the Mexican territory. The main objective, is to build a list of the most representative buildings along the country for its recognition. However, without experts collaborating with us in this research, one of the most challenging tasks is to compile a list of most relevant architectures around the country.

3.3.1 Relevant architectural structures

In order to identify specific architectural structures, we propose to build up a list of most relevant architectural structures in Mexico and then run a web-based retrieval process to populate each category in the dataset. A "relevant architecture", here stands for a representative architectural work located in any city/town in Mexico.

Any architectural work can be relevant or not for different people, it depends on the personal interest for what each one of the architectures represents. Therefore, in the search for the most relevant architectures per state, we take as a reference the amount of results produced by Google search site. The main idea is to build a list of architectural structures located in the Mexican territory for each state to produce "Google number results", this number is the index of relevance. But, why is this index relevant? It is important to clarify how we visualize this. On the web, an indexer, such as Google site, is constantly scanning websites, and efficiently stores information related to the content of any page, any publicly available document to offer fast results to users. The number of results that a "query" may offer by its search, represents, approximately, the number of times the concept has been used in any content, such as books, websites, documents, video descriptions, metadata, etc. For example, executing the queries "Mexico City, Palace of Fine Arts" and "Mexico City, Metropolitan Cathedral of Mexico City", Google search gives 16.8M and 4.9M results, respectively. Then, which one is more relevant? If we ask the reader, it still depends on his interest, but it is not the case for indexing systems such as Google search[2].

During the process, the first step was to choose the architectural structures by state, without knowing their relevance index looking into different search services, such as Google, Bing or DuckDuckGo. After filtering relevance results, we found that some architectural elements were not relevant at all (we discarded elements with less than 50,000 results). Then, resulting in a list of 372 categories, 3 different styles, distributed in 32 states. Figure 3.3 shows the distribution of selected architectural structures, most of them are concentrated at the center of the country.

3.3.2 Keywords-based image search

The process of web crawling allows downloading related videos based on keywords, i.e., combinations of architectural styles, location, and name of the architectural work. The web-crawling process is performed for each category as follows,

- Combine item attributes to design queries; style, state name, city and name.

[2]Google number results might change when we use different servers in different countries.

Figure 3.3: Territorial distribution of Mexican architectural structures in the MexCulture buildings dataset.

- Retrieve copyright-free images from Google.[3]
- Discard non-related images manually.

Performing this process for every single category in the list of architectural structures, we populated each category as well as possible. Unfortunately, a high number of images are not found for each category in the list, only 142 categories where partially populated, totaling 8,155 images, with 63±29 images per category.

3.3.3 Saliency Mexculture142 database

The Saliency MexCulture142 dataset is the result of running the psycho-visual experiment described further in section 5.5. Images in this dataset are annotatied by recording gaze fixations in order to generate real human-gaze saliency maps. The annotation was performed by PhD and Master students of the University of Bordeaux. For each image, raw tracking data is recorded. In total 23 students participated on the visualization and annotation task.

The annotated dataset is composed of the following files,

[3]We use the python package: https://github.com/hardikvasa/google-images-download, thanks to Hardik Vasa.

- Images: 142 specific buildings.
- Fixations: each file holds the fixations of all the participants.
- Density maps: the so-called saliency maps, computed as in section 2.3.5.
- Scanpaths: includes 6,532 with fixation coordinates (in pixels) and fixations lenght (in seconds).

This dataset is freely available on NAKALA server for research purposes, a detailed description of the experiment can be found on https://www.nakala.fr/data/11280/5712e468.

Chapter 4

Saliency-based content selection for style recognition

In this chapter, we compare object proposal methods for the recognition of Mexican architectural structures while training Deep Neural Networks (DNNs), specifically Convolutional Neural Networks (CNNs). The so-called saliency maps are introduced in the first stage of the training process, for content selection. We consider that the advantages of selecting content based on salient regions are the following: i) it is faster than using a semantic segmentation method, ii) it is not necessary to annotate the position of the object of interest, iii) the salient region normally contains a semantic relationship of the content of interest regarding the scene. Besides, we compare our proposal with two different methods: Selective Search (SS) and Multiscale Combinatorial Grouping (MCG). Finally, we benchmark our models on the MCBD and on a subset of ImageNet.

4.1 Introduction

The automatic recognition of cultural content is considered by different cultural institutions as a key element for the preservation and dissemination of any culture. In Mexico, these tasks are of current relevance, since institutions need to keep their documents alive and in constant use. Architecture is a clear example, it is part of a country's identity. Large cities are known for the architectural works they house and Mexico is no exception. In previous research, Llamas et al., in 2016, present a learning-based approach to the identification of architectural elements as part of the identity of different architectural influences (Llamas et al., 2016). These approaches clearly depend on a reference collection –the training set– from which they gain experience by processing. In addition, the context plays an important role in the recognition task since the objects of interest can be immersed in the scene, at different levels.

Recently, it has been proven that different methods of ML and DL have a great capacity to identify objects in complex scenes, even when the objects are partially occluded or are captured from different perspectives (Krizhevsky et al., 2012; Szegedy et al., 2015; Simonyan and Zisserman, 2014). Such deep approaches to classify images on the ImageNet dataset (Russakovsky et al., 2015) require a strong and expensive effort to label image by image even when collaborative-specific annotation tools are employed. In contrast to previous research, where the objects of interest are immersed in the scene, training is performed over crops of the input data. The image is randomly cropped, regardless of content (considering only the central-bias theory). When introducing randomly generated patches, an unstable behavior is also introduced, where the image annotated with a single label, may or may not contain the annotated objects.

Instead of using methods to generate proposals for regions that require a certain level of computational complexity and are characterized by being exhaustive, such as SS (Uijlings et al., 2013a) and MCG (Pont-Tuset et al., 2017). These kind of exhaustive methods have been used before by Ross Girshick (Girshick, 2015) as object proposal methods in Regions with Convolutional Neural Networks (R-CNN). Even with most recent approaches based on CNNs (Ghodrati et al., 2017) in scene, we consider that object proposal based on the salient regions of the image could be more effective. In addition, training by means of the exhaustive region generation methods becomes extremely heavy. Therefore, we seek a balance between computational complexity when generating regions of interest and the amount of information to train the models. Then, the performance of CNNs must increase in terms of overall accuracy when introducing the previously selected information through a criterion based on visual attention as the one employed by González-Díaz et al. in a SVM (González-Díaz et al., 2016). In this work, we follow a strong trend in research, the integration of visual attention into classifiers, such as the approaches

for content-based image retrieval (Papushoy and Bors, 2015; Carvalho Soares et al., 2012; Benois-Pineau and Le Callet, 2017).

In this chapter, the main objective is to explore the methods of content selection for the recognition of Mexican architectural structures while training DNNs, specifically CNNs. The so-called saliency maps are introduced in the first stage of the training process, for content selection. We consider that the advantages of selecting content based on salient regions are the following: i) it is faster than using a semantic segmentation method, ii) there is no need to annotate the position of the architectural structure, iii) the salient region normally contains a semantic relationship of the content of interest regarding the scene. We also benchmark this approach on a general purpose dataset such as ImageNet (Russakovsky et al., 2015).

The results and methodology described in this chapter was previously published in the paper "Architectural style classification of Mexican historical buildings using deep convolutional neural networks and sparse features", in "Journal of Electronic Imaging", of the International Society for Optics and Photonics. For more details please see Montoya Obeso et al. (Montoya Obeso et al., 2016a).

The remainder of this chapter is organized as follows. In section 4.2 the methods of content selection are described. After, in section 4.3, the data augmentation methods, the backbone architectures and the training details are presented. Later, section 4.4, summarizes the results on both tasks; for architectural structures recognition and on the general purpose dataset. Finally, in section 4.5, we conclude this chapter.

4.2 Data selection methods

This section briefly introduces the main methods used for content selection as training data.

4.2.1 Geometrical cropping

Consider the input image I, where the width and height are denoted by M and N, respectively. Now, the shorter side of the image is taken to generate a square region centered in the image, either horizontally or vertically, depending on the orientation of the input image. Finally, the square region centered on the image is resized to 256×256 pixels. In the case of images where $N = M$, the image is directly resized to 256×256.

In the case of images where there are multiple annotations, with their respective bounding boxes (such as ImageNet), a Jaccard coefficients vector is arranged with all combinations between the central region and all objects annotated in the image

and the object with the highest Jaccard coefficient is selected. Formally, consider the central region of the image as G and each annotated object as I_i. Then, compute Jaccard's coefficient $J_i = (G \cap O_i)/(G \cup O_i)$ for each combination of G and O_i. Finally, O_i^* represents $max(J_i), i = 1, ..., k$, where k denotes the number of GTobjects annotated. All the Regions of Interest (ROIs) with a Jaccard coefficient greater than 0.7 are retained. The resulting crop with this method is called Geometrical Cropping (GC) in the remaining of this document.

4.2.2 Selective search

In the context of object recognition, the Selective Search (SS) method proposed by Uijlings et al. (Uijlings et al., 2013a), allows to exhaustively generate regions of interest by exploiting the structure of the scene. They propose to use content-based region grouping methods that allows the generation of high quality regions of interest, using complementary color spaces.

The proposed algorithm is mainly composed of three parts; i) the hierarchical grouping, ii) a diversification of strategies in different color spaces and iii) the combination of different results. In the stage of diversification of strategies, the authors propose to use a collection of different color spaces and a random initialization of the regions. The color spaces are; i) the grayscale image, ii) the Hue, Saturation and Value (HSV), iii) the Red, Green and Blue color model (RGB), the CIELAB, iv) RG channels plus intensity (rgI), v) normalized RGB and vi) the hue channel from HSV. To measure similarity in the range [0, 1], they use; i) color similarity, ii) texture similarity and iii) the similarity of joint regions (holes are filled to measure regions similarity). Finally a strategy of merging regions ensures that different bounding boxes for the same region best encapsulate the this common region, also reducing the amount of ROIs proposed.

For the automatic annotation of regions of the ImageNet dataset, we use the same process presented in section 4.2.1, based on Jaccard coefficient.

4.2.3 Multiscale combinatorial grouping

The Multiscale Combinatorial Grouping (MCG) method, presented by Pont-Tuset et al. (Pont-Tuset et al., 2017), proposes a bottom-up multi-scale hierarchical segmentation for ROIs generation. An efficient normalized cuts algorithm is used in this method in combination with a multi-scale hierarchical segmentation and a grouping algorithm that takes advantage of hierarchical information in a combinatorial space to produce object proposal from images. The exploration of the multiscale combination allows for the production of a large number of regions of interest. This method does not depend on any parameter.

Figure 4.1: Saliency Based data selection method.

The image labeling process and ROIs selection is performed based on the Jaccard coefficient, such as in previous sections for the large-scale classification task.

4.2.4 Saliency-based data selection

In order to make the saliency-based content selection method easy to replicate, the choice is to use the so-called bottom-up saliency maps. Since the bottom-up methods roughly express the sensitivity of the HVS to the stimuli, mainly depending on the texture, the color and the orientations presented on the scene, they are the ideal candidate to predict saliency on natural images. In addition, there is more research in the literature about bottom-up models, such as the proposed by Itti et al. (Itti et al., 1998) and Harel's GBVS model (Harel et al., 2007). These prediction models enabled several approaches in the literature, such as for image retrieval systems, object tracking, actions recognition, semantic segmentation, among others (Papushoy and Bors, 2015; Carvalho Soares et al., 2012; Bhowmik et al., 2017; Su et al., 2014; Mahadevan and Vasconcelos, 2013; Mathe and Sminchisescu, 2012; Wang et al., 2013).

For our experiments we selected the well-kown GBVS method for our Saliency-Based (SB) content selection method. This saliency prediction method has shown better performances over others and remains with a relatively low computational cost during predictions. As shown in Fig. 4.1, the SB selection is performed as follows,

1. Predict I_{sm} through GBVS (Harel et al., 2007), the saliency map.
2. Normalize saliency values in [0,1].
3. Threshold pixel values in normalized I_{sm} where $\phi = 0.2$, resulting in a binary mask.
4. Find max and min axes limits on the salient binarized region to create a bounding box.
5. Crop the image.

As in the previous content selection methods, the label is selected using the GT annotations. In this method, the same methodology based on the Jaccard index is used to label crops. In other methods, such as in SS method and MCG method, an

automatic annotation error can be included when having multiple objects and these are located closely in the same neighborhood. Finally, with both, the GC method and the SB content selection method, it is also possible to include part of the context, which introduces an error in the annotation by cutting off portions of the objects as illustrated in the Fig. 4.1. The selection methods are summarized in Fig. 4.2.

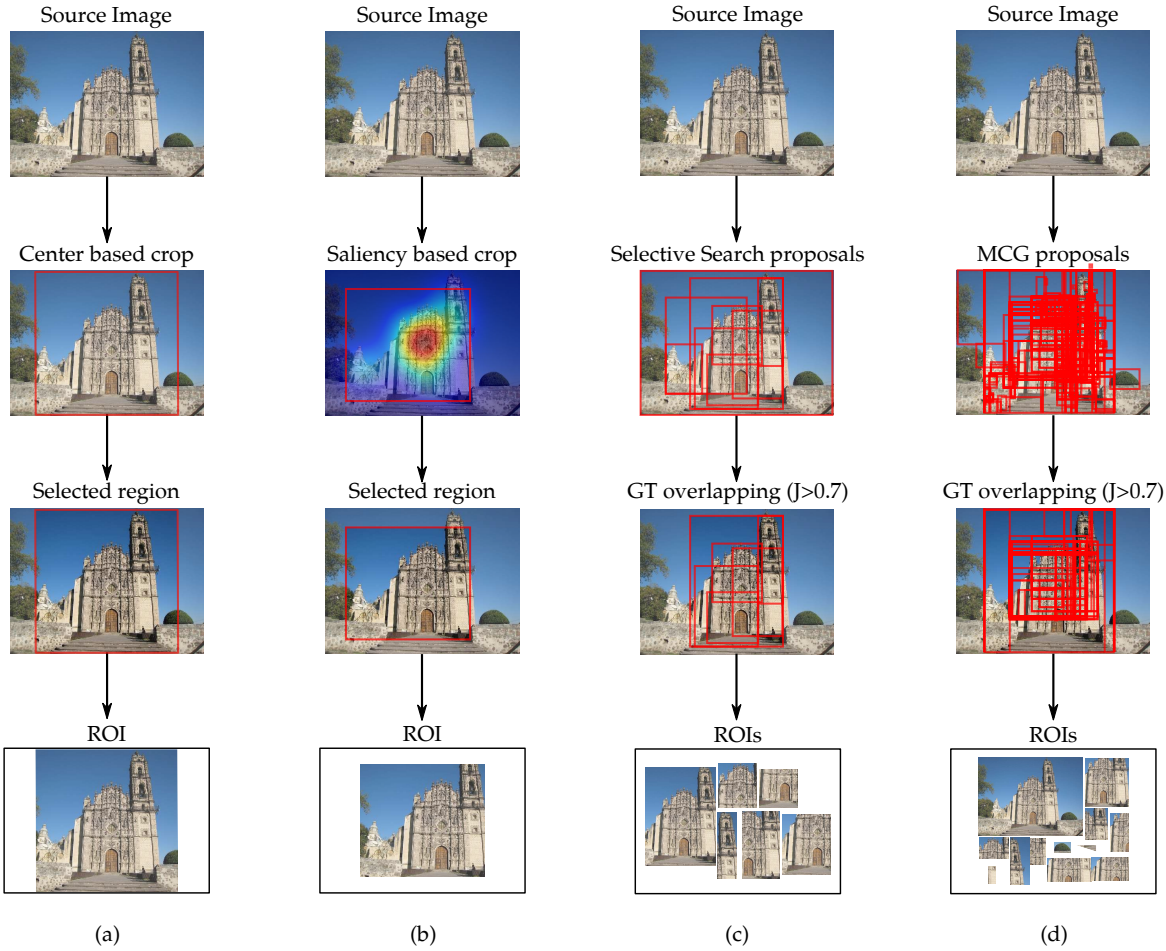


Figure 4.2: Data selection methods: (a) GC, (b) SB, (c) SS and (d) MCG.

4.3 Trainings setup

In this section, we present the main data augmentation methods, the architectures and the hyper-parameters used during trainings.

4.3.1 Data augmentation

To prevent over-ffiting and improve concepts generalization, different versions of each of the training images are introduced during training (Howard, 2013;

Krizhevsky et al., 2012). The label preserving transformations we apply to each image are the following; i) rotations of $\pm 5^{\circ}$ and $\pm 10^{\circ}$, totaling 4 additional images per sample (introducing also zoom out factor when we preserve all the image information in the resulting image) and ii) we introduce mirror flips, in all directions, i.e., vertical, horizontal and both. The latter transformation produces 3 more variants of the input image, totaling 7 variants per image.

For the second large-scale classification task, we apply the same data augmentation to each ROI. Therefore, each class will be partially invariant to these transformations. Finally, we fill out outliers when required, specifically when we rotate the image and to normalize the size to squared images, resulting easier to handle data and feed a CNN. For more details on how the outliers are filled, we invite the reader to review the article published (Montoya Obeso et al., 2016a). An example of a transformed image with different outliers filling –somethimes refered as padding– is shown in Fig. 4.3.

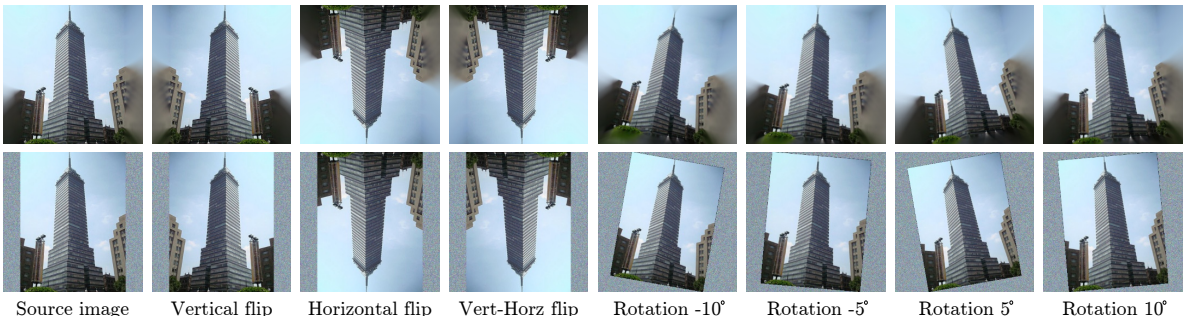


Figure 4.3: Data augmentation methods, rotation and flips. Top row: blur padding. Bottom row: Gaussian noise padding.

4.3.2 Architectures

To compare results between different architectures, we selected the well-known architectures AlexNet (Krizhevsky et al., 2012) and GoogLeNet (Szegedy et al., 2015) as backbone architectures. Since they presented outstanding results in the classification tasks in 2012 and 2014, respectively. These two architectures mainly keep a low computational cost, which is convenient for our experiments, which are located at both ends, a low-scale database for the identification of cultural content and a large-scale database for image classification. At this stage we discard other deeper networks in order to be more objective and to analyze the behavior of well-studied trained models using the generated data by the content selection methods. For both architectures, we normalize the RGB input image to 256×256. The illustration of both architectures is presented in Fig. 2.10.

4.3.3 Models training

During the training of both models, we set as initial reference the hyperparameters proposed in each of the reference papers, from Krizhevsky's and Szegedy's works. The initial learning rate is 0.01, we verified that it was adequate to find the first models convergence. We used a learning rate decay schedule at each *i*-th iteration, following a negative exponential,

$$lr = e^{-\gamma^i} \tag{4.1}$$

with $\gamma = 0.96$.

We trained both models during 90 epochs with MexCulture buildings dataset and during 30 epochs with ImageNet dataset. We find these parameters enough to validate convergence of the models.

We found that Stochastic Gradient Descent (SGD) with Nesterov's momentum (Nesterov's Accelerated Gradient (NAG) (Nesterov, 1983)) is very effective for both models training. The momentum coefficient we use is 0.9, such as in Krizhevsky et al. (Krizhevsky et al., 2012). More formally, the NAG algorithm updates the weights $\mathbf{W}$ considering the loss function $L(\mu, \mathbf{W}_t, \mathbf{V}_t)$ following the rule,

$$\begin{aligned} \mathbf{V}_{t+1} &= \mu\mathbf{V}_t - \eta\nabla L(\mathbf{W}_t + \mu\mathbf{V}_t) \\ \mathbf{W}_{t+1} &= \mathbf{W}_t + \mathbf{V}_{t+1} \end{aligned} \tag{4.2}$$

where $\mathbf{V}_t$ is a speed factor given the momentum coefficient μ, $\mathbf{W}_t$ denotes the filters coefficients and η the learning rate (such as in equation 2.14).

