UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR

MEJORA DEL RENDIMIENTO DE REDES CONVOLUCIONALES ENTRENADAS PARA EL RECONOCIMIENTO DE ESCENA MEDIANTE EL USO DE INFORMACIÓN SOBRE LOS OBJETOS COMUNES A ÉSTAS

Máster Universitario en Ingeniería de Telecomunicación

Raúl García Jiménez
Tutor: Marcos Escudero Viñolo
Ponente: Jesús Bescós Cano

-TRABAJO FIN DE MÁSTER-

Departmento de Tecnología Electrónica y las Comunicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Junio 2018

MEJORA DEL RENDIMIENTO DE REDES CONVOLUCIONALES ENTRENADAS PARA EL RECONOCIMIENTO DE ESCENA MEDIANTE EL USO DE INFORMACIÓN SOBRE LOS OBJETOS COMUNES A ÉSTAS

Raúl García Jiménez

Tutor: Marcos Escudero Viñolo
Ponente: Jesús Bescós Cano

Video Processing and Understanding Lab
Departamento de Tecnología Electónica y de las Comunicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Junio 2018

Trabajo parcialmente financiado por el Gobierno de España (MobiNetVideo, TEC2017-88169-R)

Resumen

Este trabajo estudia el rendimiento de esquemas basados en redes convolucionales en la tarea de reconocimiento automático de escenas. Comienza con un breve estudio cualitativo de cinco de las arquitecturas más utilizadas. Posteriormente se evalúa cuantitativamente el rendimiento de estas arquitecturas en la tarea de reconocimiento de escena, particularizando en la dependencia con la categoría y la arquitectura. Asimismo, se evalúan las salidas de las redes ante imágenes que representan categorías de escena no entrenadas, así como se explora la robustez de las soluciones analizadas a la existencia de ruido en la imagen de entrada.

Los resultados de este estudio cuantitativo motivan el diseño y desarrollo de un esquema para mejorar el rendimiento de las redes sin tener que reentrenarlas ni ajustarlas. Para ello, se esboza un mecanismo de refocalización de una de las redes convolucionales estudiadas, que denominaremos red convolucional de escena. Este esquema hace uso de una red neuronal complementaria que se entrena a partir de descripciones de la escena basadas en los objetos presentes en ella. Para obtener estas descripciones se hace uso de otra red convolucional que está diseñada y entrenada para obtener la segmentación semántica—asignación de cada píxel a una clase de objeto—de una imagen. Las anotaciones así obtenidas se ponderan por la focalización de la red convolucional de escena, dando más relevancia en la descripción a las zonas de la imagen con mayor impacto en la predicción.

Las predicciones de la red convolucional y la red neuronal de escena se comparan en un esquema iterativo de consenso. En este esquema, la imagen se va modificando gradualmente si las predicciones de ambas redes no coinciden, forzando a que la red convolucional utilice distintas zonas de la imagen para realizar la predicción. Los resultados preliminares en un subconjunto de la base de datos analizada son prometedores, alcanzando mejoras relativas del 8,85 % respecto al rendimiento de la red convolucional de escena.

Palabras Clave

reconcocimiento de escena, redes covolucionales, segmentación semántica, focalización.

Abstract

This work studies the performance of schemes based on convolutional networks in the task of automatic scenes recognition. The work begins with a brief qualitative study of five of the most used architectures. Subsequently, the performance of these architectures is evaluated quantitatively in the task of scene recognition, particularizing in the effect on the performance of the category and the nets architecture. Likewise, the responses of the networks to images that represent untrained scene categories are studied. Furthermore, the robustness of the analyzed solutions to image noise is also evaluated.

The results of this quantitative study motivate the design and development of a scheme to improve the performance of networks without having to retrain or adjust them. For this purpose, a refocusing mechanism of one of the studied convolutional networks is outlined. This scheme makes use of a complementary neural network that is trained from descriptions of the scene based on the objects present in it. To obtain these descriptions, another convolutional network that is designed and trained to obtain the semantic segmentation of an image is used. This network provides object-wise annotations of each image pixel. The annotations thereby obtained are weighted by a focusing information of the scene convolutional network, giving more relevance to the areas of the image with the greatest impact on prediction. The predictions of the convolutional network and this scene neural network are compared in an iterative consensus scheme. In this scheme, the image is gradually modified if the predictions of both networks do not coincide, forcing the convolutional network to rely on different areas of the image to predict the scene.

Preliminary results in a subset of the analyzed database are promising, reaching relative improvements of 8.85% with respect to the performance of the scene convolutional network.

Keywords

scene recognition, convolutional neuronal network, semantic segmentation, focusing.

Agradecimientos

En primer lugar, dar las gracias a mi tutor Marcos, por permitirme hacer este proyecto, ayudándome en todo momento en cada una de mis dudas y haciendo que llegue a la meta.

A mis padres, por apoyarme siempre durante este máster en los momentos más duros.

A mí chica Irene, que me ha acompañado en tantos días de estudio soportando el estrés conmigo y haciéndolo más llevadero.

A mis compañeros Álvaro y Jose, que tras muchas tardes en clase agobiados entre tanto estudio y tanta práctica siempre sacaban momentos de risas que hacen que lo eche de menos.

Gracias a todos.

Raúl García Jiménez.

2018.

Índice general

Índice de figuras

Índice de cuadros

Capítulo 1

Introducción

1.1. Motivación

Todas las imágenes son capturas de una determinada escena, considerando escena como el lugar que se representa en la imagen y contiene los objetos capturados. El campo de reconocimiento automático de escenas tiene como objetivo predecir la escena a partir de una imagen sin intermediación del usuario. Las potenciales aplicaciones del reconocimiento de escenas en el ámbito del análisis automático son variadas, desde aplicaciones de *life-logging*, pasando por sistemas de localización automática no basados en GPS, hasta sistemas de constricción automática para la detección de objetos basada en el contexto que suministra la escena.

Debido al amplio abanico de aplicaciones potenciales, esta tarea ha sido estudiada exhaustivamente en los últimos años, especialmente desde comienzos del siglo XXI [7]. Los recientes avances en el diseño de arquitecturas de reconocimiento basadas en redes convolucionales y la reciente creación de conjuntos de datos amplios y variados que permiten entrenarlas [6] han cambiado el paradigma.

Sin embargo, debido a la amplia variedad de sistemas de reconocimiento existentes y a la problemática de elegir una u otra en función de los requisitos de un problema particular, creemos que es necesario realizar un estudio cuantitativo de las ventajas e inconvenientes de cada sistema y modelo existentes que vaya más allá de su rendimiento global.

En particular, existe una amplia variedad de arquitecturas y modelos de redes convolucionales para el reconocimiento de escenas. En este trabajo se realiza un estudio detallado de cinco de los esquemas más utilizados, estudiando la dependencia de su rendimiento con el tipo de escena a reconocer, su robustez al ruido y la adaptabilidad de éstos esquemas para reconocer tipos de escenas no observados durante el

entrenamiento.

A partir de este estudio comparativo este trabajo también se motiva en la exploración de un esquema que permita mejorar el rendimiento de las redes haciendo uso de información sobre los objetos de la escena. Este esquema parte de la interpretación de la escena como contexto de captura, que constriñe los objetos que ésta puede contener y explora la viabilidad de establecer una relación en el otro sentido, es decir, dados los objetos, reconocer la escena.

Una de las motivaciones principales de este trabajo es también la de mejorar el rendimiento de la red evitando su re-entrenamiento, puesto que además de ser ineficiente, éste puede derivar en sistemas sobre-ajustados a las escenas observadas durante la creación del modelo.

1.2. Objetivos

En este trabajo de fin de máster se busca:

1. Realizar un estudio detallado de las ventajas e inconvenientes del mayor número posible de arquitecturas y modelos basados en redes convolucionales entrenadas para el reconocimiento de escena.

2. Analizar y evaluar las similitudes y las diferencias entre las arquitecturas y modelos evaluados.

3. Analizar la adaptabilidad y robustez al ruido de las arquitecturas y modelos evaluados.

4. Explorar y evaluar la viabilidad un esquema para mejorar el rendimiento de las arquitecturas evaluadas usando información de los objetos presentes en la escena sin necesidad de entrenar nuevos modelos.

1.3. Organización de la memoria

Este documento altera la organización ortodoxa de este tipo de trabajos: introducción, estado del arte, diseño y desarrollo, resultados y conclusiones. En particular, añade un capítulo donde se realiza el estudio comparativo entre los esquemas evaluados. Así, la organización de la memoria de este trabajo de fin de máster se estructura en seis capítulos:

- Capítulo 1. En este capítulo inicial se presenta la motivación, los objetivos y la estructura de este trabajo de fin de máster.

- Capítulo 2. En este capítulo se detalla el estado del arte, contado en qué consiste el reconocimiento de escena, explicando la base de datos utilizada, así como las características de las redes. Además, se describen las máscaras de focalización y la segmentación semántica, técnicas usadas para el desarrollo del sistema de re-focalización.
- Capítulo 3. En este capítulo de evaluación comparativa aparece el estudio realizado de las redes, detallando su funcionamiento, sus problemas, sus similitudes y diferencias, su adaptabilidad y su robustez al ruido.
- Capítulo 4. En este capítulo se presenta el sistema de re-focalización diseñado a partir de un esquema y posteriormente explicando cada módulo de este esquema.
- Capítulo 5. En este capítulo se detallan las pruebas realizadas y los resultados obtenidos.
- Capítulo 6. En este capítulo se enumeran las conclusiones alcanzadas tras el trabajo y se esboza el trabajo futuro a realizar.

Capítulo 2

Estado del arte

2.1. Reconocimiento de escena

2.1.1. Definición

El reconocimiento de escena consiste en, a partir de una imagen, obtener una predicción del lugar o la escena que en ella se muestra. Normalmente se asume que las imágenes que representan una escena son aquellas que son captadas desde una distancia mayor a 5 metros, de lo contrario se considerarían imágenes que representan objetos. En la actualidad, para esta tarea se utilizan generalmente redes convolucionales entrenadas con una base de datos de imágenes previamente etiquetadas con su correspondiente categoría. Las redes así entrenadas devuelven, dada una imagen, un vector de puntuaciones, una para cada una de las categorías para las que ha sido entrenada: a mayor puntuación, la red considera que la imagen se corresponde con mayor verosimilitud a dicha categoría.

2.1.2. Estrategias para el reconocimiento de escenas

Sistemas basados en características de la imagen

En los sistemas basados en características de la imagen como en [8] y [9] se realiza un reconocimiento de escena a partir de descriptores de tipo holístico (de toda la imagen) o de tipo local (de áreas y objetos presentes en la escena). En otros trabajos se hace uso de descriptores de la imagen que se basan en la estructura o forma de la escena, como en [7] por la envolvente de escena o como en [10] con la representación de esencia (*gist*).

Sistemas basados en el uso de redes convolucionales

Con la llegada de las redes convolucionales (ConvNets) se consiguió un gran avance en el reconocimiento de escena. Con ellas se ha conseguido analizar conjuntos de datos de tamaño medio [11] con un acierto global de casi un 70 % [12], un acierto que está en consonancia con el reconocimiento de escena humano. Actualmente también existen experimentos en los que se usan ConvNets entrenadas para el reconocimiento de escena para la tarea de reconocimiento de objetos [13]. Los resultados obtenidos sugieren que durante el entrenamiento, la red convolucional aprende los objetos comunes a una escena además de la escena. Exploraremos las soluciones existentes en la sección 2.2.

2.1.3. Bases de datos existentes

La base de datos (*dataset*) de imágenes utilizada para realizar este trabajo de fin de máster es Places365, disponible en [6]. En el Apéndice A se enumeran las categorías de escena allí definidas y se incluye una imagen de ejemplo de cada una de ellas. Esta base de datos se estructura en categorías de escena, en concreto en 365 categorías. En este trabajo usaremos todo el conjunto de validación, donde cada categoría contiene 100 imágenes, hasta un total de 36.500 imágenes, y un subconjunto de imágenes de entrenamiento, 1.000 imágenes por categoría, hasta un total de 365.000 imágenes seleccionadas en cada categoría del total de 1,8 millones de imágenes disponibles.

Cabe destacar que la mayoría de imágenes en la bases de datos son imágenes de una escena en concreto, que pueden contener personas ocluyendo parcialmente. Pero también, hay representaciones pictóricas (dibujos) de la escena, lo que implica que los esquemas de reconocimiento deben ser flexibles a estas distintas modalidades de imagen.

2.2. Definiciones generales sobre ConvNets

No es el objetivo de este trabajo fin de máster el de detallar el funcionamiento de las ConvNets, entendemos las ConvNets como son un tipo de redes neuronales capaces de clasificar un dato de entrada aportando una salida, tras un entrenamiento previo realizado a partir de una gran cantidad de datos. Están formadas por neuronas interconectadas entre sí que a su vez forman capas, siendo el número de estas capas la profundidad de la red. Estas neuronas tienen como entrada neuronas de la capa anterior y su salida alimenta una o más neuronas de la siguiente capa. Cada una de estas neuronas tiene un peso asociado que se obtiene a partir del entrenamiento de la red. En definitiva, una red entrenada está definida por su **arquitectura** (organización

de las neuronas y sus conexiones) y por los **pesos** aprendidos o ajustados durante el entrenamiento. La particularidad de las redes convolucionales es que estos pesos se organizan en filtros multi-dimensionales (*kernels*) que se aplican de manera local sobre datos multi-dimensionales a la entrada de cada capa convolucional. Además de las capas convolucionales, existen capas adicionales dentro de las arquitecturas ConvNet, en particular, algunos tipos de capas adicionales que utilizan las redes analizadas en este trabajo son:

- *Fully-connected*: Todas las neuronas de la capa se encuentran conectadas con las de la capa contigua.
- *Max-pooling*: Reduce el tamaño de la imagen.
- *ReLU*: La salida es igual a la entrada si es positiva y 0 en caso contrario.
- *Softmax*: Última capa que adapta el resultado de la red a un número determinado de salidas en un rango de valores.
- *Pyramyd-pooling*: Extrae características de la imagen en subregiones de distintos tamaños.

En este trabajo se han evaluado cinco redes para el reconocimiento de escena a partir de imágenes. Todas las redes tienen al final una capa *softmax*, que permite que éstas devuelvan una puntuación entre 0 y 1 para cada categoría de escena entrenada. Las redes han sido entrenadas con el conjunto de datos total del Places365 [6] (no con el subconjuto utilizado en este trabajo). Por tanto están orientadas al reconocimiento de las 365 categorías allí definidas. Las puntuaciones están normalizadas, de forma que las puntuaciones para las 365 categorías en una salida suman 1 entre ellas, es decir, podemos entender que las verosimilitudes a la salida de las redes son probabilidades.

2.2.1. Arquitecturas analizadas

Las arquitecturas analizadas son: Alexnet [1], VGG16 [2], Resnet18 [3], Resnet50 [3] y Densenet161 [4]. En esta sección describiremos brevemente cada una de ellas, en orden cronológico, enfatizando las diferencias respecto a las arquitecturas anteriores. El cuadro resumen Cuadro 2.1 aglutina las diferencias principales entre las redes. En ella puede observarse, por ejemplo, que la red menos compleja y profunda es Alexnet [1] y su contrapartida, la de mayor profundidad, es Densenet161 [4].

Red	Lineal/Recursivo	Nº capas	Primera capa	Kernel Size
Alexnet [1]	Lineal	11	227x227	11 y 5
VGG16 [2]	Lineal	16	224x224	3x3
Resnet18 [3]	Recursivo	18	224x224	3x3
Resnet50 [3]	Recursivo	50	224x224	3x3
Densenet161	Recursivo	161	224x224	1x1 & 3x3

Cuadro 2.1: Comparativa de las redes neuronales usadas

Alexnet

La red Alexnet [1] es una de las primeras redes convolucionales, su aparición supuso un punto de inflexión en el análisis automático de imágenes. Surge para el reconocimiento de objetos pero ha sido entrenada para diversas tareas, entre ellas el reconocimiento de escena. La red se compone de 8 capas: 5 de ellas convolucionales de resolución y profundidad variable y las otras 3 *fully-connected* de 4096 neuronas cada una. La salida de la última de estas capas se conecta con una capa *softmax* que se ajusta al problema a resolver. La arquitectura de Alexnet [1] ajustada a un problema de 1000 clases se incluye en la Figura 2.1.

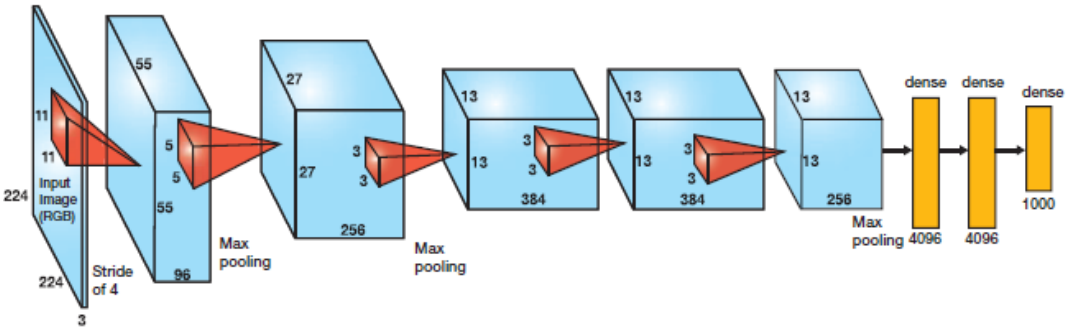

Figura 2.1: Esquema de la arquitectura de Alexnet [1]. Adaptada de [14].

VGG

Su principal ventaja frente a sus antecesores es que, al introducir una mayor profundidad (más capas), se incluyen más filtros que, en teoría, suministran una mejora de rendimiento respecto a redes anteriores. En particular, la red VGG [2] puede entenderse como una mejora de Alexnet [1], remplazando los *kernels* 11x11 y 5x5 por múltiples 3x3, lo que permite definir una red más profunda, que en teoría es capaz de modelar tareas más complejas. Inicialmente VGG [2] se diseñó para tareas de reconocimiento de imágenes, pero su adaptabilidad y la existencia de soluciones

software consolidadas han permitido su uso para múltiples tareas. La red VGG [2] es una red no recursiva al igual que Alexnet [1].

Como normalización de las imágenes a la entrada, se les resta el valor RGB medio, calculado en el conjunto de entrenamiento, de cada píxel. La imagen pasa por 5 capas convolucionales con filtros 3×3. En una de las configuraciones se utilizan filtros 1×1, que se pueden ver como una transformación lineal de los canales de entrada. A continuación siguen 5 capas *max-pooling*, a las que siguen diversas capas convolucionales. El *max-pooling* se realiza en una ventana de 2×2, con un paso de 2. Después hay 3 capas *fully-connected*: las dos primeras tienen 4096 canales y la tercera realiza una clasificación que se adapta al problema a resolver. Finalmente, la última capa es la capa *soft-max*, también adaptada al problema. Incluye varias capas de rectificación ReLU que permiten explorar relaciones no lineales. La arquitectura de la red VGG [2] se puede observar en la Figura 2.2.

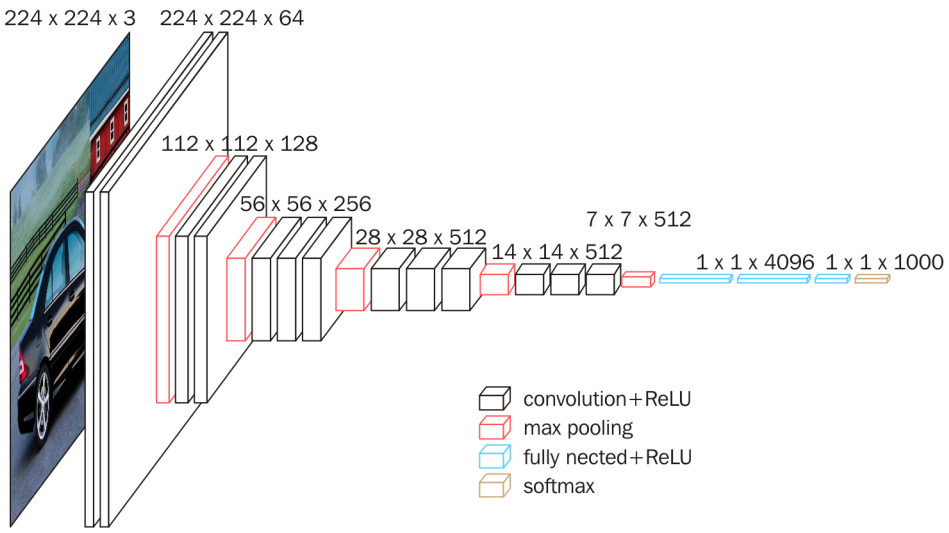

Figura 2.2: Esquema de la arquitectura de VGG [2]. Adaptada de [15].

