COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI

GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

TRABAJO DE FIN DE GRADO

INTERPRETACIÓN Y ANÁLISIS DE SENSIBILIDAD DE CNN MEDIANTE IMÁGENES SINTÉTICAS

Autora:
Gabriela Martín Carballo

Directores:
Dr. Jaime Boal Martín-Larrauri
Dr. Eugenio Sánchez Úbeda

Mayo 2019

AUTORIZACIÓN PARA LA DIGITALIZACIÓN, DEPÓSITO Y DIVULGACIÓN EN RED DE PROYECTOS FIN DE GRADO, FIN DE MÁSTER, TESINAS O MEMORIAS DE BACHILLERATO

1º. Declaración de la autoría y acreditación de la misma.

La autora **Dª. GABRIELA MARTIN CARBALLO, DEC**LARA ser el titular de los derechos de propiedad intelectual de la obra: "**INTERPRETACIÓN Y ANÁLISIS DE SENSIBILIDAD DE CNN MEDIANTE IMÁGENES SINTÉTICAS**", que ésta es una obra original, y que ostenta la condición de autor en el sentido que otorga la Ley de Propiedad Intelectual.

2º. Objeto y fines de la cesión.

Con el fin de dar la máxima difusión a la obra citada a través del Repositorio institucional de la Universidad, el autor **CEDE** a la Universidad Pontificia Comillas, de forma gratuita y no exclusiva, por el máximo plazo legal y con ámbito universal, los derechos de digitalización, de archivo, de reproducción, de distribución y de comunicación pública, incluido el derecho de puesta a disposición electrónica, tal y como se describen en la Ley de Propiedad Intelectual. El derecho de transformación se cede a los únicos efectos de lo dispuesto en la letra a) del apartado siguiente.

3º. Condiciones de la cesión y acceso

Sin perjuicio de la titularidad de la obra, que sigue correspondiendo a su autor, la cesión de derechos contemplada en esta licencia habilita para:

a) Transformarla con el fin de adaptarla a cualquier tecnología que permita incorporarla a internet y hacerla accesible; incorporar metadatos para realizar el registro de la obra e incorporar "marcas de agua" o cualquier otro sistema de seguridad o de protección.
b) Reproducirla en un soporte digital para su incorporación a una base de datos electrónica, incluyendo el derecho de reproducir y almacenar la obra en servidores, a los efectos de garantizar su seguridad, conservación y preservar el formato.
c) Comunicarla, por defecto, a través de un archivo institucional abierto, accesible de modo libre y gratuito a través de internet.
d) Cualquier otra forma de acceso (restringido, embargado, cerrado) deberá solicitarse expresamente y obedecer a causas justificadas.
e) Asignar por defecto a estos trabajos una licencia Creative Commons.
f) Asignar por defecto a estos trabajos un HANDLE (URL *persistente)*.

4º. Derechos del autor.

El autor, en tanto que titular de una obra tiene derecho a:

a) Que la Universidad identifique claramente su nombre como autor de la misma
b) Comunicar y dar publicidad a la obra en la versión que ceda y en otras posteriores a través de cualquier medio.
c) Solicitar la retirada de la obra del repositorio por causa justificada.
d) Recibir notificación fehaciente de cualquier reclamación que puedan formular terceras personas en relación con la obra y, en particular, de reclamaciones relativas a los derechos de propiedad intelectual sobre ella.

5º. Deberes del autor.

El autor se compromete a:

a) Garantizar que el compromiso que adquiere mediante el presente escrito no infringe ningún derecho de terceros, ya sean de propiedad industrial, intelectual o cualquier otro.

b) Garantizar que el contenido de las obras no atenta contra los derechos al honor, a la intimidad y a la imagen de terceros.
c) Asumir toda reclamación o responsabilidad, incluyendo las indemnizaciones por daños, que pudieran ejercitarse contra la Universidad por terceros que vieran infringidos sus derechos e intereses a causa de la cesión.
d) Asumir la responsabilidad en el caso de que las instituciones fueran condenadas por infracción de derechos derivada de las obras objeto de la cesión.

6º. Fines y funcionamiento del Repositorio Institucional.

La obra se pondrá a disposición de los usuarios para que hagan de ella un uso justo y respetuoso con los derechos del autor, según lo permitido por la legislación aplicable, y con fines de estudio, investigación, o cualquier otro fin lícito. Con dicha finalidad, la Universidad asume los siguientes deberes y se reserva las siguientes facultades:

- La Universidad informará a los usuarios del archivo sobre los usos permitidos, y no garantiza ni asume responsabilidad alguna por otras formas en que los usuarios hagan un uso posterior de las obras no conforme con la legislación vigente. El uso posterior, más allá de la copia privada, requerirá que se cite la fuente y se reconozca la autoría, que no se obtenga beneficio comercial, y que no se realicen obras derivadas.
- La Universidad no revisará el contenido de las obras, que en todo caso permanecerá bajo la responsabilidad exclusive del autor y no estará obligada a ejercitar acciones legales en nombre del autor en el supuesto de infracciones a derechos de propiedad intelectual derivados del depósito y archivo de las obras. El autor renuncia a cualquier reclamación frente a la Universidad por las formas no ajustadas a la legislación vigente en que los usuarios hagan uso de las obras.
- La Universidad adoptará las medidas necesarias para la preservación de la obra en un futuro.
- La Universidad se reserva la facultad de retirar la obra, previa notificación al autor, en supuestos suficientemente justificados, o en caso de reclamaciones de terceros.

Madrid, a 29 de Agosto de 2019.

ACEPTA

Fdo 

Motivos para solicitar el acceso restringido, cerrado o embargado del trabajo en el Repositorio Institucional:

DECLARO, bajo mi responsabilidad, que el Proyecto presentado con el título **Interpretación y análisis de sensibilidad de CNN mediante imágenes sintéticas** en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el curso académico 2018/2019 es de mi autoría, original e inédito y no ha sido presentado con anterioridad a otros efectos. El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido tomada de otros documentos está debidamente referenciada.

Fdo.: Gabriela Martín Carballo Fecha: 29/ 08/2019

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.: Jaime Boal Martín- Larrauri Fecha: 29/ 08/ 2019

Eugenio FranciscoSánchez Ubeda

COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI

GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

TRABAJO DE FIN DE GRADO

INTERPRETACIÓN Y ANÁLISIS DE SENSIBILIDAD DE CNN MEDIANTE IMÁGENES SINTÉTICAS

Autora:
Gabriela Martín Carballo

Directores:
Dr. Jaime Boal Martín-Larrauri
Dr. Eugenio Sánchez Úbeda

Mayo 2019

AGRADECIMIENTOS

El presente proyecto ha sido para mi tan apasionante como retador. Realmente, su culminación ha sido un verdadero camino de obstáculos que, sin la ayuda de muchas y muy diferentes personas, no hubiera logrado el resultado esperado.

En primer lugar, obviamente, quiero destacar la dedicación, comprensión, sensibilidad, flexibilidad y paciencia de Jaime Boal y Eugenio Sánchez, mis tutores del proyecto sin los cuales no hubiera visto la luz. Muchas gracias por todo el tiempo invertido.

En segundo lugar, a mi familia que no sólo ha soportado mis momentos de presión y/o desasosiego, sino que han sido de ayuda en todo momento.

Y por último a aquellas personas que me han ayudado cuando no sabía como seguir con el proyecto, dónde encontrar nuevas vías para mejorar. Gracias por todos los consejos.

INTERPRETACIÓN Y ANÁLISIS DE SENSIBILIDAD DE CNN MEDIANTE IMÁGENES SINTÉTICAS

Autor: **Martín Carballo, Gabriela.**
Director: Boal Martín-Larrauri, Jaime.
Sánchez Úbeda, Eugenio Francisco.
Entidad Colaboradora: ICAI – Universidad Pontificia Comillas.

RESUMEN DEL PROYECTO

1. INTRODUCCION

Uno de los pilares fundamentales del proceso de digitalización de la industria en el que nos encontramos actualmente inmersos, es la integración de fuentes de datos de procedencias diversas que permitan mejorar el conocimiento y la operación de los sistemas. Este tipo de datos puede ser desde texto hasta imágenes. Las redes neuronales convolucionales trabajan con este segundo tipo, las imágenes. Cabe recordar que los seres humanos también utilizamos la visión como nuestro sensor principal para interactuar con el entorno.

2. ESTADO DEL ARTE

Las redes neuronales convolucionales son redes computacionales organizadas jerárquicamente, y compuestas por un gran número de neuronas artificiales, las cuales tratan de simular el funcionamiento del sistema nervioso biológico.

Las redes neuronales convolucionales (en adelante, CNN, acrónimo del término anglosajón Convolutional Neural Networks) han revolucionado la forma de trabajar con imágenes, llegando incluso a superar al ser humano en tareas que nos resultan tan naturales como identificar y clasificar objetos por categorías. Las redes convolucionales se diferencian de otros tipos de redes neuronales por llevar a cabo la convolución. Se entiende por convolución al filtrado de una imagen mediante una máscara (kernel). La convolución consiste en tomar submatrices de la matriz de píxeles de la imagen de entrada, y operar mediante producto escalar con la matriz de filtrado llamada kernel. Cada píxel de salida será la combinación lineal de los pixeles de entrada.