We performed trainings on Caffe Framework (Jia et al., 2014) using NVIDIA Digits (Yeager et al., 2015) in Ubuntu 16.04.3 LTS. Our main hardware platform is a 20×Intel i7-6950X CPU@3.0GHz, 128GB RAM and 3×NVidia GeForce GPU TITAN-X 32Gb.

4.4 Results

In this section, we briefly describe the datasets and then we analyze results on both tasks; i) for architectural style classification and ii) large-scale recognition on ImageNet dataset.

4.4.1 Databases

In the experiments of this chapter we consider two datasets; the MexCulture buildings dataset (presented in section 3.2.3) and a subset of ImageNet dataset.

(a)

(b)

Figure 4.4: Samples in MexCulture buildings and ImageNet datasets.

The MexCulture buildings dataset, is composed of 16,700 labeled images, categorized into 4 categories (4,175 images per class), a discriminator class called "other" and three more for style recognition; pre-hispanic, colonial and modern. Each image in the dataset is weakly annotated with a single label and no bounding boxes were considered. This is the main limitation of applying SS and MCG on this dataset, due to this, we are not able to identify positive and negative samples during regions cropping. Some examples of this dataset are presented in 4.4 a.

For the large-scale experiment, we selected a subset of images from the ImageNet database consisting of 569 annotated categories for the object recognition task, i.e., multi-object recognition. In total, the database contains 569,440 images, with approximately –in average– 598 images per category (Russakovsky et al., 2015).

In the case of the ImageNet database, shown in Fig. 4.4 b, we apply all content selection methods described in the section 4.2. We use the same labeling strategy for all the methods in this subset, based on the Jaccard coefficient. In the case of exahustive ROI generation methods, such as SS and MCG, we only retain 3 proposals, whose provide the maximum Jaccard coefficient. Otherwise, if we take more ROIs, the number of samples would increase over 9M. This is too large for the hardware equipment we have and processing would take several days or even

Table 4.1: ROIs extracted using data selection methods from ImageNet dataset.

Method	μ	σ	min	max	Σ
GC	1	0	1	1	369,440
SB	1	0	0	1	341,365
SS	6.16	3.65	0	70	371,204
MCG	34.44	34.64	1	1,012	819,766

Table 4.2: Split of the ImageNet dataset into training, validation and test.

Method	Training	Validation	Test
GC	246,293	52,561	53,365
SB	237,294	52,561	53,365
SS	2,072,744	55,335	56,776
MCG	4,585,159	122,219	124,169

Table 4.3: Regions of interest computation time (in seconds, except Σ).

Method	μ	σ	min	max	Σ
GC	0.0050	0.0020	0.0014	0.16	00:05:20
SB	0.4477	0.1087	0.1508	14.84	01:23:46
SS	6.4179	22.1401	0.7267	3,967.24	1 day, 17:41:25
MCG	32.6324	23.6203	3.9312	1,419.10	7 days, 00:36:01

weeks. The statistics on data selection methods is presented in Table 4.1 and the split –in training, validation and test– of the ImageNet dataset is shown in Table 4.2.

Execution time is given in Table 4.3. This time is approximated using 30 parallel processes in 6 servers (CPU-only implementations). Note that, specially the MCG method, requires a tremendous computational time.

4.4.2 Classification results

In this section, we first present the results with respect to the classification task in the MexCulture buildings dataset and second, the results of the large scale classification task with the ImageNet database.

Saliency-based cropping parameterization

The SB data selection method is dependent on a single parameter; the threshold used during the binarization step, denoted by ϕ.

Table 4.4: Models training varying threshold ϕ.

Model	Dataset Aug.	ϕ	Val. Acc.	Test Acc.
SB-GoogLeNet	Rot+Flip	0.1	96.67	80.20
SB-GoogLeNet	Rot+Flip	0.2	**96.85**	**80.50**
SB-GoogLeNet	Rot+Flip	0.3	96.65	80.03

Given the nature of the saliency maps, if it is considered as a topographic map of the visual attention sparsely distributed over the captured scene, then there is a concentration of attention where the values will be maximum (the map is normalized in [0, 1]). Then, by choosing a small parameter ϕ, the binarized salient region will be larger than by choosing a larger parameter ϕ. For a graphical reference, see Fig. 4.2.4. Then, in order to select the ϕ parameter, as shown in Table 4.4, we trained three different models with $\phi = \{0.1, 0.2, 0.3\}$, where $\phi = 0.2$ gives better results during training. Note that for further experiments we use the same value, $\phi = 0.2$ in SB data selection method.

Mexican architecture classification results

The first classification task corresponds to style recognition of architectural structures. The results of both reference architectures, AlexNet and GoogLeNet are depicted in Table 4.5. The models are trained under the data selection methodology of GC and SB selection method.

For this experiments we use two approaches to fill image outliers; i) Gaussian noise padding and i) blur padding. As shown in Fig. 4.3, Gaussian noise padding induces high frequency features in the borders between content and the outliers. On the contrary, blur padding adds a border-related content in the padded image, making the rest of the scene apparently even more similar. Nevertheless, models training under blur padding performs worse than others and in some cases the convergence is not guaranteed. The reason we consider more accurate behind this behavior is the following: consider the figure 4.5 as reference, one can notice that the details that describe the scene are highlighted in such a way that the borders help in the delimitation of the useful content. When this content is extrapolated into homogeneous areas such as the sky, these regions of the images become more similar and tend to introduce fake features. This is why the phenomenon of over-fitting and non-convergence is observed in such models. The classification results are presented in Table 4.5 (best scores are in bold).

Once the best padding method is chosen, it is possible to analyze the behavior of the trained models under the different methods of content selection. In table 4.5, specifically Table 4.5 c and Table 4.5 d shows the results of trained models with

Table 4.5: Classification results on Mex-Culture dataset for style recognition.

(a) Blur Padding - Validation

Model	Dataset configuration			
	None	Rot	Flip	Rot+Flip
GC-AlexNet	94,89	95,50	95,75	**95,92**
SB-AlexNet	**89,52**	86,52	86,03	88,47
GC-GoogLeNet	96,57	96,65	**96,67**	96,65
SB-GoogLeNet	94,05	93,05	94,55	**94,75**

(b) Blur Padding - Test

Model	Dataset configuration			
	None	Rot	Flip	Rot+Flip
GC-AlexNet	**47,70**	38,58	38,42	44.67
SB-AlexNet	**45,35**	43,93	44,35	40,25
GC-GoogLeNet	81,90	80,38	**82,42**	82,08
SB-GoogLeNet	43,22	**50,05**	46,48	41,93

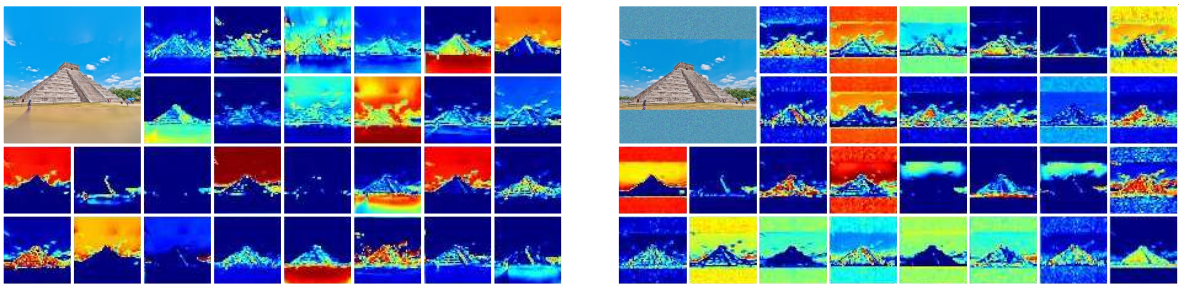
(c) Gaussian Noise Padding - Validation

Model	Dataset configuration			
	None	Rot	Flip	Rot+Flip
GC-AlexNet	96,43	96,16	**96,71**	96,38
SB-AlexNet	95,50	95,95	95,67	**96,15**
GC-GoogLeNet	96,82	**96,90**	96,82	96,57
SB-GoogLeNet	96,57	96,95	**97,12**	96,85

(d) Gaussian Noise Padding - Test

Model	Dataset configuration			
	None	Rot	Flip	Rot+Flip
GC-AlexNet	77,80	79,55	**82,97**	81,38
SB-AlexNet	72,58	75,80	75,72	**77,08**
GC-GoogLeNet	82,87	83,18	83,95	**84,78**
SB-GoogLeNet	79,62	79,55	**81,30**	80,50

Bold: maximum accuracy for each model



(a) (b)

Figure 4.5: Normalized SB-GoogLeNet output features of the first convolution layer: (a) model trained with blur padded images and (b) model trained with Gaussian noise padded images. Input image: top-left.

both cropping methods GC and SB. Note that different configurations of the dataset are possible by combining rotations and mirror flips affine transformations. Then, we can see that during validation, SB selection method outperforms others in both architectures. In AlexNet with 96.15% (with Rot+Flip dataset configuration) and GoogLeNet with 97.12% (using only Flip configuration). During testing, the victory is for GC method, where we only use 700 images that might not be representative to evaluate models.

Now, if we have a look to Table 4.6, in terms of recall and precision, using the best model SB-GoogLeNet wht Flip dataset configuration, ensures a good balance per class, presented in columns 5 and 6. Note that, in a target application point of view, colonial and modern styles are the best classified categories.

If we go further to draw some conclusions from this first experience, classification

Table 4.6: Accuracy, precision and recall of GoogLeNet models on Mex-Culture test dataset.

Models/Classes	SB-GoogLeNet								GC-GoogLeNet							
	None		Rot		Flip		Rot+Flip		None		Rot		Flip		Rot+Flip	
	P	R	P	R	P	R	P	R	P	R	P	R	P	R	P	R
Prehispanic	76,54	85,80	78,30	84,80	79,96	85,80	82,00	82,90	75,55	88,70	77,41	92,20	76,91	88,60	77,64	92,00
Colonial	83,86	89,40	87,68	85,40	87,62	85,60	93,91	78,60	88,54	89,60	87,39	90,80	87,19	91,90	88,29	92,70
Modern	79,87	87,70	76,28	88,10	82,98	87,30	76,10	93,30	91,46	88,90	89,63	87,30	94,91	89,50	91,18	93,00
Other	77,76	55,60	76,02	59,90	74,05	66,50	72,57	67,20	76,37	64,30	78,39	62,40	77,32	65,80	82,42	61,40
Accuracy (%)	79.62		79.95		81.30		80.50		82.87		83.18		83.95		84.74	

P: Precision R: Recall

scores are relatively good, over 80% in our test dataset with both architectures. Results on both methods of data selection are quite similar, then, let us compare the Jaccard coefficient between them, on the basis of GC and SB cropping methods. On the MexCulture dataset, the average Jaccard coefficient between GC and SB methods is 0.625 ± 0.089. These values are relatively high, which means that there is a variance in terms of the regions selected, but they are not so different in terms of shape and location. These values are computed without filtering the content selection where the minimum accepted value of the Jaccard coefficient is 0.7. However, we found some examples in our dataset where the Jaccard coefficient is under 0.3, which is interesting, specifically for video footage where the central bias is not always present.

The computation time to predict labels is quite short. During inference in deployment step, to pass a single image through the network takes around 5ms, which enables batch processing for metadata production. Although training time is not relevant for deployment, the required training time for AlexNet is 72:50:00 and the time required to train GoogLeNet is 19:40:00. Both during 90 epochs and a batch size of 64 images.

ImageNet classification results

The second classification task is on the ImageNet dataset. In this task, all the content selection methods described in the section refsec:dataselectionmethods are used. It is known that ideally, a CNN requires a balanced database, so for these experiments we considered two branches; i) we train the models using only 100 categories out of 569 (balanced) and ii) we train using all the 569 categories as well (unbalanced). The first set, with 100 categories, contains 327 ± 55 images in average and the second set contains 619 ± 884 images in average, totaling 369,440. Note that the standard deviation in the second set is much higher than its mean value.

Analyzing the classification results presented in the Table 4.7, we can conclude that in a larger scale database, with a higher amount of objects to identify, the methods to generate ROIs, provide better results when training CNNs. This is comprehensible since the methods to propose regions in an exhaustive way that

when combined with the content selection based on the GT (and the Jaccard coefficient), what is introduced as training data, are the regions that have been manually annotated, which corresponds to a better selection of the region of interest. However, automatic methods are faster and less expensive computationally.

Table 4.7: Top-5 classification results on ImageNet dataset and training time.

	100 categories			569 categories		
Model	Time (Σ)	Val Acc. (%)	Test Acc. (%)	Time (Σ)	Val Acc. (%)	Test Acc. (%)
GC-GoogLeNet	01:34:00	58.77	55.62	2 days, 12:00:00	82.80	71.12
SB-GoogLeNet	01:33:00	57.75	55.47	3days, 04:00:00	79.13	62.35
SS-GoogLeNet	07:36:00	61.27	60.69	4 days, 15:00:00	77.86	47.01
MCG-GoogLeNet	1 day, 23:00:00	49.06	51.52	6 days, 10:00:00	87.48	71.71

4.5 Conclusions

In this chapter four methods of content selection have been proposed and benchmarked in two different classification tasks; in a weak-labeled dataset of the Mexican culture and in the large-scale ImageNet database.

For both databases, two deep convolutional neural networks from the literature were trained, the AlexNet architecture (Krizhevsky et al., 2012) and the GoogLeNet (Szegedy et al., 2015) architecture. Both, illustrated in the figure 2.10. Concerning to content-selection methods, four are compared; i) Geometrical Cropping (GC), ii) Saliency-Based (SB), iii) Selective Search (SS) and iv) Multiscale Combinatorial Grouping (MCG).

We have shown that the approach of selecting content based on visual attention is interesting in the case where there is no clue of where the objects are (no bounding boxes are annotated) and only single labels are available to describe the whole scene. Hence, we believe that simple methods such as GC and SB cropping, are useful and represent a balance between complexity and performance. This is the case of Mexican architectural styles classification. Besides, on the side of a large-scale classification task, with the ImageNet-569 and the ImageNet-100 sets, the MCG method outperforms others with a high computational cost, which required several days of image pre-processing on modern hardware.

Chapter 5

Visual saliency integration into deep neural networks

In this chapter, we explore the integration of visual attention models into the DNNs from different perspectives. First, a saliency-based pooling layer is introduced for filtering features from a processing layer, a convolution layer. These features are chosen according to the distribution of saliency in each of the images. In this approach, different saliency prediction models are compared, trained models equipped with the saliency layers show better results than the base model. In addition, different dropping strategies –one of them based on saliency– are integrated into the deep layers of a CNN. In general, the models equipped with saliency-based processing stages present outstanding results.

5.1 Introduction

In recent years, in the computer vision community, visual attention modeling has become a popular research field (Benois-Pineau and Le Callet, 2017). Indeed, in visual attention modeling approaches, we try to emulate human understanding through perception, specially through the visual system. To understand a scene, the human brain does not need to process all the information it receives. Sometimes, decisions are taken immediately, using only the pre-attentive vision as a fast visual input. An example, is the presented by Healy and Enns (Healey and Enns, 1996), where targets are searched without much effort and in a very fast way. In previous research, oppositely, the community tried to use a lot of local features in order to describe visual scenes, mainly through the so-called Bag of Visual Words (BoVW) model (Csurka et al., 2004). Using this approach, it is possible to weight features through a saliency map as a support information for saliency-based sampling and coding (González-Díaz et al., 2016).

With DNNs in the evolving field of DL, outstanding performances have been obtained in different classification tasks. The main capacity of these models is the ability to capture abstract patterns in data related to a given task. With a deeper network, a higher level of abstraction is reached. Indeed, just a few research works have been devoted to track how relevant are pixel values with regard to the task by back-tracing activations in the CNN (Zeiler and Fergus, 2014; Simonyan et al., 2013; Li et al., 2017a). Then, we propose to propagate visual attention forward from early layers to deeper layers, i.e., end-to-end in features processing stages in CNNs. The propagated visual attention through the CNN is built upon human gaze fixations. The visual attention of the human in this task is top-down, mainly characterized by the search of the main characteristics that can describe an architectural structure and the relation with its context, while Harell's model is a bottom-up attention model. The GBVS is composed by low-level features accordingly to the FIT of Treisman and Gelade (Treisman and Gelade, 1980).

The main idea of integrating visual attention models into CNNs is that specifically top-down attention models can improve performance on specific classification tasks. However, top-down attention modeling depends entirely on the task to be solved and the interest of the observer (Buso et al., 2015). That means that for any task we want to solve a top-down saliency model should be developed. Then, speaking about architectural content, a specific saliency model is developed by Guissous and Gouet-Brunet (Guissous and Gouet-Brunet, 2017) where the approach is based on the analysis of local distribution of orientations to predict saliency maps of structured content. Later, for example, Guissous and Gouet-Brunet, integrated their saliency prediction approach in a Content-Based Image Retrieval (CBIR) system, specifically in features filtering in a Fully Convolutional Network (FCN) (Guissous and Gouet-Brunet, 2019). Nevertheless, many other features

influence focus on scene, such as the central bias hypothesis formulated by Buswell (Buswell, 1935) and other interacting elements in scene.

In order to integrate attention in CNNs, it is necessary to build a visual attention model on the basis of real gaze fixations of a sufficiently large group of observers in a psycho-visual experience. Furthermore, given that the annotation could be time-consuming and tedious for observers, we conduct this experiment using a sufficiently large dataset and a bootstrap strategy to propagate attention between similar images. Then, these resulting saliency maps are used into custom saliency layers deep CNN for the specific task of architectural structures identification.

The prediction of saliency in collections of images with similar content received the name of co-saliency detection, indicating the common and salient objects in different scenes (Jacobs et al., 2010; Li et al., 2013). In the philosophy of co-saliency detection by label propagation we use a rather strong hypothesis; if humans are focused on a specific area of an image to fulfill their semantic visual search task, then in other views of the same content they will be attracted by the same visual elements. Note that such a hypothesis is in agreement with other co-saliency detection methods which use similarity of details or regions in different images of the same or similar objects. Obviously, in real-world conditions this hypothesis is not completely hold due to the distractors, central-bias, large perspective changes, etc. The proposed approach is designed for classification of visual content with CNNs on a large amount of data and it is compared with two reference methods, graph-based visual saliency (Harel et al., 2007) and saliency maps for urban image contents (Guissous and Gouet-Brunet, 2017).

The results and methodology described in this chapter was previously published in the paper "Comparative study of visual saliency maps in the problem of classification of architectural images with Deep CNNs", in "The eighth International Conference on Image Processing Theory, Tools and Applications (IPTA 2018)", held in Xi'an, China. For more details please see Montoya Obeso et al. (Montoya Obeso et al., 2018a).

In this chapter we tackle down the problem of Mexican historical buildings classification into 67 categories. A CNN model is trained using saliency maps for sequential pooling layers. We compare three saliency models for this purpose: i) automatically predicted from general bottom-up approach (Graph-Based Visual Saliency (GBVS)), a ii) content-driven approach focused on buildings (Saliency Maps for Urban Image Contents (SMUIC)) and iii) the subjective saliency maps built upon gaze fixations (Cosaliency Maps (COSAL)).

5.2 Features filtering in CNNs

In the neural networks, the features extracted from the input data are called feature maps and are used to feed consequent layers of the network in a feed-forward way. The processing of these features during training allows the weights of each layer to be adjusted according to the usefulness of these features. A widely used layer in convolutional neural networks is pooling operation. The pooling function summarizes the input given a specific operation, where each output element will represent a neighborhood. More formally, a pooling layer, takes a set of features denoted by $\mathbf{F} \in \mathbb{R}^{C \times W \times H}$ as input and returns a lower dimensional set of features $\mathbf{F}_m \in \mathbb{R}^{C' \times W' \times H'}$. As described in section 2.2.2, the output depends on, the stride, the kernel size and the padding.

5.2.1 Pooling layer

Pooling layers are commonly integrated between features extraction layers in convolutional networks in order to reduce the size of the input and to reduce the amount of trainable parameters in the network.

There are different flavors on pooling units, e.g., the max-pooling, average pooling, L2-norm pooling, stochastic pooling among others. However, max-pooling has provided better performances in convolutional networks. Other pooling strategies, such as L2-norm pooling or average pooling were used but fallen out of favor compared to max-pooling performance. The main difference between units lies on the operation which outputs the representation of the neighborhood being processed.