Resnet

A diferencia de las redes anteriores, Resnet [3] es una red recursiva. Surge como un avance a VGG [2] que permite una mayor especialización de la red gracias a la recursividad. Esta recursividad permite en teoría alcanzar una mayor profundidad efectiva, a diferencia de lo que ocurre en una red no recursiva muy profunda, donde sus primeras capas casi no se especializan y son muy similares independientemente del problema a resolver. Al conseguir una mayor profundidad se espera un mayor rendimiento en tareas complejas. Como para las redes anteriores, la tarea principal

para la que fueron entrenadas fue el reconocimiento de imágenes, pero existen modelos Resnet para múltiples tareas.

Esta red tiene menos filtros y una menor complejidad que las redes VGG [2]. La novedad que implementa esta red es que aparecen conexiones entre capas haciendo de ésta una red residual. Las conexiones directas se pueden usar cuando la entrada de una capa y la salida de la otra tienen el mismo tamaño. Cuando el tamaño aumenta, o bien se siguen usando estas conexiones directas añadiendo ceros para aumentar el tamaño de una de ellas, o se usan conexiones de proyección para unir tamaños. En la Figura 2.3 se puede observar un ejemplo de una conexión residual entre capas.

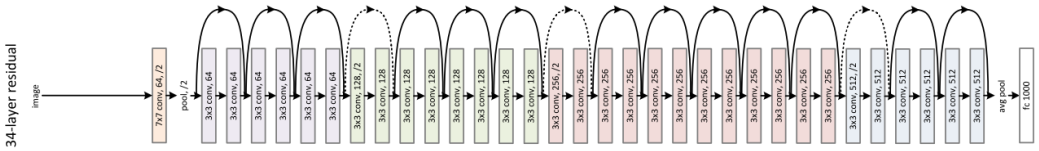

Figura 2.3: Esquema de una conexión residual entre capas de Resnet [3], en este caso de 34 capas. Adaptada de [3]

Densenet

La red Densenet [4] es también una red recursiva. En particular, surge como una mejora de las redes recursivas anteriores que permite utilizar este esquema en redes con mayor profundidad, gracias a una hipotética mayor velocidad de convergencia en entrenamiento. La idea es que en esta red todas las capas están conectadas con todas, por lo que cada capa utiliza los datos de las demás, lo que produce un mejor resultado en términos de eficiencia durante el entrenamiento. Como para las redes anteriores, la tarea principal para la que fueron entrenadas fue el reconocimiento de imágenes, pero existen modelos Densenet para otras tareas, incluyendo el reconocimiento de escena.

Esta red tiene una arquitectura que busca que todas las capas tengan la máxima información posible del resto de capas, para ello se conectan todas las capas entre ellas (con el mismo tamaño). De esta forma, cada capa tiene entradas de todas las capas anteriores y a su vez conecta su salida a todas las capas posteriores. Esto provoca que una red de L capas tenga $\frac{L(L+1)}{2}$ conexiones. Esta arquitectura además hace que sea más sencillo entrenar redes más profundas, ya que todas las capas tienen acceso directo a la información del resto de capas. En la Figura 2.4 se puede observar un ejemplo de esta arquitectura.

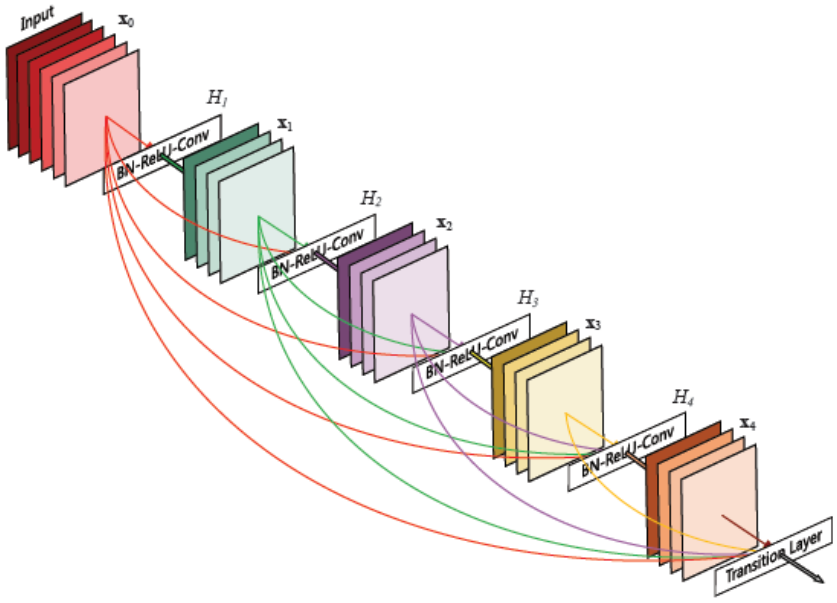

Figura 2.4: Esquema de la arquitectura de Densenet [4]. Adaptada de [4].

Soluciones utilizadas

Por su lado, Alexnet [1] y VGG16 [2] han sido ejecutadas en Matlab2017b mediante Caffe [[16]] que incluye el toolbox de Neural Network con el cual se puede cargar una red neuronal a partir de un fichero que contiene la arquitectura de la red y un fichero con los pesos. Por su parte, Resnet18 [3], Resnet50 [3] y Densenet161 [4] han sido ejecutadas en una máquina virtual de Ubuntu a partir de un *script* de Python que usa Pytorch. Además, la red Alexnet [1] también se ha podido ejecutar con Pytorch para comprobar sus diferencias, ya que en Matlab es obligatorio redimensionar las imágenes a un tamaño fijo antes de introducirlas en la red. Las similitudes y diferencias entre las redes, en términos de implementación utilizada pueden observarse en la tabla comparativa del Cuadro 2.2. Todas las redes ejecutadas en Matlab son redes no recursivas, esta funcionalidad en Matlab no se incluyó hasta 2018, cuando este trabajo ya estaba avanzado.

2.3. Máscaras de focalización

Se denominan máscaras de focalización a los mapas de activación [17] de las imágenes tras pasar por una red y aportar una predicción. Estos mapas, del mismo tamaño que la imagen evaluada, contienen un valor entre 0 a 1 en cada uno de sus píxeles, representando el grado de relevancia de ese píxel en la predicción realizada por la

Red	Solución software	Versiones
Alexnet [1]	Pytorch/Caffe	-
VGG16 [2]	Caffe	11, 13, 16 y 19 layers
Resnet18 [3]	Pytorch	18, 34, 50, 101 y 152 layers
Resnet50 [3]	Pytorch	18, 34, 50, 101 y 152 layers
Densenet161 [4]	Pytorch	121, 161, 169, 201 y 264 layers

Cuadro 2.2: Comparativa 2 de las redes neuronales usadas

red. Para obtener este mapa de activación, se vuelve para atrás en la última capa para obtener los pesos de capas anteriores que han derivado en la predicción. Esta estructura se puede observar en la Figura 2.5.

Figura 2.5: Esquema de la extración del mapa de activación de una imagen. Se observa que antes de la última capa, se obtienen los pesos de la capa anterior para extraer el mapa de activación. Adaptada de [17].

2.4. Segmentación semántica

La segmentación consiste en la división de una imagen en un conjunto de segmentos que no se superponen entre ellos. En el ámbito del reconocimiento de escenas se busca la partición en segmentos semánticamente significativos, de modo que cada segmento abarque un objeto o área en la imagen. En este trabajo, se hará uso de la red Pyramid Scene Parsing Net [5].

PSPNet

Esta red se basa en una arquitectura Resnet [3] para obtener en primer lugar el mapa de características de la imagen en la última capa convolucional, al que posterior-

mente se le aplica el módulo *pyramid pooling* obteniendo diferentes representaciones en subregiones. A continuación, hay una serie de capas de muestreo y de concatenación formando el conjunto de características final con información local y global de la imagen. Finalmente, pasa por una capa de convolución final obteniendo la predicción de cada píxel de la imagen. La red utilizada devuelve un valor o clase para cada píxel que se corresponde con la etiqueta correspondiente en el *dataset* ADE20K usado para su entrenamiento, compuesto por instancias de 150 objetos, como se detalla en el Apéndice B. Esta arquitectura de red se puede observar en la Figura 2.6.

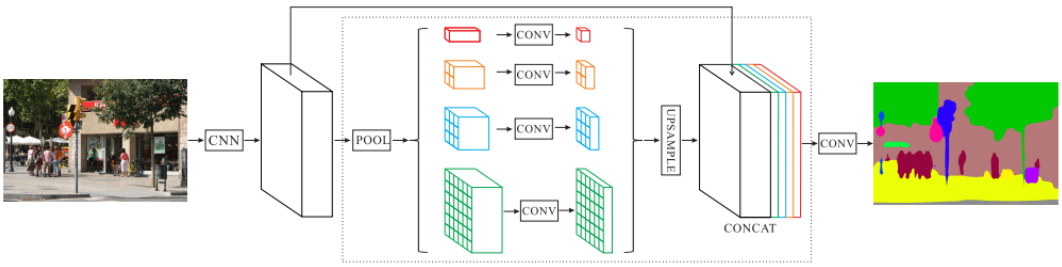

Figura 2.6: Esquema de la red PSPNet [5] en la que se observa una imagen a la izquierda, que es la entrada, a la que se le extrae su mapa de características y se introduce en el módulo *pyramid pooling* (zona interior a la línea de puntos) y a su salida se obtiene la imagen de predicción. Adaptada de[5].

Capítulo 3

Evaluación de las soluciones existentes

3.1. Protocolo de evaluación

Para evaluar el funcionamiento de las redes se han usado las imágenes de validación de la base de datos de Places365 [6], las cuales son 36.500 imágenes, 100 por cada categoría. En este capítulo se incluyen evaluaciones del rendimiento de las redes y evaluaciones sobre la robustez al ruido y a escenas no observadas durante el entrenamiento.

3.1.1. Funcionamiento de las redes

En primer lugar se evalúa el rendimiento global de las redes en términos de tasa de acierto global. Para ello, se han comprobado los resultados obtenidos al introducir las imágenes de validación en cada una de las redes preentrenadas para después comprobar si la salida con una mayor puntuación se corresponde con la correcta (que definiremos como la anotada en el *dataset*). Posteriormente se analizan problemas de las redes respecto a las categorías de escena. Para ello, se observan las salidas de las redes, pero en este caso centrándose en cada categoría en concreto, para comprobar si existe comportamiento singular de una o varias redes para una o varias clases.

Dado que las redes están entrenadas con un mismo *dataset* de imágenes de entrenamiento, éstas tenderán a tener un comportamiento similar, salvando las diferencias derivadas del uso de arquitecturas diferentes. Para evaluar estas diferencias experimentalmente, se organizan los resultados obtenidos en términos de profundidad y las arquitectura de cada red, y se mide la mejora de rendimiento obtenida al utilizar redes más complejas. Además, para cuantificar las diferencias entre redes, se ha medido la

correlación entre sus probabilidades de salida, así como el número de veces que dos redes dan la misma salida para cada categoría, ya sea acertando o equivocándose en la misma categoría errónea.

3.1.2. Adaptabilidad de las redes

Dado que las redes están entrenadas para predecir sólo las 365 categorías utilizadas en el entrenamiento, consideramos interesante observar los resultados de las redes ante imágenes de escenas no catalogadas durante el entrenamiento. De esta manera se puede comprobar si las distintas redes actúan de una forma similar o cada una obtiene distintos resultados ante categorías no observadas, así como evaluar si existen patrones dada una nueva categoría: escenas como combinación de escenas. Si bien este experimento no ha centrado el resto del trabajo fin de máster, creemos que puede ser de utilidad para posteriores trabajos. Para realizar este estudio se han evaluado todas las redes con 500 imágenes nuevas, distribuidas en 5 categorías (100 imágenes cada una) nuevas de escena: *bodega*, *canchal*, *estudio_fotos*, *frontón* y *sala_revelado*.

Finalmente, para evaluar la sensibilidad de las redes al ruido, en términos de su rendimiento, se han vuelto a analizar las salidas de las redes a versiones modificadas de las imágenes de validación, en particular, las imágenes se han modificado incluyendo píxeles nulos en proporciones del 1 %, 10 %, 25 % y 50 % del total de píxeles de la imagen. Los píxeles a anular se han seleccionado aleatoriamente con distribución uniforme, a fin de evitar la concentración de píxeles anulados en áreas de las imágenes. Si los píxeles se concentran en una zona de la imagen, podría suceder que la salida de la red variara en exceso. Por ejemplo, si la escena es *art_gallery* y los píxeles nulos se concentrasen sobre una "*obra de arte*" de la imagen podría suceder que la red cambie su salida completamente porque este objeto es de especial importancia para predecir la escena. Por otro lado, si la red se está equivocando diciendo que una imagen es *art_gallery* porque aparece una "*obra de arte*", cuando en realidad no lo es, anular estos píxeles podría mejorar artificialmente el rendimiento de la red.

3.1.3. Nomenclatura utilizada

Las arquitecturas utilizadas son Alexnet [1], VGG16 [2], Resnet18 [3], Resnet50 [3] y Densenet161 [4]. Además, dada la posibilidad de usar Alexnet [1] en Matlab y en Pytorch, se han evaluado ambas implementaciones de ésta red, denominándolas Alexnet_M [1] y Alexnet_P [1] respectivamente, aunque en los experimentos de adaptabilidad y robustez sólo se ha analizado el rendimiento de Alexnet_M [1]. También se ha evaluado el impacto de la resolución de imagen en la predicción, usando

imágenes a resolución completa (*_large*) y versiones reducidas (256x256).

3.2. Funcionamiento de las redes

3.2.1. Descripción

En primer lugar, se estudia el rendimiento de cada red en términos del número de aciertos, simplemente contando el número de veces que la categoría anotada en la base de datos coincide con la que tiene mayor puntuación a la salida de la red (Top 1). También se han tenido en cuenta las veces que la categoría correcta está entre las tres (Top 3) y entre las diez (Top 10) de mayor puntuación.

Cabría esperar que a mayor profundidad de la red se obtuviese una mayor tasa de acierto. Para cuantificar el beneficio de complicar la arquitectura, se propone calcular la mejora relativa que proporciona una red sobre la red cuyo acierto global es inmediatamente inferior:

$$\triangle_{21} = \frac{\triangle_2 - \triangle_1}{\triangle_1} \tag{3.1}$$

, donde $\triangle_1$ es el número de aciertos de una red y $\triangle_2$ es el número de aciertos de la siguiente red con mayor acierto global.

Asimismo, se explora la similitud entre cualesquiera dos redes analizando las coincidencias entre la salida de mayor puntuación obtenida en cada red y se desglosan las categorías que provocan mayor y menor coincidencia entre pares de redes. Para ello, se mide el número de veces que aciertan dos redes y el número de veces que se equivocan prediciendo la misma categoría. Finalmente, la dependencia con la categoría se particularizará en dos categorías: *building_facade* y *river*. La elección de estas categorías se debe al hecho de que ambas son para las que se obtiene un menor acierto tanto global como particular para cada red.

En las redes Resnet18 [3], Resnet50 [3] y Alexnet_P [1] se ha observado un comportamiento peculiar en algunas categorías, en concreto en tres categorías de lugares de desierto (*desert_sand*, *desert_vegetation* y *desert_road*) y dos de campo (*field_wild* y *field_road*). Las salidas de las redes para estas categorías se encontraban intercambiadas entre ellas, obteniendo una puntuación alta para otra de ellas distinta a la anotada. Este hecho parece apuntar a que existe un error durante el entrenamiento de estas redes, ya que, claramente clasifica las imágenes de una categoría como otra categoría. Por esta razón, para estos y el resto de experimentos se han intercambiado estas salidas.

Red	Top-1	Top-3	Top-10
Alexnet_M	42.80 %	64.94 %	84.12 %
Alexnet_P	48.28 %	70.59 %	87.33 %
Alexnet_M_large	43.33 %	65.58 %	84.33 %
Alexnet_P_large	45.33 %	67.26 %	85.31 %
VGG16	46.01 %	68.47 %	86.81 %
Resnet18	54.45 %	76.97 %	91.91 %
Resnet50	55.54 %	78.22 %	92.89 %
Resnet50_large	53.80 %	76.55 %	92.04 %
Densenet161	56.10 %	78.65 %	93.14 %

Cuadro 3.1: Porcentajes de aciertos de las redes teniendo en cuenta que la categoría correcta está en el Top-1, 3 o 10.

	Profundidad	Nº Aciertos	$\triangle_{21}$
Alexnet	8	15.622	-
VGG16	16	16.794	7,50 %
Resnet18	18	19.874	18,34 %
Resnet50	50	20.272	2,00 %
Densenet161	161	20.477	1,01 %

Cuadro 3.2: Comparativa de las redes entre su profundidad y su número de aciertos.

3.2.2. Resultados comparativos

El Cuadro 3.1 representa el porcentaje de aciertos de cada una de las arquitecturas y modelos evaluados. En la Figura 3.1, se observan los aciertos de cada red en todas las categorías. El Cuadro 3.2 presenta una comparativa entre la profundidad de las redes y el número de aciertos. La Figura 3.2 muestra la matriz de confusión de las salidas de las redes, de forma que a mayor valor, más salidas coinciden. En el siguiente Cuadro 3.3 se pueden ver las categorías con menos coincidencias entre redes y en el Cuadro 3.4, las categorías con mayor coincidencias entre redes.

Si nos centramos en las categorías con menos aciertos en todas las redes, que son *building_facade* y *river*, observamos que para estas categorías se obtiene un acierto menor al 10 % independientemente de la red evaluada. El Cuadro 3.5 resume este comportamiento comparando el número de aciertos de cada red para las categorías *building_facade* y *river*. En la figuras 3.3 y 3.4 se incluyen las puntuaciones obtenidas en cada categoría para las 100 imágenes de *building_facade* y *river* respectivamente.

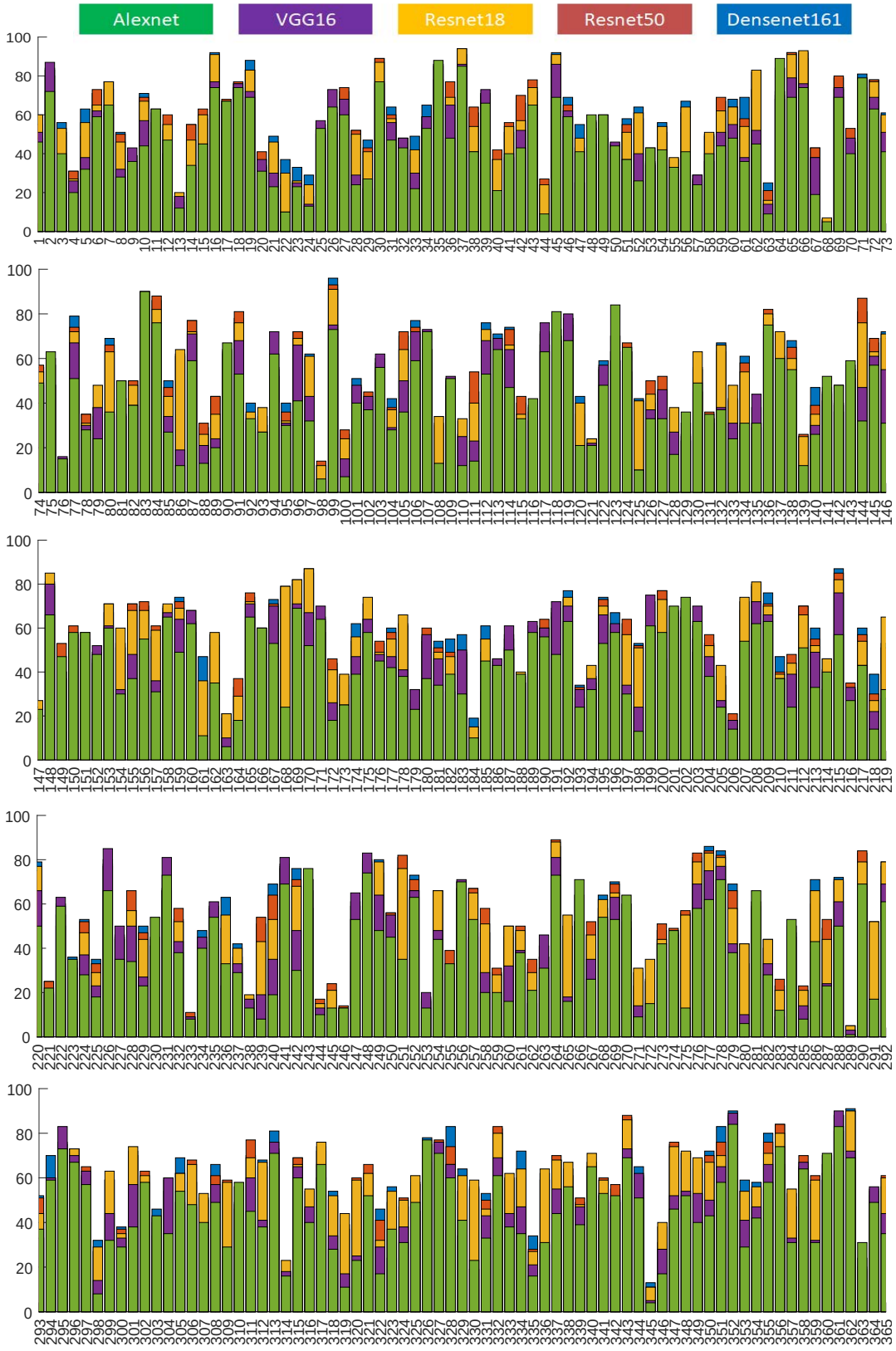

Figura 3.1: Gráfica de barras en las que el eje X muestra categorías y el eje Y el número de aciertos de cada red. Los resultados se superponen: Los resultados de la red de mejor rendimiento (Densenet161 [4]) se sitúan detrás de los resultados de los demás.