Las capas principales de las que consta una red convolucional son:

- **Capa convolucional:** Lleva a cabo la convolución con las matrices de filtrado (*kernel*).
- **Capa de activación ReLU:** El fin de esta capa es introducir no-linealidad a un sistema donde solamente se han realizado operaciones lineales.
- **Capa de Pooling o Subsampling:** Reduce el tamaño de las imágenes tomando cuatro pixeles cercanos y seleccionando el de mayor valor.
- Por último, solo quedaría aplanar la salida de estas capas para así conectarla a una red neuronal y realizar la clasificación. La salida de esta última red será el número de clases que estamos clasificando.

Además, según el número de capas de las que conste la red aumenta la complejidad de la red y los patrones detectados también lo hacen. A continuación, se especifica lo detectado por cada red:

- Las primeras capas extraen las características más generales, principalmente bordes y contornos.
- Las segundas capas detectan matices como texturas, colores y matices complejos
- Las siguientes capas son capaces de reconocer piezas de objetos, como pueden ser las diferentes partes de la cara de una persona u animal.
- Las últimas capas detectan patrones más específicos que pueden incluso no tener formas exactas, son patrones más complejos, en ocasiones indescriptibles por un humano. Estas capas alcanzan la detección de objetos completos, considerando varios matices.

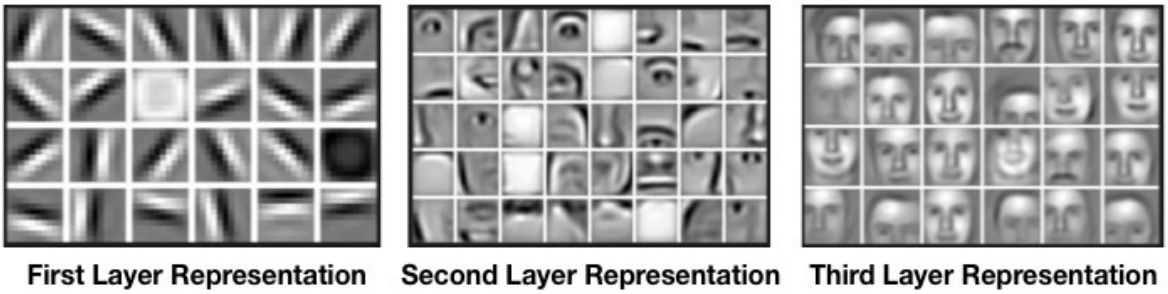


Figura 1 Patrones o partes detectadas por las distintas capas de la red.

A lo largo de los tiempos se ha trabajado con datasets y arquitecturas conocidas las cuales han sio investigadas con el fin de alcanzar objetivos en la clasificación y análisis. Algunas de estas arquitecturas conocidas son Lenet (1998), AlexNet (2012), ResNet (2015) y GoogleNet (2014), entre otras.

Los resultados obtenidos de estudios variados utilizando estas conocidas arquitecturas o muchas otras han mostrado un futuro esperanzador en el uso de redes neuronales convoluciones. Las CNN están llamadas a desempeñar un papel fundamental en la revolución digital de la industria, puesto que las cámaras son los sensores con mejor relación información-coste y ya se están utilizando de forma masiva en aplicaciones de control de calidad, en sistemas automáticos de seguridad o en los vehículos autónomos, por poner solo algunos ejemplos. Sin embargo, el principal inconveniente de las CNN es que se trata de modelos muy complejos y escasamente interpretables, lo cual los hace proclives a presentar sesgos no deseados.

Es por ello por lo que los objetivos de este proyecto se orientan con el fin de entender el aprendizaje de estas redes a partir de modelos sencillos y datasets controlados. Habitualmente se trabaja con datasets formados por imágenes complejas y no controladas. Por ello se ha utilizado un conjunto de datos sintético, generado *ex profeso* para este proyecto, compuesto por figuras geométricas (triángulos, rectángulos y círculos).

3. METODOLOGÍA

El procedimiento llevado a cabo para analizar el proceso de aprendizaje de las redes convolucionales ha consistido en varios pasos que van a ser explicados a continuación. Este proceso

ha sido aplicado a cuatro casos de estudio. Tras analizar estos casos por separado, las ideas principales han sido puestas en común con el fin de llegar a conclusiones que nos acerquen más a conocer el proceso de aprendizaje de las redes convolucionales.

Para el análisis de la sensibilidad de la red se han utilizado dos métodos paralelamente: la observación de los mapas de activación para todos los filtros de la red y su comparación, y el método de la poda de filtros hasta conocer los más representativos para la clasificación.

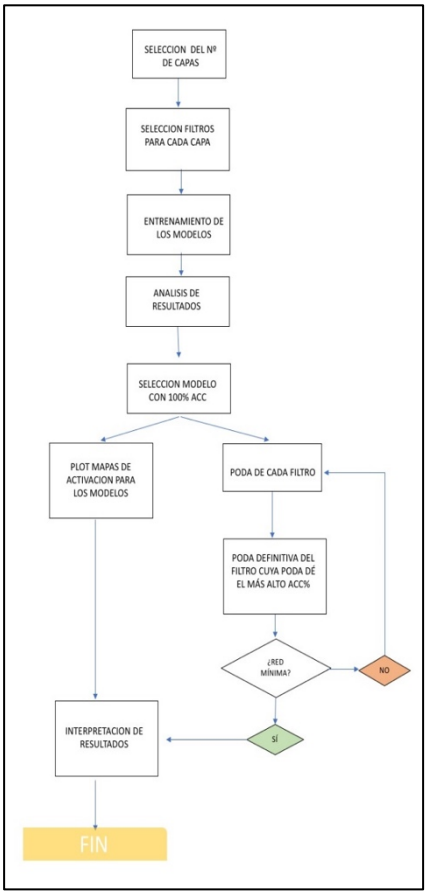
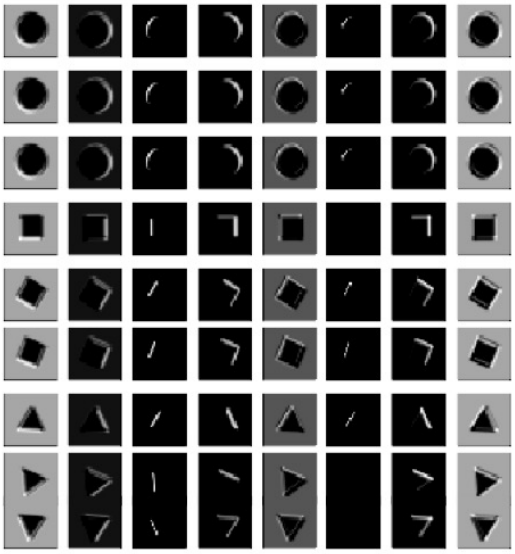


Figura 2 Procedimiento de análisis de las redes convolucionales

En la Figura 4 se puede observar el procedimiento llevado a cabo. El primer paso consiste en seleccionar el número de capas, el cual en el análisis principal se han utilizado dos capas, ya que una era insuficiente, y con dos la red ha aprendo en un gran número de casos, por lo que es suficiente. Depués se ha seleccionado el número de filtros para cada capa. Una vez generado el modelo se ha procedido al entrenamiento de éste. Esto proceso se ha realizado repetitivamente para tener varios modelos entrenados. Una vez ovbservados los datos de predicción se han seleccionado dos modelos de *accuracy* 100%, es decir, donde el aprendizaje ha resultado en una óptima clasificación. Con estos modelos seleccionados se ha llevado a cabo el análisis de la sensibilidad con dos métodos paralelamente: el análisis visual de los mapas de activación y el método basado en la poda de filtros hasta alcanzar la red mínima. Con los resultados de ambos métodos se ha llevado a cabo una comparativa con el fin de llegar a conclusiones generales del aprendizaje de la red.

En la siguiente figura se representan los dos métodos desarrollados en este proyecto. En la parte superior de la figura, hay un ejemplo de los mapas de activación obtenidos en una capa. Para cada imagen de entrada, se generará un mapa de características por filtro. En la parte inferior hay un ejemplo del método de poda, donde en cada paso el filtro que elimina la reducción de precisión más baja se poda definitivamente y este proceso se repite hasta llegar a la red más baja.

MAPAS DE ACTIVACION- METODO DE ANÁLISIS VISUAL



MÉTODO DE PODA HASTA LA RED MINIMA

	ACC% PODA 1	ACC% PODA 2	ACC% PODA 3	ACC% PODA 4
1ª CAPA- F1	0.3333	0.3333	0.3333	0.3333
1ª CAPA- F2	0.6116	0.66511	0.70566	-
2ª CAPA- F1	0.4583	0.563167	0.624667	0.624666
2ª CAPA- F2	0.6406	0.7056667	-	-
2ª CAPA- F3	0.6333	0.663	0.66566	0.66656
2ª CAPA -F4	0.6652	-	-	-

Figura 3 Mapas de activación y método de poda.

4. CASOS ESTUDIO

Primeramente, se ha comenzado trabajando con el dataset 1.2, el cual consiste en imágenes de elipses, rectángulos y triángulos de diversos tamaños. Sin embargo, se ha detectado que había imágenes que habían sido generadas de tamaño tan pequeño que eran irreconocibles incluso para el ojo humano. Es. Por ello que se perfeccionó esta dataset creando el dataset 1.3.