To illustrate a pooling layer, the max-pooling operation summarizes a neighborhood as,

$$\mathbf{F}_m(x, y) = \underset{(k,s,p)}{maxpool}(\mathbf{F}(\bar{x}, \bar{y})), \tag{5.1}$$

where (x, y) are the coordinates and $(\bar{x}, \bar{y})$ the neighborhood to be summarized. Coordinates (x, y) are sampled in the input feature maps, based on stride parameter s, with a kernel size k and padding p.

5.2.2 Saliency-based pooling layer

As feature selection method in pooling layers, we propose to use a pruning process of visual data for high saliency areas in feature maps. This method takes inspiration from Vig et al. (Vig et al., 2012) and González-Díaz et al. (González-Díaz et al., 2016). We select features from feature maps based on a random sampling process guided by the Cummulative Weibull Distribution Function (CWDF) and the saliency map of the

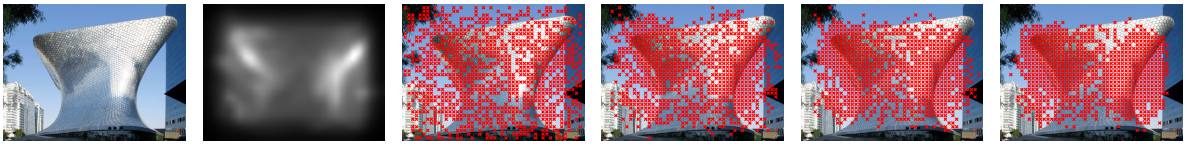


Figure 5.1: Process of random sampling for different k values. Selected features for max-pooling are denoted in red, otherwise avgpooling is applied. The source image is shown instead a features map in order to identify relevant sampled regions given the saliency map.

input image. The CWDF is better for this purpose since it allows a parameterization in which it resembles the Gaussian distribution.

Accordingly to Vig et al. (Vig et al., 2012) we consider saliency maps to follow a CWDF,

$$P(r; k, \lambda) = 1 - e^{-(\frac{r}{\lambda})^k}, \tag{5.2}$$

and a random variable $r \in [0, 1]$ follows the uniform distribution, $k > 0$ gives the distribution shape and $\lambda > 0$ the scale factor of the Weibull distribution.

Then, with the inverse of the CWDF we obtain a response,

$$P(r; k, \lambda) = \lambda \sqrt[k]{-log(1 - r)}, \tag{5.3}$$

based on random variable r. Thus, we consider the input features map $\mathbf{F}$ (coming from a previous layer) and the saliency map as $\mathbf{S}_m$ to follow the rule designed to prune regions in function of $P(r)$,

$$\mathbf{F}_m(x, y) = \begin{cases} \underset{(k,s,p)}{maxpool}(\mathbf{F}(\bar{x}, \bar{y})) & \text{if } P(r) \leq S_m(x, y), \\ \underset{(k,s,p)}{avgpool}(\mathbf{F}(\bar{x}, \bar{y})) & \text{otherwise,} \end{cases} \tag{5.4}$$

where, as well as in Eq. 5.1, $(\bar{x}, \bar{y})$ denotes the neighborhood around regularly sampled coordinates (x, y). Hence, in highly salient regions of the input features map the baseline max-pooling is applied, in other regions average pooling will propagate forward non-salient features from input $\mathbf{F}$.

As shown in Fig. 5.1, k controls how dense is max-pooling applied over salient regions. Given lower values of k the CWDF takes a shape where low values in saliency map have high probabilities to remain during the sampling process. The shape of the CWDF with a high value of k gives a balance on pruning between low and high values of saliency. Besides, λ limits the minimum area to be processed, by shifting the CWDF within the range [0,1]. After analyzing several images, we see that for a low k value max-pooling is sparsely applied, see Fig. 5.1. Thus, for a better selection of features to pool, we fixed $\lambda = 0.3$ and $k = 3.5$.

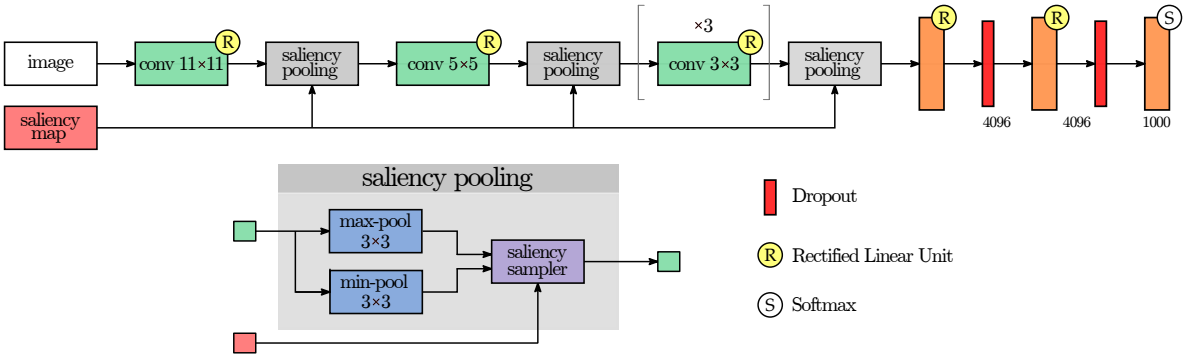


Figure 5.2: Integration of visual attention in a Convolutional Neural Network. The saliency map is resized to match incoming feature maps shape in saliency sampler block. The main operation in this block is described in equation 5.4.

Saliency maps are introduced in all pooling layers of the network, as illustrated in Fig. 5.2. Then, taking the saliency map from the input layer of the network, we resize it by conventional Gaussian low-pass filtering to match the size of feature maps at different stages of the network. The resize is performed in the saliency sampler block in saliency pooling layers in Fig. 5.2. The scale parameter σ of Gaussian filter in each layer is computed accordingly to the filter size $l = \rho - 1$, with ρ the stride hyperparameter. The saliency maps are only included inside pooling layers.

5.3 Regularization methods

A central problem in DL is that models should perform well over both sets of data, training and validation. Many strategies we currently use in DL have been proposed to reduce this error. According to Goodfellow et al. (Goodfellow et al., 2016), the term regularization is defined as "any modification we make to a learning algorithm that is intended to reduce its generalization error but not its training error" (generalization error is computed on unseen data, i.e., test data). There are many regularization strategies. Some of them, establish restrictions to the parameters of the algorithm, while others integrate additional elements in the objective functions, usually, related to the recognition task, with a prior knowledge of the data. If these strategies are chosen correctly, they can lead models to find better performances on the test set.

In the literature, it is well-known that dropout (Srivastava et al., 2014) provides an inexpensive and powerful method to regularize a broad family of models. For instance, dropout can be thought as a bagging method for practical ensembles of many different versions of deep models. It is common to make ensemble models and gain in performances, such as presented by Szegedy et al. (Szegedy et al., 2015). Bagging different trained versions of the model could be impractical, and

computationally expensive. Dropout, provides an inexpensive strategy to evaluate a bagged ensemble of many neural networks. Although the literature on the subject is extensive, in this section we address some regularization techniques based on saliency maps but inspired by dropout.

5.3.1 Dropout

According to Krizhevsky et al. (Krizhevsky et al., 2012) and Szegedy et al. (Srivastava et al., 2014), dropout allows to ignore neurons during training. These neurons, which have been randomly chosen, are not considered during the forward propagation, shutting down sections of the network, reducing the response in activations and generating different versions of the architecture each time. Thus, using only one base architecture, but having different configurations in each iteration, we can get an ensemble of different models.

Accordingly to Srivastava et al. (Srivastava et al., 2014), the process is to multiply hidden activations by Bernoulli random variables where each variable has probability p of being 1,

$$P(k;p) = \begin{cases} p & \text{if } k = 1, \\ 1-p & \text{if } k = 0. \end{cases} \tag{5.5}$$

In this case, a given neuron is turned off or not depending on the dropping probability given by p. Then, we take $\mathbf{y}$ from layer l as input for the next layer $l+1$ to to drop activations before feeding next layer,

$$r_i^l \sim P(k;p), \tag{5.6}$$
$$\tilde{\mathbf{y}}^{(l)} = \mathbf{r}^{(l)} * \mathbf{y}^{(l)}, \tag{5.7}$$

then, we feed the next layer with $\tilde{\mathbf{y}}$ as follows,

$$\mathbf{z}^{(l+1)} = \mathbf{w}_i^{(l+1)} \tilde{\mathbf{y}}^{(l)} + b_i^{(l+1)}, \tag{5.8}$$
$$\mathbf{y}^{(l+1)} = f\left(z_i^{(l+1)}\right). \tag{5.9}$$

This should make neurons more robust by creating useful features without relying on other neurons and correct their mistakes. In a regular neural network, the update received for each parameter tells it how to change to reduce the loss function based on what all other units are doing, forcing neurons to create strong co-adaptations which do not generalize to unseen data and leading the model to over-fitting. As illustrated in Fig. 5.3, dropout layers are integrated between

fully-connected layers.

5.3.2 Spatial dropout

In a similar way that regular dropout is integrated, we add a spatial dropout before fully connected layers, randomly selecting neurons, where each one will rely on threshold r. Instead a Bernoulli distribution we use the uniform distribution given a random variable n as follows,

$$P(n) = \begin{cases} 1 & \text{for } n > r, \\ 0 & \text{otherwise}, \end{cases} \tag{5.10}$$

where n is paremeterized with $a = 0$ and $b = 1$. The threshold r defines if the neuron is active in activations map A_m when $P(a, b; n) > r$ or inactive if $P(a, b; n) \leq r$, as follows,

$$A_m(x, y) = \begin{cases} A_m(x, y), & \text{if } P(a, b; n) > r, \\ 0, & \text{otherwise}, \end{cases} \tag{5.11}$$

here, A_m denotes a single activations map is being processed at each coordinates (x, y). In a CNN, dropping is applied to every single map along channel axis in activation maps $\mathbf{A}_m$.

By dropping neurons based on this rule we keep under control of the percentage of dropped neurons by setting r. In our experiments, we experimentally found that $r = 0.2$, dropping 20% of activations in average, leads to better performances. As shown in Fig. 5.3, this layer is only integrated before fully-connected layers.

5.3.3 Saliency-based spatial dropout

The saliency-based spatial dropout is a variant of the saliency-based pooling strategy we previously presented in section 5.2.2. The main difference resides in the features selection strategy. Both are based on the same intuition, we aim to randomly select relevant features based on saliency maps. Once $P(r)$ is computed, such as in equation 5.3, we drop activations in $\mathbf{A}_m$ as follows,

$$A_m(x, y) = \begin{cases} A_m(x, y), & \text{if } P(r) < S_m(x, y), \\ 0, & \text{otherwise}, \end{cases} \tag{5.12}$$

where, such as in section 5.3.2, A_m denotes a single activations map. This process is applied to all channels in a collection of activation maps $\mathbf{A}_m$. In contrast with spatial dropping strategy, presented in previous section, some activations in "relevant

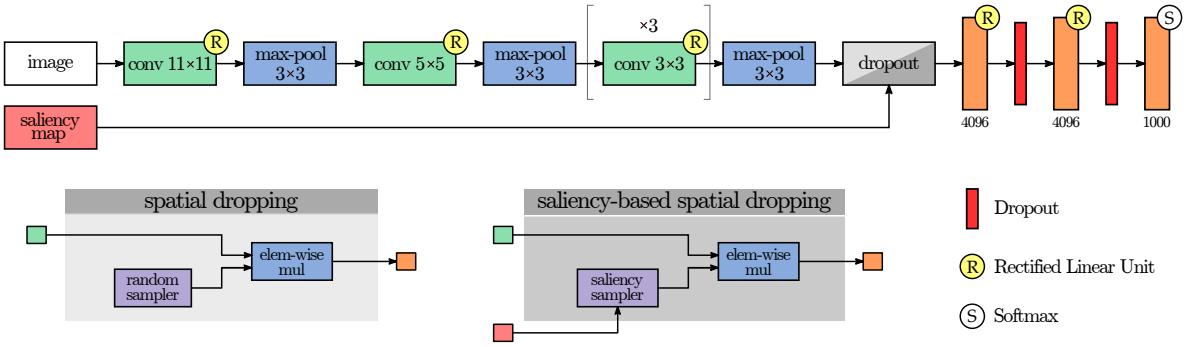


Figure 5.3: Dropping strategies integration in a Convolutional Neural Network. In the position of gray dropping block, spatial strategies are integrated; spatial dropping and saliency-based spatial dropping. Random sampler in spatial dropping block relies on the rule presented in equation 5.10 and saliency sampler in the saliency-based spatial dropping block on the rule in equation 5.12.

regions" survive based on the high probability given the saliency map while others located ni non-salient regions are dropped.

5.3.4 Dropping strategies integration in a CNN

Each of the dropping strategies is integrated into the final chain of the CNN. Dropping strategies for spatial dropping –spatial dropping and saliency-based spatial dropping– are integrated after the last pooling layer, while common dropout strategies, are integrated between fully-connected layers. The main configurations we consider are presented in Table 5.1 and illustrated in Fig. 5.3.

5.4 Visual attention models

In this work, the input images are combined with saliency maps as support data during processing of features. We compare the influence during models training using our task-driven generated saliency maps with two reference methods: i) the GBVS and ii) the method for the generation of SMUIC.

Our proposal is called COSAL. Where the maps are derived from a psycho-visual experiment and the propagation of gaze fixations through homography estimation between similar images from the same instance. More details are given in the following sections.

The models GBVS and SMUIC are models to predict saliency maps directly from any input image. We have computed the saliency maps for all the samples in the dataset.

Table 5.1: Architectures configuration. The variants of dropping strategies are denoted in **bold**. Parameters for each layer are denoted following its name, with kernel/window size and number of output channels, respectively, in convolution and max-pooling layers. In fully-connected layers the number of units follows each name.

Architecture Configuration					
#1	#2	#3	#4	#5	#6
Input RGB image (256x256)					
conv11/96	conv11/96	conv11/96	conv11/96	conv11/96	conv11/96
maxpool					
conv5/256	conv5/256	conv5/256	conv5/256	conv5/256	conv5/256
maxpool					
conv3/384	conv3/384	conv3/384	conv3/384	conv3/384	conv3/384
conv3/384	conv3/384	conv3/384	conv3/384	conv3/384	conv3/384
conv3/384	conv3/384	conv3/384	conv3/384	conv3/384	conv3/384
maxpool					
		spdropout	**spdropout**	**saldropout**	**saldropout**
fully-connected4096					
	dropout		**dropout**		**dropout**
fully-connected4096					
	dropout		**dropout**		**dropout**
fully-connected67					
softmax					

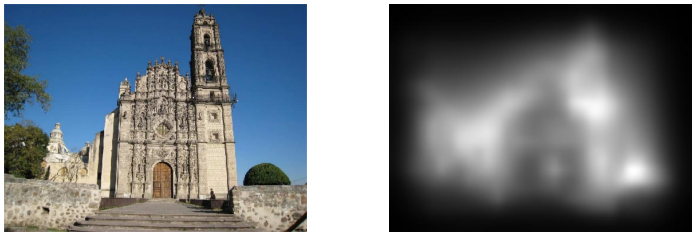
5.4.1 Graph-based visual saliency

The GBVS, is presented in section 2.4.2. This approach is a plausible biologically inspired method which produces a normalized bottom-up saliency map by highlighting a handful of significant locations where the image is informative based of the criterion of human fixations (Harel et al., 2007).

The method consists on two steps: i) uses a novel application of ideas from graph theory to concentrate mass on activation maps, and ii) to form activation maps from raw features. The authors, present this method as a simple model to predict continuous activation maps on natural images. An example of a predicted saliency map is shown in Fig. 5.4 b.

5.4.2 Saliency maps for urban image contents

Accordingly to Guissous et al. (Guissous and Gouet-Brunet, 2017), the Saliency Maps for Urban Image Contents (SMUIC), is an approach based on the analysis of local distribution of edges orientation. It is focused to predict salient images



(a) (b)

Figure 5.4: A sample of Graph-Based Visual Saliency predicted map. In (a) the source image and in (b) the GBVS map is presented.

corresponding to facades of buildings, monuments or street-view images, i.e., structured content.

This method, illustrated in Fig. 5.5, consists on three main steps: i) detection of edges, ii) local dominant orientation estimation and iii) saliency prediction.

(a) (b) (c)

Figure 5.5: Saliency Maps for Urban Image Contents. (a) Source image. (b) Source image and merged saliency maps, Red: One Dominant Direction (ODD); Green: Two Dominant Directions (TDD); Blue: Multi Directions (MD). (c) ODD+TDD saliency map over source image. Source: (Montoya Obeso et al., 2018a). Thanks to Kamel Guissous for the illustrations.

The prediction process is pixel-size designed. First, the Line Segment Detector (LSD) algorithm (Von Gioi et al., 2010) is used to detect line segments in the image. Then, for each pixel, a circular window (with radius r) is considered as region of interest to identify line segments and their orientations. After, orientations distribution is analyzed through a 315 bins histogram at each pixel location and they performs a circular convolution of the histogram and a rectangular window of $\frac{\pi}{8}$ width.

Finally, saliency prediction is on the basis of the local distribution of segment pixel orientations. The authors focused on three particular types of distribution: unimodal, bimodal and multi-modal, corresponding to one peak, two peaks and the absence of peaks on the convolution curve, respectively. The authors, found that these three configurations contain most relevant information about structured content in images.

In the original paper, each type of distribution corresponds to one saliency map, named ODD, TDD and MD. If we consider Fig. 5.5, one can see that all obtained saliency maps are semantically correlated in urban imagery; i) ODD with architectural structures, ii) TDD with windows and iii) MD with vegetation.

The ODD and TDD saliency maps combinations provides a higher performance in the task of image retrieval, performed by the authors. Therefore, this joint is used in our experiments to predict a continuous saliency map using the ration of two first dominants peaks as saliency factor. Then, the normalization and smoothing by applying a Gaussian filter generates a saliency map. This adaptation was implemented by original authors in (Montoya Obeso et al., 2018a). We call obtained saliency maps SMUIC, in the follow-up of this chapter.

5.5 Psycho-visual experiment

The psycho-visual experiment is conducted to record visual attention for the specific task of identifying representative characteristics of Mexican architectural structures. We use a High-speed Cambridge Eye-tracker system (250Hz) to record gaze from participants to generate gaze fixation density maps, our ground truth. The goal was to identify relevant regions on images when participants executed a visual task of recognition of only architectural styles of Mexican culture heritage into three main categories; Prehispanic, Colonial and Modern. For each category, we gather images with different views of different architectural structures. During this experiment, we randomly selected two images per category from the Saliency MexCulure Buildings Dataset (SMCBD).

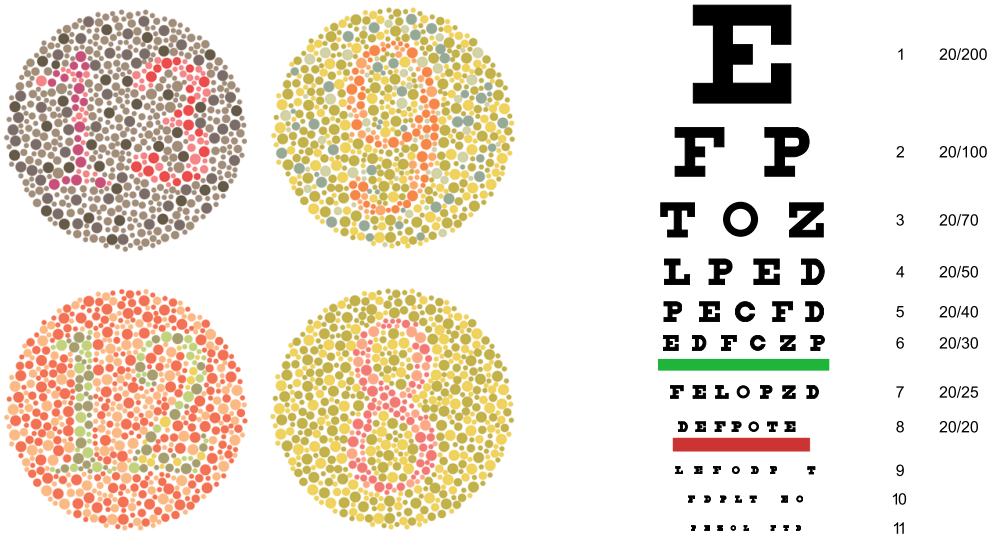
5.5.1 Experiment protocol

During the experiment we follow several steps to filter participants before recordings. First, we perform visual tests such as Ishihara test (4 patterns) and Snellen chart, shown in Fig. 5.6. If the participant is able to complete them, is accepted to continue. Next, experiment instructions are given as a lecture of 2 minutes with graphic examples about architectural styles and the eye-tracker setup. Once all instructions are clear, we proceed with the eye-tracker calibration and gaze recordings. We split gaze recordings in two parts, to give participants some minutes to rest. We ran this experience with the collaboration of 23 participants, with 23 years old in average. Each session takes around 30 minutes. The protocol is illustrated in Table 5.2

Table 5.2: Psycho-visual experiment protocol

Step	Activity	Description	Time
1	Ishihara test	Display 4 patterns, the participant should identify at least 3 patterns to continue	1 min
2	Snellen chart	Perform Snellen chart test, participant should be able to reach level 20/50	1 min
3	Instructions	Participant reads instructions	2 min
4	System calibration	During calibration, the participant should see directly some points on the screen	1 min
5	Gaze recordings*	We display images and record gaze as raw data	20 min

*Each image is displayed for 3 seconds, then a gray screen is shown to reset their attention for 1 second.