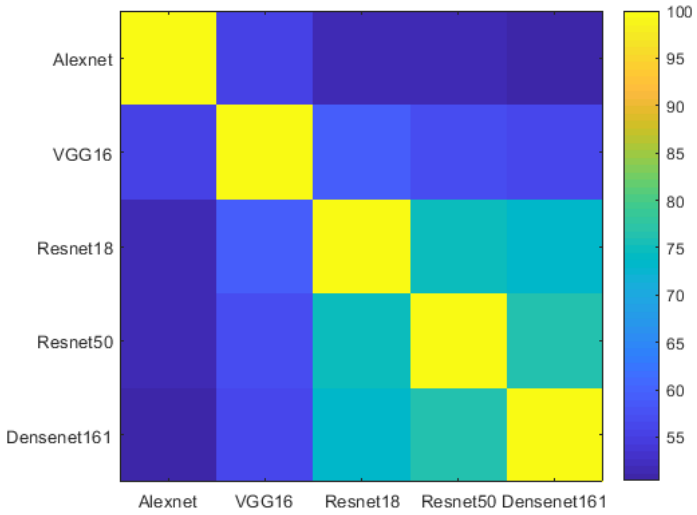

Figura 3.2: Matriz de confusión de la correlación entre las redes en las que el valor que relaciona cada red representa el porcentaje de salidas que coinciden entre las redes, de forma que ambas aciertan o se equivocan de igual forma.

	Alexnet	VGG16	Resnet18	Resnet50	Densenet161
Alexnet	-	*science_museum* (29)	*music_studio* (19)	*music_studio* (16)	*science_museum* (15)
VGG16	*science_museum* (29)	-	*pub/indoor* (17)	*pub/indoor* (16)	*pub/indoor* (21)
Resnet18	*music_studio* (19)	*pub/indoor* (17)	-	*mansion* (53)	*artist_loft* (51)
Resnet50	*music_studio* (16)	*pub/indoor* (16)	*mansion* (53)	-	*valley* (56)
Densenet161	*science_museum* (15)	*pub/indoor* (21)	*artist_loft* (51)	*valley* (56)	-

Cuadro 3.3: Cuadro que muestra en cada relación entre redes la categoría con menos coincidencias y entre paréntesis se muestran el número de veces que coinciden en las imágenes de validación. El número de imágenes por categoría es 100.

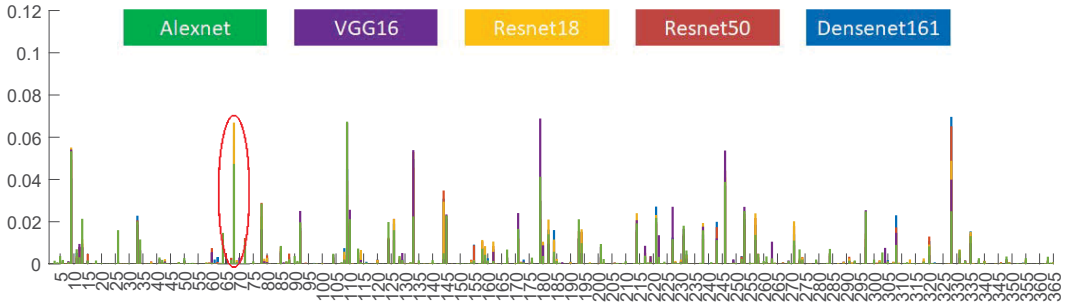

Figura 3.3: Gráfica en la que el eje X muestra categorías y el eje Y las puntuaciones obtenidas en cada categoría en las 100 imágenes de validación para la categoría *building_facade*, rodeada con un círculo rojo.

	Alexnet	VGG16	Resnet18	Resnet50	Densenet161
Alexnet	-	*volleyball_court/outdoor* (89)	*car_interior*, *volleyball_court/outdoor* (86)	*car_interior* (87)	*volleyball_court/outdoor* (87)
VGG16	*volleyball_court/outdoor* (89)	-	*volleyball_court/outdoor* (92)	*wind_farm* (89)	*volleyball_court/outdoor* (91)
Resnet18	*car_interior*, *volleyball_court/outdoor* (86)	*volleyball_court/outdoor* (92)	-	*cockpit* (98)	*bowling_alley* (95)
Resnet50	*car_interior* (87)	*wind_farm* (89)	*cockpit* (98)	-	*arena/hockey* (97)
Densenet161	*volleyball_court/outdoor* (87)	*volleyball_court/outdoor* (91)	*bowling_alley* (95)	*arena/hockey* (97)	-

Cuadro 3.4: Cuadro que muestra en cada relación entre redes la categoría con más coincidencias y entre paréntesis se muestran el número de veces que coinciden en las imágenes de validación. El número de imágenes por categoría es 100.

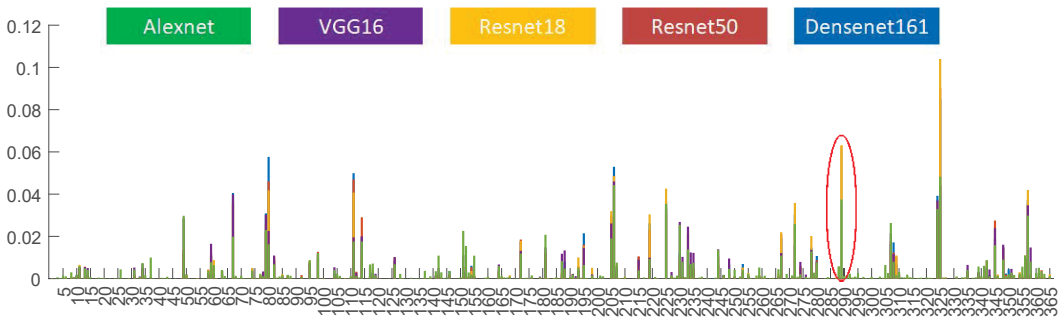

Figura 3.4: Gráfica en la que el eje X muestra categorías y el eje Y las puntuaciones obtenidas en cada categoría en las 100 imágenes de validación para la categoría *river*, rodeada con un círculo rojo.

Red	*building_facade*	*river*
Alexnet	5	1
VGG16	3	3
Resnet18	7	5
Resnet50	6	5
Densenet161	6	3

Cuadro 3.5: Número de aciertos de las redes en las 100 imágenes de validación en las escenas *building_facade* y *river*.

3.2.3. Discusión

Rendimiento en función de la implementación utilizada:

En el Cuadro 3.1 se observa que en comparación, el resultado de Alexnet_P [1] es mejor que de Alexnet_M [1]. Esto puede deberse al redimensionado de las imágenes a la entrada de la red en Matlab, hecho que también puede ser el causante del poco cambio en los resultados obtenidos en Matlab con las imágenes High Resolution. Además, destaca que con Alexnet_P [1] y con Resnet50 [3] se obtenga un mejor resultado con las imágenes de tamaño 256x256 que con las imágenes en su tamaño original en Alexnet_P_large [1] y Resnet50_large [3] respectivamente. A la vista de estos resultados se puede concluir que Densenet161 [4] es la red con mejor resultado, como era de esperar, ya que es la red de mayor profundidad y por lo tanto más compleja. Aún así, una red de menor profundidad como Resnet18 [3], aporta un resultado similar a Densenet161 [4] con menor profundidad y una predicción por lo tanto más eficiente. Además, se observa un salto en los resultados entre las redes Alexnet [1] y VGG16 [2] y las redes no lineales Resnet18 [3], Resnet50 [3] y Densenet161 [4], por lo que se puede concluir que el redimensionado que se realiza obligatoriamente en Matlab perjudica los resultados.

Rendimiento en función de la arquitectura:

En el Cuadro 3.1 se puede observar como aumenta el porcentaje de acierto a mayor profundidad de la red. Aún así, se puede observar que la mejora no es proporcional al número de capas añadidas, por ejemplo, las distintas versiones de Resnet [3], a pesar de una de ellas tener 32 capas más sólo aumenta un 1,09 % en el Top-1 y prácticamente igual en el Top-3 y el Top-10. Entre Resnet50 [3] y Densenet161 [4] una diferencia de 111 capas y un incremento absoluto del 0,56 % más de acierto. En términos de incremento relativo, tal y como se observa en el Cuadro 3.2, al complicar la arquitectura y aumentar la profundidad de las redes se obtiene una mejora relativa

de los resultados, pero esta mejora cada vez va siendo menor, por lo que llega un momento que ya no compensa modificar más la red ya que la mejora es mínima respecto del coste de eficiencia que produce este cambio: el incremento relativo de acierto no es proporcional al aumento de profundidad. En relación a la correlación entre las salidas de las redes, en el Cuadro 3.2 se observa que las redes Alexnet [1] y VGG16 [2] son las que menor puntuación tienen en términos de correlación, tanto entre ellas como con las demás, mientras que Resnet18 [3], Resnet50 [3] y Densenet161 [4] muestran una alta correlación entre ellas. Las dos redes con peor relación entre sí son Alexnet [1] y Densenet161 [4] y, por otro lado, las dos redes que mayor relación tienen entre ellas son Resnet50 [3] y Densenet161 [4], como cabría esperar, puesto que son las dos redes con más aciertos, lo que provoca que haya más probabilidad de correlación.

En base a esta discusión se ha seleccionado la red Resnet50 [3] como solución de compromiso entre eficacia y eficiencia. Esta arquitectura de utilizará en los experimentos del capítulo 5.

Rendimiento en función de la categoría:

Se observa en la Figura 3.3 y en la Figura 3.4 cómo las categorías que resultan en una tasa de acierto baja presentan una distribución multi-modal de puntuaciones a la salida de las redes, es decir existen máximos locales adicionales en otras categorías además de para la correcta. Además se observa que todas las redes se equivocan en estas categorías, lo que podría sugerir que los datos usados para el entrenamiento no han sido los *adecuados*. Como se observa en el Cuadro 3.5, en la categoría *building_facade*, Densenet161 [4] a pesar de ser la red más profunda no es la que mejor resultado retorna, sino que acierta el mismo número de veces que Resnet50 [3]. La que mejor resultado retorna para estas categorías es en cambio Resnet18, que acierta una vez más que Resnet50. En la categoría *river,* al igual que con la categoría *building_facade* la red más profunda no es la que mejor resultado retorna. En este caso las mejores son las dos redes Resnet, que a pesar de tener distinta profundidad tienen el mismo número de aciertos. Por tanto se puede establecer que, en líneas generales, al aumentar la profundidad de una red mejora la tasa de aciertos, pero en aquellas categorías que presentan problemas, introducir mayor complejidad en la red no mejora sustancialmente el rendimiento. Esta idea se refuerza en los resultados del Cuadro 3.3, donde las categorías con menor número de coincidencias son categorías con baja tasa de acierto general. Lo contrario sucede con las categorías de mayor número de aciertos global, que resultan ser aquellas para las que se produce un mayor número de coincidencias, como se indica en el Cuadro 3.4. Ambos resultados parecen sugerir

Figura 3.5: En esta figura se representa un ejemplo de una de las imágenes usadas, siendo de izquierda a derecha: La imagen original, con 1 %, con 10 %, con 25 % y con 50 %.

que estas categorías pueden considerarse *difíciles* (o *fáciles*) de discriminar del resto. En definitiva, la categoría parece determinante para determinar el rendimiento.

3.3. Adaptabilidad de las redes

3.3.1. Descripción

A partir de imágenes que no cumplen con ninguna de las categorías contempladas se ha observado el comportamiento de las redes, para ver si es similar o distinto en distintas redes y si dan una categoría con mucha puntuación o por el contrario, no existe una puntuación máxima clara en una única categoría. Las categorías que se han usado son: *bodega, canchal, estudio_fotográfico, frontón y sala_revelado.*

Para evaluar la robustez al ruido de las redes, se han evaluado las imágenes de validación, modificando un número determinado de píxeles poniendo los tres canales RGB a 0, para luego comprobar si ha variado la predicción. La prueba se ha hecho anulando un 1 %, 10 %, 25 %, y 50 % de los píxeles de las imágenes como se observa en la figura 3.5.

3.3.2. Resultados comparativos

Adaptabilidad a categorías no entrenadas

En la figura 3.6 se muestra una imagen de ejemplo de *bodega* y se puede ver las puntuaciones obtenidas por las redes evaluadas para las 100 imágenes de *bodega*. En la figura 3.7 se muestra una imagen de ejemplo de *canchal* y se puede ver las puntuaciones obtenidas por las redes evaluadas para las 100 imágenes de *canchal*. En la figura 3.8 se muestra una imagen de ejemplo de *estudio_fotográfico* y se puede ver las puntuaciones obtenidas por las redes evaluadas para las 100 imágenes de *estudio_fotográfico*. En la figura 3.9 se muestra una imagen de ejemplo de *frontón* y se puede ver las puntuaciones obtenidas por las redes evaluadas para las 100 imágenes

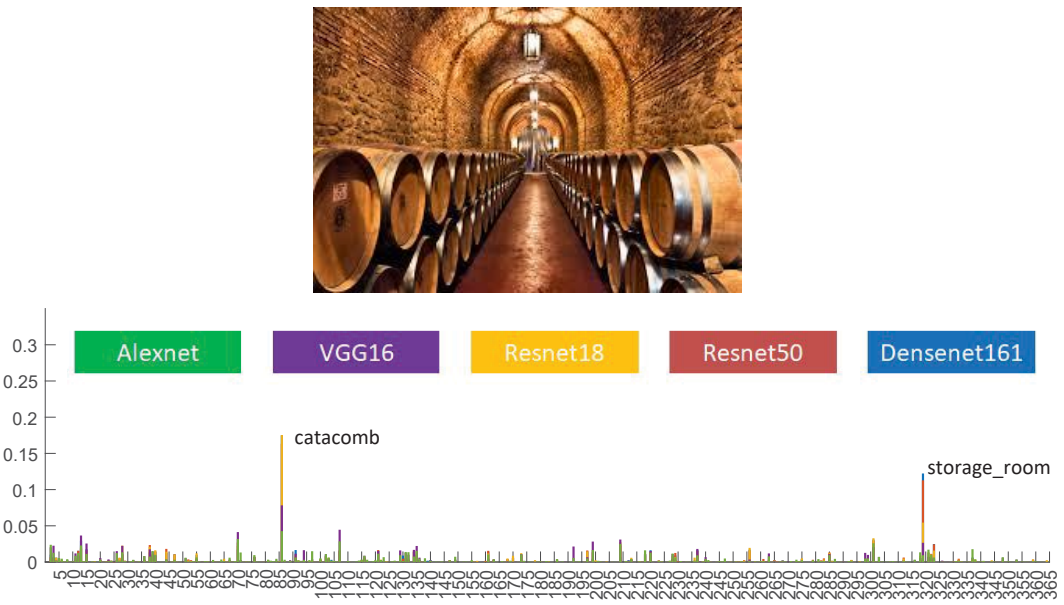

Figura 3.6: La imagen de arriba es un ejemplo de la categoría de *bodega*. La gráfica de abajo muestra la media de las salidas de las 5 redes con imágenes de *bodega*. El eje X muestra las categorías y el eje Y el valor medio de cada una de las categorías a la salida de la red.

de *frontón*. En la Figura 3.10 se muestra una imagen de ejemplo de *sala_revelado* y se puede ver las puntuaciones obtenidas por las redes evaluadas para las 100 imágenes de *sala_revelado*.

Robustez al ruido

Las Figura de 3.11 muestra los resultados de la robustez al ruido para Resnet50. El rendimiento de las otras redes puede observarse en el apéndice C.

3.3.3. Discusión

Con lo visto en este apartado se puede establecer que las redes actúan de forma similar ante unas mismas imágenes no catalogadas, sobre todo haciendo una distinción entre las redes Alexnet [1] y VGG16 [2] por un lado y Resnet18 [3], Resnet50 [3] y Densenet161 [4] por otro, ya que se parecen más los resultados dentro de estos dos grupos.

En la Figura 3.6 de *bodega* destaca *catacomb* pero hay más repetidas. Lo que sí es destacable es que a pesar de ser la más puntuada por todas las redes, en algunas la puntuación es mucho mayor que en otras. Por otro lado, en la Figura 3.7 de *canchal,* también parece ser que las redes se comportan de forma similar, aunque una de ellas

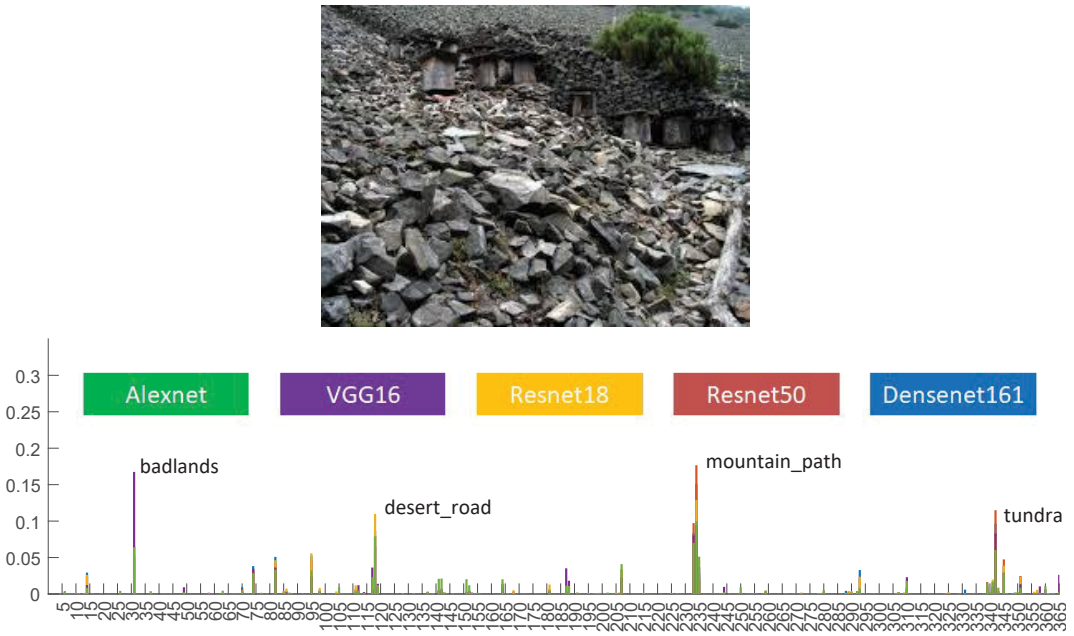

Figura 3.7: La imagen de arriba es un ejemplo de la categoría de *canchal*. La gráfica de abajo muestra la media de las salidas de las 5 redes con imágenes de *canchal*. El eje X muestra las categorías y el eje Y el valor medio de cada una de las categorías a la salida de la red.

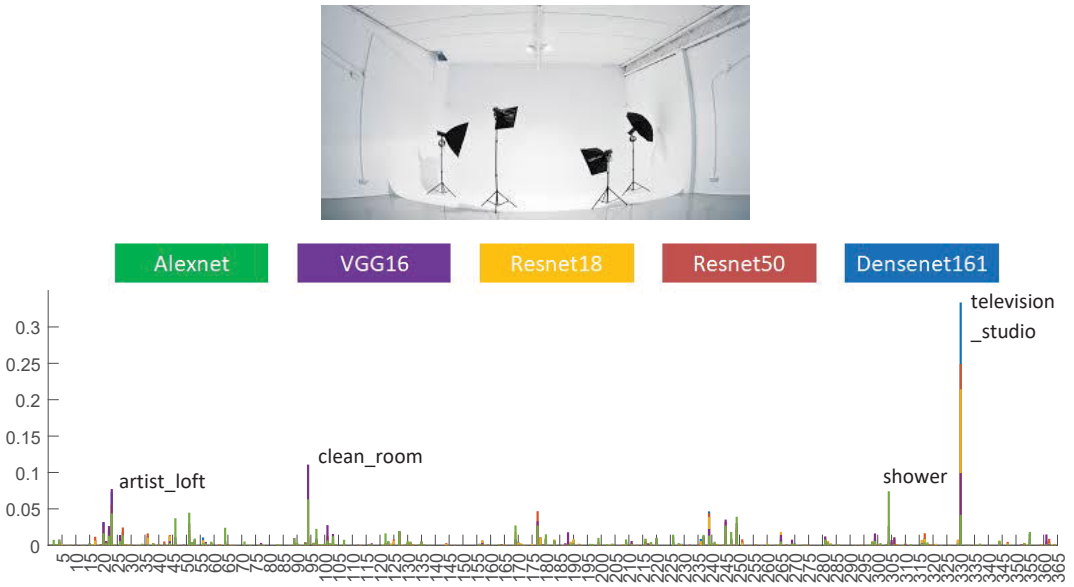

Figura 3.8: La imagen de arriba es un ejemplo de la categoría de *estudio_fotográfico*. La gráfica de abajo muestra la media de las salidas de las 5 redes con imágenes de *estudio_fotográfico*. El eje X muestra las categorías y el eje Y el valor medio de cada una de las categorías a la salida de la red.