Tras el entrenamiento de diversos modelos con distintas combinaciones de filtros, se ha observado cómo afecta el hecho de que los valores iniciales de los filtros sean aleatorios puesto que para mismas combinaciones la red puede aprender o no aprender.

Después se han seleccionado dos casos en los que la red ha aprendido, un de ellos de 16-4 filtros respectivamente en la primera y segunda capa, y otro de 2-16 filtros. Se ha llevado a cabo el análisis de los mapas de activación, y el análisis de la poda de filtros.

El mismo procedimiento se ha llevado a cabo para el análisis del dataset 2.1, el cual consiste en imágenes del mismo tamaño de figuras geométricas (elipses, rectángulos y triángulos) que han sido rotadas un ángulo aleatorio.

Tras de comparar los resultados obtenidos en los cuatro estudios de caso, se ha observado en los mapas de características las características comunes que se pueden encontrar para ambos estudios de conjuntos de datos. El objetivo en esta parte del análisis es llegar a conclusiones que se apliquen a todo el proceso de aprendizaje de las CNN.

5. INTERPRETACIÓN DE RESULTADOS

La interpretación de resultados engloba el análisis visual de los mapas de activación y de las tablas del método de podas para cada caso. A continuación, se presentan las características principales extraídas gracias a este análisis:

- **Caso 1. Análisis de los mapas de activación del caso 16-4, entrenado con dataset 1.3**:

  En este caso se ha observado un gran número de filtros como partes activadas similares, es decir, es posible agrupar los filtros según la activación realizada. En varios casos se ha podido observar una activación mayor que en otros a pesar de ser las mismas partes de la imagen, y los resultados de la poda son coherentes con la observación. Según se va realizando la poda los filtros que se mantienen hasta los últimos pasos son los que muestran mayor activación. Además, dado que la variación en el dataset 1.3 es de tamaño, se puede observar en los mapas de activación que un gran numero de filtros activa únicamente una zona de la imagen como puede ser la izquierda, derecha, superior o inferior, y en ocasiones estos filtros son representativos en la clasificación, lo que implica que dichas partes son suficientes para la distinción de figuras.

- **Caso 2. Análisis de los mapas de activación del caso 2-16, entrenado con dataset 1.3**:

  En este análisis se ha dado el caso de un filtro que activaba todos los contornos de las figuras de forma prácticamente total, y observando el método de podas se ha podido comprobar la relevancia de dicho filtro en la red mínima. Por lo tanto, existe la posibilidad que para ciertos valores iniciales de los filtros, y tras su variación en el entrenamiento del modelo, un filtro active todas las partes de las figuras. La poda para este caso de estudio es notable, ya que la red puede ser reducida en un total de quince filtros, manteniendo una predicción correcta superior al noventa por ciento.

- **Caso 3. Análisis de los mapas de activación del caso 2-16, entrenado con dataset 2.1**:

En este caso la variación de las figuras geométricas es la rotación aleatoria. En el análisis de los mapas de activación se puede observar que, a diferencia de los requerimientos de activación del dataset 1.3 (variaciones de tamaño), en este caso es necesario la activación de dos de los lados de los rectángulos y triángulos, dado que la activación de un único lado puede llevar a confusión entre ambos. Además, cabe resaltar del método de poda de este caso, un aumento muy drástico del resultado de *accuracy* tras la poda de un filtro lo que muestra cierta interacción entre filtros.

- **Caso 4. Análisis de los mapas de activación del caso 8-4, entrenado con dataset 2.1**:

  En este caso, como ocurría en el caso 1, la red consta de filtros muy similares si se observa la activación de las figuras. La mayoría de ellos activan un mínimo de dos lados en rectángulos y triángulos, y en el caso de las elipses una parte de su curvatura. Aquellos casos en los que la activación es menos clara, o activan un único lado en rectángulos y triángulos, resultan en una menor reducción en la *accuracy* por lo que, coherentemente, son menos representativos en la clasificación de la red.

6. CONCLUSIONES Y FUTUROS DESARROLLOS

Respecto a las conclusiones alcanzadas tras el análisis de los casos estudios, y realizando un análisis global de todos los resultados obtenidos se ha considerado importante resaltar:

- **La similitud entre filtros**.
  Este factor puede llevar a la redundancia que puede suponer para la red, y por tanto reducir su eficiencia. Para evitar esta similitud sería posible hacer una comparación automática que mostrase el porcentaje de similitud entre filtros para poder considerar si algunos de ellos deben ser o no podados.

- **La interacción entre filtros**.
  Este hecho puede afectar de forma muy considerable, dado que los patrones extraídos por los filtros pueden ser contradictorios entre sí, y afectar al aprendizaje de la red de forma negativa sin mostrar dicha interacción desde el primer momento de la poda. Por tanto, este factor podría ser analizado con mayor profundidad.

- **Coherencia entre los resultados obtenidos con el método de análisis visual de los mapas de activación y el método de poda filtro a filtro.**
  En todos los casos estudiados se ha comprado la concordancia entre ambos métodos, principalmente en el caso de la primera capa, por la dificultad que supone el análisis visual de patrones de la segunda capa.

Respecto a los futuros desarrollos propuestos en este proyecto cabe resaltar:

- **Análisis de la sensibilidad mediante el método L1-norm.**
  A lo largo de este proyecto esta idea se ha comenzado a desarrollar y programar sin ser aplicada, dado que se decidió utilizar los métodos analizados, pero podría continuar siendo desarrollada para tener otra manera numérica de comparar los filtros, además del método de

podas. Este proceso consiste en la suma de valores absolutos de los pesos de las matrices kernel para comprobar la relevancia de dichos filtros en la clasificación.

- **Estudiar cómo evitar mínimos locales.**
  El problema de que la precisión caiga en mínimos locales de los que no sea capaz de salir ha sido estudiado por diversos investigadores de las CNN desde su surgimiento. En los casos de redes con un número elevado de capas, este problema se solventa gracias al gran número de parámetros que forman la red. Sin embargo, en el caso de este proyecto, este problema podría afectado dado el tamaño de la red, por lo que sería bueno realizar dicha comprobación puesto que podría ser que en casos en los que la red ha dejado de aprender podría haber continuado el aprendizaje.

- **Introducción de nuevas variaciones.**
  Otra posible vía para continuar el proyecto podría ser la introducción de nuevas variaciones como la deformación de figuras, o introducción de dos variaciones al mismo tiempo.

INTERPRETATION AND SENSITIVITY ANALYSIS OF CNN THROUGH SYNTHETIC IMAGES

Author: Martín Carballo, Gabriela.
Supervisors: Boal Martín-Larrauri, Jaime.
Sánchez Úbeda, Eugenio Francisco.
Collaborating Entity:: ICAI – Universidad Pontificia Comillas.

1. INTRODUCTION

One of the essential pillars of the industry digitalization process in which we are currently immersed, is the integration of data sources from diverse sources that improve the knowledge and operation of the systems. This type of data varies from text to images. The convolutional neural networks operate with this second type, the images. It should be remembered that human beings also use vision as our main sensor to interact with the environment.

2. STATE-OF-THE-ART

The convolutional neural networks are hierarchically organized computational networks, and they are composed of a large number of artificial neurons, which try to simulate the biological nervous system performance.

The convolutional neural networks (hereinafter, CNN, acronym for the Anglo-Saxon term Convolutional Neural Networks) have revolutionized the way of working with images, even surpassing the human being in tasks that are as natural to us as identifying and classifying objects by categories. The convolutional networks differ from other types of neural networks by carrying out the convolution. Convolution is understood as the filtering of an image through a mask (kernel). The convolution consists of taking sub-matrixes from the pixel matrix of the input image, and operating using the scalar product with the filtering matrix called kernel. Each output pixel will be the linear combination of the input pixels.

The main layers that form a convolutional neural network are:

- **Convolutional layer:** It carries out the convolution with the filtering matrices (kernel)
- **ReLU activation layer:** The purpose of this layer is to introduce non-linearity to a system where only linear operations have been performed.
- **Pooling or Subsampling layer:** Reduce the size of the images by taking four nearby pixels and selecting the one with the highest value.
- Finally, it would only be to flatten the output of these layers to connect it to a neural network and perform the classification. The output of this last network will be the number of classes we are classifying.

In addition, depending on the number of layers of the network, the complexity of the network increases and the patterns detected do so. Next, what each network detects is specified:

• The first layers extract the most general characteristics, specially edges and contours.
• The second layers detect nuances such as textures, colors and complex nuances
• The following layers are able to recognize pieces of objects, such as the different parts of the face of a person or animal.
• The last layers detect more specific patterns that may not even have exact shapes, are more complex patterns, sometimes indescribable by a human. These layers reach the detection of complete objects, considering several nuances.