(a) (b)

Figure 5.6: Visual tests performed during the psycho-visual experiment. (a) Ishihara test and (b) Snellen chart. Image credits: Wikimedia Commons.

5.5.2 Gaze fixations propagation

Gaze fixation annotation of images through a psycho-visual experience is time-consuming, tedious and tiring for participants. Therefore, our core idea is to propagate fixations between similar images, based on the following: if a person is looking at particular characteristics of an image to perform a specific task, such as object identification, then this person will most probably seek for similar characteristics in other image of the same object of interest, even from different perspectives. This assumption has the main limitation about high perspective changes due to that the detection range in terms of perspectives changes is unknown.

Then, given two sets, a set of pairs of reference images and their fixations $\mathbf{R}_c = [(I_1, \mathbf{M}_1), (I_2, \mathbf{M}_2)]$ and a target set $\mathbf{T}_c = [I_1, ..., I_k]$, where $c = [1, ...142]$ denotes the class number and k the number of samples in target set. Note that, we only have fixations for reference images.

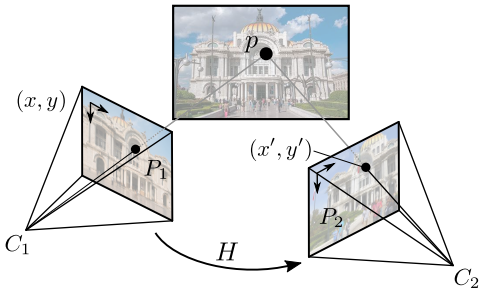
As illustrated in Fig. 5.7, the proposed solution to propagate saliency maps to similar images is to use Scale Invariant Features Transform (SIFT) keypoints

Figure 5.7: Saliency map generation through homography estimation. Green polygon over target image shows the transformation given estimated homography when the polygon condition is satisfied. The red lines depicts keypoints matching. For saliency maps over images: red means more attention and blue less attention (jet colormap is used).

to compute the homography given two images, reference and target images, and transform coordinates of reference fixations to build a new saliency map for the target image. The process to propagate saliency maps through SIFT keypoints matching is as follows,

1. **Features matching**: given a set of keypoints for each image, a K-NN ($k = 2$) algorithm is used to filter up a pair of high related matching keypoints. Then, closest point is selected.
2. **Homography estimation:** the homography estimation gives the perspective transformation between matching SIFT keypoints. The estimation is performed using a set of matching keypoints. Once we compute H as the homography matrix (see Fig. 5.8), we perform a geometrical test by transforming only the corners of the reference image into the target image plane. Testing the internal angles of the resulting projected polygon, we verify if H estimated is correct, when the polygon is convex.
3. **Fixations transformation:** we use estimated H to transform gaze fixations coordinates from reference to target image and we finally build up a new saliency map upon new gaze fixations coordinates.

We perform a geometry test of the resulting projection of reference frame corners on target image plane. Due to that matching keypoints are note always well associated when some context changes appear, such as scaling and contrast changes between similar images. Then the estimated homography is not always correct. Hence, projecting original frame corner coordinates into target plane, we should always obtain a convex polygon, i.e., each intern angle is under 180°. Such as the one drawn in green in Fig. 5.9. Of course, some small variations will appear



$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = H \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Figure 5.8: Homography estimation. H is estimated given a set of matching keypoints $((x, y), (x', y'))$ from planes captured by cameras C_1 and C_2.

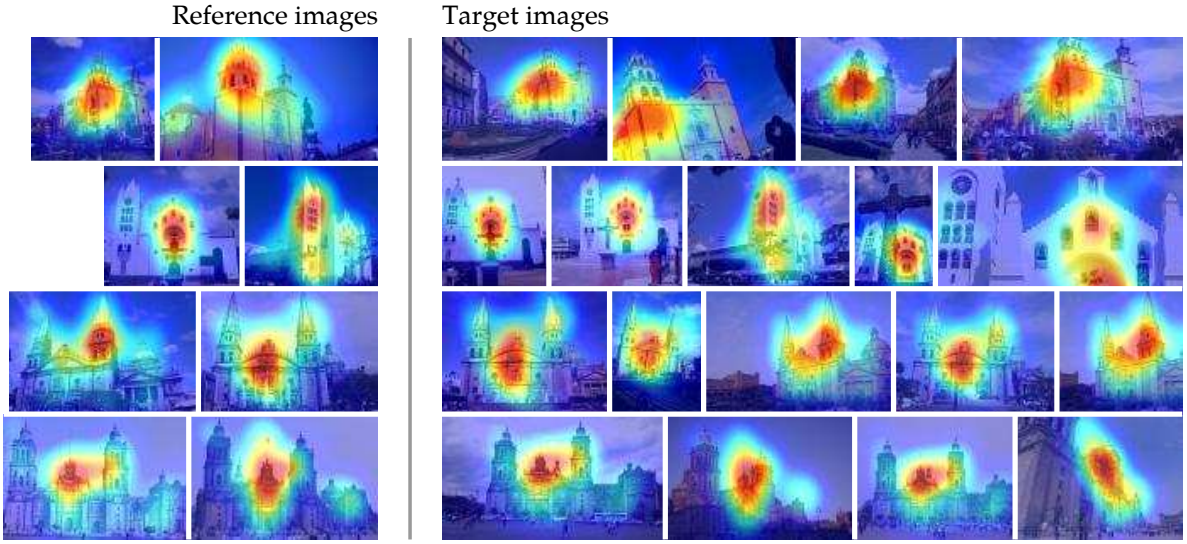


Figure 5.9: Gaze fixations propagation through homography estimation results. Read this figure row by row, where left column shows subjective saliency maps over reference images and right column shows projected saliency maps over target images. Saliency maps over images: red, more attention and blue, less attention.

due to the homography estimation process is not deterministic. To handle this variations we add an internal angle hysteresis ($\beta = 20°$) to allow the detection of a limited number of different perspectives between samples. In the follow-up of this chapter, the obtained maps upon gaze fixations and the maps generated after fixations propagation, are denoted as COSAL. Several samples of propagating fixations between similar images are presented in Fig. 5.9.

The propagation of saliency maps is done iteratively taking previous results as reference data. This scheme defines the system as causal where we also propagate saliency maps from target images. The iterative process consists in two phases; i) propagate only reference saliency maps to target images and ii) propagate already propagated target saliency maps to remaining target images. More formally, the idea is to move images and projected fixations from $\mathbf{T}_c$ to a new set $\hat{\mathbf{T}}_c$ when we find a

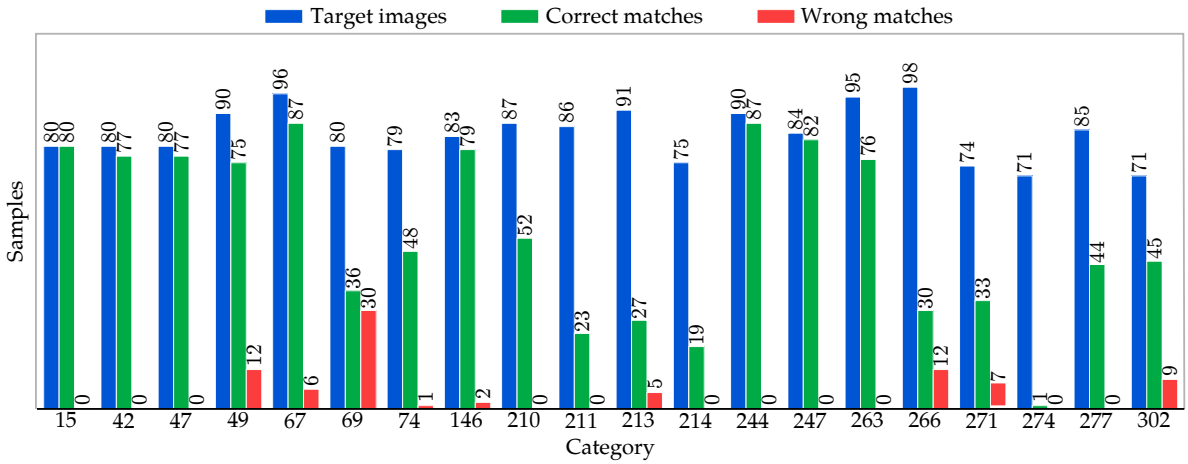


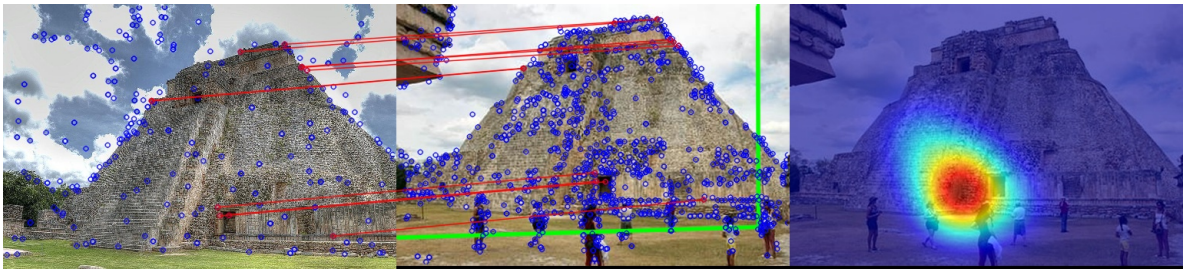
Figure 5.10: Propagated saliency maps distribution rates, only for 20 classes of 142. Where initial $\mathbf{T}_c$ contains 1,675 images, we found 1,108 correct projections, 54 wrong projections and only 513 images remain without projection results.

correct transformation, changing the matching criteria iteratively, i.e., β parameter, varying internal angles criteria allows rather small but useful perspective changes detection.

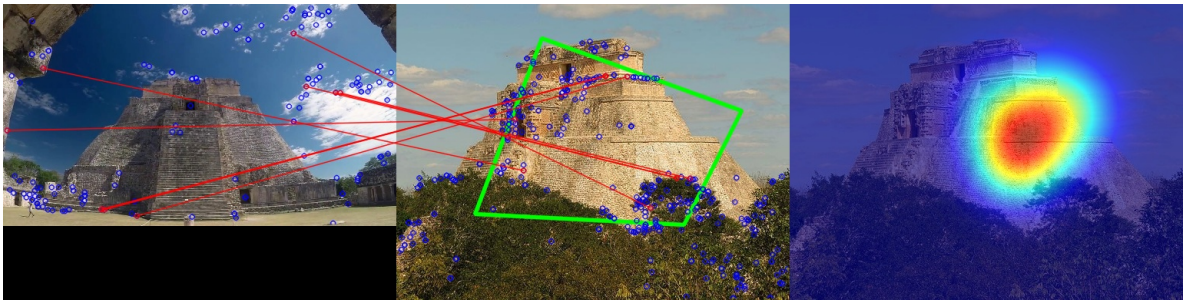
The iterative process we designed to apply our method on target set is as follows:

1. **Phase 1**, $\mathbf{R}_c$ versus $\mathbf{T}_c$: we vary the angle test range $\alpha = 90° \pm \delta°$ where $\delta \in \{10, 15, 20\}$, resulting in three iterations for this phase. With $\delta = 10$ we force the method to obtain very similar matches between reference and target images, when $\delta = 20$ the method is more permissive, more perspective changes are allowed. This phase allows to obtain strong matches to use them in the next phase. All correct projected fixations (k samples) are stored in $\hat{\mathbf{T}}_c$ as a pair $(I_k, \hat{\mathbf{M}}_k)$.
2. **Phase 2**, $\mathbf{R}_c + \hat{\mathbf{T}}_c$ versus $\mathbf{T}_c$: the same process of varying α is performed in this phase but the previously projected fixations are used as reference set. The disadvantage of taking $\hat{\mathbf{T}}_c$ as part of reference set is that is that fixations estimated given a incorrect homography estimation will produce a chain of incorrectly projected fixations.

In order to compute statistics abut how the method is working on our dataset, we randomly selected 20 categories and classified correct and incorrect fixations projections, using the visual scheme presented in Fig. 5.7. The current criteria does take into account that for projections, the aspect ratio from source image should be similar. Some projections are taken as valid even when the projection is a convex polygon but the transformation matrix is incorrect. However, as shown in Fig. 5.10 we consider that the number of correct projected fixations remains high; 1,162 images



(a)



(b)

Figure 5.11: Two cases of wrong results with a successful validation of the transformed corners of reference frame. In (a), keypoints matching and the estimated homography are correct, but the previous transformation of the reference image was incorrect. Then, the resulting saliency map is inaccurate. In (b) we show an example when the polygon is valid but matching keypoints are incorrect, then the result is not correct. In this case, it is remarkable that the perspective of the target architectural structure is quite different to the reference and keypoints matching is incorrect, the method is limited for this cases.

of 1,675 (69.37%), where only 54 matches are wrong (4.6%) (see two examples in Fig. 5.11) and 513 images (26.03%) remain without results.

5.6 Results

5.6.1 Specific buildings dataset

From propagation results using the method described before, we selected only 67 categories from 142. We found that for some categories the number of samples is very low. Then, we selected a subset with 5,327 images, containing 79.5±17.17 samples per class. The dataset is randomly split as follows; 70% for training, 20% for validation and 10% for test. Then, in order to increase the number of samples in the dataset and prevent over-fitting we perform data augmentation, such as in chapter 4, section 4.3.1. We introduced rotations and mirror flips, adding 7 image variations per sample. Data augmentation is only applied to training set. In this dataset, each RGB image is associated with different saliency maps; i) GBVS, ii) SMUIC and iii)

Table 5.3: Models validation and testing results

Model	Saliency Maps	Validation Acc.	Testing Acc.
Baseline	None	73.45%	73.74%
GBVS	Harel et al., 2007	77.99%	83.85±0.34%
SMUIC	Guissous and Gouet-Brunet, 2017	75.62%	80.98±0.39%
COSAL	Our	83.22%	88.80±0.40%

COSAL, and the label for specific architectural recognition.

5.6.2 Pooling strategies

The classification task in this chapter is tackled down with the well-known architecture AlexNet (Krizhevsky et al., 2012). The integration of visual attention saliency through saliency pooling layers in the CNN is presented in Fig. 5.2.

The process of pooling with saliency-based pooling layers is not deterministic. Due to this, we tested the models 1,000 times to present the mean accuracy and its standard deviation. We summarize results for validation and testing in Table 5.3.

In terms of accuracy during test phase, the model where we integrate COSAL maps surpasses others with an accuracy of 88.80±0.40%, followed by the model with GBVS maps with an accuracy of 83.85±0.34%, finally the model equipped with SMUIC maps reached an accuracy of 80.98±0.39%. All models where we integrated attention maps, surpasses baseline model. The error rates of all models are presented in Fig. 5.12 within the range $[0, 1]$ for 100 epochs. The confusion matrix of best model is presented in Fig. 5.13.

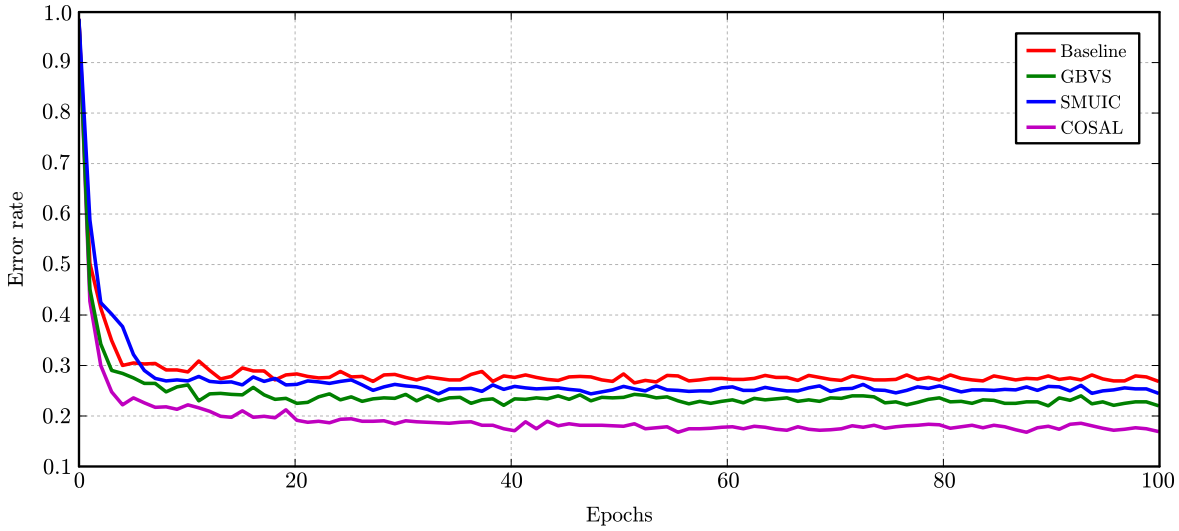


Figure 5.12: Error rate during models validation. COSAL model reached a minimum error of 0.1678 during validation and and error of 0.1120 during testing phase. Low values correspond to a better performance of the model.

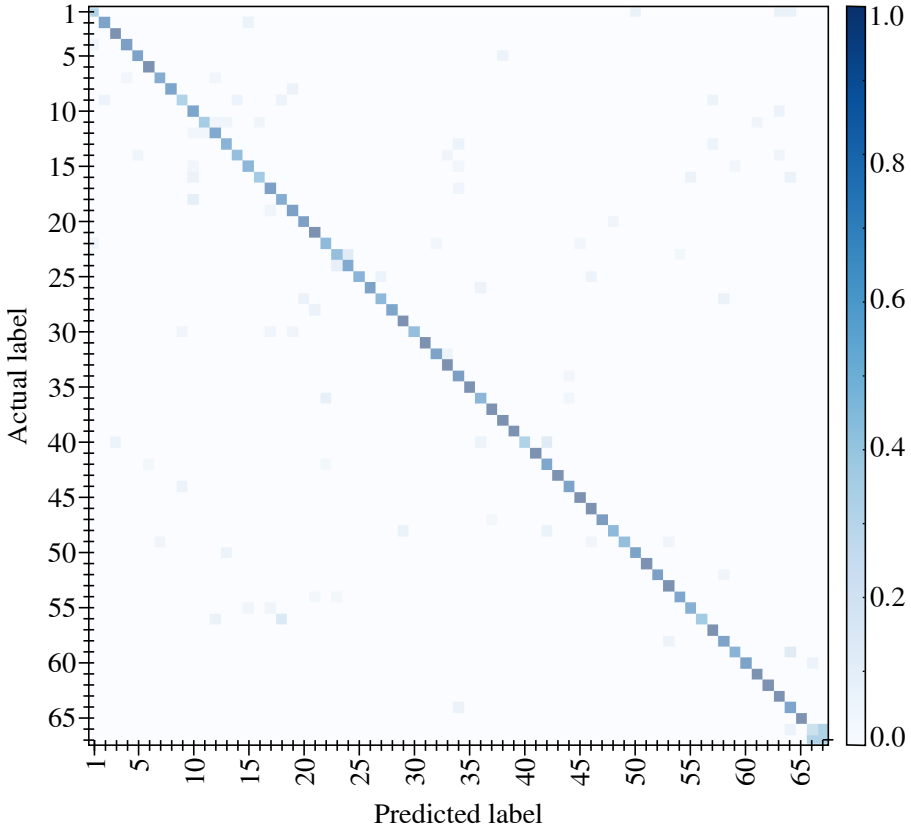


Figure 5.13: COSAL model confusion matrix, classification of 67 categories.