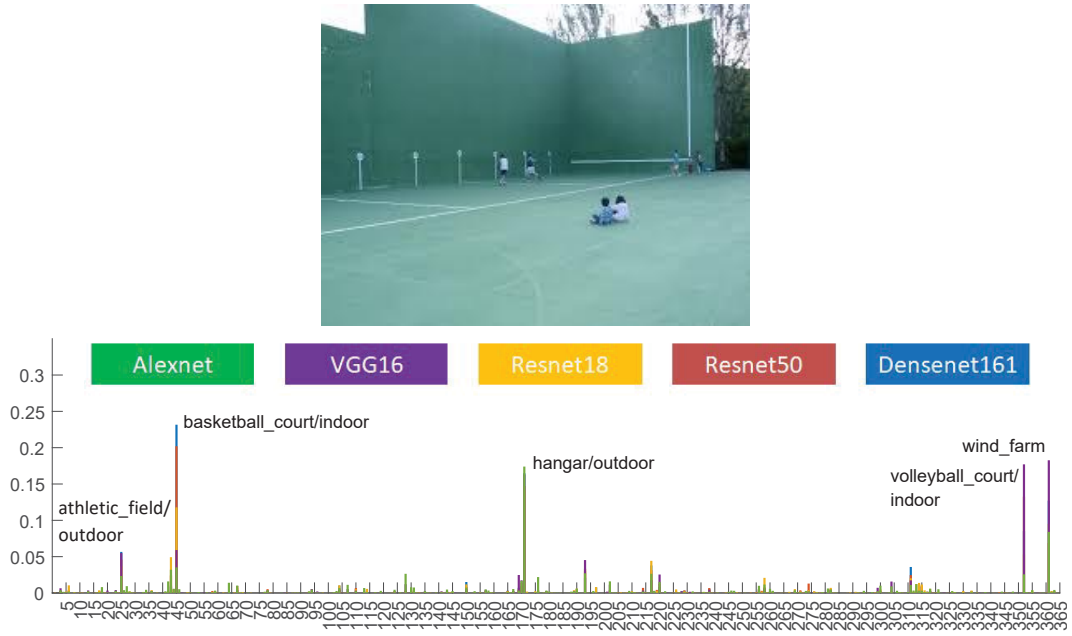

Figura 3.9: La imagen de arriba es un ejemplo de la categoría de *frontón*. La gráfica de abajo muestra la media de las salidas de las 5 redes con imágenes de *frontón*. El eje X muestra las categorías y el eje Y el valor medio de cada una de las categorías a la salida de la red.

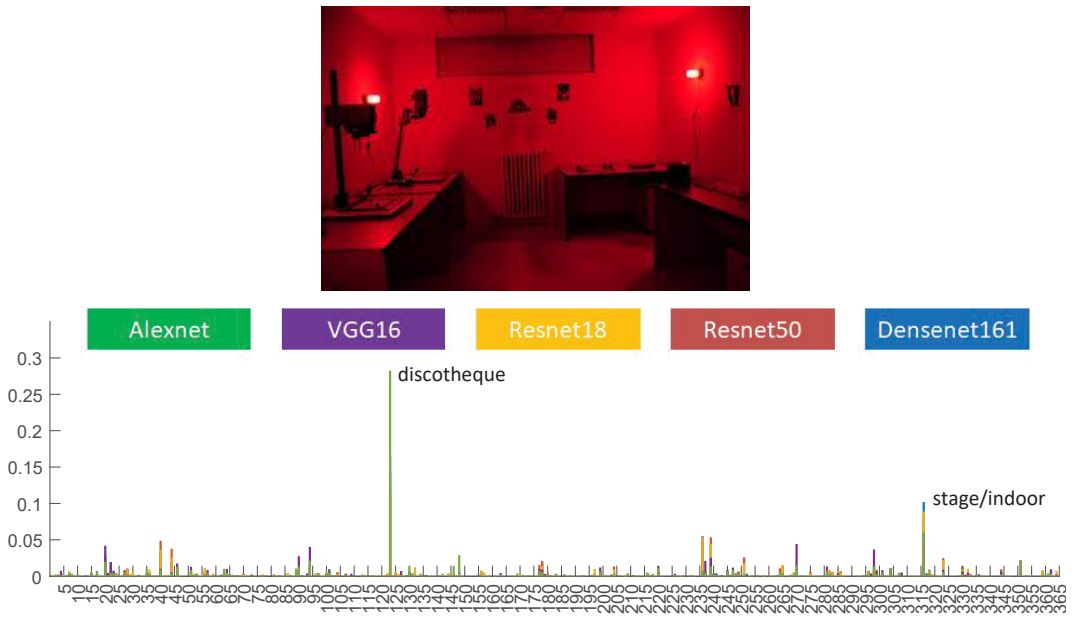

Figura 3.10: La imagen de arriba es un ejemplo de la categoría de *sala_revelado*. La gráfica de abajo muestra la media de las salidas de las 5 redes con imágenes de *sala_revelado*. El eje X muestra las categorías y el eje Y el valor medio de cada una de las categorías a la salida de la red.

no tenga el máximo en la misma categoría que las demás, sí tiene esa categoría entre las de mayor puntuación. En la categoría *estudio_fotográfico* vista en la Figura 3.8, aparece una gran diferencia entre las dos redes lineales (Alexnet [1] y VGG16 [2]) y las recursivas (Resnet [3] y Densenet [4]), ya que las dos primeras dan la mayor puntuación a otra categoría y tienen las puntuaciones más repartidas entre las categorías, mientras que las otras redes tienen un máximo muy claro en la categoría *television_studio*. En la Figura 3.9 de *frontón,* parece que hay menos consenso entre las redes, aunque también aparecen las puntuaciones más repartidas entre distintas categorías. Aún así vemos que las categorías con mayor puntuación aparecen repetidas en las distintas redes. En la Figura 3.10 que muestra los resultados de *sala_revelado,* destaca *discotheque* como categoría con mayor puntuación, sobre todo en las redes de Alexnet [1] y VGG16 [2], mientras que en las otras también está en primer lugar pero con una puntuación más distribuida entre las demás categorías.

Se observa en la figura 3.11 cómo empeora el rendimiento de las redes, algo que se nota en algunas categorías con sólo el 1 % de los píxeles anulados. Además, con el 50 % de píxeles anulados, casi no aciertan en ninguna imagen, por lo que se puede establecer que las redes son sensibles al ruido y por lo tanto no muy robustas a éste. Un resultado interesante es que existen algunas categorías que mejoran el número de aciertos al anular píxeles, lo que sugiere que puede conseguirse una mejora de la red alterando las imágenes, tal y como se pretende con el sistema propuesto en el siguiente capítulo 4.

3.4. Resumen y oportunidades de la evaluación

Tras lo visto en esta evaluación, se puede concluir que las redes con mayor complejidad y profundidad obtienen un mejor resultado, lo que provoca que sus salidas estén más correladas entre ellas. A pesar de esto, parece que la categoría es factor discriminante en el rendimiento, puesto que para algunas categorías, el rendimiento es pobre con independencia de la red evaluada.

En cuanto a la adaptabilidad de las redes a imágenes no catalogadas, se concluye que las redes actúan de forma similar, aunque depende del tipo de imágenes introducidas, ya que, en algunos casos, las redes evalúan las imágenes de una categoría no catalogada como una misma categoría y en otros casos dan salidas similares en múltiples categorías.

Respecto a la robustez al ruido, se observa que las redes disminuyen su rendimiento cuando las imágenes contienen un determindado nivel de ruido, mientras que si el nivel de ruido no es muy alto, en algunas categorías, las redes mejoran su resultado. En

base a estos resultados, en el capítulo 4 se implementa un esquema que se inspira en este experimento para la mejora de los resultados de las redes. A partir de lo observado en el experimento de robustez, se propone estudiar la posibilidad de mejorar la tasa de aciertos de una red sin reentrenarla a partir de la modificación de las imágenes previamente a introducirlas en la red. El capítulo 4 de este trabajo propone un esquema para realizar esta mejora. El capítulo 5 muestra resultados iniciales en la evaluación de la bondad del esquema diseñado.

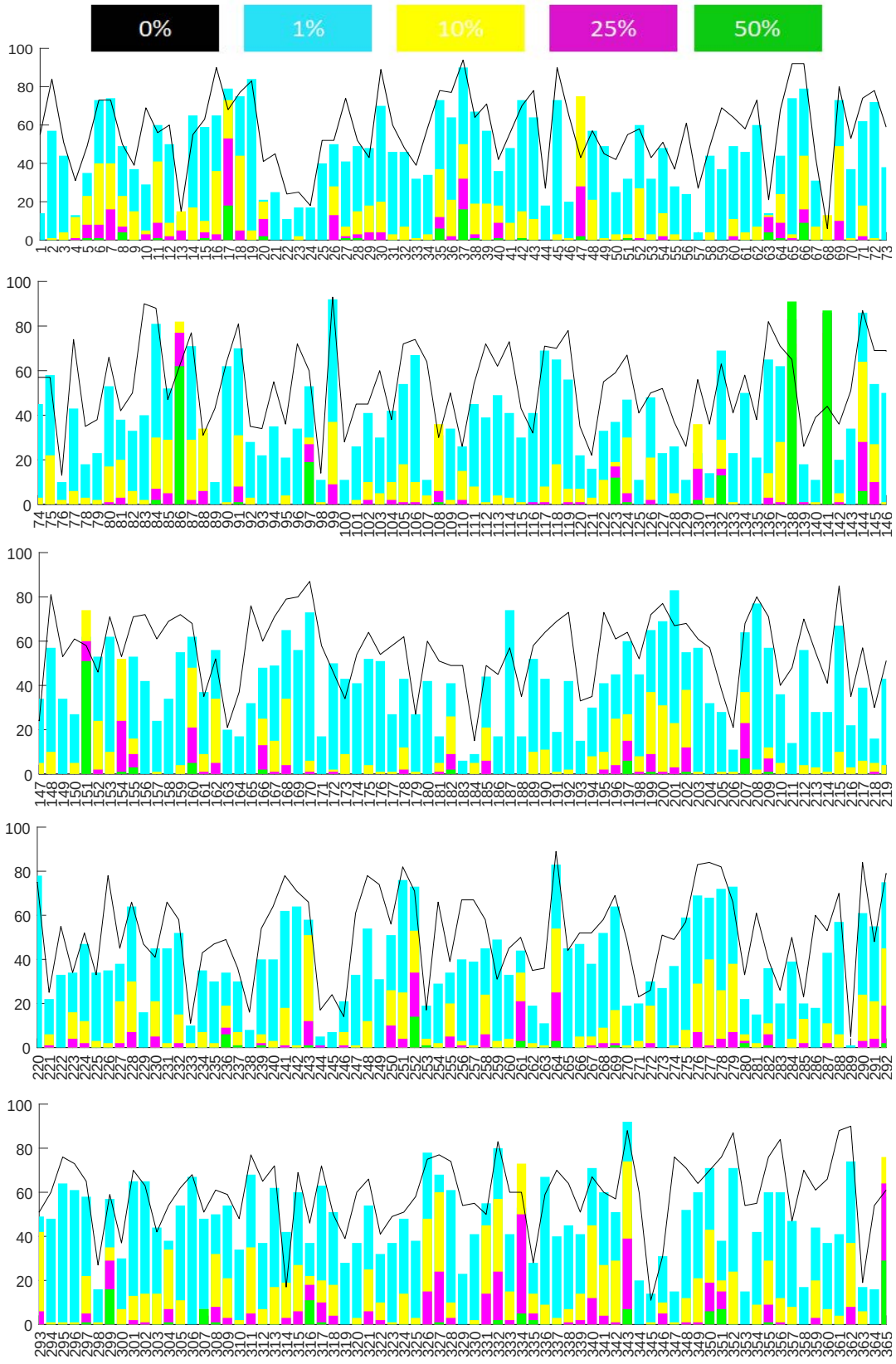

Figura 3.11: Gráfica de barras en las que el eje X muestra categorías y el eje Y el número de aciertos en cada porcentaje de píxeles a 0 con la red Resnet50.

Capítulo 4

Método de refocalización (sistema, diseño y desarrollo)

4.1. Introducción

En este capítulo se describe el sistema implementado con el cual se explora la viabilidad de mejorar el rendimiento de las redes a partir de información de los objetos contenidos en la escena. En primer lugar se presenta el sistema general a partir de un esquema modular para posteriormente describir en cada apartado el funcionamiento de cada módulo.

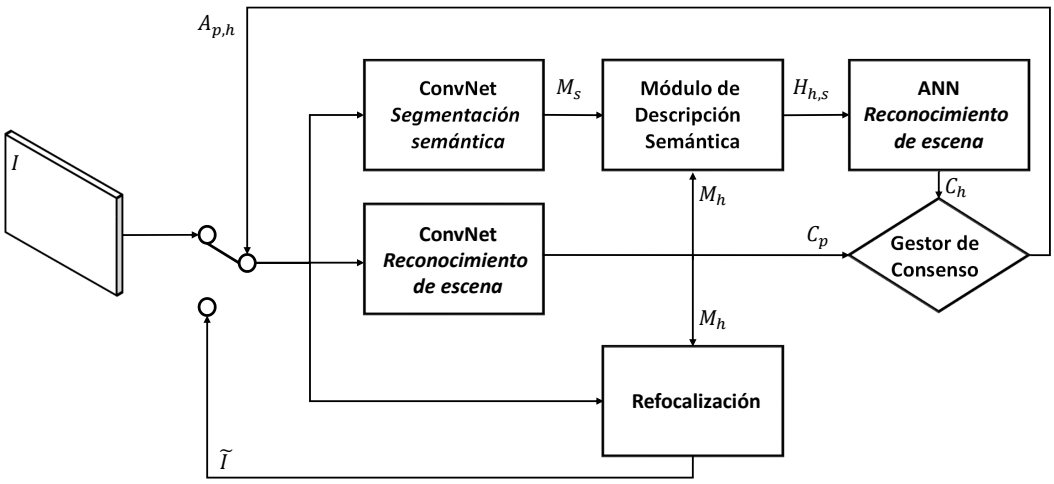

Figura 4.1: Esquema del modelo de re-focalización implementado.

4.2. Esquema del sistema desarrollado

En la figura 4.1 se muestra el esquema general del sistema, donde se utiliza la siguiente nomenclatura:

- I es la imagen de entrada al sistema.
- M_s es el mapa semántico obtenido por el módulo de segmentación semántica (ver sección 4.4).
- M_h es la máscara del mapa de focalización obtenida por el módulo de reconocimiento de escena (ver sección 4.4).
- $H_{h,s}$ es el histograma semántico con pesos del mapa de focalización que se obtiene en el módulo de descripción semántico (ver sección 4.5).
- C_p es la categoría con mayor puntuación a la salida del módulo de reconocimiento de escena (ver sección 4.3) siendo $C_p \epsilon [0, 364]$.
- C_h es la categoría con mayor puntuación a la salida de la red neuronal artificial (ver sección 4.4) siendo $C_h \epsilon [0, 364]$.
- $A_{p,h}$ es el acuerdo establecido entre las categorías C_h y C_p establecido en el módulo gestor de consenso (ver sección 4.7).

El proceso iterativo seguido por el sistema de refocalización puede dividirse en cinco etapas:

1. Se evalúa la imagen de entrada en la red convolucional entrenada para el reconocimiento de escena (ver sección 4.3), obteniendo la categoría predicha por la red (C_p) como la categoría para la que se obtiene la mayor puntuación o probabilidad. Asimismo, se extrae también la máscara de focalización (M_h). En paralelo, la imagen es segmentada la red convolucional entrenada para la segmentación semántica (ver sección 4.4) obteniendo el mapa semántico (M_s) de la imagen.

2. A partir del mapa semántico (M_s) se calcula un histograma (H_{hs}) de los objetos que aparecen en la imagen en el módulo de descripción semántica (ver sección 4.5). En el cálculo de este histograma se pondera cada objeto en función de su puntuación en la máscara de focalización (M_h).

3. El histograma ponderado ($H_{h,s}$) se evalúa en una red neuronal artificial (ANN) previamente entrenada para el reconocimiento de escenas, obteniendo así una predicción C_h (ver sección 4.6).

4. En el gestor de consenso (ver sección 4.7) se comparan las categorías C_h y C_p. En caso de que ambas sean la misma, se establece que la categoría predicha por la red convolucional de escenas es correcta. En caso contrario se presupone que el sistema ha errado en su predicción y se devuelve el acuerdo A_{ph} para activar el proceso de refocalización.

5. En el módulo de refocalización se parte de la máscara de focalización (M_h) y de la imagen, introduciendo valores aleatorios en los píxeles de la imagen en las posiciones con mayor puntuación en la máscara de focalización (M_h). La imagen así modificada se introduce de nuevo en el proceso, volviendo al paso 1.

Los pasos 1-5 se repiten hasta coincidencia entre las categorías C_h y C_p o hasta que se alcance un número máximo de iteraciones.

4.3. Red convolucional para el reconocimiento de escena

En este módulo se utiliza la red Resnet50 [3] entrenada con el conjunto completo de datos de entrenamiento Places365 [6]. A partir de la imagen, la red retorna un identificador de la escena que se predice: C_p, la puntuación asociada a la predicción no se utiliza en la aproximación diseñada. Además se utiliza el mecanismo de extracción de la máscara de focalización descrito en la sección 2.3 para obtener M_h, una máscara o mapa de 2 dimensiones con valores entre 0 y 1, siendo mayor en aquellas regiones de la imagen de estimada mayor relevancia para la predicción. Un ejemplo de máscara de focalización puede observarse en la primera fila de la Figura 4.2.

Dado un píxel $\mathbf{p}$, denominaremos $v(\mathbf{p}) \in [0, 1] \subset \mathbb{R}$ a la puntuación que recibe ese píxel en la máscara de focalización.

4.4. Red convolucional para la segmentación semántica

En este módulo se utiliza la red PSPNet (2.4) entrenada con el conjunto completo de datos de entrenamiento AD20K (ver apéndice B). A partir de la imagen, la red retorna un mapa semántico de clases M_s en forma de imagen, en la cual cada píxel tiene un valor entre 1 y 150, los cuales representan las 150 etiquetas de los objetos de AD20K. Un ejemplo de mapa semántico puede observarse en la primera fila de la Figura 4.2.

Dado un píxel $\mathbf{p}$, denominaremos $c(\mathbf{p}) \in [1, 150] \subset \mathbb{N}$ a la clase asociada a ese píxel en el mapa semántico.

4.5. Módulo de descripción semántica

A partir del mapa semántico M_s, puede estimarse un histograma $H_s = \{h_1, ...h_j, ..., h_{150}\}$ donde cada h_j represente el número de píxeles en M_s etiquetados como el objeto j. Este histograma podría entenderse como una descripción de los objetos presentes en la imagen.

Para incluir en esta descripción información de los objetos más relevantes para la predicción de escena realizada por la red convolucional, proponemos crear una versión de H_s ponderada por el valor de cada píxel en la máscara de focalización M_h. Cada posición $\dot{h}_j$ de este histograma ponderado $H_{h,s} = \{\dot{h}_1, ...\dot{h}_j, ..., \dot{h}_{150}\}$ se obtiene sumando para cada píxel etiquetado como el objeto j el valor de su puntuación en el mapa:

$$\dot{h}_j = \sum_{\forall \mathbf{p} | c(\mathbf{p}) \longrightarrow j} v(\mathbf{p}) \tag{4.1}$$

Utilizaremos $H_{h,s}$ como un descriptor de imagen, que contiene información, no sólo de los objetos presentes en la escena, sino de aquellos que son más relevantes en la predicción realizada por la red Resnet50 [3]. Un ejemplo de $H_{h,s}$ puede observarse en la segunda fila de la Figura 4.2.

4.6. Red neuronal para el reconocimiento de escena

Se propone entrenar una red neuronal artificial [18] con 10 capas ocultas, donde todas las neuronas están conectadas entre sí. La capa de entrada tendrá 150 neuronas, donde se introducen cada uno de las posiciones de $H_{h,s}$, que se utilizan como vectores de caracterización de las imágenes de entrenamiento. La capa de salida estará adaptada al número de categorías de escena a reconocer. El objetivo es entrenar un modelo que permita predecir la clase de escena (C_h) a partir de estos vectores de caracterización. El modelo así entrenado se documentará en la sección 5.3.