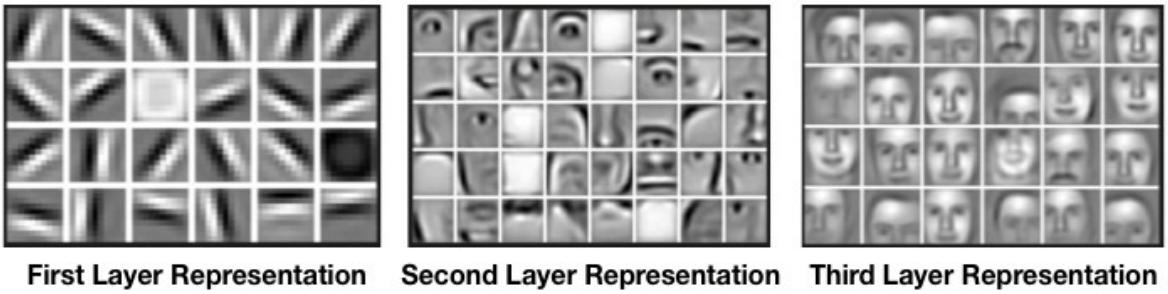


Figure 1 Patterns and parts extracted by different layers of the convolutional neural network

Over the years, we have worked with known datasets and architectures which have been investigated in order to reach goals in classification and analysis. Some of these known architectures are Lenet (1998), AlexNet (2012), ResNet (2015) y GoogleNet (2014), among others.

The results obtained by many studies developed from these architectures and many others have shown a bright future of CNN developments. Furthermore, they are called to play an important role in the industry digital revolution, since cameras are the sensors with the best information-cost relationship and are already being used massively in quality control applications, in automatic security systems or in autonomous vehicles, to give just a few examples. However, the main drawback of CNNs is that they are very complex and poorly interpretable models, which makes them prone to unwanted biases.

That is why the objectives of this project are oriented in order to understand the learning process of these networks from simple models and controlled datasets. It is common to work with datasets consisting of complex and uncontrolled images. Therefore, a synthetic data set has been used, generated ex profeso for this project, consisting of geometric figures (triangles, rectangles and circles).

3. METHODOLOGY

The process of analyzing the convolutional neural networks learning process has consisted of some steps that are going to be explained below. This process has been applied in four case studies. After analyzing separately these four cases, the main ideas extracted from the results have been considered all together with the aim of reaching conclusions that bring us closer to know more about the learning process of convolutional neural networks.

For the analysis of the sensitivity of the network two methods have been used in parallel: the observation of the activation maps for all the filters of the network and their comparison, and the method of pruning filters until the most representative ones for classification are known.

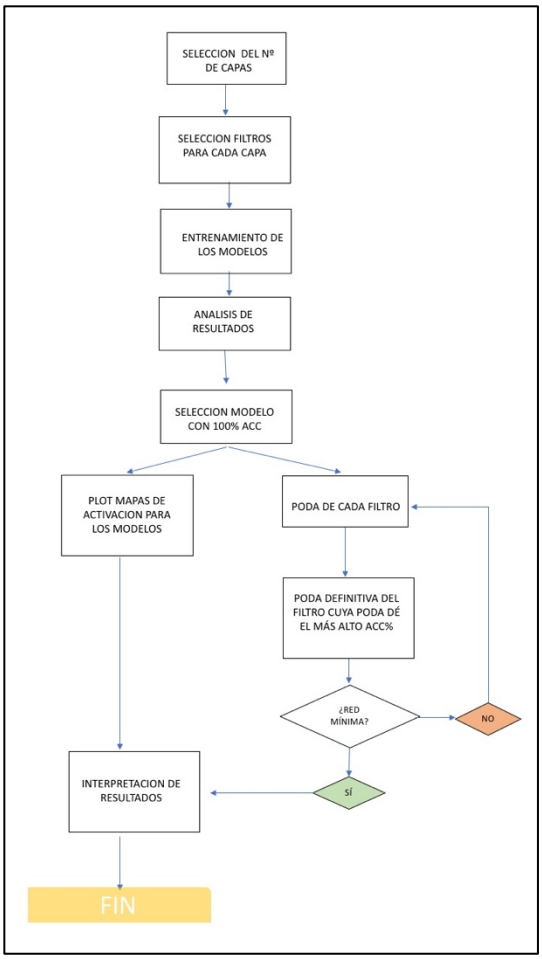
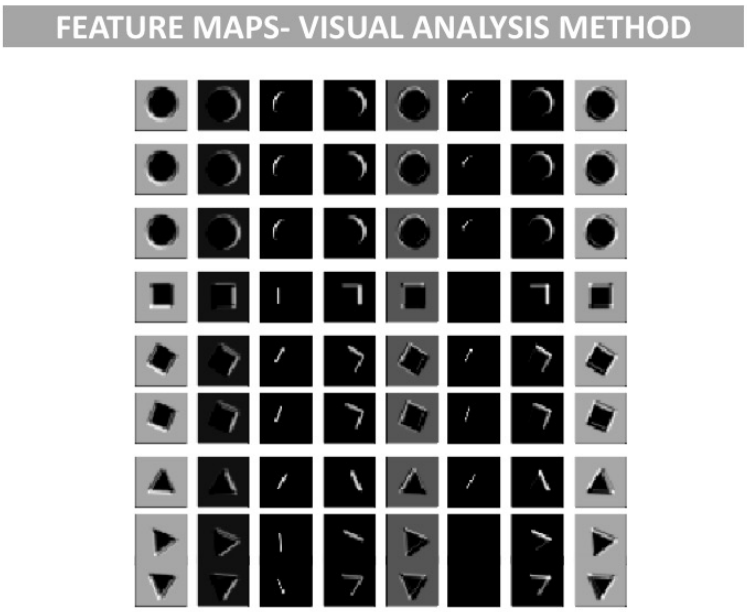


Figure 2 Convolutional Neural Network analysis process.

Figure 2 shows the procedure carried out. The first step consists of selecting the number of layers, which in the main analysis have used two layers, since one was insufficient, and with two the network has learned in a large number of cases, so it is sufficient. Then the number of filters for each layer has been selected. Once the model has been generated, it has been trained. This process has been done repeatedly to have several trained models. Once the prediction data has been observed, two models of 100% accuracy have been selected, that is, where learning has resulted in an optimal classification. With these selected models, sensitivity analysis was carried out with two methods in parallel: the visual analysis of the activation maps and the method based on the pruning of filters until reaching the minimum network. With the results of both methods a comparison has been carried out in order to reach general conclusions of the network learning .

In the following figure it is represented the two methods developed in this project. At the top of the figure, there is an example of the activation maps obtained in one layer. For each input image, there are going to be generated one feature map per filter. At the bottom there is an example of the pruning method, where in each step the filter which prune results in the lowest accuracy reduction is definately pruned and this process is repeated until reaching the lowest net.



	ACC% PRUNE 1	ACC% PRUNE 2	ACC% PRUNE 3	ACC% PRUNE 4
1st LAYER- F1	0.3333	0.3333	0.3333	0.3333
1st LAYER - F2	0.6116	0.66511	0.70566	-
2nd LAYER - F1	0.4583	0.563167	0.624667	0.624666
2nd LAYER - F2	0.6406	0.7056667	-	-
2nd LAYER - F3	0.6333	0.663	0.66566	0.66656
2nd LAYER -F4	0.6652	-	-	-

Figure 3 Feature maps and pruning methods.

4. CASE STUDIES

Firstly, it has begun working with dataset 1.2, which consists of images of ellipses, rectangles and triangles of various sizes. However, it has been detected that there were images that had been generated so small that they were unrecognizable even to the human eye. That is why this dataset was perfected by creating dataset 1.3.

After training different models with different combinations of filters, it has been observed how it affects the fact that the initial values of the filters are random since for the same combinations the network can learn or not.

Then two cases have been selected in which the network has learned, one of them with 16-4 filters respectively in the first and second layers, and another of 2-16 filters. The activation map analysis and filter pruning analysis have been carried out for these two cases.

The same procedure has been carried out for the analysis of dataset 2.1, which consists of images of the same size of geometric figures (ellipses, rectangles and triangles) that have been rotated by a random angle.

Then two cases have been selected in which the network has learned, one of them with 16-16 filters respectively in the first and second layers, and another of 8-4 filters. The activation map analysis and filter pruning analysis have been carried out for these two cases.

After comparing the results obtained the four case studies, it has been observed in the feature maps the common characteristics that could be found for both datasets studies. The goal in this part of the analysis is to reach conclusions that apply for all learning process of CNN.

5. RESULTS INTERPRETATION

The interpretation of results encompasses the visual analysis of the activation maps and the pruning method tables for each case. Below are the main features extracted thanks to this analysis:

• Case 1. Analysis of the activation maps of case 16-4, trained with dataset 1.3:

In this case, a large number of filters have been observed as similar activated parts, that is, it is possible to group the filters according to the activation performed. In several cases, it has been possible to observe a greater activation than in others despite being the same parts of the image, and the pruning results are consistent with the observation. As pruning is done, the filters that are maintained until the last steps are the ones that show the greatest activation. In addition, since the variation in dataset 1.3 is of size, the activation maps show that a large number of filters activate only one area of the image such as the left, right, upper or lower sides, and sometimes these filters are representative in the classification, which implies that these parts are sufficient for the distinction of figures.

• Case 2. Analysis of the activation maps of case 2-16, trained with dataset 1.3:

In this analysis, the case of a filter that activated all the contours of the figures has been practically completed, and observing the method of It was possible to verify the relevance of this filter in the minimum network. Therefore, there is a possibility that for certain initial values, and the training of the model, a filter activates all parts of the figures. Pruning for this case study is remarkable, since the network can be reduced to fifteen filters, maintaining a correct prediction greater than ninety percent.