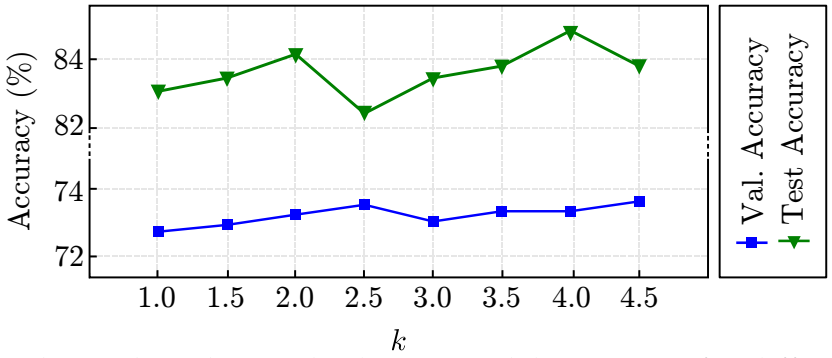


Figure 5.14: Saliency-based spatial saliency models accuracy for different values of k. We considered $k = 4$ as the best model with an accuracy of 84.86% during testing phase.

5.6.3 Dropping strategies

During experiments with dropping activations, we compare dropping strategies presented in section 5.3. Classification results of trained modes are presented in Table 5.4. Besides, they are described in detail in Table 5.1. From top (model #1) to bottom (model #6), we trained models adding, dropout, spatial dropout and saliency-based spatial dropout.

We trained 8 models varying k parameter in saliency-based spatial dropping method, validation and test accuracies are presented in Fig. 5.14. We found that $k = 4$ gives better results during testing, with a model accuracy of 84.86%. Despite the accuracy of the model with $k = 4.5$ is higher during validation the performance during testing is less than the model trained with $k = 4.0$.

In terms of testing accuracy, model #6 outperforms all other models with an accuracy of 84.86%. Generally speaking, deep models present better performances when we integrate saliency-based dropping strategies, such as in the case of model #5 and #6. Besides, by integrating only saliency-based spatial dropping in model #5, the performance is better than models from #1 to #4. Finally, model #6 is over baseline model (#2) by 11.12%.

We can state that all the models where regular dropout is integrated, outperformed their versions without it. In general, regular dropout gives a considerably important advantage to increase accuracy of trained models. The differences during testing are as follows:

- from model #1 to #2: 5.91%,
- from model #3 to #4: 7.30%,
- from model #5 to #6: 5.73%.

Table 5.4: Trained models validation and test accuracy

#	Architecture configuration	Val. Acc. (%)	Test Acc. (%)	Iteration
1	No Dropout	67.82	67.83	6,291
2	Dropout	73.44	73.74	11,883
3	Spatial dropout	68.31	68.87	16,310
4	Spatial dropout + dropout	73.35	76.17	19,106
5	Saliency-based dropout	76.90	79.13	9,320
6	Saliency-based dropout + dropout	82.42	84.86	7,456

5.6.4 Saliency-based backward propagation

In order to evaluate the performance of integrating visual saliency in forward and backward training of a convolutional neural network for image classification, we experiment using the AlexNet architecture (Krizhevsky et al., 2012). As shown in equation 5.13, we use the SGD, with Nesterov's momentum with the proposed modification for saliency propagation.

The attention mechanisms described in section 6.2 are only to process features during forward propagation in deep neural networks. Then, once the loss is computed for a given batch of images, the optimization process is performed such as in other neural networks, by back-propagating errors.

We select relevant features during random sampling process (in forward pass), based on the saliency map denoted by S. Then, considering the selection rule, $\bar{\Psi}_{\lambda^*}(\mathbf{u}) \leq \mathbf{S}$, given in equation 5.4 we obtain a binary map at each layer that is useful to point out to local gradients during backward propagation. Then, we can denote this binary map $\mathbf{S}_s$ as the binary map generated by $\bar{\Psi}_{\lambda^*}(\mathbf{u}) \leq \mathbf{S}$ as in the equation 5.4. The main idea is to limit the propagation of non-relevant local gradients to non-task-relevant regions in images at each layer l. This, can be seen as a hard-attention constrain to local gradients, while max-pool or avg-pool back-propagation strategy in saliency-based pooling layer is softer.

We consider that the visual attention integration into the back propagation process can be introduced locally by the binary mask to manage local gradients at each layer in the network,

$$\mathbf{V}_{i+1}^{l} = \mu \mathbf{V}_{i}^{l} - \eta \mathbf{S}^{l} \nabla \mathcal{L}(\mathbf{W}_{i}^{l} + \mu \mathbf{V}_{i}^{l}) \tag{5.13}$$

$$\mathbf{W}_{i+1}^{l} = \mathbf{W}_{i}^{l} + \mathbf{V}_{i+1}^{l} \tag{5.14}$$

where η denotes the learning rate parameter, $\mathbf{W}_i^l$ the filters coefficients at iteration i, $\mathbf{V}_t$ is a speed factor given the momentum coefficient μ and $\mathbf{S}^l$ is the binary map generated by selection process layer l.

In Table 5.5, one can see that if we compare the best model with the baseline architecture, we obtain an improvement of around 15% on testing set, that is saliency-based forward-only model with 88.80% on test set. Thus, the full

Table 5.5: Results in the task of specific buildings classification. AlexNet architecture (Krizhevsky et al., 2012) is used as the backbone. "Epoch" denotes both, when maximum accuracy and convergence are reached.

Model	Saliency maps?	Val. Acc.	Test Acc.	Epoch
AlexNet	N/A	73.45%	73.74%	49
AlexNet + Saliency-pooling	Forward	83.22%	88.80%	65
AlexNet + Saliency-pooling	Forward + Backward	82.19%	83.50%	48

forward-backward model converges faster than other models, but with a lower accuracy than best model. Although the full forward-backward gave worst results than forward-only model, we believe that it could be interesting to continue the research to identify which configurations in both stages allow a better selection of characteristics during the forward phase and also the selection of local gradients during the back-propagation phase.

5.7 Conclusions

In this chapter, we experimentally compared the influence of top-down and bottom-up attention models during training a convolutional neural network. We specially integrate saliency maps in a custom saliency-based pooling layer where we filter out selected features for deeper layers to identify structures of the Mexican architecture. Besides, in order to populate our saliency annotated reference dataset, we propose a method to propagate fixations between similar images through homography estimation. In this method, we constrain results on estimated homography for each sample in order to detect possible wrong transformations. We increased the set of reference images, containing 134 images (67 categories) up to 5,327, with an average of 79.5$\pm$17.17 images per class and keeping a very low rate of wrong projections.

We consider these classification results promising, trained models achieved a quite high testing accuracy. Note that in Fig. 5.12, our model converges faster than others reaching a minimum error of 0.1678 in the range [0,1]. Our proposal, the model equipped with COSAL maps, reached a maximum accuracy of $88.80 \pm 0.40\%$ outperforming other models. This shows, that despite an explicit knowledge on the structures in images used in the design of SMUIC, the use of a top-down model obtained through a real experience of image visualization and attention recordings, the maps built upon gaze fixations remains as the most promising option to integrate attention models in neural networks for this specific recognition task. In our case was possible given the number of images obtained after automatic propagation of fixations between similar samples in our dataset.

We compared dropping strategies for features filtering in deep layers. A random

activations dropping mechanism and a saliency-based spatial dropping method during training process of CNNs. First, based on the rule of activation pruning described in section 5.3.2, we improved accuracy regarding to the baseline in terms of testing accuracy. The second approach described in section 5.3.3, we brought formal information through saliency maps into as an attention mechanism in the CNN, improving and outperforming baseline and spatial dropping methods.

Finally, it is clear that for our classification task of Mexican architectural structures, accuracies are rather high and better performances are reached when filter up features in early layers with saliency-based pooling layers.

Chapter 6

Visual attention mechanisms in deep neural networks

In this chapter, we analyze the role of visual attention and attention mechanisms when they are combined with common layers into deep neural networks. On one hand, externally generated visual attention maps, given as a result of modeling a top-down process or a psycho-visual experiment of gaze recording using eye-trackers are integrated in special layers to filter out relevant features during forward pass. On the other hand, we also integrate attention mechanisms, which are automatically adapted during training process based on data patterns. Nowadays, visual attention prediction and attention mechanisms are widely studied problems in computer science for multiple tasks, such as object recognition, image classification and saliency prediction. However, most of these approaches aim to extract the features automatically during training in order to interpret better the training data and predict on new unseen data. A priori information is not considered with respect to the type of task to be solved and only temporary extracted features are treated to find out if they are relevant or not. If this knowledge is provided beforehand as input as well as images (i.e. saliency maps), we are able to focus on both, regions of interest and features during model optimization.

6.1 Introduction

In the last years, deep neural networks have gained a lot of attention in computer science community. The performance of deep neural networks to solve complex tasks has increased unexpectedly fast in many domains, e.g., image classification, semantic segmentation or natural language processing. Humans have a great capacity to process signals from different biological sensors, such as sight (Koch et al., 2006). This great capacity to process around 10^8-10^9 bits per second with our visual cortex, is possible because our biological attention mechanism attentively selects relevant parts in a given stimuli, instead processing the whole scene (Eriksen and Hoffman, 1972; Connor et al., 2004).

Several studies have been conducted to mimic visual attention mechanisms in computer vision, dating back to (Itti et al., 1998) and (Harel et al., 2007), these mechanisms generate biologically inspired results, helping on the field to understand and predict visual attention and motivated the integration the human visual system into computer vision models (Borji et al., 2015; Jiang et al., 2015; Li et al., 2017b).

In CNNs, it is well-known that early layers capture patterns such as the orientation of edges or smooth color transitions, in a spatial representation of dimensions similar to that of the input image, later, in intermediate layers, more complex patterns are extracted by the combinations of shapes, corners and contours. Finally, in the last layers an abstract representation is generated of different instances, such as object parts. This deep representation is given by presenting to the network different instances of each category and their natural variants in different contexts (Goodfellow et al., 2016; Krizhevsky et al., 2012). Then, with deep neural networks breaking records and coming to the forefront in computer vision, the trend is to integrate visual attention mechanisms into the learning pipeline to automatically select relevant features, by selecting regions, parts or sections of a given input, such as in images or in feature maps. Thus, attention mechanisms are designed to focus neural networks on the most relevant patterns given a specific task to solve.

Comparing automatic visual attention against human attention could be very complex and unfair. In most of the benchmark datasets, human visual attention is recorded under free-viewing conditions (Bylinskii et al., 2012), which makes annotation ambiguous for different classification tasks in images and mostly because objects may be ignored during task-driven annotation during annotation. Now, the first question arises, what in images is meaningful in classification tasks? We know that what is meaningful depends on the context of the task to solve, the objects to detect and the way they can interact (e.g., with lanes, people, and street signs in self-driving cars or hands and kitchen objects in daily life activities recognition task). Hence, given a context and a task, the objects might or might not be relevant to solve the task at hand.

In computer vision, attention models have become a very active topic in research as they allow for selective processing based on the selective process of human perception, driving analysis on regions of interest or salient in scene. In the literature, some approaches have been adopted to integrate attention models into neural networks, such as in natural language processing (Wang et al., 2016; Vaswani et al., 2017), automatic translation (Bahdanau et al., 2014), image classification and action recognition in videos (Wang et al., 2018; Chen et al., 2018), automatic image and video captioning (Xu et al., 2015; Zhu and Jiang, 2019), audio tagging (Yin et al., 2019), image and text correlation (Liu et al., 2019a), to discover semantic relationships of objects in images (Zheng et al., 2019), brain activities detection given a stimuli (Zhong et al., 2019), users interest predictor in images (Chaabouni and Precioso, 2019), for no-reference image quality assessment tasks (Yang et al., 2019), audio and face fusion for speaker naming (Liu et al., 2019b), among other applications.

Most of the existing methods use attention mechanisms that are systematically adapted from data during training. In the case of attention attention supported by a psycho-visual experiment (i.e. gaze recording), there is a specific task to solve. This, results in a top-down attention modeling, which is guided by a specific task on a voluntary basis, where high level cognitive aspects are involved to relate the information semantically, involving context characteristics, emotions, preferences and personal interest. Thus, our main question here is; are externally attention-based mechanisms better than automatic attention mechanisms when integrated in CNNs for classification tasks?

The results and methodology described in this chapter was previously published in the paper "Forward-backward visual saliency propagation in Deep NNs vs internal attentional mechanisms", in "The Ninth International Conference on Image Processing Theory, Tools and Applications (IPTA 2019)", in Istanbul, Turkey. For more details please see Montoya Obeso et al. (Montoya Obeso et al., 2019).

In this chapter, we explore whether visual attention models from a psycho-visual experiment are better than automatic attention mechanisms when integrated into CNNs. Three attention mechanisms are considered; i) the squeeze-and exitation, ii) double attention and iii) saliency-based attention. The proposal is to propagate forward and backward the visual attention on a simple architecture, i.e., the AlexNet (Krizhevsky et al., 2012). Then we also compare the influence of each attention mechanisms in a deeper neural network, the ResNet-26 (He et al., 2016).

This chapter is organized as follows. In section 6.2, we introduce the main attention mechanisms and our proposal, the saliency-based attention mechanism. Next, in section 6.3, we describe the backbone architectures, AlexNet and ResNet, the dataset, how we integrate attention blocks in neural networks. Then, results are summarized in section 6.5. Finally, we conclude this chapter in section 6.6.

6.2 Attention mechanisms

In this section we first give a general introduction about how attention mechanisms are integrated in neural networks. Visual attention, accordingly to McMains & Kastner (McMains and Kastner, 2009), corresponds to the cognitive process that allow filtering out relevant information in the field of vision, mainly in cluttered scenes.

The term visual attention has been used in a wide range of topics in the area concerning this research, computer vision. Attention is commonly associated with eye gaze fixations. However, its definition goes much further, since eye gaze fixations does not fully represent the attention of a subject in the scene under established conditions, such as the type of task and viewing conditions. Then, in computer vision, visual attention is derived in two different attention mechanisms; i) bottom-up, which is involuntary and driven by signals (features) and ii) top-down, a voluntary and task-driven (Benois-Pineau and Le Callet, 2017).

Then, speaking about features in neural networks as a collection of activations generated by specific layers; such as convolution layers. Let $\mathbf{I} \in \mathbb{R}^{h \times w \times d}$ be the input tensor (the input image), $\mathbf{X} \in \mathbb{R}^{H \times W \times D}$ a set of feature maps derived from $\mathbf{I}$, a soft attention map is denoted as $\mathbf{A} \in [0,1]^{H \times W}$ and $S \in [0,1]^{H \times W}$, an externally generated saliency map.

In the case of classification tasks, the attention neural network $\mathcal{F} : \mathbb{R}^{h \times w \times d} \rightarrow \mathbb{R}^{C}$ learns to map an input tensor to a prediction vector of probabilities $\mathbf{Y} = \mathcal{F}(\mathbf{X}) \in \mathbb{R}^{C}$, where C denotes the number of categories. Then, being $\mathbf{X}$ a set of feature maps, coming from a previous processing block in the network, attention mechanisms can be generically implemented as follows,

$$\mathbf{A} = \psi(\mathbf{X}), \tag{6.1}$$

$$\tilde{\mathbf{X}} = \mathbf{A} \odot \mathbf{X}, \tag{6.2}$$

where, $\psi(\cdot)$ denotes the attention mechanism, $\tilde{\mathbf{X}}$ the output of attention layer and $\odot$ the element-wise multiplication operation.

6.2.1 Squeeze-and-excitation

This channel weighting mechanism was first introduced to take advantage of feature maps channel inter-dependencies (Hu et al., 2018). The core idea is to add independent parameters for each channel in the block, thus, the networks learns to weight the relevance of each channel during optimization process. This model, assume that the global averaging of features maps posses patterns which can help in classification tasks. This mechanism, as shown in Fig. 6.1, is composed of three main sections; i) summarizing, ii) excitation and iii) features scaling, denoted by $\mathbf{F}_{\text{squeeze}}$,

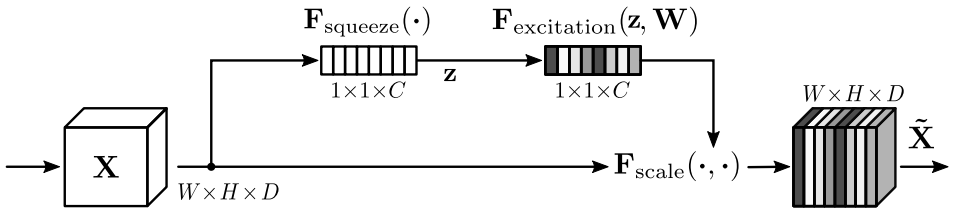


Figure 6.1: Squeeze-and-excitation block. Features are squeezed by averaging each channel to feed a very shallow 2-layer depth neural network, then original features are scaled by C learnt coefficients. Redrawn from (Hu et al., 2018).

$\mathbf{F}_{\text{excitation}}$ and $\mathbf{F}_{\text{scale}}$, respectively.

Given a set of features $\mathbf{X}$, we obtain the global average pooling per channel as follows,

$$\begin{aligned} z_c &= \mathbf{F}_{\text{squeeze}}(x_c), \\ &= \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} x_c(i,j), \end{aligned} \tag{6.3}$$

where, c denotes the number of channels in the input and $\mathbf{z} = [z_1, z_2, ..., z_c]$ denotes a channel-wise descriptor.

Then, the adaptive channel recalibration of features is given as,

$$\begin{aligned} \mathbf{s} &= \mathbf{F}_{\text{excitation}}(\mathbf{z}, \mathbf{W}), \\ &= \sigma(\mathbf{W}_\sigma \phi(\mathbf{W}_\phi \mathbf{z})), \end{aligned} \tag{6.4}$$

here, the pair ($\mathbf{W}_\sigma$, $\mathbf{W}_\phi$) are the weights of two fully-connected layers, where σ and ϕ denotes the Sigmoid and ReLU activation functions, respectively.

Finally, the output is computed by scaling $\mathbf{X}$ with $\mathbf{s}$ as follows,

$$\tilde{\mathbf{x}}_c = \mathbf{F}_{\text{scale}}(\mathbf{x}_c, s_c) = \mathbf{x}_c \cdot s_c, \tag{6.5}$$

where, $\tilde{\mathbf{X}} = [\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, ..., \tilde{\mathbf{x}}_c]$ are the weighted features and $\mathbf{F}_{\text{scale}}(\mathbf{x}_c, s_c)$ denotes the channel-wise multiplication between the scalar s_c and the features map $\mathbf{x}_c$

6.2.2 Double attention

This mechanism, proposed by Yunpeng et al. (Chen et al., 2018) is designed to capture long-range features inter-dependencies. The core idea is to gather and distribute features spatially, it was inspired by squeeze-and-excitation networks, not only for channel-wise weighting but spatial filtering and enhancement of features. A network armed with double attention blocks is able to generate "attention maps" and "attention vectors", as part of the main processing in each block.

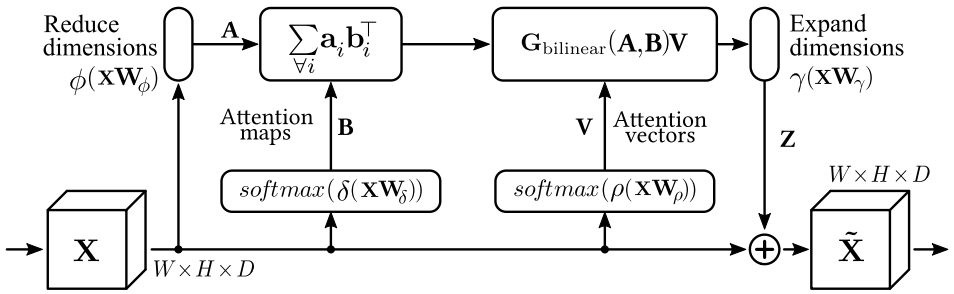


Figure 6.2: A^2: double attention block. Attention maps and attention vectors are generated by convolutional layers, denoted as $\phi(\cdot)$, $\delta(\cdot)$ and $\rho(\cdot)$, then are combined and expanded with $\gamma(\cdot)$ to be integrated back with original features $\mathbf{X}$. Redrawn from (Chen et al., 2018).

As shown in Fig. 6.2, from left to right, we first reduce dimensions as $\mathbf{A}$, extract attention maps as $\mathbf{B}$ and attention vectors as $\mathbf{V}$,

$$\mathbf{A} = \phi(\mathbf{X}\mathbf{W}_\phi), \tag{6.6}$$

$$\mathbf{B} = softmax(\delta(\mathbf{X}\mathbf{W}_\delta)), \tag{6.7}$$

$$\mathbf{V} = softmax(\rho(\mathbf{X}\mathbf{W}_\rho)), \tag{6.8}$$

In order to capture global features in the first step, features gathering is performed as,

$$\mathbf{G}_{\text{bilinear}}(\mathbf{A},\mathbf{B}) = \mathbf{A}\mathbf{B}^\top = \sum_{\forall i} \mathbf{a}_i \mathbf{b}_i^\top \tag{6.9}$$

where $\mathbf{a}_i$ and $\mathbf{b}_i$ are HWD-shaped vectors from convolution layers; $\phi(\mathbf{X}\mathbf{W}_\phi)$ and $\delta(\mathbf{X}\mathbf{W}_\delta)$, respectively.