4.7. Gestor de consenso

Una vez se tienen las dos categorías predichas C_h y C_p se comparan entre sí. En caso de que sean iguales se establece que la categoría predicha por el sistema es C_p, en caso contrario se establece una señal $A_{p,h}$ que activa el mecanismo de refocalización. Este proceso se repetirá, al igual que el mecanismo de refocalización, siempre y cuando las categorías predichas no sean iguales, hasta que ambas sean iguales o se alcance un número máximo de iteraciones.

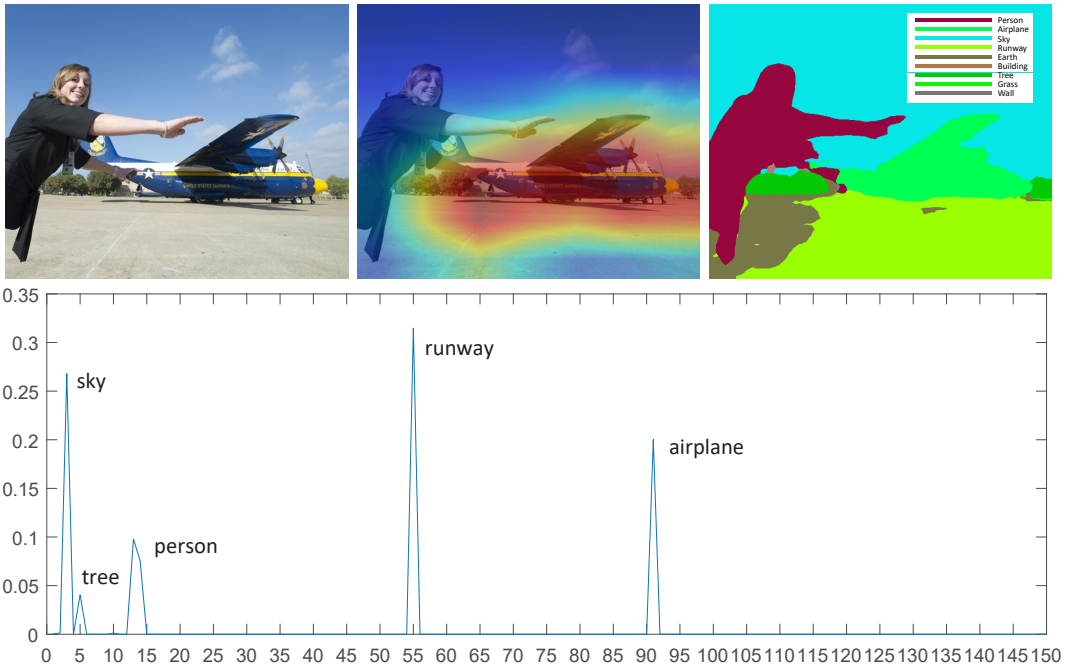

Figura 4.2: En esta figura se muestra un ejemplo de una imagen con sus máscaras y su histograma ponderado. La primera fila incluye, en la primera columna la imagen de entrada I. En la segunda columna la máscara de focalización M_h, en la cual los píxeles con mayor puntuación son los representados en rojo y los de menos puntuación azules, en este caso, la categoría predicha por Resnet50 [3] es *runway*, aunque la imagen está etiquetada como *airfield*. En la tercera columna se presenta el mapa semántico M_s, en el cual cada color se corresponde a un objeto (en la leyenda sólo se incluyen aquellos objetos que aparecen en el mapa). En la fila inferior se muestra el histograma ponderado $H_{h,s}$ con los valores ponderados de los objetos que aparecen en la escena.

4.8. Mecanismo de refocalización

Si las dos categorías predichas C_h y C_p no son iguales, se realiza el mecanismo de refocalización. El mecanismo consiste en la modificación de los píxeles de la imagen a los que se les ha asignado valores de focalización por encima de un valor determinado,γ, en la máscara de focalización M_h. En particular, se crea una nueva imagen $\tilde{I}$, cuyos colores son idénticos a los de la imagen original en los píxeles con valor de focalización menor o igual que γ y aleatorios en aquellos píxeles con valor de focalización mayor que γ:

$$\begin{cases} \tilde{I}(\mathbf{p}) = I(\mathbf{p}) & , \ \mathbf{p}|v(\mathbf{p}) \leq \gamma \\ \tilde{I}(\mathbf{p}) = f(\mathbf{p}) & , \ \mathbf{p}|v(\mathbf{p}) > \gamma \end{cases} , \tag{4.2}$$

donde $f(\mathbf{p})$ es una función que devuelve un color aleatorio en el rango de la imagen.

Tras realizar esta modificación, se evalúa en la red la nueva imagen $\tilde{I}$ obteniendo una nueva categoría predicha C_p. Si esta nueva categoría tampoco coincide con la categoría C_h, se vuelve a realizar el mecanismo de refocalización disminuyendo el valor de γ, de manera que el número de píxeles de la nueva imagen $\tilde{I}$ con valores de color aleatorio distintos de los de la imagen original es mayor.

La idea detrás de este esquema es que, dado que la predicción realizada por la red toma en mayor relevancia los píxeles con mayor valor en la máscara de focalización M_h, si la red se equivoca en la predicción, son estos píxeles los que tienen mayor influencia en esta predicción errónea. Cambiando estos píxeles, en teoría, la red se fijará en otras áreas de la imagen para establecer la predicción.

Capítulo 5

Experimentos realizados y resultados

En este capítulo se expone una prueba representativa del sistema implementado como se expone en el capítulo 4.

5.1. Conjunto de datos seleccionado

Como prueba inicial de concepto, se han escogido únicamente 10 categorías de las 365: *airfield, bedroom*, *building_facade*, *coast*, *creek*, *fabric_store*, *field/cultivated*, *glacier*, *mountain*, *ocean*. Esto se debe a que empíricamente se comprobó que los histogramas ponderados ($H_{h,s}$) no eran lo suficientemente discriminativos para que la red neuronal artificial entrenada distinguiese entre las 365 categorías originales. La selección de estas categorías, y no de otras, se debe a dos factores principales: la amplia distancia semántica entre ellas y la oportunidad de mejora, en base a los resultados del estudio comparativo del capítulo 3 (ver Figura 3.11).

De cada una de estas 10 categorías se seleccionan 1.000 imágenes de entrenamiento por categoría (para un total de 10.000) del conjunto de entrenamiento definido en Places365 [6] y 100 imágenes de test por categoría (para un total de 1.000), éstas extraídas del conjunto de validación definido en Places365 [6]. Para todas las imágenes se establece su clase real de escena, C_{GT}, como la anotada en Places365 [6]. Todas las imágenes escogidas, tanto de entrenamiento como de test, son las imágenes en su tamaño original.

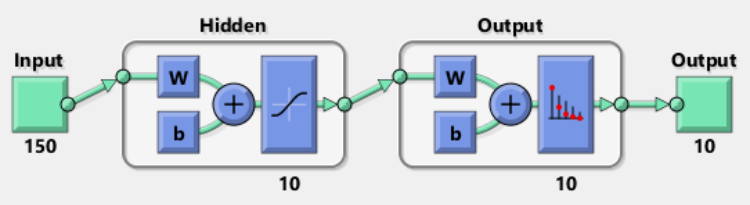

Figura 5.1: Esquema de la red neuronal artificial entrenada.

5.2. Descripción de las pruebas realizadas

El procedimiento seguido es el siguiente:

1. Se evalúan con Resnet50 [3] las imágenes de entrenamiento obteniendo las predicciones C_p y las máscaras de focalización M_h, además de con la red PSPNet [5] para obtener las máscaras semánticas M_s.

2. A partir de las máscaras M_s y M_h se obtienen los histogramas ponderados $H_{h,s}$ del set de imágenes de entrenamiento.

3. Se entrena la red neuronal artificial con los histogramas ponderados $H_{h,s}$ de las imágenes de entrenamiento.

4. Se evalúan las imágenes de test siguiendo el método explicado en el capítulo 4 hasta que las dos predicciones C_p y C_h coincidan o hasta que se alcance un número máximo de iteraciones.

5. En caso de que ambas coincidan se comprueba que C_p es igual a la categoría anotada C_{GT} , si es así se considera un acierto.

6. Finalmente se comprueba si el número de aciertos inicial (sin el sistema de refocalización) ha variado respecto al nuevo sistema, y si lo ha hecho, si ha sido para mejor o para peor.

5.3. Configuración del sistema

Utilizando los histogramas ponderados como datos de caracterización se entrena la red neuronal artificial (ANN) con los histogramas ponderados de cada imagen como vectores de caracterización de 150 valores. Para ello se ha utilizado una arquitectura de entrenamiento y clasificación disponible en Matlab, en particular *patternnet* [18]. Se configura la red neuronal con 10 capas ocultas y 10 salidas (una por categoría). El esquema de esta configuración se observa en la Figura 5.1.

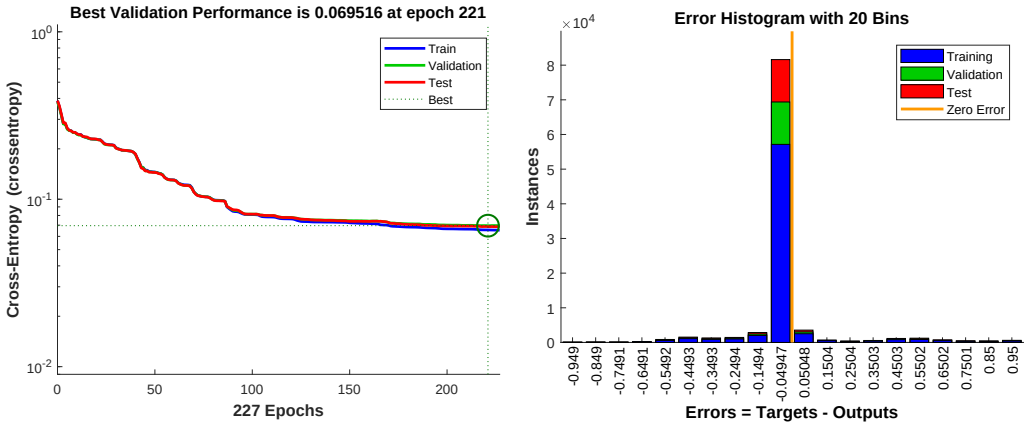

Figura 5.2: La imagen de la izquierda se corresponde con la gráfica de rendimiento de la red neuronal artificial, la cual usa *cross-entropy* como medida de rendimiento, sienedo el eje X el número de iteraciones y el eje Y el valor del *cross-entropy*. La imagen de la derecha es el histograma de error, el cual tiene en el eje X el porcentaje de error como la resta de los valores reales de los datos menos la salida obtenida de la red y el eje Y como la cantidad de datos en cada porcentaje de error.

Para el entrenamiento, Matlab por defecto divide los datos de entrada en conjuntos de test, validación y entrenamiento, en proporciones respectivas del 15 % , 15 % y 70 %. La convergencia del entrenamiento de la red neuronal se establece a partir del valor de la entropía cruzada, *cross-entropy*, $H(p, q)$, que se calculan, siendo p y q las distintas distribuciones de entrenamiento, test o validación como:

$$H(p, q) = -\sum_{x} p(x) log(q(x)) \tag{5.1}$$

La red neuronal entrenada converge tras 221 épocas, obteniendo un valor de *cross-entropy* igual a 0.069516 entre cada par de distribuciones (ver 5.2). En la Figura 5.2, se incluye además, el histograma de error respecto a las anotaciones (*Target*), que muestra que la mayoría de las predicciones resultan en errores de predicción cercanos al 0, lo que sugiere un buen resultado. Como complemento a estos resultados, en la Figura 5.3 se incluyen las diferentes curvas ROC para cada conjunto de datos de entrenamiento. Las curvas muestran un rendimiento adecuado, ya que el punto de trabajo de todas ellas están cerca del rendimiento óptimo: 1 en el *True Positive Rate* cuando el *False Positive Rate* es 0.

Por último reseñar que esta red, por si sóla, resulta en un sorprendente 70,20 % de acierto en la predicción de escena cuando se evalúa con el conjunto de datos de test definido en la sección (5.3). Este resultado se desglosa por categoría en la primera

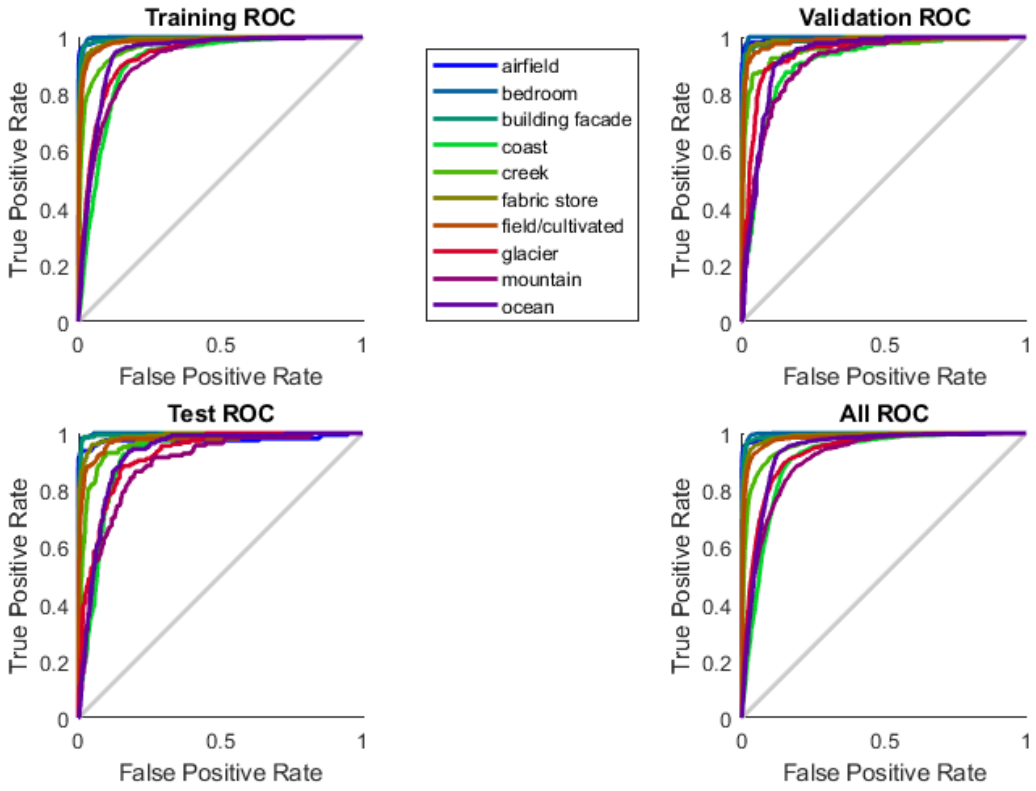

Figura 5.3: Gráficas de las curvas ROC de los datos de entrenamiento de la red neuronal artificial que se dividen en test, validación y entrenamiento. En concreto, la gráfica de arriba a la izquierda representa las curvas ROC de los datos usados como entrenamiento, la gráfica de arriba a la derecha representa las curvas ROC de los datos usados como validación, la gráfica de abajo a la izquierda representa las curvas ROC de los datos usados como test y la gráfica de abajo a la derecha representa las curvas ROC de los todos los datos.

columna del Cuadro 5.1.

al principio con las 1.000 imágenes de test (100 imágenes por categoría).

5.4. Resultados obtenidos de las pruebas

Este apartado recoge los resultados de las 1.000 imágenes de test de las 10 escenas escogidas.

5.4.1. Cuantitativos

A partir del procedimiento expuesto en el apartado 5.2 se han obtenido los resultados expuestos en el Cuadro 5.1.

	ANN	Resnet50	Resnet50 + focalización	Mejora absoluta	Mejora relativa
airfield	93	54	70	+16	+30
bedroom	95	39	42	+3	+8
building_facade	90	10	9	-1	-10
coast	27	11	4	-7	-64
creek	83	53	55	+2	+4
fabric_store	79	66	69	+3	+5
field/cultivated	86	48	61	+13	+27
glacier	61	28	34	+6	+21
mountain	26	12	6	-6	-50
ocean	62	18	19	+1	+6
GLOBAL	**70,20**	**33,90**	**36,90**	**+3**	**8,85**

Cuadro 5.1: Tabla que muestra para cada escena, el número de aciertos obtenidos con la red neuronal artificial entrenada en la primera columna, sólo con la red Resnet50 en la segunda columna, el número de aciertos con el sistema implementado en la tercera columna, la mejora absoluta del número de aciertos en la cuarta columna y la mejora relativa respecto a los aciertos previos al sistema de focalización. Todos los valores son para las 100 imágenes de validación de Places365 [6], por lo tanto, todos los números representan % de acierto.

Discusión

Como se observa en el Cuadro 5.1, todas las categorías han variado su número de aciertos para bien o para mal. La que mejor resultado ha dado es *airfield* con una mejora de 16 imágenes acertadas más en las 100 totales y una mejora relativa del 30 %. La que peor resultado ha dado es *coast* que ahora tiene 7 aciertos menos, lo que supone un empeoramiento del 64 % respecto a los aciertos previos al sistema. Observando los aciertos anteriores al sistema, las categorías que han empeorado son aquellas que tienen una tasa de acierto más bajo, luego parece ser que si la tasa de acierto es mala, este sistema no consigue buenos resultados. Mientras que en categorías con una

tasa de acierto aceptable sí se consigue una mejora. Destaca que en las categorías *coast y mountain*, con baja tasa de acierto con Resnet50, también se obtenga un mal resultado con la red neuronal artificial, en contraposición a *building_facade*, que en la red neuronal artificial tiene un acierto del 90 %, y a pesar de ello se obtiene un empeoramiento relativo de acierto del 10 %, lo que indica que Resnet50 funciona mal con esta categoría aunque se fije en distintas zonas de las imágenes. Sería intersante modificar el gestor de consenso para casos como este, en el cual la red neuronal artificial tiene muy claro que la categoría de esas imágenes es *building_facade*, para así obtener un resultado más alto con el sistema.

5.4.2. Cualitativos

Para comprobar los resultados cualitativos del sistema, se observa la evolución de imágenes de test introducidas en el sistema, así como sus respectivas máscaras de focalización, comprobando cómo al variar la imagen, la máscara de focalización también varía dando más relevancia a zonas de la imagen a las que antes no les daba, provocando en algunos casos que la predicción aportada por la red sea distinta a la inicial. Estos resultados se incluyen en la Figuras 5.4, 5.5, 5.6, 5.7 y 5.8.

Discusión

El cambio en la predicción varia según la imagen, hay casos en los que se necesitan muy pocas iteraciones para que varíe la predicción y otros casos en los que se necesitan más iteraciones.

En la Figura 5.4 se muestra el caso de una imagen que predice *runway* y pasa a predecir *airfield*. Esto se puede deber a que, como se observa en sus máscaras de focalización, la red, que en principio se fija en una gran zona de carretera, pasa a fijarse únicamente en los vehículos de la escena. El caso de la Figura 5.5 es más peculiar, ya que se obtiene la categoría anotada en pocas iteraciones, pero se muestra que la red ha pasado por otra predicción distinta a la original y a la final y simplemente modificando un pequeño número de píxeles en comparación al total de la imagen. En particular, parece que la máscara de focalización final está centrada en los cojines, objetos que definen mejor la escena (*bedroom*). En la Figura 5.6 la red también ha llegado a una iteración alta, con la peculiaridad de que en este caso el mecanismo de refocalización elimina objetos que pueden ser distractores, consiguiendo con la modificación de los píxeles que la red se fije en las zonas contrarias a las que se fijaba en un principio. En la Figura 5.7 se ve un caso en el que el mecanismo de refocalización ha tenido que ir hasta una iteración alta para alcanzar la predicción anotada, pasando por otra (u

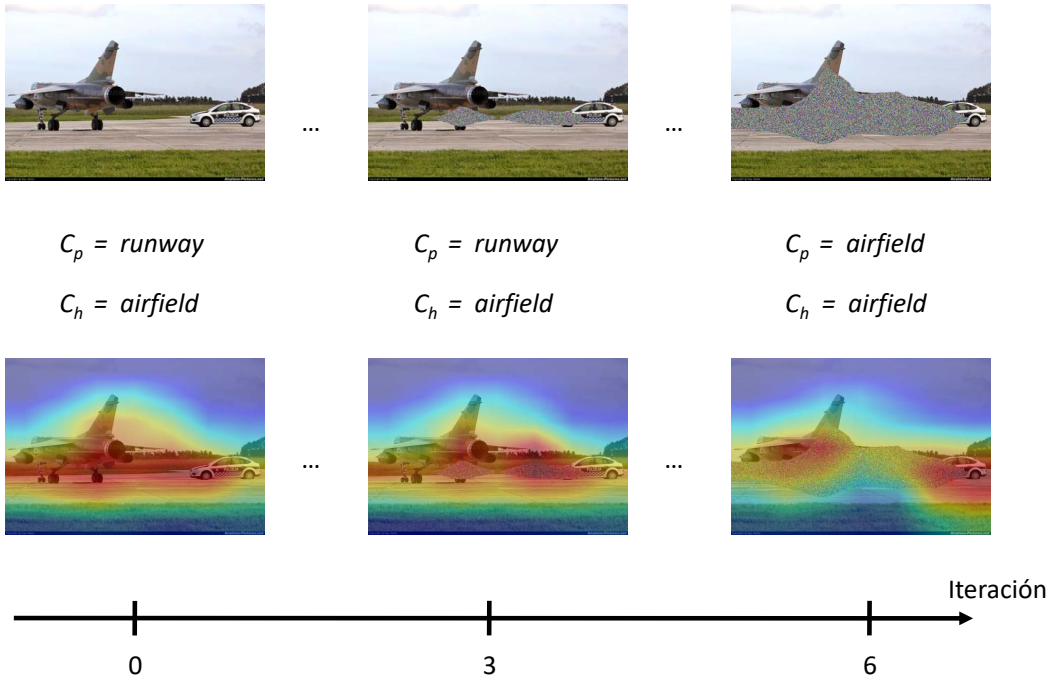

Figura 5.4: En esta figura se muestra un ejemplo de una imagen que se ha procesado en el sistema. La imagen de arriba a la izquierda es la original de la que se parte. La segunda de arriba es la imagen en la iteración 3, con $\gamma = 0{,}89$. La tercera imagen de arriba es la imagen en la iteración 6, con $\gamma = 0{,}74$. Las imágenes de abajo corresponden a las máscaras de focalización de las respectivas de arriba. Además se muestran las categorías predichas C_h y C_p, mostrando que no es hasta la iteración 6 en la que coinciden ambas predicciones y el sistema acierta.