• Case 3. Analysis of the activation maps of case 2-16, trained with dataset 2.1:

In this case the variation of the geometric figures is random rotation. In the analysis of the activation maps it can be observed that, unlike the activation requirements of the dataset 1.3 (size variations), in this case it is necessary to activate two sides of the rectangles and triangles, since the single side activation can lead to confusion between the two of them. In addition, it is worth highlighting in the pruning method a very drastic increase in the accuracy result after the pruning of a filter that can mean some interaction between filters.

• Case 4. Analysis of the activation maps of case 8-4, trained with dataset 2.1:

In this case, as in case 1, the network consists of very similar filters if the activation of the figures is observed. Most of them activate a minimum of two sides in rectangles and triangles, and in the case of ellipses a part of their curvature. Those cases in which the activation is less clear, or they activate a single side in rectangles and triangles, result in a smaller reduction in accuracy, which is why they are coherently less representative in the classification of the network.

6. CONCLUSIONS AND FUTURE WORK

Regarding the conclusions reached after the analysis of the case studies, and carrying out a global analysis of all the results obtained, it is important highlighting:

• **The similarity between filters.**
This factor can lead to the redundancy that can be assumed for the network, and therefore reduce its efficiency. To avoid this similarity it would be possible to make an automatic comparison programmed with Python that showed the percentage of similarity between filters to be able to consider whether some of them should be pruned or not.

• **The interaction between filters.**
This fact can affect in a very considerable way, given that the patterns extracted by the filters can be contradictory to each other, and affect the learning of the network in a negative way without showing such interaction from the first moment of pruning. Therefore, this factor could be analyzed in greater depth.

• **Consistency between results obtained with visual analysis method of activation maps and pruning method filter to filter.**
In all the cases studied, the concordance between the two methods has been purchased, mainly in the case of the first layer, due to the difficulty of visual analysis of patterns of the second layer.

Regarding the future developments proposed in this project, it is worth highlighting:

• **Sensitivity analysis using the L1-norm method.**
Throughout this project this idea has begun to be developed and programmed without being applied, since it was decided to use the analyzed methods, but it could continue to be developed to have another numerical way to compare the filters, in addition to the pruning method. This process consists of the sum of absolute values of the weights of the kernel matrices to verify the relevance of these filters in the classification.

• **Study how to avoid local minimums.**
The problem that accuracy falls to local minimums that are not able to leave has been studied by various CNN researchers since its emergence. In the case of networks with a high number of layers, this problem is solved thanks to the large number of parameters that make up the network. However, in the case of this project, this problem could be affected given the size of the network, so it would be good to perform such a check since it could be that in cases where the network has stopped learning it could have continued learning .

• **Introduction of new variations.**

Another possible way to continue the project could be the introduction of new variations such as deformation of figures, or introduction of two variations at the same time.

Tabla de Contenidos

INDICE DE FIGURAS

INDICE DE TABLAS

CAPITULO 1

INTRODUCCION
ESTADO DEL ARTE
MOTIVACIÓN
OBJETIVOS
RECURSOS

Introducción

Uno de los pilares fundamentales del proceso de digitalización de la industria en el que nos encontramos actualmente inmersos, es la integración de fuentes de datos de procedencias diversas que permitan mejorar el conocimiento y la operación de los sistemas. Este tipo de datos son generalmente no estructurados (texto o imágenes) y su procesamiento para extraer información que aporte valor al negocio requiere de nuevas técnicas. Simultáneamente, la reducción del coste de algunas tecnologías de sensorización, junto con el auge del Internet de las cosas ha favorecido la instalación de multitud de nuevos sistemas de percepción para monitorizar variables que antes no eran tenidas en cuenta. En concreto, las cámaras han proliferado en multitud de aplicaciones dado su bajo coste con relación a la información que potencialmente se puede extraer de ellas. Conviene recordar que los seres humanos también utilizamos la visión como nuestro sensor principal para interactuar con el entorno.

Inspiradas en un experimento realizado por Hubel y Wiesel en 1959 para entender cómo funcionaba el córtex visual de los gatos [1], no ha sido hasta principios de la esta década cuando las redes neuronales convolucionales (o CNN, del inglés *Convolutional Neural Networks*) han revolucionado la forma de trabajar con imágenes, llegando incluso a superar al ser humano en tareas que nos resultan tan naturales como identificar y clasificar objetos por categorías.

Estado del arte

Redes neuronales convolucionales (CNN)

Para conocer las CNN nos remontamos a la aparición de las redes neuronales artificiales, y el objetivo de su creación. Las redes neuronales pueden ser definidas como redes computacionales organizadas jerárquicamente, y compuestas por un gran número de parámetros, las cuales tratan de simular el funcionamiento del sistema nervioso biológico. Una de las características más valoradas de este tipo de redes es el aprendizaje adaptativo, es decir, la capacidad de aprendizaje a partir de un previo entrenamiento [7]. La estructura general de una red neuronal viene dada por una capa de entrada, capas ocultas, y una capa de salida como se observa en la **Figura** 1. Estas redes están compuestas por cientos de neuronas (que tienen entradas, utilizan pesos y generan salidas).

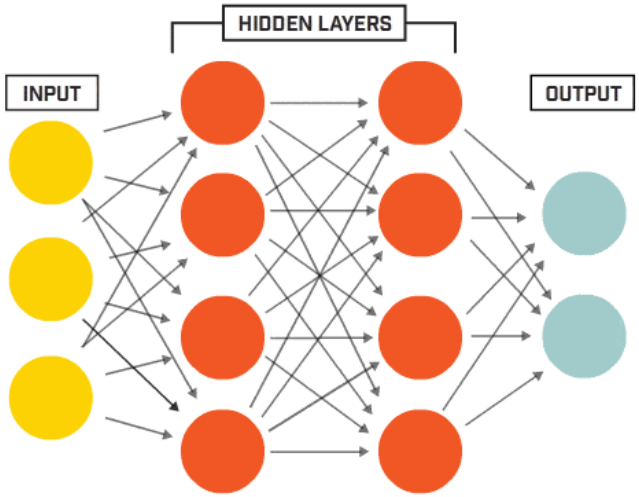


Figura 1 Red neuronal interconectada [21]

Las redes convolucionales son un tipo de redes neuronales que se diferencian por realizar la aplicación con entradas multidimensionales. Se utilizan para el análisis de imágenes principalmente, y en vez de tomar todos los valores de entrada de una imagen, aplican matrices multidimensionales que operan las entradas (filtros) y extrae carracterísticas de la imagen.

Respecto a la topología de las redes convolucionales son redes multicapa con múltiples filtros convolucionales, que realizan la aplicación de funciones con matrices bidimensionales

que es lo que conocemos como convolución, y posteriormente se aplica una función de activación para realizar un mapeo causal no lineal. Y la capa de salida consta de neuronas sencillas que realizan la clasificación.

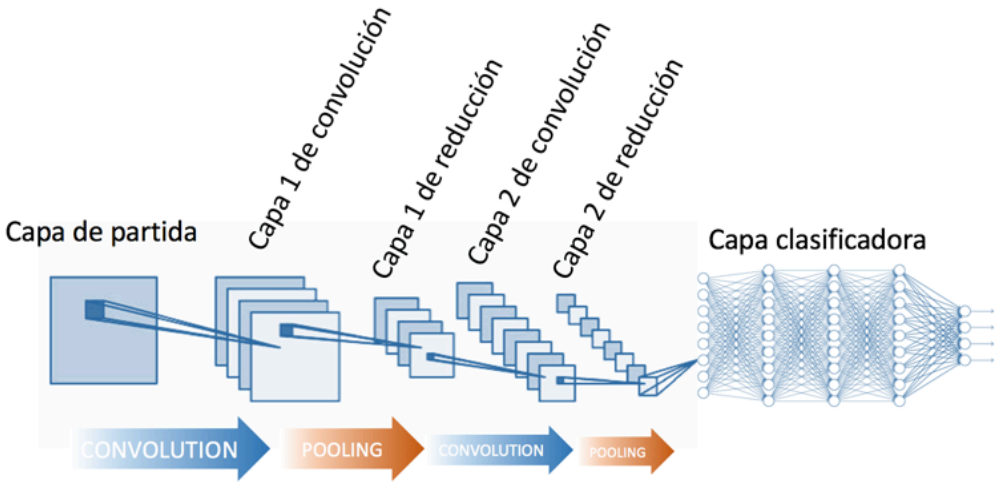


Figura 2 Proceso llevado a cabo por las CNN [25]

El proceso realizado por las redes convolucionales para la detección de características de una imagen y posterior clasificación viene representado en la **Figura** 2. Constaría primero del establecimiento de patrones, seguido de capas que hacen combinaciones en formas más simples. Después en las capas más profundas se trata el detalle de la imagen como posición de objetos, iluminación, escalas, etc. Por ultimo, en las capas últimas se intenta hacer coincidir dicha imagen obtenida con los patrones para obtener una predicción final.