Then, features distribution is given by,

$$\mathbf{F}_{\text{distr}}(\mathbf{X},\mathbf{V}) = \mathbf{G}_{\text{bilinear}}(\mathbf{A},\mathbf{B})\mathbf{V}, \tag{6.10}$$

where $\mathbf{V} = [v_1, v_2, ..., v_i]$ denotes a bag of visual primitives to capture more complex relations between features.

Finally, the output is given by the addition of the output of a last convolutional layer which is used to expand dimensions and with original features,

$$\tilde{\mathbf{X}} = \mathbf{X} + \gamma(\mathbf{F}_{\text{distr}}(\mathbf{X},\mathbf{V}),\mathbf{W}_\gamma). \tag{6.11}$$

Besides squeeze-and-excitation blocks, this mechanism focuses on the inter-dependencies that may exist in $\mathbf{X}$, not only measuring how relevant each channel is, but spatially adding new features to original features in $\mathbf{X}$.

6.2.3 Saliency-based attention

The Saliency Attention (SA) mechanism seeks to identify high saliency regions and give priority over others in images. As shown in Fig. 6.3, this module is composed of two branches; i) a random adaptive sampling process based on the accumulated Weibull distribution function and ii) a features pooling process, constrained on salient or non-salient regions.

Consider any set of feature maps $\mathbf{X}$, then the saliency-based attention module can be constructed as $\tilde{\mathbf{X}} = \mathbf{A}_{\text{sb}}(\mathbf{X}, S)$. Where S denotes a single channel saliency map for the input sample. If we compare this module against squeeze-and-excitation and double attention mechanisms, our mechanism requires an externally computed saliency map. In this case we consider the propagated saliency maps presented before in chapter 5, section 5.5.

A saliency map is a smooth function with values in [0,1]. Thus, the first part is to assign a high priority to salient regions while we maintain a random behavior in transitions where local values are close to the saliency map mean (μ_s).

As well as in chapter 5, we consider the maps to follow a cumulative Weibull function. Then, as well as in (González-Díaz et al., 2016), the cumulative Weibull distribution is defined as,

$$\mathbf{\Psi}_\lambda(x) = f(x; \lambda, k) = 1 - \exp^{-(x/\lambda)^k}, \tag{6.12}$$

where, with k and λ, we scale and fit the shape of the Weibull distribution. We ignore k in $\mathbf{\Psi}_\lambda$ as we set it as constant $k = 3.5$.

The aim is now to make the sampling process adaptive. For this, it is necessary to find a specific value of λ where the maximum value of the cumulative distribution function is reached when $x = \mu_s$, which corresponds to the average value of intensities in the saliency map.

Then, to find λ^*, we approximate the exponential part of eq. 6.12 very close to zero, we obtain,

$$\lambda^* = \frac{\mu_s}{|\log(0.0001)|^{\frac{1}{k}}}, \tag{6.13}$$

then, our adapted cumulative distribution $\mathbf{\Psi}_\lambda$ must be very close to 1 when $x = \mu_s$. With λ^* we can randomize the selection of features that are under μ_s, paying more attention to features in high-valued salient areas in S. For each sample in training process we compute μ_s, thus the mechanism adapts this parameter to fit and normalize in this way the behavior of random process.

Given a new matrix of random numbers $\mathbf{u} \in \mathbb{R}^{H \times W}$ following a uniform distribution, in range $[0, <1]$, we obtain a response given the inverted cumulative

distribution function,

$$\bar{\Psi}_{\lambda^*}(\mathbf{u}) = \lambda^* \sqrt[k]{-\log(1-\mathbf{u})}, \tag{6.14}$$

here, $\bar{\Psi}_{\lambda^*}$ correspond to the inverted version of the adaptive cumulative distribution function (Eq. 6.12). Before comparing $\bar{\Psi}_{\lambda^*}$ against saliency maps values, we extend $S \rightarrow \mathbf{S} \in \mathbb{R}^{H\times W\times D}$ to enable element-wise operations. Attention map $\mathbf{A}_m$ is composed then as,

$$\mathbf{A}_m = \begin{cases} 1, & \text{if } \bar{\Psi}_{\lambda^*}(\mathbf{u}) \leq \mathbf{S}, \\ 0, & \text{otherwise}, \end{cases} \tag{6.15}$$

here, in $\mathbf{A}_m \in \mathbb{R}^{H\times W\times D}$ ones represent salient regions and non-salient regions are labeled with zeros.

Then, the next step consists on the filtering of $\mathbf{X}$ given $\mathbf{A}_m$. First, we generate a pair of features; i) $\mathbf{P}_1$ for max-pooling and ii) $\mathbf{P}_2$ for average pooling as,

$$\mathbf{P}_1 = \underset{(k,s,p)}{\text{maxpool}}(\mathbf{X}), \tag{6.16}$$

$$\mathbf{P}_2 = \underset{(k,s,p)}{\text{avgpool}}(\mathbf{X}), \tag{6.17}$$

with k as the window size (3×3), s the stride parameter (1×1) and p the padding factor (1×1), generating a set of features with the same shape as the input.

The core idea is to take relevant features and reduce attention on features which are "out-of-focus", given the performed task. Thus, the saliency-based attention is computed as,

$$\tilde{\mathbf{X}} = \mathbf{A}_{\mathbf{sb}}(\mathbf{X}, S) = (\mathbf{P}_2 \odot \mathbf{A}_m) \oplus (\mathbf{P}_2 \odot \bar{\mathbf{A}}_m) \tag{6.18}$$

where, $\odot$ denotes element-wise multiplication and $\oplus$ element-wise sum.

6.3 Classification experiments setup

In this section we describe the experimental setup for architectural structures classification.

6.3.1 Residual network

Using deeper networks, the backbone is the ResNet-26 architecture (He et al., 2016). As shown in Table 6.1, the input size for the network is $256 \times 256 \times 3$, followed by a convolution layer and a pooling layer. Then, four stages of residual blocks are stacked, with two residual units each. In Table 6.1 residual blocks are denoted

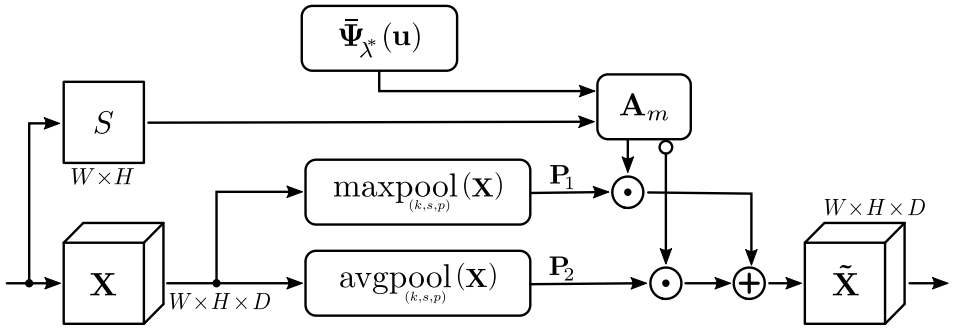


Figure 6.3: Saliency-based attention block. The saliency map is extended $S \rightarrow \mathbf{S} \in \mathbb{R}^{W \times H \times D}$ before generating $\mathbf{A}_m$. The output is composed by the sum of triggered max and average features given the input $\mathbf{X}$. Note the not logical output of $\mathbf{A}_m$ for avgpool features selection.

in brackets, where the number of output channels is 128, 256, 512, 1024, for each residual block respectively. Finally, 2D average pooling uses a window of shape 7×7 and the output layer (fully-connected) reduces its input to the number of categories (67).

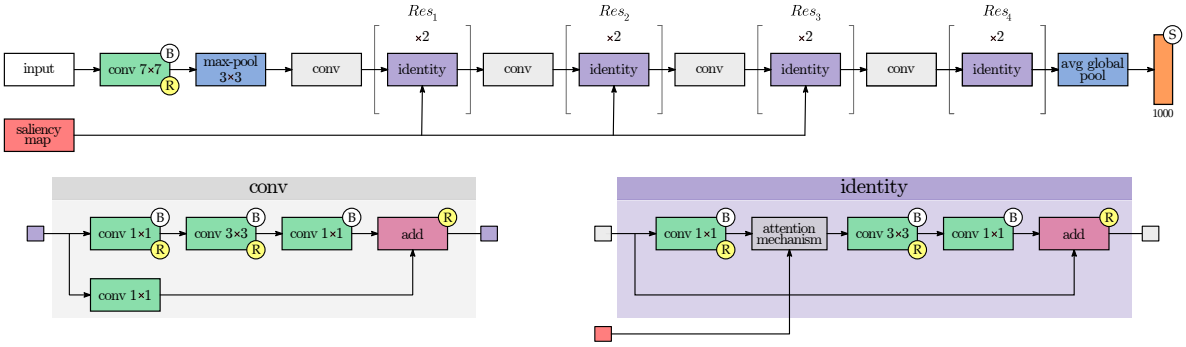


Figure 6.4: Attention mechanisms integration into the ResNet-26 architecture. Saliency-based attention mechanisms and double attention mechanisms are integrated in attention mechanism bloc in indentity blocks. Squeeze-and-excitation blocks are integrated after 3×3 convolution in identity block.

6.3.2 Dataset

Such as in previous chapter, here we use the Saliency MexCulure Buildings Dataset (SMCBD) dataset for architectural recognition, described in in section 5.6.1. This is a subset of the MexCulture Buildings Dataset (MCBD) (Montoya Obeso et al., 2016a) previously collected and labelled into styles as "Prehispanic", "Colonial" and "Modern". In total, 142 categories for the identification of specific architectural structures are considered to compose the dataset and each category stands for a specific building we aim to identify from Mexican culture.

Table 6.1: The baseline ResNet-26 for image classification. The dimensions for 3D output feature maps are in $W \times H \times D$. 2D kernels are shown in $W \times H$. The input is $3 \times 256 \times 256$. Residual units are shown in brackets.

	Layer parameters	Output size
$Data$	—	$256 \times 256 \times 3$
$Conv_1$	$16 \times 7 \times 7$, stride 2×2	$128 \times 128 \times 16$
$Pool_1$	$16 \times 3 \times 3$, stride 2×2	$64 \times 64 \times 16$
Res_1	$\begin{bmatrix} 1 \times 1, & 32 \\ 3 \times 3, & 32 \\ 1 \times 1, & 128 \end{bmatrix} \times 2$	$64 \times 64 \times 128$
Res_2	$\begin{bmatrix} 1 \times 1, & 64 \\ 3 \times 3, & 64 \\ 1 \times 1, & 256 \end{bmatrix} \times 2$	$32 \times 32 \times 256$
Res_3	$\begin{bmatrix} 1 \times 1, & 128 \\ 3 \times 3, & 128 \\ 1 \times 1, & 512 \end{bmatrix} \times 2$	$16 \times 16 \times 512$
Res_4	$\begin{bmatrix} 1 \times 1, & 256 \\ 3 \times 3, & 256 \\ 1 \times 1, & 1024 \end{bmatrix} \times 2$	$8 \times 8 \times 1024$
	global average pooling	$1 \times 1 \times 1024$
	fully-connected	67

In the base dataset, each category only contains two ground truth reference images (gaze fixations density maps) annotated through the Psycho-visual experiment (described in section 5.5). Then, gaze fixations are propagated between similar images to ensure correspondence and populate each category, following the strategy of homography estimation, described in (Montoya Obeso et al., 2018a). Finally, from the 142 categories, we only retain 67 categories as not all of them resulted well populated after attention propagation. In total, we gathered, in average 79.5±17.17 images per category, totaling 5,327 images. We split the dataset into training, validation and test, with 3,728, 1065 and 532, respectively.

6.3.3 Attention blocks integration

All attention mechanisms described in section 6.2: squeeze-and-excitation, double attention methods and saliency-based attention, are integrated in residual blocks in ResNet (see table 6.1). Different configurations are adopted accordingly to ablation studies in reference papers. Squeeze-and-excitation blocks are integrated after

the last convolution inside each residual unit (Hu et al., 2018). Non-local blocks are integrated just before the last unit of the stage (Wang et al., 2018). Finally, double attention and saliency-based attention blocks are added only after the first convolution in the first unit at the stage, such as in (Chen et al., 2018). Residual stages are labeled as Res_1, Res_2, Res_3 and Res_4, in table 6.1. Having such a small feature maps, we discard any experimentation by integrating attention mechanisms in Res_4. An illustrative example of how mechanisms are integrated is presented in Fig. 6.4.

6.3.4 Training parameters

For all the experiments we obtained better results when we use SGD with Nesterov's momentum. We started all the experiments with the same base learning rate ($l_0 = 0.01$), and we decrease it following the exponential decay function $l = l_0 * e^{-\gamma i}$, we fixed $\gamma = 0.97$ experimentally. We train all our models for 100 epochs, with a batch size of 64 images, in total 46,735 iterations.

All experiments were conducted in MXNET (Chen et al., 2015), in a single server with a Xeon E5-2620 v4 (16×@2.10GHz) and 2 × GeForce GTX 1080 NVIDIA GPUs.

6.4 Organizing content with deep features

Since DNNs became a reference for solving classification tasks, such as in the ImageNet Large Scale Visual Recognition Competition (ILSVRC), it has been proven that features extracted from intermediate layers are useful and discriminative to perform other tasks, even in tasks for which the models were not specifically trained.

In this context, Babenko et al. (Babenko et al., 2014), presented an image retrieval scheme based on extracted features using the well-known AlexNet architecture (Krizhevsky et al., 2012). The so-called "neural codes", are feature vectors extracted given by fully-connected layers. They found that even when the neural codes are compressed, with a trivial Principal Component Analysis (PCA), neural codes are still useful for retrieval tasks. Furthermore, Hoang et al. (Hoang et al., 2017), in a more intensive evaluation of neural codes, propose a framework for image retrieval given different neural codes, obtained from different layers in a CNN. The composition of their models is a mixture of masking schemes for features selection, aggregation models and embedding, reaching high rated results on different datasets.

6.4.1 Features extraction with attentive CNNs

The neural codes or deep features, are extracted using pre-trained models. Let us consider a pre-trained model of a ResNet-26 derived from the architecture described before in section 6.3.1. This model is capable to receive an input and generate a series of internal features immediately after each layer, we are interested on deeper layers. Besides, this model is not a plain ResNet version, we consider in this case, a ResNet-26 equipped with saliency-based attention mechanisms.

6.4.2 Features clustering

For clustering, we use the well-known k-means method (MacQueen et al., 1967). This is one of the simplest algorithm that solve the clustering problem. We define K clusters as far away as possible to each other, then, each data point is associated to a nearest cluster. When every point is associated, the first step is completed and a first grouping is done. At this point, the centroids are recalculated accordingly the data points that have acquired in previous step. This generates a loop, where data points are associated to each centroid and centroids are adjusted over and over. Finally, the centroids change their location step by step until no more significant changes are done (or another stop condition is satisfied). The k-means parameterization is about the number of clusters, the total iterations for fitting on data, as well the as the centroids initialization strategy.

We fixed the main k-means parameters as follows, 3 clusters for style recognition. We found that only 100 iterations were enough reach a stable clustering while fitting on data and we use the "kmeans++" method as centroids initialization strategy, based on its convergence acceleration advantage.

For evaluation purposes, we first define the following,

- the dataset is composed of N samples,
- the classes $C = \{c_i \mid i = 1, ..., n\}$ and,
- the clusters $K = \{k_j \mid j = 1, ..., m\}$.

Consider Z to describe a clustering solution table produced by the k-means algorithm on data. Such that, elements in $Z = \{z_{ij}\}$ are the number of data points which are members of class c_i and elements of cluster k_j.

Metrics: we consider 4 metrics for clustering evaluation: i) homogeneity, ii) completeness, iii) v-measure and iv) silhouette coefficient. More details about these metrics in annex B.

6.4.3 Implementation details

All experiments are conducted in a single server with 3 $\times$ RTX2080 Ti GPUs. For ResNets training, we use Nesterov's Stochastic Gradient Descent, base learning rate is initialized to 0.01 and scheduled to decrease following an exponential decay with a gamma value of 0.97. Neural Networks trainings are conducted on MXNet framework (Chen et al., 2015). Besides, the PCA algorithm and K-means clustering we use, are part of the Scikit-Learn framework in Python (Pedregosa et al., 2011).

6.5 Results

In this section, we compare performances of the following attention mechanisms; squeeze-and-excitation, double-attention and the saliency-based attention mechanism in a ResNet-26 deep architecture. Besides, we analyze clustering results by using deep features for cultural content organization.

6.5.1 Architectural structures recognition

We found that visual saliency-based mechanism outperforms others with a top-1 accuracy of 96.35% and a top-5 accuracy of 99.47% at epoch 92. This means that the proposal converges faster than others, except for the ResNet-26 + A^2 equipped with attention blocks at stage @$Res_{1,2,3}$. In the case of double-attention networks we see that the performance is worst than others. However, when more attention blocks are added at deeper stages, the accuracy increases. This model reached a maximum accuracy of 89.58% at top-1 and 98.26% at top-5, at epoch 26.

Furthermore, in Fig. 6.5 and in Fig. 6.6, we illustrate two examples of extracted features from, double attention blocks and saliency-based attention blocks, respectively. $\mathbf{X}$ and $\tilde{\mathbf{X}}$, denotes in both the input and the output of the attention block. We can notice that both attention methods add similar features to input feature maps in a very similar but specific way. In the case of double attention block, relevant features focuses on high frequency regions. Moreover, our proposal focuses more on salient regions given by the input image components. If we have a look to $\tilde{\mathbf{X}}$ in Fig. 6.6, that features are enhanced by max-pooling while in non-relevant regions, features are no dropped to zero but a bit vanished. We consider that saliency-based attention mechanisms captures better those elements in feature maps that are relevant to the tasks of classification. We also understand that the observation task that gave birth to our top-down model –based on gaze recordings– is better represented in our dataset.

Table 6.2: Results in the classification task of architectural structures classification. Attention blocks are integrated after first residual unit in the stage if it applies. ResNet-26 is the backbone architecture such as in Chen et al., 2018. Accuracy is reported at Top1 and at Top5 for 67 categories. "Epoch" denotes both, when maximum accuracy and convergence are reached.

Model	+1 Block	# Params	@Top-1	@Top-5	Epoch
ResNet-26	-	44.5 M	92.19%	98.26%	70
ResNet-26 + SE	-	44.6 M	92.36%	98.43%	64
ResNet-26 + A^2	@Res_1	44.6 M	88.28%	97.56%	54
	@$Res_{1,2}$	44.9 M	89.41%	97.57%	81
	@$Res_{1,2,3}$	45.9 M	89.57%	98.26%	26
ResNet-26 + SA	@Res_1	44.5 M	95.14%	98.78%	53
	@$Res_{1,2}$	44.5 M	96.35%	98.95%	47
	@$Res_{1,2,3}$	44.5 M	96.35%	99.47%	92

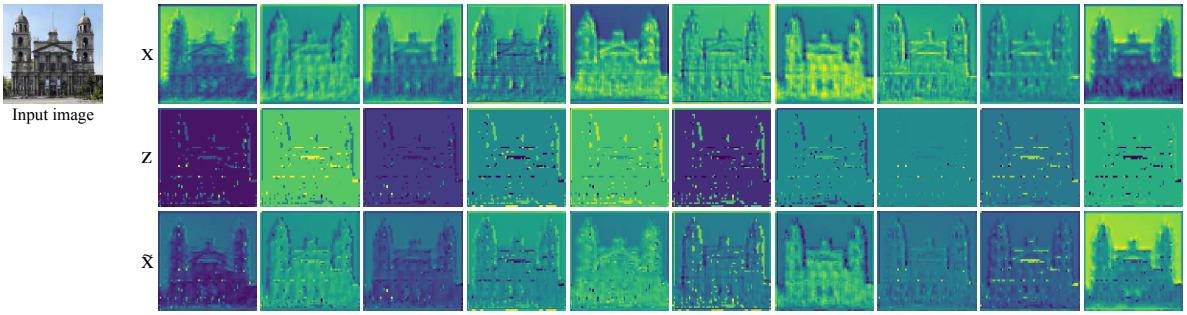


Figure 6.5: Double attention feature maps (A^2-Net). Feature maps are denoted as $\mathbf{X}$, attention ($\mathbf{Z}$) is given by both mechanisms, features gathering and distribution of features. $\tilde{\mathbf{X}}$ denotes the output of the double attention block. Features are extracted from stage @Res_2. Source: (Montoya Obeso et al., 2019).