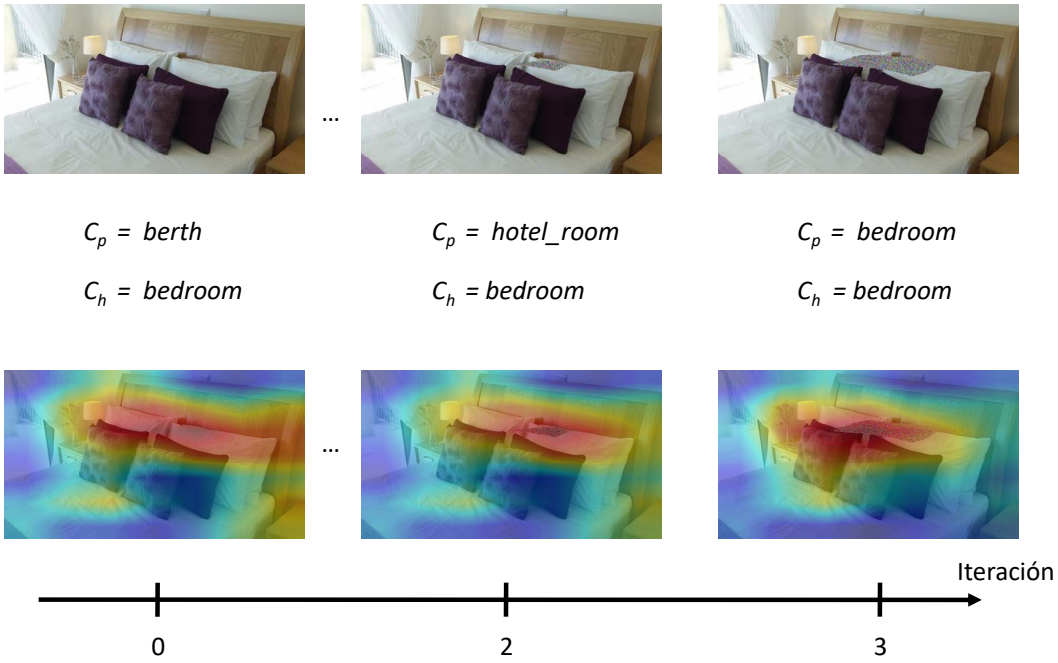

Figura 5.5: En esta figura se muestra un ejemplo de una imagen que se ha procesado en el sistema. La imagen de arriba a la izquierda es la original de la que se parte. La segunda de arriba es la imagen en la iteración 2, con $\gamma = 0{,}94$. La tercera imagen de arriba es la imagen en la iteración 3, con $\gamma = 0{,}89$. Las imágenes de abajo corresponden a las máscaras de focalización de las respectivas de arriba. Además se muestran las categorías predichas C_h y C_p, mostrando que no es hasta la iteración 3 en la que coinciden ambas predicciones y el sistema acierta.

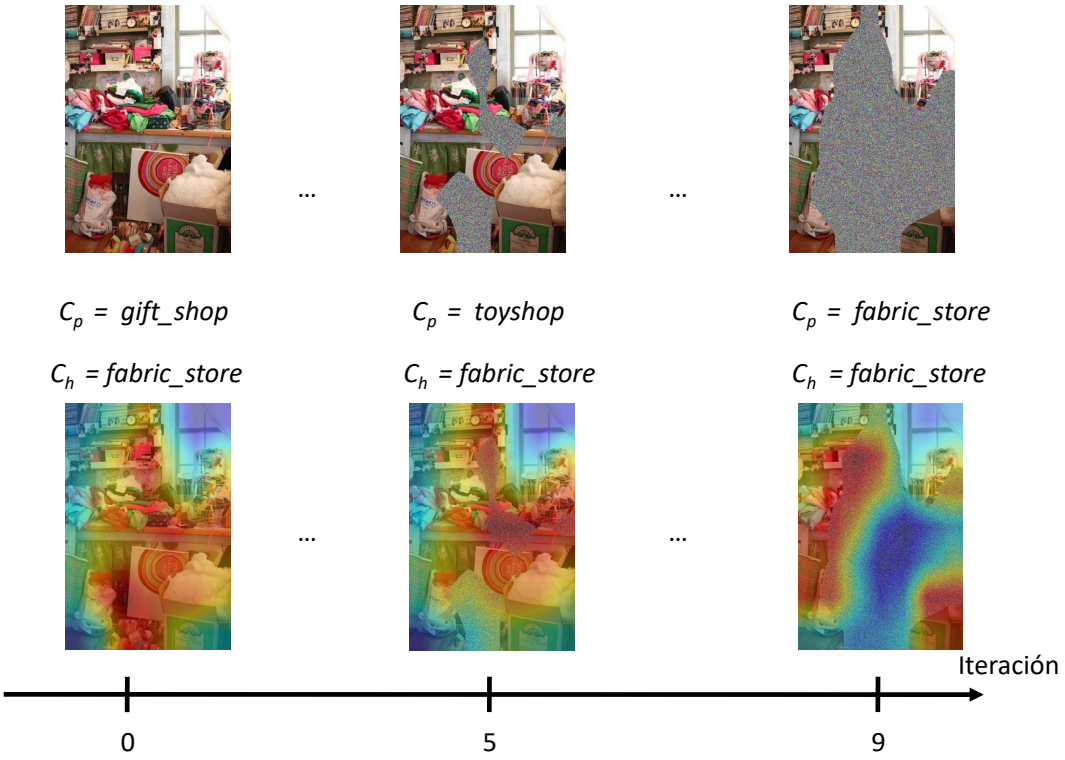

Figura 5.6: En esta figura se muestra un ejemplo de una imagen que se ha procesado en el sistema. La imagen de arriba a la izquierda es la original de la que se parte. La segunda de arriba es la imagen en la iteración 5, con $\gamma = 0{,}79$. La tercera imagen de arriba es la imagen en la iteración 9, con $\gamma = 0{,}59$. Las imágenes de abajo corresponden a las máscaras de focalización de las respectivas de arriba. Además se muestran las categorías predichas C_h y C_p, mostrando que no es hasta la iteración 9 en la que coinciden ambas predicciones y el sistema acierta.

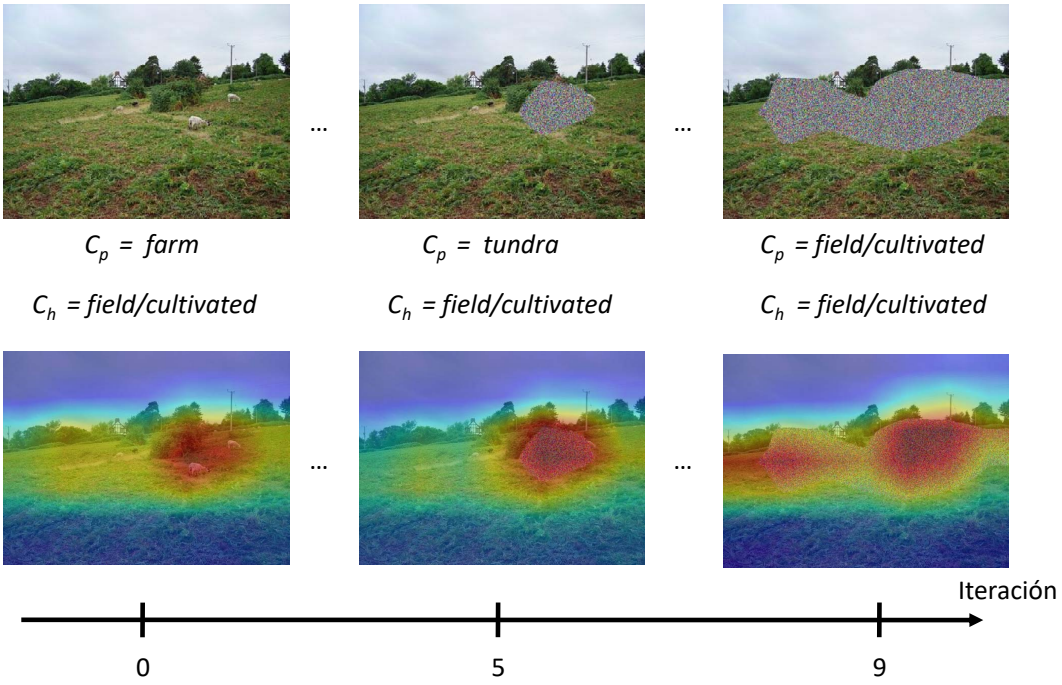

Figura 5.7: En esta figura se muestra un ejemplo de una imagen que se ha procesado en el sistema. La imagen de arriba a la izquierda es la original de la que se parte. La segunda de arriba es la imagen en la iteración 5, con $\gamma = 0{,}79$. La tercera imagen de arriba es la imagen en la iteración 9, con $\gamma = 0{,}59$. Las imágenes de abajo corresponden a las máscaras de focalización de las respectivas de arriba. Además se muestran las categorías predichas C_h y C_p, mostrando que no es hasta la iteración 9 en la que coinciden ambas predicciones y el sistema acierta.

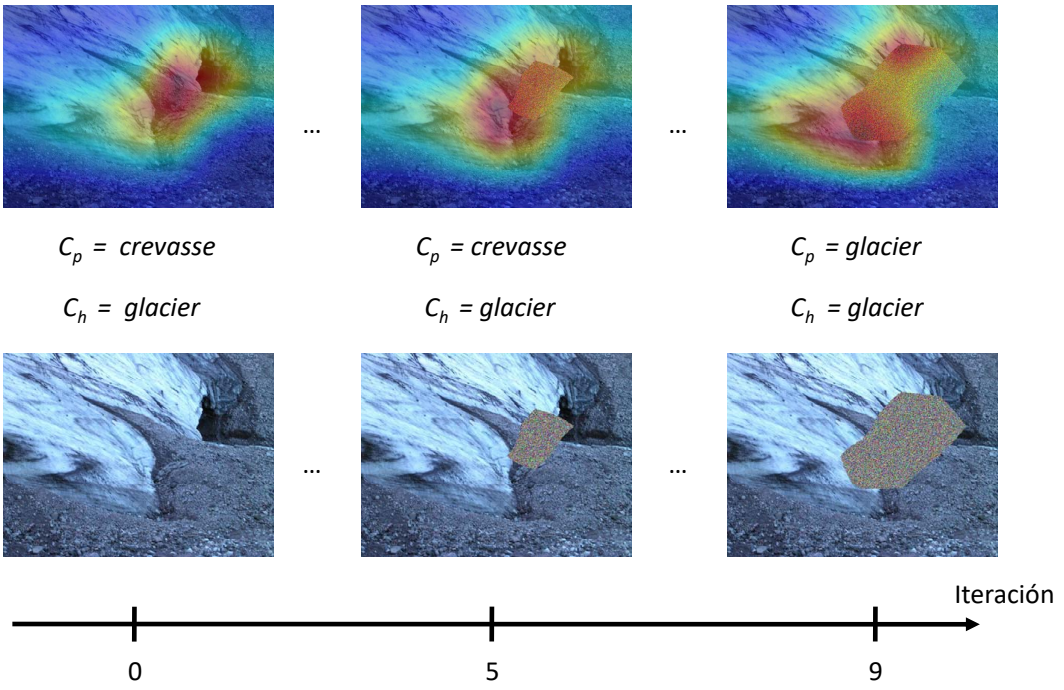

Figura 5.8: En esta figura se muestra un ejemplo de una imagen que se ha procesado en el sistema. La imagen de arriba a la izquierda es la original de la que se parte. La segunda de arriba es la imagen en la iteración 5, con $\gamma = 0{,}79$. La tercera imagen de arriba es la imagen en la iteración 9, con $\gamma = 0{,}59$. Las imágenes de abajo corresponden a las máscaras de focalización de las respectivas de arriba. Además se muestran las categorías predichas C_h y C_p, mostrando que no es hasta la iteración 9 en la que coinciden ambas predicciones y el sistema acierta.

otras en las demás iteraciones). En particular, en este caso parece que el foco principal en la primera imagen, la roca, que podría confundirse con una oveja, podría hacer que la predicción inicial (*farm*) fuese equivocada, eliminando este objeto, el resultado mejora. Un caso similar de eliminación de zonas conflictivas se muestra en la Figura 5.8, donde aparece un caso en el que la red se fijaba en una zona muy concreta de la imagen y al modificar los píxeles de la imagen, la red amplia la zona en la que se fija, incluyendo aún píxeles modificados, pero consiguiendo que la categoría predicha sea la misma que la categoría devuelta por la red convolucional de segmentación semántica.

5.5. Discusión general de los resultados

Observando la tabla 5.1, se observa una mejora en el número de aciertos, sobre todo en las escenas de *airfield* y *field/cultivated*, con 16 y 13 aciertos más respectivamente. En el resto también existe una mejora salvo en *building_facade, coast* y *mountain*, que son las escenas con un número de aciertos más bajo y los aciertos disminuyen. Por lo tanto, en las escenas con un número de aciertos bajo con la red Resnet50, no se obtiene una mejora, sino que se obtiene un peor resultado, mientras que en escenas con un mayor número de aciertos se obtiene un mejor resultado.

Por otro lado, tras lo visto en las Figuras 5.4, 5.5, 5.6, 5.7 y 5.8, se comprueba como con este sistema la red va modificando sus máscaras de focalización y aportando categorías distintas a la inicial, consiguiendo en algunos casos que coincida con la categoría devuelta por la red convolucional de segmentación semántica, que si ha acertado, acierta también el sistema diseñado.

En definitiva, con los resultados obtenidos, se establece que el sistema implementado es correcto y funciona, ya que sí se puede obtener una mejora del número de aciertos, pero aún tiene un gran margen de mejora para evitar estas disminuciones de aciertos. Por ejemplo si se desarrollaran histogramas ponderados considerando un mayor número de objetos, estos serían más discriminativos y por lo tanto, se obtendría una red neuronal artificial con mejores resultados, provocando que se pueda alcanzar un número de aciertos mayor con el sistema.

Capítulo 6

Conclusiones y trabajo futuro

En este apartado se presentan las conclusiones y trabajos futuros a partir de lo visto en la presente memoria.

6.1. Conclusiones

Tras todo lo expuesto, este trabajo de fin de máster presenta las siguientes conclusiones:

1. Las aquitecturas más complejas y con mayor profundidad son las que mejor tasa de acierto aportan, aunque todas ellas, al estar entrenadas con las mismas imágenes, tienen un comportamiento similar.

2. Ante imágenes de escenas no presentes en el *dataset* de entrenamiento los modelos se comportan de forma similar, dando una misma salida o bien dando puntuaciones similares a distintas salidas. Por otro lado, se observa que los modelos pueden mejorar el número de aciertos ante una alteración de las imágenes, lo que indica que se puede conseguir un mejor resultado sin re-entrenar las redes.

3. El sistema de re-focalización basado en el uso de la información semántica de las imágenes aporta mejor tasa de aciertos en algunas escenas, mientras que en otras escenas con peor tasa de acierto empeora. Se concluye que el sistema funciona pero que aún tiene margen de mejora en distintos aspectos, sobre todo obteniendo un método de obtención de H_{hs} que sea más discriminatorio entre las distintas escenas.

6.2. Trabajo futuro

Tras lo visto en este trabajo, se pueden establecer una serie de puntos para un trabajo futuro:

1. Conseguir los mapas de activación de la red Densenet161 [4], ya que es la que mejor tasa de acierto tiene.

2. Buscar un método mejor para modificar las imágenes que no sea cargarse las zonas del mapa de activación con mayor valor.

3. Obtener un sistema de obtención de H_{hs} más discriminatorio de modo que se pueda crear una red NN con mejor resultado entre todas las 365 escenas.

Apéndice A

Categorías de Places365

En este apéndice se muestra una lista de las categorías de Places365 [6] que aportan las redes a su salida ordenadas alfabéticamente con el número que les corresponde:

1. airfield
2. airplane cabin
3. airport terminal
4. alcove
5. alley
6. amphitheater
7. amusement arcade
8. amusement park
9. apartment building/outdoor
10. aquarium
11. aqueduct
12. arcade
13. arch
14. archaelogical excavation
15. archive
16. arena/hockey
17. arena/performance
18. arena/rodeo
19. army base
20. art gallery
21. art school
22. art studio
23. artists loft
24. assembly line
25. athletic field/outdoor
26. atrium/public
27. attic
28. auditorium
29. auto factory
30. auto showroom
31. badlands
32. bakery/shop
33. balcony/exterior
34. balcony/interior
35. ball pit
36. ballroom
37. bamboo forest
38. bank vault
39. banquet hall
40. bar
41. barn
42. barndoor
43. baseball field
44. basement
45. basketball court/indoor
46. bathroom
47. bazaar/indoor
48. bazaar/outdoor
49. beach
50. beach house
51. beauty salon
52. bedchamber
53. bedroom
54. beer garden
55. beer hall
56. berth
57. biology laboratory
58. boardwalk
59. boat deck
60. boathouse
61. bookstore
62. booth/indoor
63. botanical garden
64. bow window/indoor
65. bowling alley
66. boxing ring
67. bridge
68. building facade
69. bullring
70. burial chamber
71. bus interior
72. bus station/indoor
73. butchers shop
74. butte
75. cabin/outdoor
76. cafeteria
77. campsite
78. campus
79. canal/natural
80. canal/urban
81. candy store
82. canyon
83. car interior
84. carrousel
85. castle
86. catacomb
87. cemetery
88. chalet
89. chemistry lab
90. childs room
91. church/indoor
92. church/outdoor
93. classroom
94. clean room
95. cliff
96. closet
97. clothing store
98. coast
99. cockpit
100.coffee shop
101.computer room
102.conference center
103.conference room
104.construction site
105.corn field
106.corral
107.corridor
108.cottage
109.courthouse
110.courtyard
111.creek
112.crevasse
113.crosswalk
114.dam
115.delicatessen
116.department store
117.desert/sand
118.desert/vegetation
119.desert road
120.diner/outdoor
121.dining hall
122.dining room
123.discotheque
124.doorway/outdoor
125.dorm room
126.downtown
127.dressing room
128.driveway
129.drugstore
130.elevator/door
131.elevator lobby
132.elevator shaft
133.embassy
134.engine room
135.entrance hall
136.escalator/indoor
137.excavation
138.fabric store
139.farm
140.fastfood restaurant
141.field/cultivated
142.field/wild
143.field road
144.fire escape
145.fire station
146.fishpond
147.flea market/indoor
148.florist shop/indoor
149.food court
150.football field
151.forest/broadleaf
152.forest path
153.forest road
154.formal garden
155.fountain
156.galley
157.garage/indoor
158.garage/outdoor
159.gas station
160.gazebo/exterior

Figura A.1: Lista de las categorías del Places365 [6] con su número asociado de la 1 a la 160.