- ## Capas

Como se puede observar en la **Figura** 1, las redes convolucionales pasan por tres tipos de capas principales. El esquema general del proceso se podría resumir en que una imagen de entrada pasa por la capa de convolución, después por capa de *pooling* o de reducción, y, por último, las capas de clasificación. Como se indica en la **Figura** 1 se suele repetir para que la propia red aprenda lo que anteriormente no ha aprendido siendo la entrada de estas capas la salida de las mismas capas usadas anteriormente. Puede repetirse tantas veces como se quiera, aunque se deberá buscar el numero óptimo de repeticiones para no caer en el error de que la red sobre-aprenda y los resultados de la clasificación sean erróneos.

- **Capa convolucional:** Es la capa que diferencia este tipo de redes de las usadas anteriormente. Se entiende por convolución al filtrado de una imagen mediante una

máscara (*kernel)*. La convolución consiste en tomar submatrices de la matriz de píxeles de la imagen de entrada, y operar mediante producto escalar con la matriz de filtrado llamada *kernel.* Este proceso esta representado de forma visual en la **Figura** 3. En la convolución, cada píxel de salida será la combinación lineal de los pixeles de entrada. En la convolución se toman se operan matemáticamente haciendo el producto escalar a grupos cercanos de píxeles. Tras la convolución se obtienen varias matrices de salida dado que no aplicamos solo un *kernel* (filtrado), obtendremos tantas matrices de salida como kernels aplicados. Este conjunto se llama feature mapping. Estas matrices de salida guardan lo que serian ciertas características de la imagen que permitirán diferenciar posteriormente los objetos. Esta matriz de salida será la entrada a las capas de neuronas ocultas. Además, si la imagen fuese en color, el kernel no sería una matriz 3x3 sino que sería 3 filtros kernel, por lo que sería 3x3x3, que posteriormente se sumarán. En definitiva, la salida de la convolución es como obtener muchas imágenes nuevas filtrada que resaltan características de la original. [9]

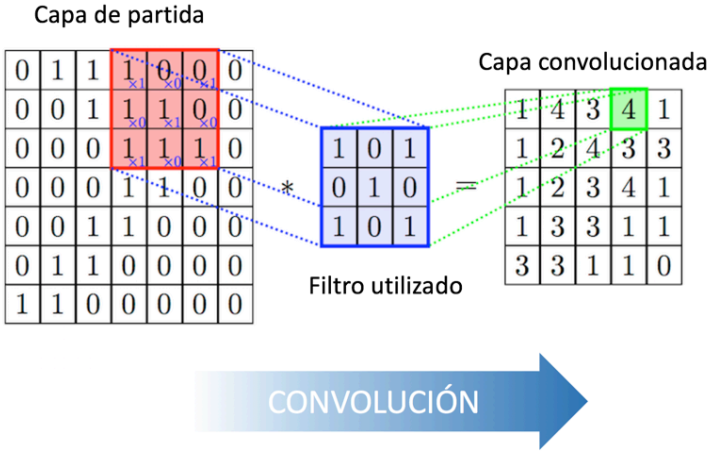


Figura 3 Aplicación de la convolución a una imagen de entrada [25]

- **Capa de activación (ReLU):** Las siglas de "ReLU" corresponden al Rectified Linear Unit y consiste en f(x) = max(0, x). Después de cada capa de convolución se aplica esta capa. El fin de esta capa es introducir no-linealidad a un sistema donde solo se ha realizado operaciones lineales. En las redes anteriores a las convolucionales, se solían introducir no linealidades aplicado funciones como tanh o sigmoides, pero el uso de la capa de activación ReLU es más rápido y eficiente. El problema encontrado con las funciones tanh o sigmoides es que saturan. Es posible encontrar más información en la siguiente referencia. [35]

- **Pooling o Subsampling:** Después de pasar la imagen de entrada por la capa convolucional, como se ha visto se obtiene un número muy elevado de imágenes partir de la de entrada (podemos imaginar cuando nuestra entrada sea una gran cantidad de imágenes, de diversos tamaños y de color, por ejemplo) [9]. Por ejemplo, de una imagen 28x28, podremos obtener 32 mapas de características, lo que corresponde a 25.088 parámetros. Por ello debemos filtrar dichas imágenes de los mapas de características para conservar las características más importantes resaltadas tras la convolución. Este proceso es conocido como *subsampling*, y el más usado entre los procesos de este tipo es *el Max-Pooling*. *Max-Pooling* recorre la imagen de salida (*kernel*) tomando sub-matrices 2x2 (4 pixeles) y coge los valores más altos. Por lo tanto, la matriz obtenida tras el *maxpooling* sería una matriz reducida a la mitad, como se puede observar en la **Figura** 4. Las capas *pooling* o *subsampling* hace una reducción de las dimensiones, pero al tomar los valores más altos, no pierde en profundidad.

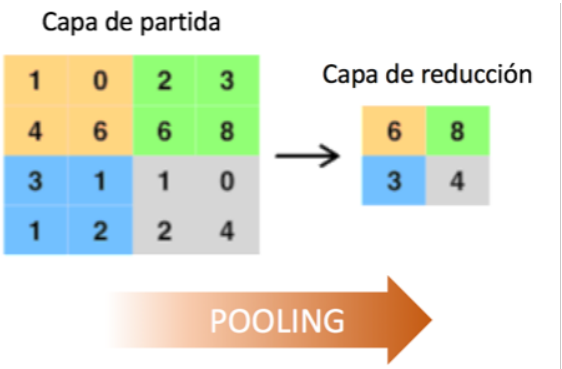


Figura 4 Reducción a través de aplicación *pooling [25]*

Las redes neuronales ganan complejidad en función del numero de capas ocultas de las que este compuesta. Para ello es importante conocer el concepto de pesos. Las capas ocultas lo que hacen es crear interconexiones entre variables, y generando muchas relaciones con una porcentaje o probabilidad de importancia en el resultado final, y esto es lo que conocemos como pesos. Los pesos inicialmente toman valores aleatorios, y según la red va a aprendiendo, estos pesos toman valores más adecuados.

Cuantas más capas ocultas formen la red convolucional, más aprende nuestra red, aunque ha que tener en cuenta que no se puede añadir infinitas capas. Existe un límite de convoluciones posibles, dado por el límite de reducción *pooling* que podemos realizar a la matriz [9].

- Por último, tras pasar la imagen por todas las convolucionales de la red ya solo quedaría conectar la salida a una red neuronal tradicional, para "aplanar" la salida, y así poder conectar dicha salida a una red neuronal sencilla y realizar la clasificación. La salida de esta última red será el número de clases que estamos clasificando. En estas redes tradicionales es importante conocer ciertos conceptos:

    1. **One-hot-encoding:** Las salidas del entrenamiento tienen este forma, el cual indica las clases como valores de 0s y 1s. Por tanto, si las clases de la red fueran perros y gatos, las salidas serían: [0, 1] gatos o [1, 0] perros.
    2. **Función Softmax:** Añade a la salida la probabilidad de pertenecer a cada clase, por tanto, un salida sería [0.63, 0.37] siendo 63% y 37% las probabilidades de ser gato o perro respectivamente.

- ## Arquitecturas

Como se ha visto previamente, una red convolucional consta de varias capas ocultas, que es lo que vamos a llamar niveles. El numero de estos niveles y como son es lo que conocemos con arquitectura de la red. Desde el comienzo de las redes convolucionales en 1989 se han desarrollado una gran variedad de arquitecturas útiles para distintos objetivos. Vamos a hacer un breve resumen de las arquitecturas más conocidas y los problemas de clasificación que ofrecen cada una de ellas.

- **LeNet:** Arquitectura desarrollada por LeCun, et al. (creador de la primera CNN) en 1998. Es una red pionera de 7 niveles que es aplicada para la detección de números y escritura a mano o máquina. Para procesar una mayor resolución de imágenes requeriría más capas, por lo que esta restringida la posibilidad de utilizar diversos recursos computacionales. Consecuentemente, como se ha mencionado anteriormente, la mejora de las GPUs también ha sido un factor muy importante en el desarrollo de las redes neuronales [8].

Layer		Feature Map	Size	Kernel Size	Stride	Activation
Input	Image	1	32x32	-	-	-
1	Convolution	6	28x28	5x5	1	tanh
2	Average Pooling	6	14x14	2x2	2	tanh
3	Convolution	16	10x10	5x5	1	tanh
4	Average Pooling	16	5x5	2x2	2	tanh
5	Convolution	120	1x1	5x5	1	tanh
6	FC	-	84	-	-	tanh
Output	FC	-	10	-	-	softmax

Figura 5 Arquitectura LeNet [26]

- **AlexNet**: En 2012, Alex Krizhevsky, Geoffrey Hinton, e Ilya Sutskever crean una arquitectura similar a la ofrecida por LeCun (LeNet) pero que reducía la tasa de error un 11%. La forma de alcanzar una red simple como LeCun, reduciendo la tasa de error es que añadieron filtros a cada capa además de apilar las capas convolucionales. Después de cada capa convolucional se realizaba la activación ReLU.
  Consta de cinco capas convolucionales y tres capas sencillas [3].