6.5.2 Cultural heritage clustering

Clustering results are evaluated based on metrics described in annex B. We present clustering results on test dataset (SMCBD) based on PCA-reduced deep features, from a compression level going from 10% to 90%.

Regarding the first two metrics, homogeneity and completeness, describe how well organized the clusters are, based on their membership to both, classes and clusters. Then, in Table 6.3, one can notice the maximum reached values are 0.7443 and 74.22 using only the 50% of deep features (512 components). Thus, the validity score (v-measure) in this case is 0.7422 and matches as the higher value if compared with other levels of compression. Contrarily, in the case of silhouette coefficient, this is not the best case. The best score of silhouette coefficient is reached when we compress up features up to 90%, equals to 0.3994. We are convinced that is because

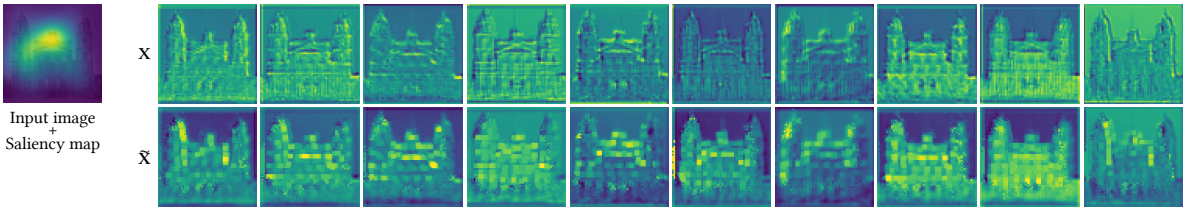


Figure 6.6: Saliency-based feature maps. Feature maps are denoted as $\mathbf{X}$, the input and $\tilde{\mathbf{X}}$ denotes the output of the saliency-based attention block. Feature maps are extracted from stage @Res_2. Source: (Montoya Obeso et al., 2019).

Table 6.3: K-means clustering results on testing dataset. The initial length of input vectors is 1024 elements, we compress features until only 10% are remaining.

Features	Hom.	Comp.	V-meas.	Silh.	PCA time	K-means time
100%	0.7330	0.7381	0.7373	0.3834	-	0.1648s
90%	0.6446	0.5778	0.5879	0.3171	0.4749s	0.1120s
80%	0.6859	0.6428	0.6496	0.3524	0.3406s	0.1257s
70%	0.6667	0.6070	0.6162	0.3313	0.3611s	0.0526s
60%	0.6781	0.6300	0.6375	0.3467	0.4257s	0.0823s
50%	0.7401	0.7443	0.7436	0.3847	0.3639s	0.0409s
40%	0.7330	0.7381	0.7373	0.3862	0.3329s	0.0334s
30%	0.6436	0.5765	0.5867	0.3212	0.3332s	0.0317s
20%	0.7330	0.7381	0.7373	0.3921	0.3602s	0.0542s
10%	0.7330	0.7381	0.7373	0.3994	0.3491s	0.0230s

PCA compression looks for the largest possible variance in data, which results easier to achieve when a lower dimensional is presented in such sparse data. Furthermore, in a general perspective, measured coefficients are rather high which indicates two things; i) such data coming from different sources and variable resolutions is well describing each category and ii) the attentive CNN model provides useful deep features for clustering that could be used in a high performance retrieval system.

In Table 6.3, we present PCA-compressing time and dataset clustering. The time required to describe by batches with the ResNet testing dataset is 1.5677s, running on GPU.

6.6 Conclusions

In this chapter, we studied different attention mechanisms and their influence when they are integrated in deep neural networks. Our proposal, the saliency-based attention mechanism, have shown better performances over others, such as the squeeze-and-excitation block and the double attention block. The attention mechanisms in ResNets shown better performances than other models trained before in our previous research. The architectures were studied to solve the task of classification of architectural structures –buildings– given the taxonomy of only

67 categories, composed of images from different views of each building (object of interest).

We found that integrating real attention top-down models lead deep neural networks to reach better performances, even when compared to automatic attention models. The main task for training these models is the generation of reference annotated data, commonly generated using eye-trackers systems. These devices are currently mass produced and have become cheaper. Therefore, it is possible to make use of them to generate information that feeds these deep models and generate useful predictions in specific use systems, such as recognition or image retrieval systems.

Concerning to cultural content organization task, we present a first approach on deep features clustering given by trained modes where we integrated attention mechanisms. Based on different metric scores, we found that deep features have discriminant power for structuring new data of Mexican culture. Quality metrics for clustering evaluation have shown that even with PCA-compressed features, lower dimensional vectors are still useful to identify groups and structuring new data based on learnt features.

Chapter 7

General conclusion

In this thesis, we studied the classification of specific Mexican architectural structures. We integrated models of visual saliency in Convolutional Neural Networks in order to focus and filter features based on regions with more attention in images, giving us the recognition of specific architectural structures in Mexican cultural images. The studies we have performed show how saliency maps improve performances in the particular implementation of custom saliency pooling layers. First, we studied how people look at architecture in images in the task of style recognition. We ran a psycho-visual experiment with the aim of identifying relevant characteristics for architectural style recognition. We found interesting advantages of using visual models into deep learning and explored an interesting field of research about eye-tracking from visual experiments for new data generation. Then, we used eye-tracking data to build up subjective saliency maps for each image displayed to participants. The generation of saliency maps is not a contribution by itself, our contribution is on the propagation of these gaze fixations between similar images. We propose a method to ensure this propagation into similar images in a new set of images we did not display to participants, generation a larger dataset in less annotation time by gaze recording. Finally, we developed a saliency pooling layer where we include saliency maps during the training process.

In the remaining part of this chapter, we will review the main contributions and we will give a brief analysis of the published research.

7.1 Main contributions

Pursuing the goal of Mexican architecture recognition, we introduced several contributions in different domains.

7.1.1 Specific purpose datasets

First of all, regarding the datasets for models training, we have generated two specific-purpose datasets; i) for the identification of architectural structures by style and ii) for the identification of specific buildings. In general, the first dataset composed of more than 16,000 images with variable resolution. Images were retrieved from a large corpus of Mexican cultural institutions, Mexican universities and from online platforms as well. The second dataset, for specific building recognition, is composed of 142 categories from the most relevant architectures in Mexico. Each category contains several samples for specific architectural identification. Even when this dataset is relatively small, we ran a psycho-visual experiment in order to improve the significance of this data in the field of computer vision by getting gaze fixations for each image with the aim of identifying relevant regions in images of architectural structures.

7.1.2 Psycho-visual experiment

Since one of our objectives is to integrate visual saliency models into deep learning models for specific architectural recognition, we propose a scheme to generate subjective saliency maps from a psycho-visual experiment by recording gaze fixations from participants. As a result of this experimentation, we provide two reference images per class, gaze fixations, gaze fixations density maps (subjective Wooding saliency maps) and scanpaths (we do not use scanpaths in this work). This dataset is now publicly available for research community (see section 5.5).

7.1.3 Subjective saliency maps propagation

Starting from gaze fixations, given by the psycho-visual experiment we conducted, we propose a gaze fixations propagation method from annotated images to images we did not show to participants during the experiment, labeled as target images in section 5.5.2.

Preliminary results shown that the propagation of subjective saliency maps is a fast way to increase the number of samples avoiding human annotation by eye-tracking data, which is hard and time-consuming. We found around 79.5 ± 17.17 samples per class reaching around 5,360 images in total for training. Hence, the proposed method to validate homography estimation from the projected polygon gives a high rate (69.37%) to find correct estimations of homography from matching points. The automatic dataset population allow us to increase the training set.

7.1.4 Visual attention models integration into CNNs

Concerning the integration of saliency maps into a deep learning model, in this research we present the first attempt to explicitly introduce predicted saliency maps into a deep CNN framework for features filtering without increasing considerably the complexity of the system. We found that models trained on a basis of externally generated saliency maps – obtained through a psycho-visual experiment– allows a more accurate classification. The integration of a custom saliency-based pooling layer allows the introduction of different saliency maps in the training scheme. In these layers, saliency maps are used to sample each of the input feature maps, which are generated by the immediate previous layer. The performance of the proposed model, equipped with the provided attention maps –the COSAL maps– presents outstanding results over the other models.

7.1.5 Attention mechanisms in CNNs

One of the last contributions of this research is focused on the study of attention mechanisms deep residual neural networks. We integrated these attention mechanisms into residual networks. These mechanisms, strategically integrated in the residual units, allow selecting relevant characteristics –for the task to be solved– during training process. Using this scheme, our proposal is a saliency-based attention mechanism. This mechanism allows to filter out relevant characteristics for subsequent processing stages. Like the double attention mechanism, our mechanism seeks to capture the spatial relationships in training data and the attention that each of the zones receives during a real recognition task performed by humans. The saliency map is a key piece to establish which types of characteristics, in a subjective way, allow to identify relationships or characteristics that allow models to reach a better performance.

This top-down model that is integrated within our attention mechanism based on saliency maps, allows to obtain superior results compared with automatic attention mechanisms. Finally, the residual neural network is capable of classifying 67 architectural structures with an overall accuracy of 96.35%, while the baseline model has an accuracy of 92.19%.

7.2 A short analysis of published research

In this section, we analyze the articles published during this research and we highlight the contributions in each publication. The published research corresponds to the integration of visual attention methods in several levels of the reference scheme of automatic learning.

As a first contribution, in Montoya Obeso et al. (2017b), content selection methods were introduced in the feeding step of the learning model. It was proposed to select relevant information peaks of relevant regions of visual saliency maps (using GBVS). Then, it was compared with the data selection of the image center, looking for an advantage of what is known as "central bias hypothesis". The results show that the use of saliency as a content selection method contributes to the increase in the accuracy of trained models with respect to selecting the central content of the image. This problematic has been addressed by Uijilings and Arbeláez (Uijlings et al., 2013b; Arbeláez et al., 2014), where the objective is to propose multiple regions of interest relevant for object classification tasks. In contrast, our work seeks to expedite the proposal of regions of interest for classification. When saliency-based selection method is compared with the method used by Girshick (Girshick, 2015), for generation of regions in the database ImageNet (Russakovsky et al., 2015) the generation of region proposals is faster. However, the performance of proposed model of content selection is lower than the selection method proposed by Arbeláez (9.36%) and greater with respect to the Uijlings method (15.34%), this is derived from the quality and quantity of regions that the Arbeláez method generates. In general, the presented contributions are interesting and are derived from extensive experimentation with both databases, for the identification of Mexican architectural structures and in the ImageNet database.

After, in Montoya Obeso et al. (2017a), we published our first contributions regarding to the integration of saliency maps as part of models optimization process. In this paper, we added an additional map of features to input images, in this case we used GBVS for each sample in the database. We trained two reference architectures; AlexNet and GoogLeNet. We found that GoogLeNet outperforms AlexNet very little and Saliency-Based data selection slightly outperforms Center-Based selection data. These contributions motivated us to continue with new implementations in the task of integration of visual attention maps in models of deep learning.

Then, in Montoya Obeso et al. (2018b), we propose our first attempt to introduce saliency maps in the main pipeline of data processing during training a CNN, explicitly in pooling layers. In this work, we use saliency maps as support data for taking relevant features into deeper layers. A saliency pooling layer takes as input features maps of previous layer and the saliency map of the current sample. Then, assuming that saliency maps follows a Weibull distribution, we randomly sample regions to max or min pooling. We replaced max-pooling layers in an AlexNet-like network obtaining very similar results to the baseline model. The main difference with baseline in terms of performance is that our model converges faster. All this models we trained were focused to identify architectural styles in Mexican architecture, relatively small datasets with a small number of categories as well.

In Montoya Obeso et al. (2018a), we present a study about the influence of

different saliency maps in the task o recognition of specific buildings on the Mexican culture. This time, introducing 67 categories and the so-called co-saliency maps, generated by running a psycho-visual experiment. We found that integrating both; task-driven generated saliency maps and saliency pooling layers in CNNs, we can reach a better performance, an increase of ∼6%, compared with GBVS and SMUIC models. Finally, in Montoya Obeso et al. (2019), we studied different attention mechanisms and their influence when they are integrated in deep neural networks. Our main proposal, the saliency-based attention mechanism, have shown better performances over others, such as the squeeze-and-excitation block and the double attention block. The attention mechanisms in ResNets shown better performances than other models trained before in our previous research. The architectures were studied to solve the task of classification of architectural structures –buildings– given the taxonomy of only 67 categories, composed of images from different views of each building (object of interest).

Bibliography

Arbeláez, Pablo et al. (2014). "Multiscale combinatorial grouping". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 328–335.

Babenko, Artem et al. (2014). "Neural codes for image retrieval". In: *European conference on computer vision*. Springer, pp. 584–599.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2014). "Neural machine translation by jointly learning to align and translate". In: *International Conference on Learning Representations*.

Baluja, Shumeet and Dean A Pomerleau (1995). "Using a saliency map for active spatial selective attention: Implementation & initial results". In: *Advances in Neural Information Processing Systems*, pp. 451–458.

Benois-Pineau, Jenny and Patrick Le Callet (2017). *Visual Content Indexing and Retrieval with Psycho-Visual Models*. Springer.

Bhowmik, Neelanjan et al. (2017). "Combination of image descriptors for the exploration of cultural photographic collections". In: *Journal of Electronic Imaging* 26.1, pp. 011019–011019.

Borji, Ali (2012). "Boosting bottom-up and top-down visual features for saliency estimation". In: *2012 ieee conference on computer vision and pattern recognition*. IEEE, pp. 438–445.

Borji, Ali et al. (2015). "Salient object detection: A benchmark". In: *IEEE transactions on image processing* 24.12, pp. 5706–5722.

Buso, Vincent, Iván González-Díaz, and Jenny Benois-Pineau (2015). "Goal-oriented top-down probabilistic visual attention model for recognition of manipulated objects in egocentric videos". In: *Signal Processing: Image Communication* 39, pp. 418–431.

Buswell, Guy Thomas (1935). "How people look at pictures: a study of the psychology and perception in art." In:

Bylinskii, Zoya et al. (2012). *MIT Saliency Benchmark*. http://saliency.mit.edu/.

Carrasco, Marisa (2011). "Visual attention: The past 25 years". In: *Vision research* 51.13, pp. 1484–1525.

Carvalho Soares, Robson de, Ilmerio Reis da Silva, and Denise Guliato (2012). "Spatial locality weighting of features using saliency map with a

bag-of-visual-words approach". In: *Tools with Artificial Intelligence (ICTAI), 2012 IEEE 24th International Conference on*. Vol. 1. IEEE, pp. 1070–1075.

Cerf, Moran et al. (2008). "Predicting human gaze using low-level saliency combined with face detection". In: *Advances in neural information processing systems*, pp. 241–248.

Chaabouni, Souad, Jenny Benois-Pineau, and Chokri Ben Amar (2019). "ChaboNet: Design of a deep CNN for prediction of visual saliency in natural video". In: *Journal of Visual Communication and Image Representation* 60, pp. 79–93.

Chaabouni, Souad and Frederic Precioso (2019). "Impact of Saliency and Gaze Features on Visual Control: Gaze-Saliency Interest Estimator". In: *Proceedings of the 27th ACM International Conference on Multimedia*. MM '19. Nice, France: ACM, pp. 1367–1374. ISBN: 978-1-4503-6889-6. DOI: 10.1145/3343031.3350964. URL: http://doi.acm.org/10.1145/3343031.3350964.

Chen, Tianqi et al. (2015). "Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems". In: *arXiv preprint arXiv:1512.01274*.

Chen, Yunpeng et al. (2018). "A^2-Nets: Double Attention Networks". In: *Advances in Neural Information Processing Systems 31*. Ed. by S. Bengio et al. Curran Associates, Inc., pp. 352–361. URL: http://papers.nips.cc/paper/7318-a2-nets-double-attention-networks.pdf.

Chi, Jianning et al. (2019). "Saliency detection via integrating deep learning architecture and low-level features". In: *Neurocomputing* 352, pp. 75–92.

Chollet, François (2017). "Xception: Deep learning with depthwise separable convolutions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1251–1258.

Connor, Charles E, Howard E Egeth, and Steven Yantis (2004). "Visual attention: bottom-up versus top-down". In: *Current biology* 14.19, R850–R852.

Cortes, Corinna and Vladimir Vapnik (1995). "Support-vector networks". In: *Machine learning* 20.3, pp. 273–297.

Coutrot, Antoine and Nathalie Guyader (2014). "How saliency, faces, and sound influence gaze in dynamic social scenes". In: *Journal of vision* 14.8, pp. 5–5.

Csurka, Gabriella et al. (2004). "Visual categorization with bags of keypoints". In: *Workshop on statistical learning in computer vision, ECCV*. Vol. 1. 1-22. Prague, pp. 1–2.

Eriksen, Charles W and James E Hoffman (1972). "Temporal and spatial characteristics of selective encoding from visual displays". In: *Perception & psychophysics* 12.2, pp. 201–204.

Europeana (2008). *Europeana collections*. URL: http://www.europeana.eu (visited on 02/10/2019).

Fukushima, Kunihiko (1980). "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position". In: *Biological cybernetics* 36.4, pp. 193–202.

Ghodrati, Amir et al. (2017). "DeepProposals: Hunting Objects and Actions by Cascading Deep Convolutional Layers". In: *International Journal of Computer Vision* 124.2, pp. 115–131. ISSN: 1573-1405. DOI: 10.1007/s11263-017-1006-x. URL: https://doi.org/10.1007/s11263-017-1006-x.

Girshick, Ross (2015). "Fast R-CNN". In: *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448.

Glorot, Xavier, Antoine Bordes, and Yoshua Bengio (2011). "Deep sparse rectifier neural networks". In: *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 315–323.

González-Díaz, Iván, Vincent Buso, and Jenny Benois-Pineau (2016). "Perceptual modeling in the problem of active object recognition in visual scenes". In: *Pattern Recognition* 56, pp. 129–141.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep learning*. MIT press.

Guissous, Kamel and Valérie Gouet-Brunet (2017). "Image retrieval based on saliency for urban image contents". In: *2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pp. 1–6.

— (2019). "Saliency and Burstiness for Feature Selection in CBIR". In: *2019 8th European Workshop on Visual Information Processing (EUVIP)*. IEEE, pp. 111–116.

Harel, Jonathan, Christof Koch, and Pietro Perona (2007). "Graph-based visual saliency". In: *Advances in neural information processing systems*, pp. 545–552.

He, Kaiming et al. (2016). "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Healey C. G.and Booth, K. S. and J. T. Enns (1996). "High-Speed Visual Estimation Using Preattentive Processing". In: *ACM Transactions on Human Computer Interaction* 3, pp. 107–135.

Hebb, Donald Olding (1949). *The organization of behavior: A neuropsychological theory*. John Wiley & Sons.

Hoang, Tuan et al. (2017). "Selective deep convolutional features for image retrieval". In: *Proceedings of the 25th ACM international conference on Multimedia*, pp. 1600–1608.

Howard, Andrew G (2013). "Some improvements on deep convolutional neural network based image classification". In: *arXiv preprint arXiv:1312.5402*.

Hu, Jie, Li Shen, and Gang Sun (2018). "Squeeze-and-excitation networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132–7141.

I-TREASURES (2013). *i-treasures capturing the intangible*. URL: http://i-treasures.eu/ (visited on 02/10/2019).

Itti, Laurent, Christof Koch, and Ernst Niebur (1998). "A model of saliency-based visual attention for rapid scene analysis". In: *IEEE Transactions on pattern analysis and machine intelligence* 20.11, pp. 1254–1259.

Jacobs, David E., Dan B. Goldman, and Eli Shechtman (2010). "Cosaliency: where people look when comparing images". In: *Proceedings of the 23rd Annual ACM Symposium on User Interface Software and Technology, New York, NY, USA, October 3-6, 2010*, pp. 219–228. DOI: 10.1145/1866029.1866066. URL: http://doi.acm.org/10.1145/1866029.1866066.

Jia, Yangqing et al. (2014). "Caffe: Convolutional architecture for fast feature embedding". In: *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, pp. 675–678.

Jiang, Ming et al. (2015). "SALICON: Saliency in Context". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Judd, Tilke, Frédo Durand, and Antonio Torralba (2012). "A benchmark of computational models of saliency to predict human fixations". In:

Judd, Tilke et al. (2009). "Learning to predict where humans look". In: *2009 IEEE 12th international conference on computer vision*. IEEE, pp. 2106–2113.