En las siguientes figuras desde A.4 hasta A.13 se observa una imagen de ejemplo de cada escena:

161.general store/indoor
162.general store/outdoor
163.gift shop
164.glacier
165.golf course
166.greenhouse/indoor
167.greenhouse/outdoor
168.grotto
169.gymnasium/indoor
170.hangar/indoor
171.hangar/outdoor
172.harbor
173.hardware store
174.hayfield
175.heliport
176.highway
177.home office
178.home theater
179.hospital
180.hospital room
181.hot spring
182.hotel/outdoor
183.hotel room
184.house
185.hunting lodge/outdoor
186.ice cream parlor
187.ice floe
188.ice shelf
189.ice skating rink/indoor
190.ice skating rink/outdoor
191.iceberg
192.igloo
193.industrial area
194.inn/outdoor
195.islet
196.jacuzzi/indoor
197.jail cell
198.japanese garden
199.jewelry shop
200.junkyard

201.kasbah
202.kennel/outdoor
203.kindergarden classroom
204.kitchen
205.lagoon
206.lake/natural
207.landfill
208.landing deck
209.laundromat
210.lawn
211.lecture room
212.legislative chamber
213.library/indoor
214.library/outdoor
215.lighthouse
216.living room
217.loading dock
218.lobby
219.lock chamber
220.locker room
221.mansion
222.manufactured home
223.market/indoor
224.market/outdoor
225.marsh
226.martial arts gym
227.mausoleum
228.medina
229.mezzanine
230.moat/water
231.mosque/outdoor
232.motel
233.mountain
234.mountain path
235.mountain snowy
236.movie theater/indoor
237.museum/indoor
238.museum/outdoor
239.music studio
240.natural history museum

241.nursery
242.nursing home
243.oast house
244.ocean
245.office
246.office building
247.office cubicles
248.oilrig
249.operating room
250.orchard
251.orchestra pit
252.pagoda
253.palace
254.pantry
255.park
256.parking garage/indoor
257.parking garage/outdoor
258.parking lot
259.pasture
260.patio
261.pavilion
262.pet shop
263.pharmacy
264.phone booth
265.physics laboratory
266.picnic area
267.pier
268.pizzeria
269.playground
270.playroom
271.plaza
272.pond
273.porch
274.promenade
275.pub/indoor
276.racecourse
277.raceway
278.raft
279.railroad track
280.rainforest

281.reception
282.recreation room
283.repair shop
284.residential neighborhood
285.restaurant
286.restaurant kitchen
287.restaurant patio
288.rice paddy
289.river
290.rock arch
291.roof garden
292.rope bridge
293.ruin
294.runway
295.sandbox
296.sauna
297.schoolhouse
298.science museum
299.server room
300.shed
301.shoe shop
302.shopfront
303.shopping mall/indoor
304.shower
305.ski resort
306.ski slope
307.sky
308.skyscraper
309.slum
310.snowfield
311.soccer field
312.stable
313.stadium/baseball
314.stadium/football
315.stadium/soccer
316.stage/indoor
317.stage/outdoor
318.staircase
319.storage room
320.street

Figura A.2: Lista de las categorías del Places365 [6] con su número asociado de la 161 a la 320.

321.subway station/platform
322.supermarket
323.sushi bar
324.swamp
325.swimming hole
326.swimming pool/indoor
327.swimming pool/outdoor
328.synagogue/outdoor
329.television room
330.television studio
331.temple/asia
332.throne room
333.ticket booth
334.topiary garden
335.tower
336.toyshop
337.train interior
338.train station/platform
339.tree farm
340.tree house
341.trench
342.tundra
343.underwater/ocean deep
344.utility room
345.valley
346.vegetable garden
347.veterinarians office
348.viaduct
349.village
350.vineyard
351.volcano
352.volleyball court/outdoor
353.waiting room
354.water park
355.water tower
356.waterfall
357.watering hole
358.wave
359.wet bar
360.wheat field
361.wind farm
362.windmill
363.yard
364.youth hostel
365.zen garden

Figura A.3: Lista de las categorías del Places365 [6] con su número asociado de la 321 a la 365.

Figura A.4: En esta figura se representa un ejemplo de una de las imágenes de cada categoria, siendo la primera la categoría 1 y la última la 40.

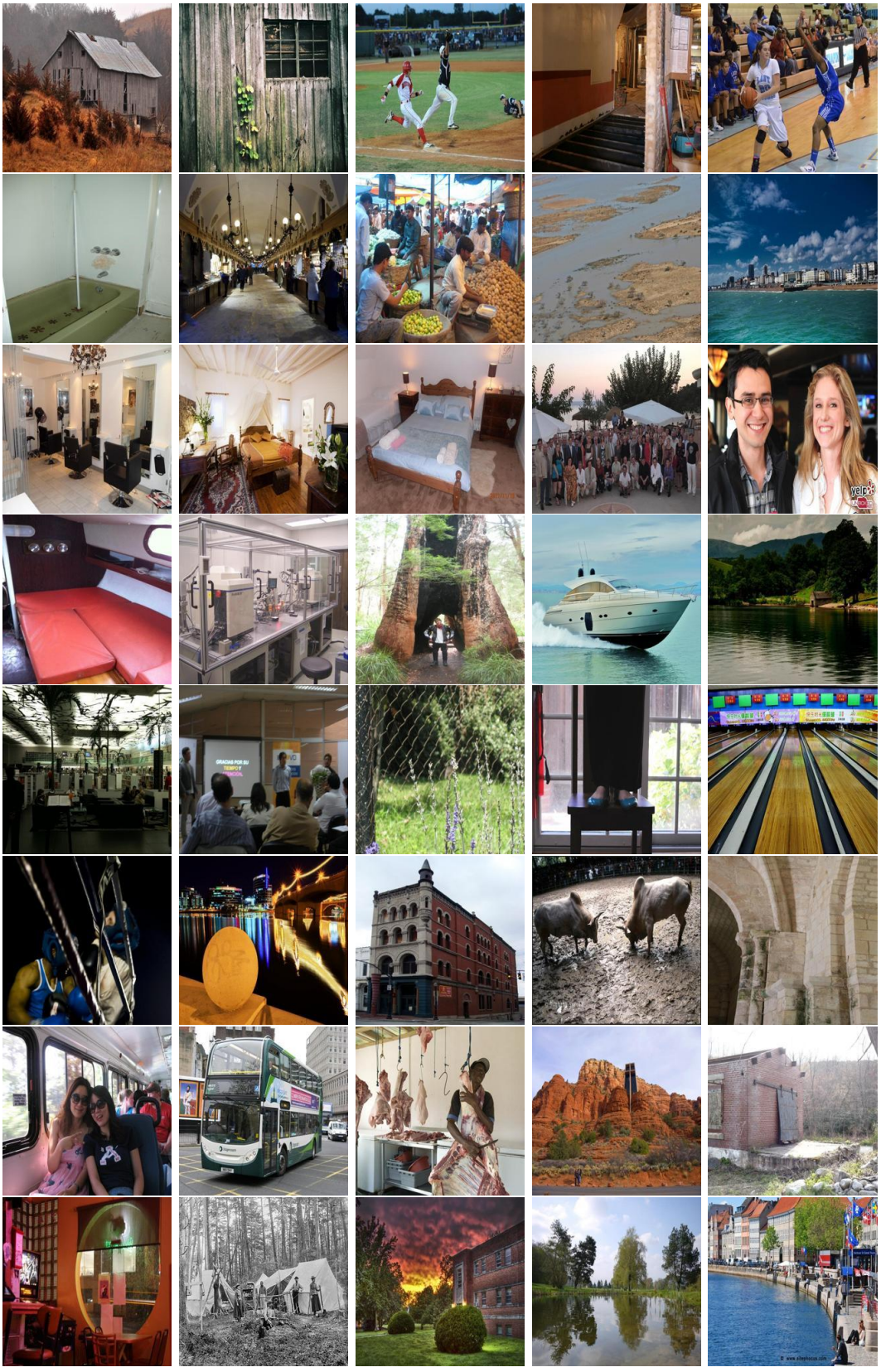

Figura A.5: En esta figura se representa un ejemplo de una de las imágenes de cada categoria, siendo la primera la categoría 41 y la última la 80.

Figura A.6: En esta figura se representa un ejemplo de una de las imágenes de cada categoria, siendo la primera la categoría 81 y la última la 120.

Figura A.7: En esta figura se representa un ejemplo de una de las imágenes de cada categoria, siendo la primera la categoría 121 y la última la 160.

Figura A.8: En esta figura se representa un ejemplo de una de las imágenes de cada categoria, siendo la primera la categoría 161 y la última la 200.

Figura A.9: En esta figura se representa un ejemplo de una de las imágenes de cada categoria, siendo la primera la categoría 201 y la última la 240.

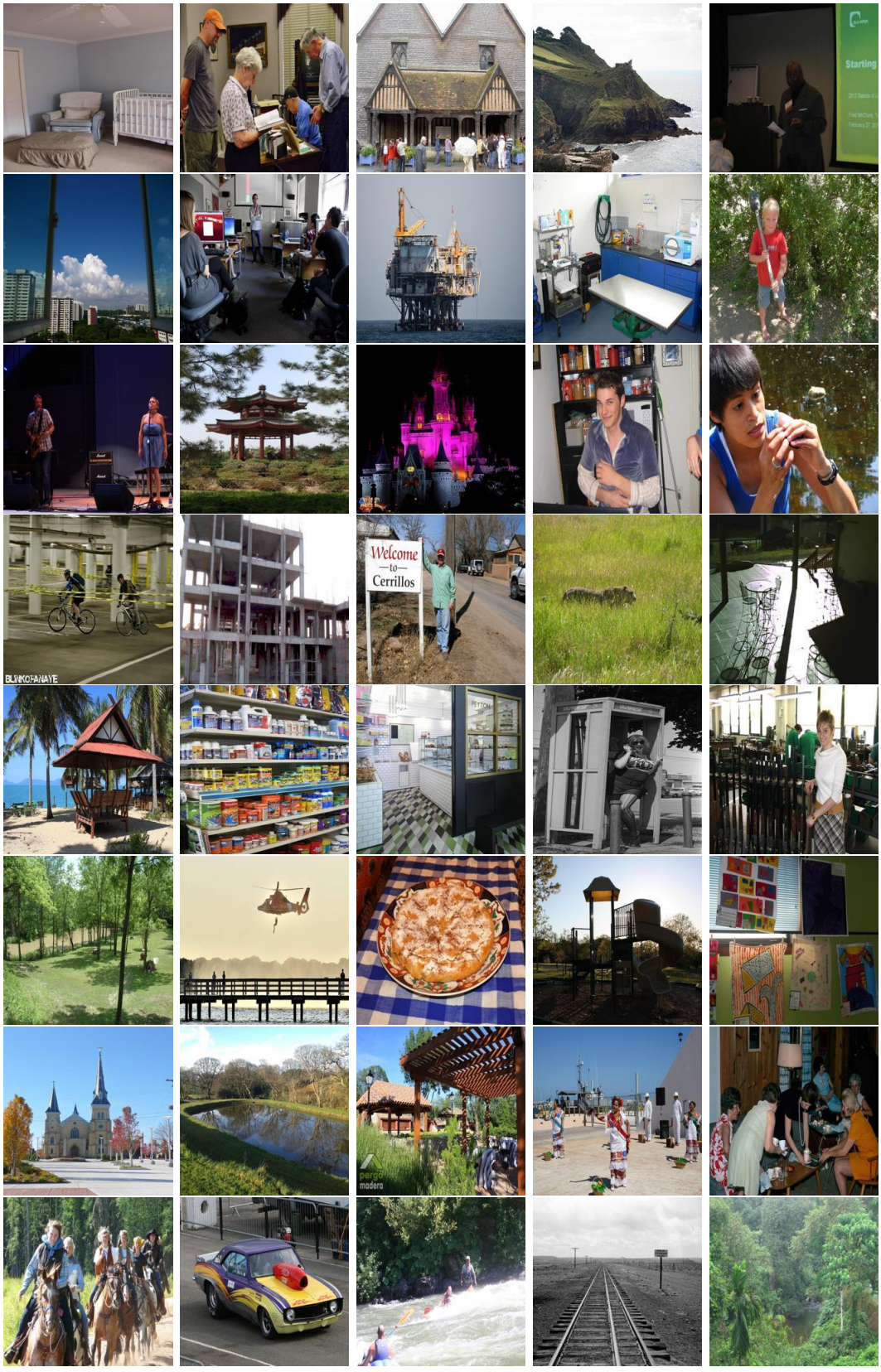

Figura A.10: En esta figura se representa un ejemplo de una de las imágenes de cada categoria, siendo la primera la categoría 241 y la última la 280.

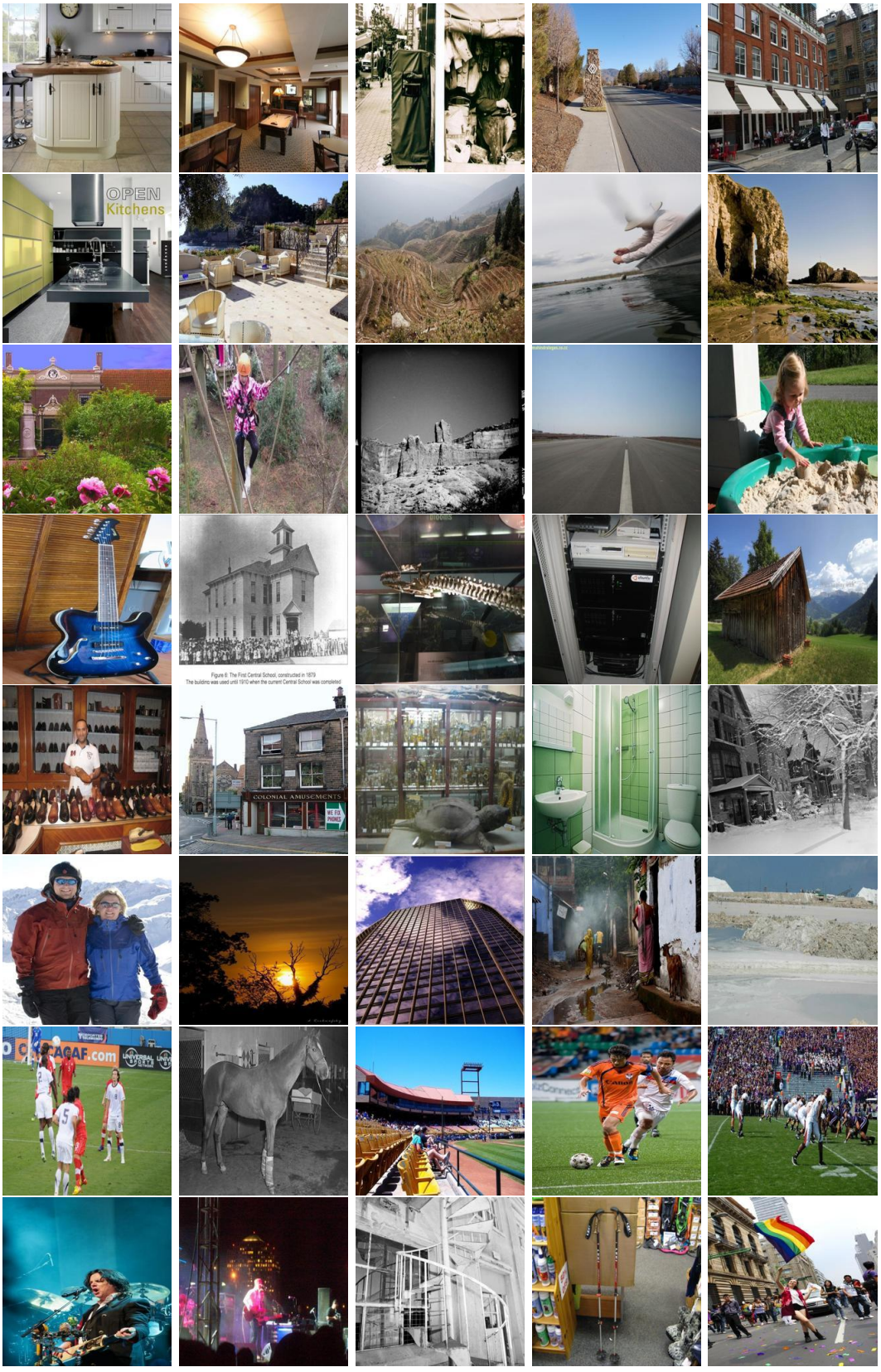

Figura A.11: En esta figura se representa un ejemplo de una de las imágenes de cada categoria, siendo la primera la categoría 281 y la última la 320.

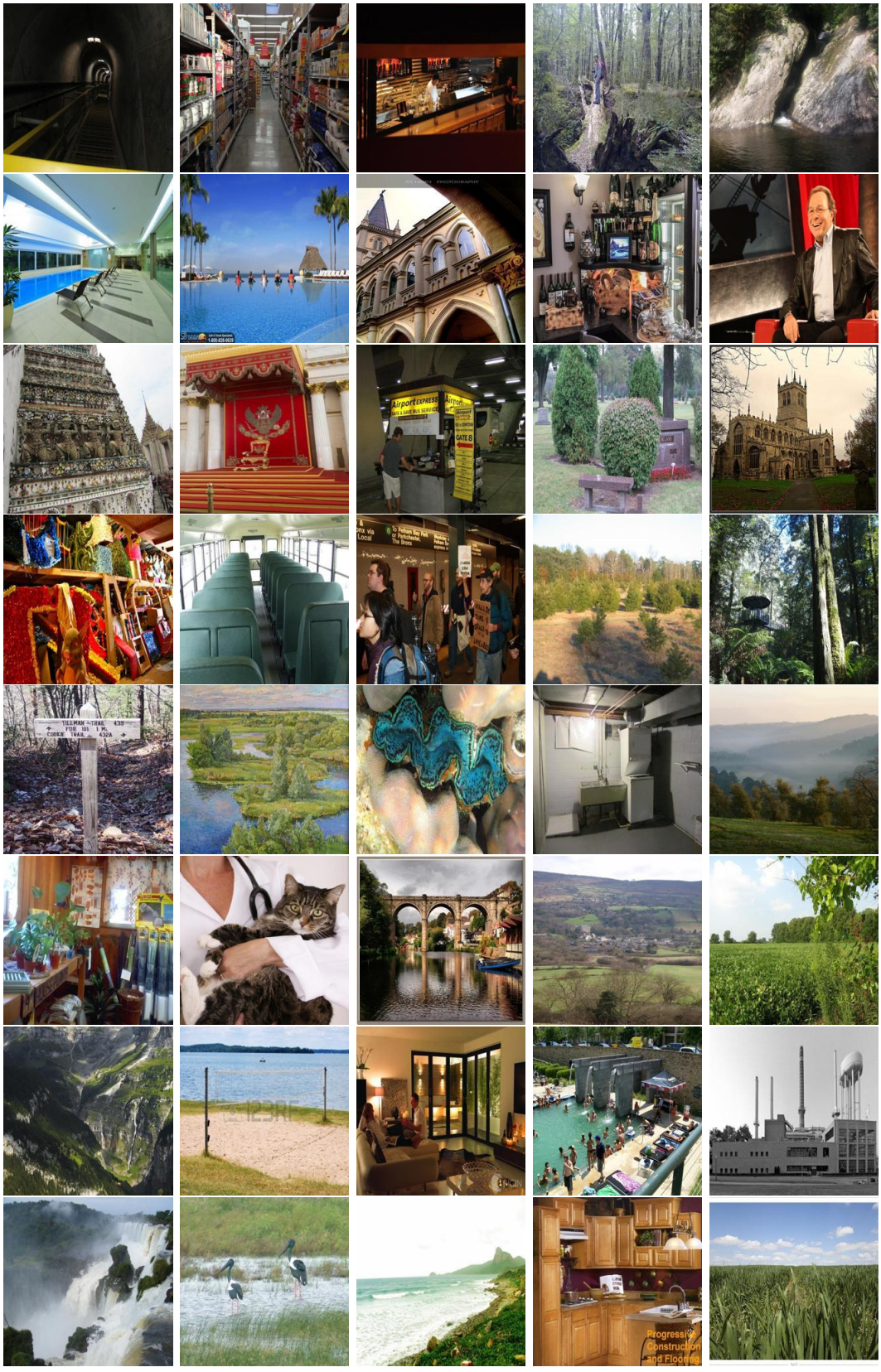

Figura A.12: En esta figura se representa un ejemplo de una de las imágenes de cada categoria, siendo la primera la categoría 321 y la última la 360.

Figura A.13: En esta figura se representa un ejemplo de una de las imágenes de cada categoria, siendo la primera la categoría 361 y la última la 365.