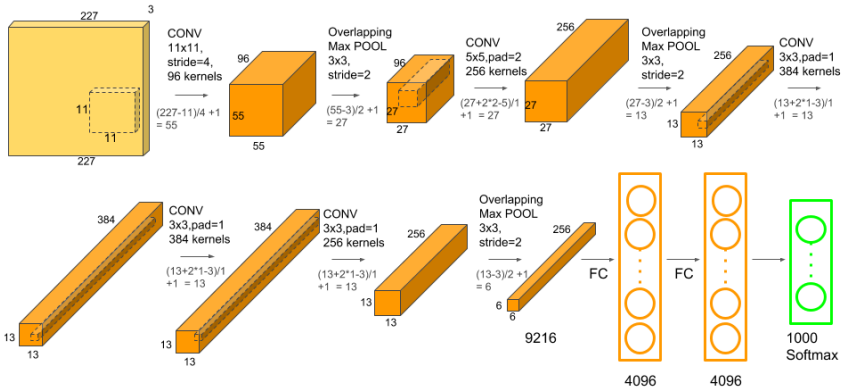


Figura 6 Arquitectura AlexNet [27]

Las capas *Overlapping Max Pooling* que se encuentran tras cada capa convolucional se utilizan para reducir los tensores manteniendo su profundidad.

- **ZFNet:** Esta arquitectura desarrollada en 2013 alcanzo una tasa de error en el top-5 del 14,8%, lo que es prácticamente la mitad de la tasa de error obtenida en los inicios de las redes convolucionales. Una capa deconvolucional se añade a cada capa para obtener siempre un camino de vuelta al numero de pixeles original [13].

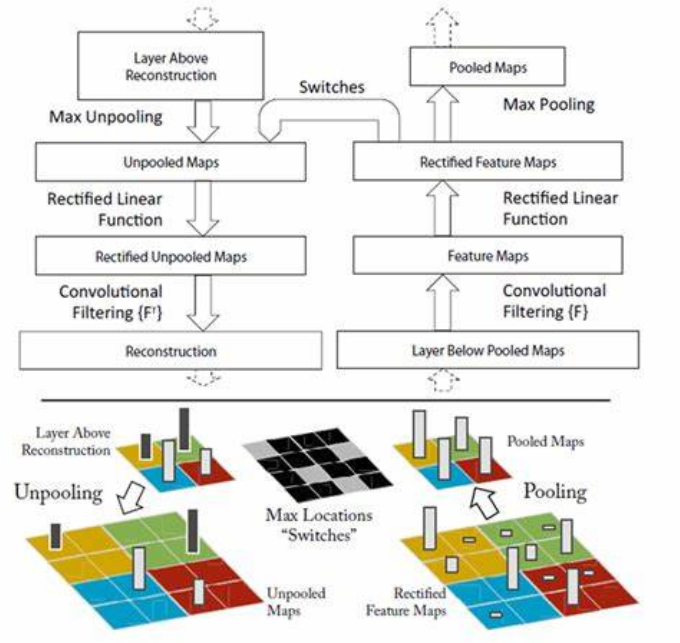


Figura 7 Arquitectura ZFNet [28]

- **GoogLeNet:** Arquitectura desarrollada en 2014 por Google. Esta arquitectura reduce el top-5 ratio de error al 6.67%. Con entrenamiento humano, se consiguió alcanzar un top5 ratio de error del 5.1%. Esta arquitectura parte de la base de LeNet, añadiendo a esta un modulo de incepción, el cual se basa en pequeñas convoluciones en paralelo para reducir el número de parámetros. Esta arquitectura consta de 22 capas profundas, pero reduce el número de parámetros a 4 millones, en comparación con los 60 millones con los que trabaja la arquitectura AlexNet. Es característico de ésta los procesos realizados en paralelo.

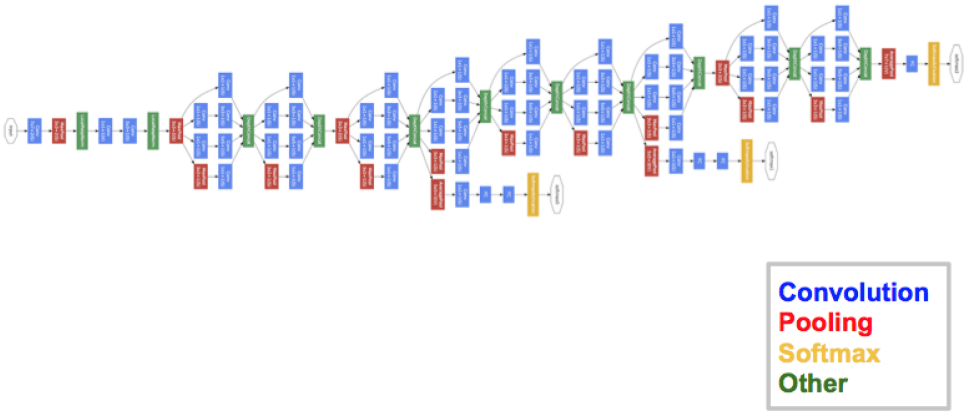


Figura 8 Arquitectura GoogleNet [29]

- **VGG:** Desarrollado por Simonyan and Zisserman en 2014, esta arquitectura consta de 16 capas convolucionales, y es una arquitectura basada en la apilación de forma uniforme. Es característico de esta arquitectura el uso de convoluciones comunes 3x3, pero constando de muchos filtros. Es una de las más utilizadas para la extracción de características de imágenes. Negativamente, consta de cientos de millones de parámetros con los que trabaja lo que resulta dificultoso y menos eficaz.

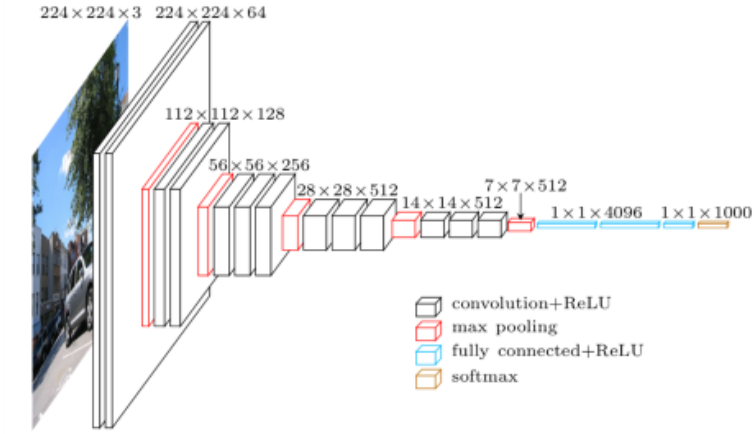


Figura 9 Arquitectura VGG. [30]

- **ResNet:** Desarrollada en 2015 también conocida como red neuronal de residuos, esta arquitectura fue desarrollada por Kaiming He. Esta arquitectura es característica por introducir el concepto de conexiones de salida, similares a las aplicadas en las redes RNNs. Lo que hace esta arquitectura es crear micro arquitecturas, creando un modulo residual que va actualizando su valor Gracias a estas conexiones es posible entrenar un numero elevado de capas (152) conservando una baja complejidad, lo que resulta en un error del 3.57% en el top 5, es decir, que solo en el 3,57% de las ocasiones, la red no predice la categoría de la imagen correctamente entre sus cinco opciones más probables. [14]

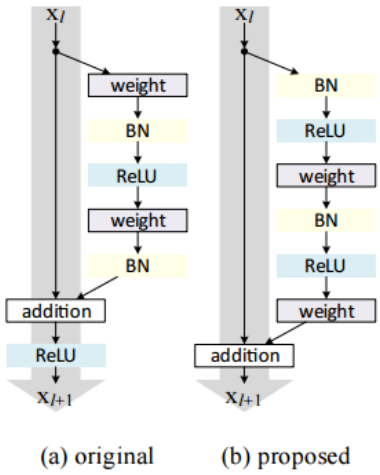


Figura 10 Modulo residual de las arquitecturas ResNet [31]

Interpretación de CNN

Para una óptima interpretación de los resultados presentados por una red convolucional lo primero es comprender el funcionamiento de los filtrados en las capas convolucionales. Las características extraídas por la red convolucional han sido clasificadas en seis tipos: patrones, objetos, texturas, colores, piezas y escenas [15]. Los matices de objetos y piezas son analizados por las capas convolucionales iniciales, los cuales miden patrones de piezas con contornos específicos, mientras que las capas convolucionales bajas llevan a cabo la detección de patrones de texturas sin formas o contornos específicos, es decir, características más complejas. Estos patrones detectados por la red pueden ser incluso no visibles o diferenciables con el ojo humano.

En resumen, según la capa de la arquitectura de la red, se detectarán:

- Las primeras capas extraen las características más generales, principalmente bordes y contornos.
- Las segundas capas detectan matices como texturas, colores y matices complejos
- Las siguientes capas son capaces de reconocer piezas de objetos, como pueden ser las diferentes partes de la cara de una persona u animal.
- Las últimas capas detectan patrones más específicos que pueden incluso no tener formas exactas, son patrones más complejos, en ocasiones indescriptibles por un humano. Estas capas alcanzan la detección de objetos completos, considerando varios matices.