Kanan, Christopher et al. (2009). "SUN: Top-down saliency using natural statistics". In: *Visual cognition* 17.6-7, pp. 979–1003.

Katsuki, Fumi and Christos Constantinidis (2014). "Bottom-up and top-down attention: different processes and overlapping neural systems". In: *The Neuroscientist* 20.5, pp. 509–521.

Koch, Christof and Shimon Ullman (1987). "Shifts in selective visual attention: towards the underlying neural circuitry". In: *Matters of intelligence*. Springer, pp. 115–141.

Koch, Kristin et al. (2006). "How much the eye tells the brain". In: *Current Biology* 16.14, pp. 1428–1434.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*, pp. 1097–1105.

LeCun, Yann et al. (1998). "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11, pp. 2278–2324.

Li, Heyi, Klaus Mueller, and Xin Chen (2017a). "Beyond saliency: understanding convolutional neural networks from saliency prediction on layer-wise relevance propagation". In: *CoRR* abs/1712.08268.

Li, Hongliang, Fanman Meng, and King Ngi Ngan (2013). "Co-salient object detection from multiple images". In: *IEEE Transactions on Multimedia* 15.8, pp. 1896–1909. DOI: 10.1109/TMM.2013.2271476.

Li, Jia, Changqun Xia, and Xiaowu Chen (2017b). "A benchmark dataset and saliency-guided stacked autoencoders for video-based salient object detection". In: *IEEE Transactions on Image Processing* 27.1, pp. 349–364.

Liu, Chunxiao et al. (2019a). "Focus Your Attention: A Bidirectional Focal Attention Network for Image-Text Matching". In: *Proceedings of the 27th ACM International Conference on Multimedia*. MM '19. Nice, France: ACM, pp. 3–11. ISBN: 978-1-4503-6889-6.

Liu, Xin et al. (2019b). "Attention guided deep audio-face fusion for efficient speaker naming". In: *Pattern Recognition* 88, pp. 557–568.

Llamas, Jose et al. (2016). "Applying Deep Learning Techniques to Cultural Heritage Images Within the INCEPTION Project". In: *Euro-Mediterranean Conference*. Springer, pp. 25–32.

Ma, Yu-Fei et al. (2005). "A generic framework of user attention model and its application in video summarization". In: *IEEE transactions on multimedia* 7.5, pp. 907–919.

MacQueen, James et al. (1967). "Some methods for classification and analysis of multivariate observations". In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. Vol. 1. 14. Oakland, CA, USA, pp. 281–297.

Mahadevan, Vijay and Nuno Vasconcelos (2013). "Biologically inspired object tracking using center-surround saliency mechanisms". In: *IEEE transactions on pattern analysis and machine intelligence* 35.3, pp. 541–554.

Mathe, Stefan and Cristian Sminchisescu (2012). "Dynamic eye movement datasets and learnt saliency models for visual action recognition". In: *Computer Vision–ECCV 2012*, pp. 842–856.

McMains, Stephanie A. and Sabine Kastner (2009). "Visual Attention". In: *Encyclopedia of Neuroscience*. Ed. by Marc D. Binder, Nobutaka Hirokawa, and Uwe Windhorst. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 4296–4302.

Menabrea, Luigi Federico and Ada Lovelace (1842). *Sketch of the analytical engine invented by Charles Babbage*.

Montoya Obeso, Abraham et al. (2016a). "Architectural style classification of Mexican historical buildings using deep convolutional neural networks and sparse features". In: *Journal of Electronic Imaging* 26.1, p. 011016.

Montoya Obeso, Abraham et al. (2016b). "Image annotation for Mexican buildings database". In: *SPIE Optical Engineering+ Applications*. International Society for Optics and Photonics, 99700Y–99700Y.

Montoya Obeso, Abraham et al. (2017a). "Connoisseur: classification of styles of Mexican architectural heritage with deep learning and visual attention prediction". In: *Proceedings of the 15th International Workshop on Content-Based Multimedia Indexing*. ACM, p. 16.

Montoya Obeso, Abraham et al. (2017b). "Saliency-based selection of visual content for deep convolutional neural networks". In: *Journal of Multimedia Tools and Applications*, pp. 1–24.

Montoya Obeso, Abraham et al. (2018a). "Comparative study of visual saliency maps in the problem of classification of architectural images with Deep CNNs". In: *2018 Eighth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pp. 1–6.

Montoya Obeso, Abraham et al. (2018b). "Introduction of Explicit Visual Saliency in Training of Deep CNNs: Application to Architectural Styles Classification". In: *2018 International Conference on Content-Based Multimedia Indexing (CBMI)*, pp. 1–5.

— (2019). "Forward-backward visual saliency propagation in Deep NNs vs internal attentional mechanisms". In: *2019 Ninth International Conference on Image Processing Theory, Tools and Applications (IPTA)*. IEEE, pp. 1–6.

Nesterov, Yurii E (1983). "A method for solving the convex programming problem with convergence rate O (1/kˆ 2)". In: *Dokl. akad. nauk Sssr*. Vol. 269, pp. 543–547.

Pan, Junting et al. (2017). "Salgan: Visual saliency prediction with adversarial networks". In: *CVPR Scene Understanding Workshop (SUNw)*.

Papushoy, Alex and Adrian G Bors (2015). "Image retrieval based on query by saliency content". In: *Digital Signal Processing* 36, pp. 156–173.

Pedregosa, F. et al. (2011). "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12, pp. 2825–2830.

Pont-Tuset, J. et al. (2017). "Multiscale Combinatorial Grouping for Image Segmentation and Object Proposal Generation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.1, pp. 128–140. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2016.2537320.

Presious (2013). *The PRESIOUS Project*. URL: http://www.presious.eu (visited on 02/10/2019).

Ramírez, Alejandro et al. (2015). "The Mex-Culture Multimedia platform: Preservation and dissemination of the Mexican Culture". In: *2015 13th International Workshop on Content-Based Multimedia Indexing (CBMI)*. IEEE, pp. 1–6.

Rosenberg, Andrew and Julia Hirschberg (2007). "V-measure: A conditional entropy-based external cluster evaluation measure". In: *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, pp. 410–420.

Rosenblatt, Frank (1958). "The perceptron: a probabilistic model for information storage and organization in the brain." In: *Psychological review* 65.6, p. 386.

Rousseeuw, Peter J (1987). "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis". In: *Journal of computational and applied mathematics* 20, pp. 53–65.

Rumelhart, David E, Geoffrey E Hinton, and Ronald J Williams (1986). "Learning representations by back-propagating errors". In: *nature* 323.6088, pp. 533–536.

Russakovsky, Olga et al. (2015). "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision (IJCV)* 115.3, pp. 211–252. DOI: 10.1007/s11263-015-0816-y.

Seo, Hae Jong and Peyman Milanfar (2009). "Static and space-time visual saliency detection by self-resemblance". In: *Journal of vision* 9.12, pp. 15–15.

Shen, Chengyao and Qi Zhao (2014). "Learning to predict eye fixations for semantic contents using multi-layer sparse network". In: *Neurocomputing* 138, pp. 61–68.

Simonyan, Karen, Andrea Vedaldi, and Andrew Zisserman (2013). "Deep inside convolutional networks: Visualising image classification models and saliency maps". In: *arXiv preprint arXiv:1312.6034*.

Simonyan, Karen and Andrew Zisserman (2014). "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556*.

Srivastava, Nitish et al. (2014). "Dropout: a simple way to prevent neural networks from overfitting". In: *The journal of machine learning research* 15.1, pp. 1929–1958.

Su, Yingya et al. (2014). "Abrupt motion tracking using a visual saliency embedded particle filter". In: *Pattern Recognition* 47.5, pp. 1826–1834.

Szegedy, Christian et al. (2015). "Going deeper with convolutions". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9.

Szegedy, Christian et al. (2017). "Inception-v4, inception-resnet and the impact of residual connections on learning". In: *Thirty-first AAAI conference on artificial intelligence*.

Tagcloud (2013). *TAG CLOUD project*. URL: http://www.tagcloudproject.eu/ (visited on 02/10/2019).

Tatler, Benjamin W, Roland J Baddeley, and Iain D Gilchrist (2005). "Visual correlates of fixation selection: Effects of scale and time". In: *Vision research* 45.5, pp. 643–659.

Tian, Huawei et al. (2014). "Salient region detection by fusing bottom-up and top-down features extracted from a single image". In: *IEEE Transactions on Image processing* 23.10, pp. 4389–4398.

Torralba, Antonio et al. (2006). "Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search." In: *Psychological review* 113.4, p. 766.

Treisman, Anne M and Garry Gelade (1980). "A feature-integration theory of attention". In: *Cognitive Psychology* 12.1, pp. 97–136. ISSN: 0010-0285.

Tsotsos, John K et al. (1995). "Modeling visual attention via selective tuning". In: *Artificial intelligence* 78.1-2, pp. 507–545.

Uijlings, Jasper RR et al. (2013a). "Selective search for object recognition". In: *International journal of computer vision* 104.2, pp. 154–171.

Uijlings, Jasper RR et al. (2013b). "Selective search for object recognition". In: *International journal of computer vision* 104.2, pp. 154–171.

Vaswani, Ashish et al. (2017). "Attention is all you need". In: *Advances in neural information processing systems*, pp. 5998–6008.

Vig, Eleonora, Michael Dorr, and David Cox (2012). "Space-variant descriptor sampling for action recognition based on saliency and eye movements". In: *European conference on computer vision*. Springer, pp. 84–97.

Von Gioi, R.G. et al. (2010). "LSD: A fast line segment detector with a false detection control". In: *Pattern Analysis and Machine Intelligence* 32.4, pp. 722–732.

Wang, Qi et al. (2013). "Saliency detection by multiple-instance learning". In: *IEEE transactions on cybernetics* 43.2, pp. 660–672.

Wang, Xiaolong et al. (2018). "Non-local neural networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7794–7803.

Wang, Yequan, Minlie Huang, Li Zhao, et al. (2016). "Attention-based LSTM for aspect-level sentiment classification". In: *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp. 606–615.

Wittenburg, Peter et al. (2006). "ELAN: a professional framework for multimodality research". In: *5th International Conference on Language Resources and Evaluation (LREC 2006)*, pp. 1556–1559.

Wooding, David S (2002). "Eye movements of large populations: II. Deriving regions of interest, coverage, and similarity using fixation maps". In: *Behavior Research Methods, Instruments, & Computers* 34.4, pp. 518–528.

Xia, Chen, Fei Qi, and Guangming Shi (2016). "Bottom–up visual saliency estimation with deep autoencoder-based sparse reconstruction". In: *IEEE transactions on neural networks and learning systems* 27.6, pp. 1227–1240.

Xie, Saining et al. (2017). "Aggregated residual transformations for deep neural networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1492–1500.

Xu, Kelvin et al. (2015). "Show, attend and tell: Neural image caption generation with visual attention". In: *International conference on machine learning*, pp. 2048–2057.

Yang, Sheng et al. (2019). "SGDNet: An End-to-End Saliency-Guided Deep Neural Network for No-Reference Image Quality Assessment". In: *Proceedings of the 27th ACM International Conference on Multimedia*. MM '19. Nice, France: ACM, pp. 1383–1391. ISBN: 978-1-4503-6889-6. DOI: 10.1145/3343031.3350990. URL: http://doi.acm.org/10.1145/3343031.3350990.

Yarbus, Alfred L (2013). *Eye movements and vision*. Springer.

Yeager, Luke et al. (2015). "Digits: the deep learning gpu training system". In: *ICML 2015 AutoML Workshop*.

Yin, Yifang et al. (2019). "Multi-Level Fusion Based Class-aware Attention Model for Weakly Labeled Audio Tagging". In: *Proceedings of the 27th ACM International*

Conference on Multimedia. MM '19. Nice, France: ACM, pp. 1304–1312. ISBN: 978-1-4503-6889-6. DOI: 10.1145/3343031.3351090. URL: http://doi.acm.org/10.1145/3343031.3351090.

Zeiler, Matthew D and Rob Fergus (2014). "Visualizing and understanding convolutional networks". In: *European conference on computer vision*. Springer, pp. 818–833.

Zheng, Sipeng, Shizhe Chen, and Qin Jin (2019). "Visual Relation Detection with Multi-Level Attention". In: *Proceedings of the 27th ACM International Conference on Multimedia*. MM '19. Nice, France: ACM, pp. 121–129. ISBN: 978-1-4503-6889-6.

Zhong, Sheng-hua, Ahmed Fares, and Jianmin Jiang (2019). "An Attentional-LSTM for Improved Classification of Brain Activities Evoked by Images". In: *Proceedings of the 27th ACM International Conference on Multimedia*. MM '19. Nice, France: ACM, pp. 1295–1303. ISBN: 978-1-4503-6889-6. DOI: 10.1145/3343031.3350886. URL: http://doi.acm.org/10.1145/3343031.3350886.

Zhu, Guokang, Qi Wang, and Yuan Yuan (2014). "Tag-Saliency: Combining bottom-up and top-down information for saliency detection". In: *Computer Vision and Image Understanding* 118, pp. 40–49.

Zhu, Yongqing and Shuqiang Jiang (2019). "Attention-based Densely Connected LSTM for Video Captioning". In: *Proceedings of the 27th ACM International Conference on Multimedia*. MM '19. Nice, France: ACM, pp. 802–810. ISBN: 978-1-4503-6889-6.

Appendix A

Scientific products

International Journal

- A. Montoya, J. Benois-Pineau, M. S. G. Vázquez, and A. R. Acosta, "**Saliency-based Selection of Visual Content for Deep Convolutional Neural Networks**", in Journal of Multimedia Tools and Applications, 2018.

International Conferences

- A. Montoya, J. Benois-Pineau, M. S. G. Vázquez, and A. R. Acosta, "**Connosieur: Classification of Styles of Mexican Architectural Heritage with Deep Learning and Visual Attention Prediction**", in Proceedings of the 15th International Conference on Content-Based Multimedia Indexing (CBMI), Florence, Italy, June 19-21, 2017 (oral presentation).

- A. Montoya, J. Benois-Pineau, M. S. G. Vázquez, and A. R. Acosta, "**Introduction of Explicit Visual Saliency in Training of Deep CNNs: Application to Architectural Styles Classification**", in Proceedings of the 16th International Conference on Content-Based Multimedia Indexing (CBMI), La Rochelle, France, September 4-6, 2018 (oral presentation).

- A. Montoya, J. Benois-Pineau, M. S. G. Vázquez, and A. R. Acosta, K. Guissous and V. Gouet-Brunet, "**Comparative Study of Visual Saliency Maps in the Problem of Classification of Architectural Images with Deep CNNs**", in Proceedings of the 8th International Conference on Image Processing Theory, Tools and Applications (IPTA), Xian, China, November 7-10, 2018 (oral presentation).

- A. Montoya, J. Benois-Pineau, M. S. G. Vázquez, and A. R. Acosta, "**Dropping Activations in Convolutional Neural Networks with Visual Attention Maps**", in Proceedings of the 17th International Conference on Content-Based Multimedia Indexing (CBMI), Dublin, Ireland, September 4-6, 2019 (poster presentation).

- A. Montoya, J. Benois-Pineau, M. S. G. Vázquez, and A. R. Acosta, "**Organizing Cultural Heritage with Deep Features**", in 1st Workshop on Structuring and Understanding of Multimedia heritAge Contents (SUMAC), Nice, France, October 21, 2019 (poster presentation).
- A. Montoya, J. Benois-Pineau, M. S. G. Vázquez, and A. R. Acosta, "**Forward-backward Visual Saliency Propagation in Deep NNs vs Internal Attentional Mechanisms**", in Proceedings of the 9th International Conference on Image Processing Theory, Tools and Applications (IPTA), Istanbul, Turkey, November 6-9, 2019 (oral presentation).

Dataset

- A. Montoya, J. Benois-Pineau, M. S. G. Vázquez, and A. R. Acosta, "**Mexculture142**", a dataset in NAKALA Repository, https://www.nakala.fr/data/11280/5712e468.

Appendix B

Clustering evaluation metrics

For evaluation purposes, we first establish the following notation given a clustering solution,

- the set is composed of N samples,
- the classes $C = \{c_i \mid i = 1, ..., n\}$ and,
- the clusters $K = \{k_j \mid j = 1, ..., m\}$.

Then, consider Z denote a table produced by the clustering algorithm representing the grouping solution on the data. Such that, elements in $Z = \{z_{ij}\}$ are denoted as the number of data points which are members of c_i and elements of k_j.

Given any clustering solution, one can consider the following evaluation metrics to objectively discover how good the cluster solution is,

- homogeneity,
- completeness,
- v-measure and,
- silhouette coefficient.

For some metrics, the GT is required, i.e., homogeneity and completeness, in order to know the membership of any data point between clusters and classes. Then, we use GT to identify miss-classified samples given a clustering solution (Rosenberg and Hirschberg, 2007). To compute the silhouette coefficient, GT labels are not required.

Homogeneity

The clustering solution will satisfy homogeneity only if all the samples contained in each cluster are members of a single class. That is, zero entropy.

To determine how close is a clustering to the ideal solution, we compute the conditional entropy of class distribution given the clustering solution $H(C|K)$. Then, the coefficient is normalized by the maximum entropy that the clustering solution could provide, that is $H(C)$. Finally, the homogeneity coefficient is in the range $[0, 1]$, where the desired case to have a perfect homogeneity is 1.
Therefore, the homogeneity coefficient is defined as,

$$h = 1 - \frac{H(C|K)}{H(C)} \tag{B.1}$$

where $H(C|K)$ is the conditional entropy of the classes given clusters,

$$H(C|K) = -\sum_{c=1}^{|C|}\sum_{k=1}^{|K|}\frac{z_{c,k}}{n}\log\left(\frac{z_{c,k}}{n_k}\right), \tag{B.2}$$

and $H(C)$ the entropy of clustering solution,

$$H(C) = -\sum_{c=1}^{|C|}\frac{n_c}{n}\log\left(\frac{n_c}{n}\right) \tag{B.3}$$

In Eq. B.2 and Eq. B.3, n denotes the total number of samples, n_c and n_k the number of samples belonging to class c and cluster k, |C| denotes the number of classes and |K| the number of clusters. Finally, $z_{c,k}$ denotes the number of samples from class c assigned to cluster k, coming from table Z, the clustering solution.

Completeness

A clustering result will satisfy completeness when all the samples which are members of a single class are elements in a single cluster. In the case of conditional entropy of clusters given classes $H(K|C)$ and the entropy of clusters $H(K)$ are defined symmetrically to homogeneity. Then the completeness coefficient is as follows,

$$c = 1 - \frac{H(K|C)}{H(K)} \tag{B.4}$$

As in the case of homogeneity, the completeness coefficient is in the range $[0, 1]$, where 1 represents a perfect completeness.

V-measure[1]

This entropy-based measure explicitly evaluates how successfully previous criteria have been satisfied, homogeneity and completeness. This coefficient is computed by the harmonic mean of homogeneity and completeness,

$$V_\beta = \frac{(1+\beta) * h * c}{(\beta * h) + c} \tag{B.5}$$

here, β weights homogeneity or completeness. If β is greater than 1, completeness is weighted more strongly, in other case homogeneity is weighted more strongly if β is less than 1.

Silhouette

This coefficient is useful for the interpretation of clusters consistency. This value is in the range of $[-1, 1]$, where higher values represent well defined clusters, high density and well separated from other clusters (Rousseeuw, 1987).
The silhouette coefficient for a sample $i \in K_i$ is computed in two parts, similarity and dissimilarity,

- $a(i)$: measures the distance between a sample and all other samples in the same class, the similarity.
- $b(i)$: measures the mean distance between a(i) and all other samples in the nearest cluster, the dissimilarity.

First, for a single sample $i \in K_i$, we compute the mean distance between i and all other members of the same cluster to measure how well this value was assigned to its cluster, small is better,

$$a(i) = \frac{1}{|K_i| - 1} \sum_{j \in K_i, i \neq j} d(i, j). \tag{B.6}$$

Second, we define the dissimilarity of i as the mean distance between i and all the members in the closest cluster,

$$b(i) = \min \frac{1}{|K_i|} \sum_{j \in K_i, j \neq i} d(i, j). \tag{B.7}$$

Then, the silhouette value for i is defined as,

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \text{if } |K_i| > 1. \tag{B.8}$$

[1] "v" stands for validity.

Finally, the mean silhouette coefficient is computed over all $i \in K_i$.

The mean silhouette coefficient over all the samples measures the "shape" of the clusters, that is how well separated and dense are the samples clustered. A silhouette coefficient close to 1 tells that clusters are dense and well separated. Otherwise, with a silhouette coefficient close to -1, clusters contains sparse members and may be overlapped with close clusters in the solution.