Apéndice B

Categorías semánticas de AD20K

En este apéndice hay una lista de los objetos que aparecen en las máscaras semánticas junto con el valor que tiene dicho objeto en la máscara como aparecen en [19]:

1. 'wall'
2. 'building'
3. 'sky'
4. 'floor'
5. 'tree'
6. 'ceiling'
7. 'road'
8. 'bed'
9. 'windowpane'
10. 'grass'
11. 'cabinet'
12. 'sidewalk'
13. 'person'
14. 'earth'
15. 'door'
16. 'table'
17. 'mountain'
18. 'plant'
19. 'curtain'
20. 'chair'
21. 'car'
22. 'water'
23. 'painting'
24. 'sofa'
25. 'shelf'
26. 'house'
27. 'sea'
28. 'mirror'
29. 'rug'
30. 'field'
31. 'armchair'
32. 'seat'
33. 'fence'
34. 'desk'
35. 'rock'
36. 'wardrobe'
37. 'lamp'
38. 'bathtub'
39. 'railing'
40. 'cushion'
41. 'base'
42. 'box'
43. 'column'
44. 'signboard'
45. 'chest of drawers'
46. 'counter'
47. 'sand'
48. 'sink'
49. 'skyscraper'
50. 'fireplace'
51. 'refrigerator'
52. 'grandstand'
53. 'path'
54. 'stairs'
55. 'runway'
56. 'case'
57. 'pool table'
58. 'pillow'
59. 'screen door'
60. 'stairway'
61. 'river'
62. 'bridge'
63. 'bookcase'
64. 'blind'
65. 'coffee table'
66. 'toilet'
67. 'flower'
68. 'book'
69. 'hill'
70. 'bench'
71. 'countertop'
72. 'stove'
73. 'palm'
74. 'kitchen island'
75. 'computer'
76. 'swivel chair'
77. 'boat'
78. 'bar'
79. 'arcade machine'
80. 'hovel'
81. 'bus'
82. 'towel'
83. 'light'
84. 'truck'
85. 'tower'
86. 'chandelier'
87. 'awning'
88. 'streetlight'
89. 'booth'
90. 'television receiver'
91. 'airplane'
92. 'dirt track'
93. 'apparel'
94. 'pole'
95. 'land'
96. 'bannister'
97. 'escalator'
98. 'ottoman'
99. 'bottle'
100.'buffet'
101.'poster'
102.'stage'
103.'van'
104.'ship'
105.'fountain'
106.'conveyer belt'
107.'canopy'
108.'washer'
109.'plaything'
110.'swimming pool'
111.'stool'
112.'barrel'
113.'basket'
114.'waterfall'
115.'tent'
116.'bag'
117.'minibike'
118.'cradle'
119.'oven'
120.'ball'
121.'food'
122.'step'
123.'tank'
124.'trade name'
125.'microwave'
126.'pot'
127.'animal'
128.'bicycle'
129.'lake'
130.'dishwasher'
131.'screen'
132.'blanket'
133.'sculpture'
134.'hood'
135.'sconce'
136.'vase'
137.'traffic light'
138.'tray'
139.'ashcan'
140.'fan'
141.'pier'
142.'crt screen'
143.'plate'
144.'monitor'
145.'bulletin board'
146.'shower'
147.'radiator'
148.'glass'
149.'clock'
150.'flag'

Figura B.1: Lista de las categorías semánticas de AD20K.

Apéndice C

Figuras de robustez del resto de redes

- Alexnet

Las figuras de C.1 muestran los resultados de la robustez al ruido para Alexnet.

- VGG16

Las figuras de C.2 muestran los resultados de la robustez al ruido para VGG16.

- Resnet18

Las figuras de C.3 muestran los resultados de la robustez al ruido para Resnet18.

- Densenet161

Las figuras de C.4 muestran los resultados de la robustez al ruido para Densenet161.

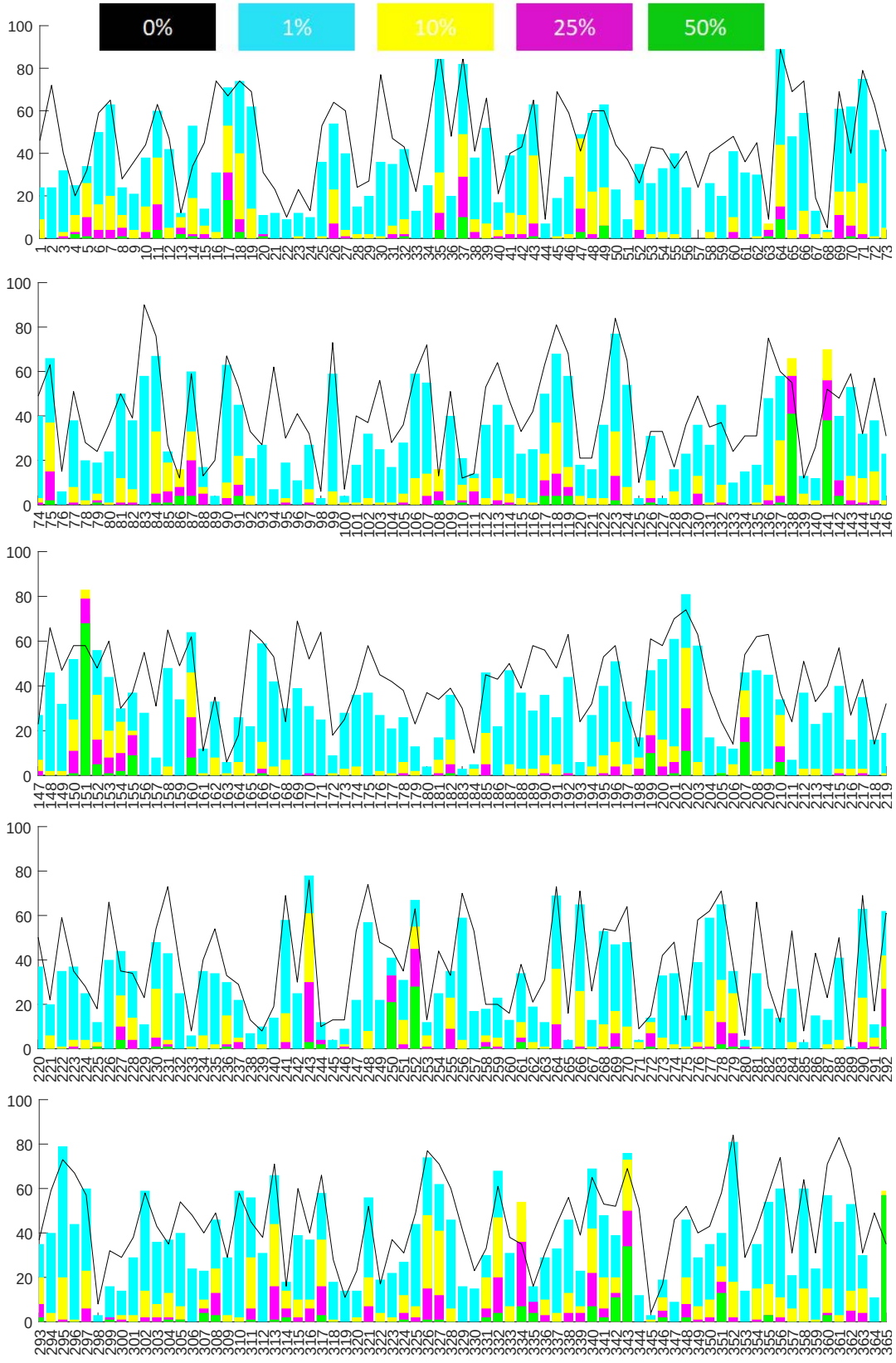

Figura C.1: Gráfica de barras en las que el eje X muestra categorías y el eje Y el número de aciertos en cada porcentaje de píxeles a 0 con la red Alexnet.

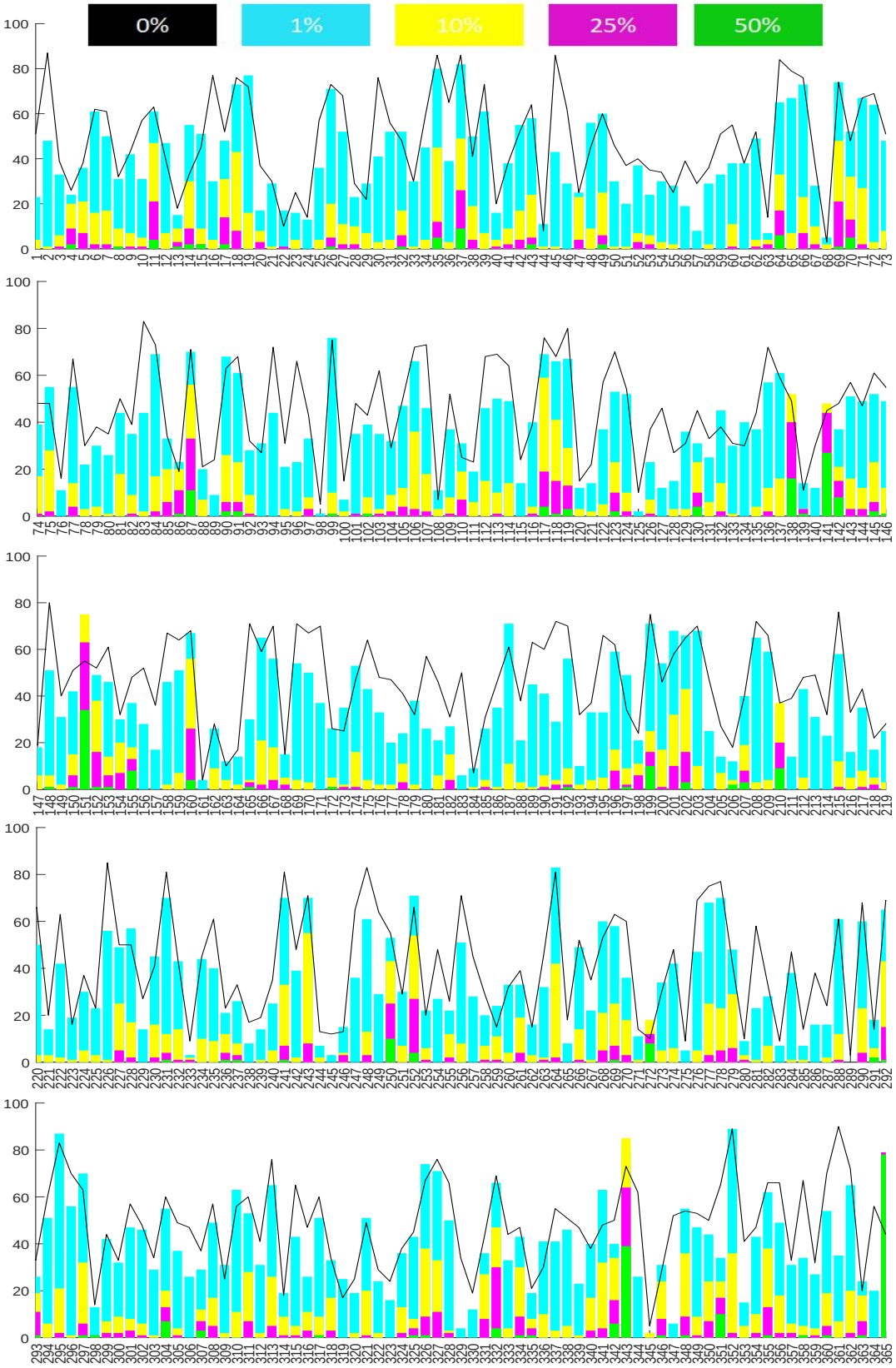

Figura C.2: Gráfica de barras en las que el eje X muestra categorías y el eje Y el número de aciertos en cada porcentaje de píxeles a 0 con la red VGG16.

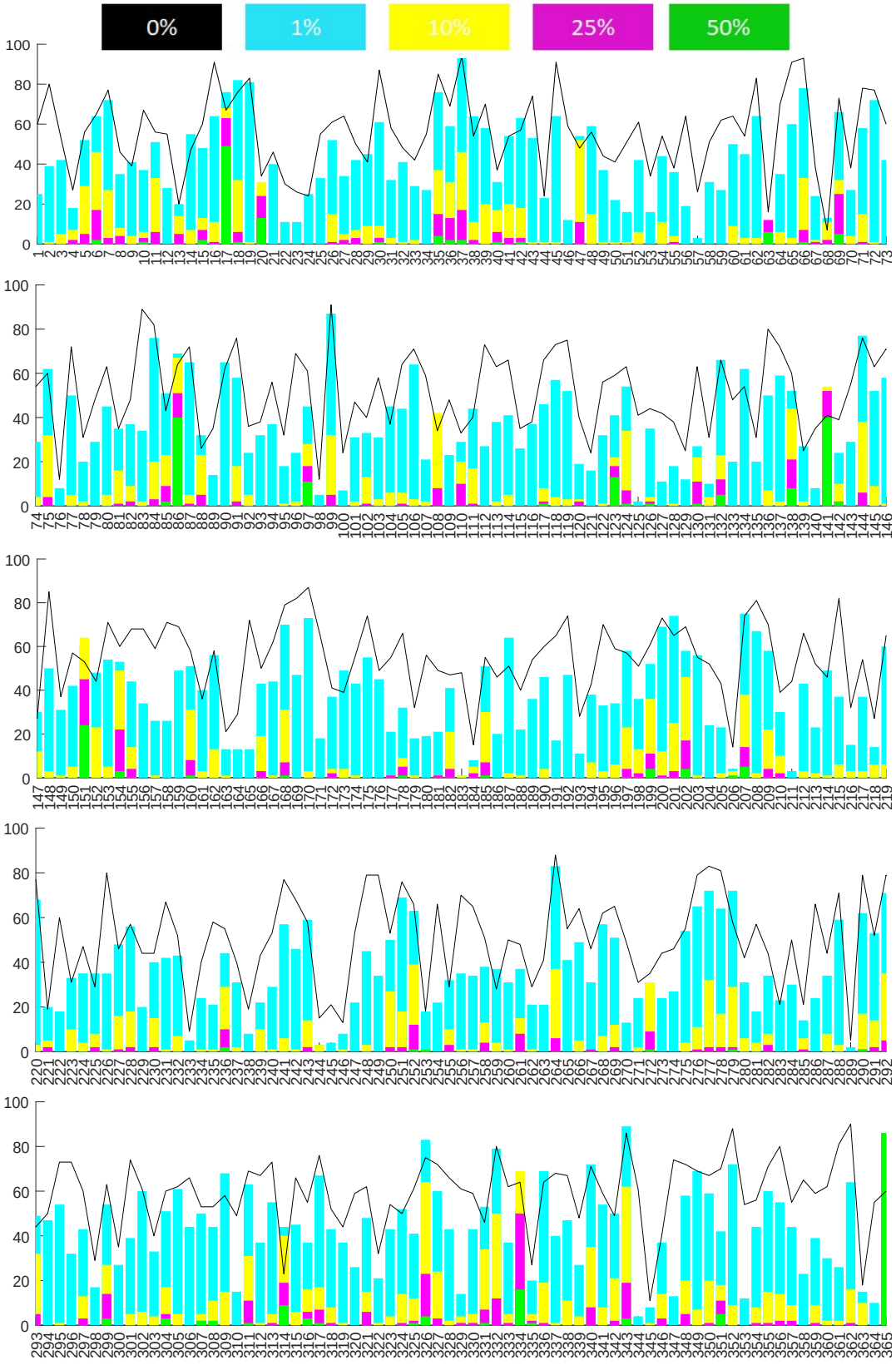

Figura C.3: Gráfica de barras en las que el eje X muestra categorías y el eje Y el número de aciertos en cada porcentaje de píxeles a 0 con la red Resnet18.

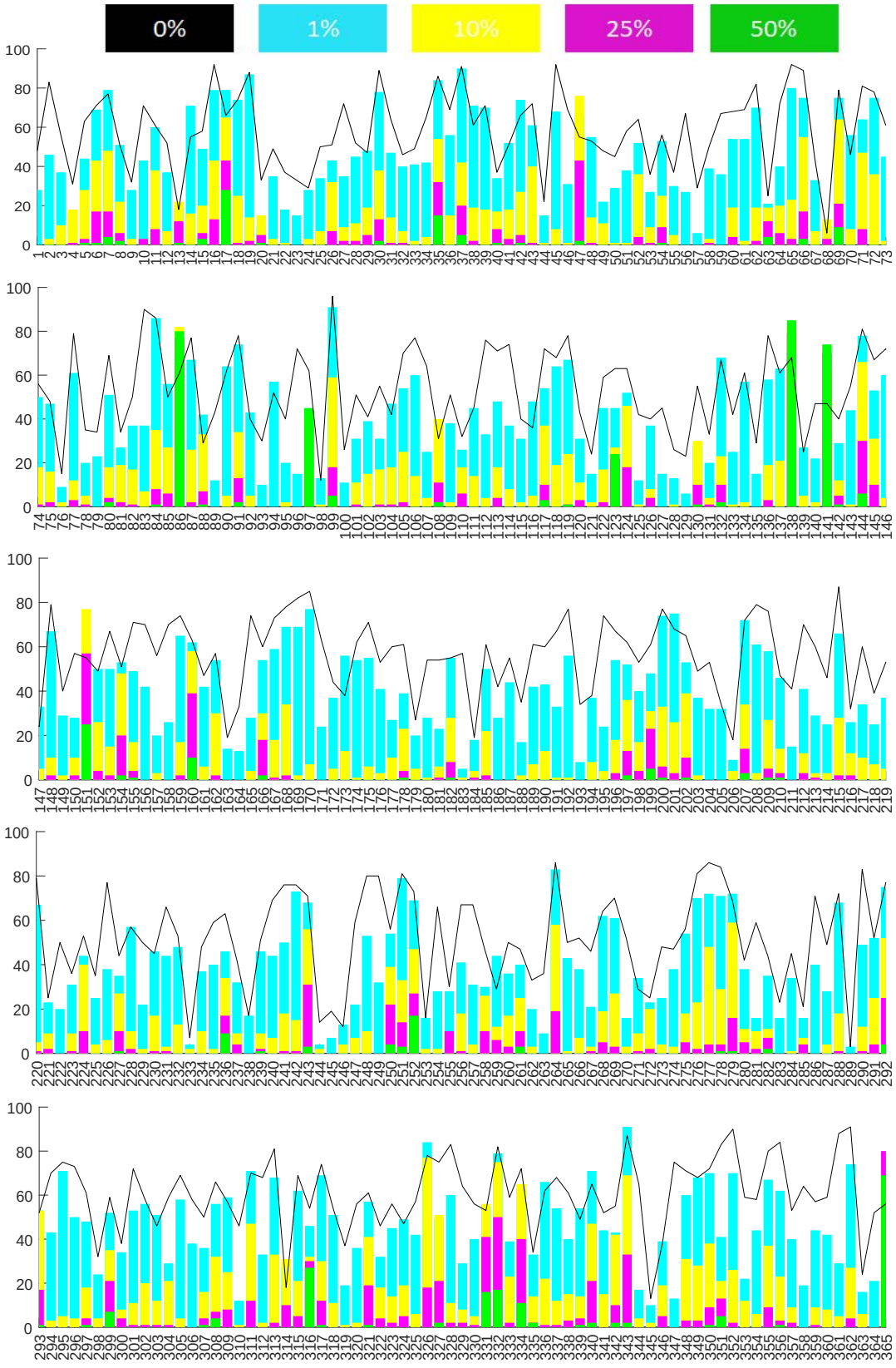

Figura C.4: Gráfica de barras en las que el eje X muestra categorías y el eje Y el número de aciertos en cada porcentaje de píxeles a 0 con la red Densenet161.

Bibliografía

[1] A. Krizhevsky, I. Sutskever, G. E. Hinton, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, no. 1, pp. 1097–1105, 2012.

[2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large scale image recognition," *International Conference on Learning Representations, ICLR*, 2015.

[3] e. a. He, Kaiming, "Deep residual learning for image recognition," *Proceedings of the IEEE conference on computer vision and pattern recognition, CVPR*, 2016.

[4] e. a. Huang, Gao, "Densely connected convolutional networks," *Proceedings of the IEEE conference on computer vision and pattern recognition. CVPR*, 2017.

[5] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," *CoRR*, 2016.

[6] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, no. 99, pp. 1–1, 2017.

[7] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *Int. J. Comp. Vis.*, vol. 42, pp. 145–175, Mar 2001.

[8] L. Fei-Fei and P. Perona, "A bayesian hierarchical model for learning natural scene categories," *Computer Vision and Pattern Recognition*, pp. 524–531, 2005.

[9] K. Ho and P. Newman, "Detecting loop closure with scene sequences," *Int. J. Comp. Vis.*, vol. 74, pp. 261–286, Mar 2007.

[10] A. e. a. Torralba, "Context-based vision system for place and object recognition," *IEEE International Conference on Computer Vision*, pp. 273–280, 2003.

[11] e. a. K. He, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," *Computer Vision and Pattern Recognition*, pp. 1026–1034, 2015.

[12] e. a. Zhou, B., "Places: An image database for deep scene understanding," *arXiv:1610.02055*, 2016.

[13] e. a. Zhou, B., "Object detectors emerge in deep scene cnns," *arXiv:1412.6856*, 2014.

[14] "Understand resnet alexnet vgg inception."

[15] "A brief report of the heuritech deep learning meetup."

[16] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional Architecture for Fast Feature Embedding," *ArXiv e-prints*, June 2014.

[17] B. Zhou, A. Khosla, L. A., A. Oliva, and A. Torralba, "Learning deep features for discriminative localization.," *CVPR*, 2016.

[18] "Mathworks documentation: Patternnet."

[19] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Semantic understanding of scenes through the ade20k dataset," *CoRR*, vol. 1608, 2016.