Como se puede observar en la **Figura** 11 los mapas de características de las primeras capas son representaciones simples relativas a la detección de bordes. Las divisiones

verticales, horizontales y diagonales que se aprecian en la primera capa de filtrado indican una gran convergencia en la red En las segundas capas, los filtrados detectan objetos y partes en las imágenes. En la primera capa de filtros, se pueden analizar los mapas de activación por las líneas que podían aparecer . La detección de bordes y contornos es clara. Según aumentan las capas, la lectura de los mapas de características son más complicados de analizar a simple vista, dado la mezcla de patrones de matices.

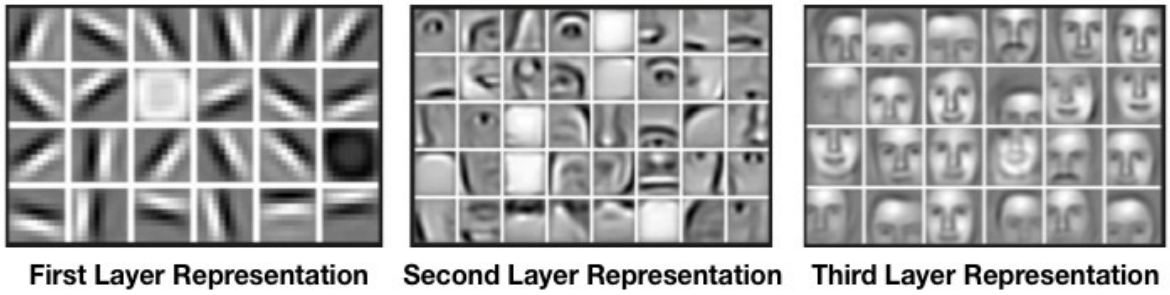


Figura 11 Mapa de características en las sucesivas capas de la CNN [32]

Las características de extraídas de una imagen se pueden resumir en dos grupos, uno las partes y objetos, y por otro lado los patrones de texturas, grupo que engloba texturas, escenas, colores materiales. En las capas convolucionales bajas se describen los patrones de texturas, mientras que los filtros en las capas convolucionales altas son los que representan patrones de partes, los cuales tienen patrones de contornos para detectar partes específicas o regiones.

Con el fin de detectar diferentes patrones específicos en las imágenes, surgen las redes interpretables, redes en las cuales se llevan a cabo métodos para tener cierto control en las activaciones y patrones de salida de la red a partir de los valores de entrada. En las redes tradicionales, los filtros se activaban para diferentes partes, sin embargo, en las CNN interpretables, los filtros se activan para partes específicas, como puede ser una parte del cuerpo. Es decir, las redes interpretables son capaces de detectar una misma parte u objeto y hacer una clasificación de ello. Sin embargo, las redes tradicionales solían detectar en un mismo patrón varias partes pero que compartían una determinada característica.

Como se comentaba anteriormente, uno de los grandes problemas de las redes convolucionales es el desconocimiento de cómo aprende la red, lo que hace que, a pesar de poder alcanzar un aprendizaje de nivel alto, como puede ser la detección facial, las redes son muy complejas, y es complicado simplificarlas por el hecho de desconocer como aprende la

red. Sin embargo, han sido muchos los investigadores que han buscado metodologías para entender la interpretabilidad de la red y optimizarla lo máximo posible.

Un ejemplo de ello es la metodología propuesta por Quanshi Zhang et al. [15] para mejorar la interpretabilidad de la red consiste en añadir tras las capas convolucionales altas, unas capas de pérdidas, que provoquen que cada filtro lleve a cabo la representación de una parte, activando el filtro solo para una categoría.

La mejor forma de analizar la interpretabilidad de la red es inspeccionar los filtros aplicados en las capas profundas para conocer los patrones que detecta dicho filtro. Muchos han sido los métodos propuestos para analizar la interpretabilidad de la red. Bau et al. propuso el análisis de la interpretabilidad de cada filtro mediante el método de gradientes, lo que supone la visualización de las activaciones más fuertes producidas por cada filtro, mediante los máximos gradientes. Dosovitskiy et al.[36] propone invertir el proceso y a partir de los mapas de características obtenidos de la red, reconstruir la imagen inicial para ver la fiabilidad de la extracción de características. Ambos métodos, son utilizados actualmente para el análisis de redes.

Otros investigadores de las redes convolucionales han trabajado en encontrar métodos para la extracción de las características de nivel medio como escenas o partes del objeto a través de ciertos filtros de manera supervisada.

Como se puede observar muchos han sido los métodos propuestos para el análisis de la interpretabilidad y la extracción de unidades o medidas para su utilización. Entre los métodos más generales utilizados resaltan el LME y el SHAP, los cuales extraen unidades que la red utilizada para los resultados de predicción.

En [13], Matthew D. Zeiler y Rob Fergus proponen un método para la visualización e interpretación de los mapas de características de cada capa interna y revela que factores de las entradas estimulan los mapas de características de salida. Para el análisis de la sensibilidad de clasificación, propone bloquear porciones de la imagen de entrada para revelar que partes son importantes en la imagen para su clasificación.

La propuesta de Erhan et al [34] para encontrar los óptimos estímulos para cada unidad, consiste en generar el descenso de gradiente en el espacio imagen para maximizar la activación de dicha unidad. Para llevar a cabo este proceso es necesaria una buena inicialización de parámetros.

Aparte de obtener los parámetros óptimos, es necesario para el correcto análisis conocer la invariancia para lo que Le Cun propone el análisis de la matriz Hessiana y estudia como ejecuta numéricamente las unidades a partir de la respuesta optima y teniendo en cuenta las invariancias. Dado que en las capas más altas la complejidad aumenta considerablemente, proponen una forma visual de realizar el análisis de estas.

Una forma de llevar a cabo el análisis de interpretabilidad y sensibilidad [13] es entrenar un conjunto de datos en la red, y para poder ver las activaciones en la imagen, una capa deconvolucional, es decir, que lleva a cabo el proceso inverso a la capa convolucional tratando de reconstruir la imagen de entrada. De esta forma podemos observar que características de la imagen están activadas.

Respecto a la forma de conocer la sensibilidad de la interpretabilidad de la red, [13] proponen bloquear aquellas partes de la imagen que contengan el objeto a identificar, para ver si los valores de clasificación correcta disminuyen considerablemente como correspondería.

La forma visual que permite el análisis de la interpretabilidad explicado anteriormente, son los mapas de calor. Estas representaciones muestran en una escala de colores las zonas más activadas para cada filtro. Además, las matrices del *kernel* nos permiten ver los pesos que se aplican en dicho filtro para una correcta posterior inicialización. Los mapas de activación son las representaciones de las partes activadas generadas a partir de redes deconvolucionales, explicado previamente.

Motivación

Los resultados que se consiguen en problemas de procesamiento de imágenes con redes neuronales convolucionales son, sin ningún género de dudas, superiores a los que se pueden obtener con técnicas tradicionales de visión artificial, e incluso en algunas circunstancias mejoran las capacidades de los seres humanos. Así pues, las CNN están llamadas a desempeñar un papel fundamental en la revolución digital de la industria, puesto que las cámaras son los sensores con mejor relación información-coste y ya se están utilizando de forma masiva en aplicaciones de control de calidad, en sistemas automáticos de seguridad o en los vehículos autónomos, por poner solo algunos ejemplos. Sin embargo, el principal inconveniente de las CNN es que se trata de modelos muy complejos y escasamente interpretables, lo cual los hace proclives a presentar sesgos no deseados.

Objetivos

El objetivo de este proyecto es determinar cómo aprende una red neuronal convolucional y estudiar qué tipo de filtros y arquitecturas son los más adecuados para entrenar una red desde cero. Para realizar este tipo de análisis, generalmente se utilizan imágenes reales, pero al no existir control acerca de los objetos que aparecen en el fondo, se introduce mucho ruido que complica la interpretación de los parámetros de la red. Por ese motivo, se utilizará un conjunto de datos sintético, generado *ex profeso* para este proyecto y ya disponible, compuesto por figuras geométricas (triángulos, rectángulos y círculos).

Recursos

Se ha optado por Python como lenguaje de programación, dada su facilidad para desarrollar aplicaciones de *machine learning* usando las dos librerías más populares para redes neuronales convolucionales, como son Keras [4] y TensorFlow [5]. Adicionalmente, como los tiempos de entrenamiento de este tipo de modelos son bastante elevados si se realizan en un ordenador personal, se utilizará **Google Colab** [6] con el fin de utilizar tarjetas gráficas (GPU) para acelerar el proceso.

CAPITULO 2

PROCEDIMIENTO DE ENTRENAMIENTO DE LA RED Y ANÁLISIS.

Procedimiento de entrenamiento de la red y análisis

Una de las dificultades presentadas en el análisis de redes convolucionales es que las imágenes de entrada utilizadas habitualmente son imágenes aleatorias complejas, de diversos objetos reales. Con el fin de controlar el dataset de entrada en este proyecto se han utilizado dos datasets con imágenes simples de figuras geométricas (triángulos, cuadrados y elipses), con distintos tamaños (datasets 1.2 y. 1.3) y ángulos . Es posible encontrar más información al respecto en el apartado de Anexos/Datasets al final del proyecto.

El procedimiento llevado a cabo en la búsqueda de una red validad para la detección de las figuras geométricas y para conocer la sensibilidad de dicha red previamente entrenada se encuentra en la **Figura** 12 mostrada a continuación:

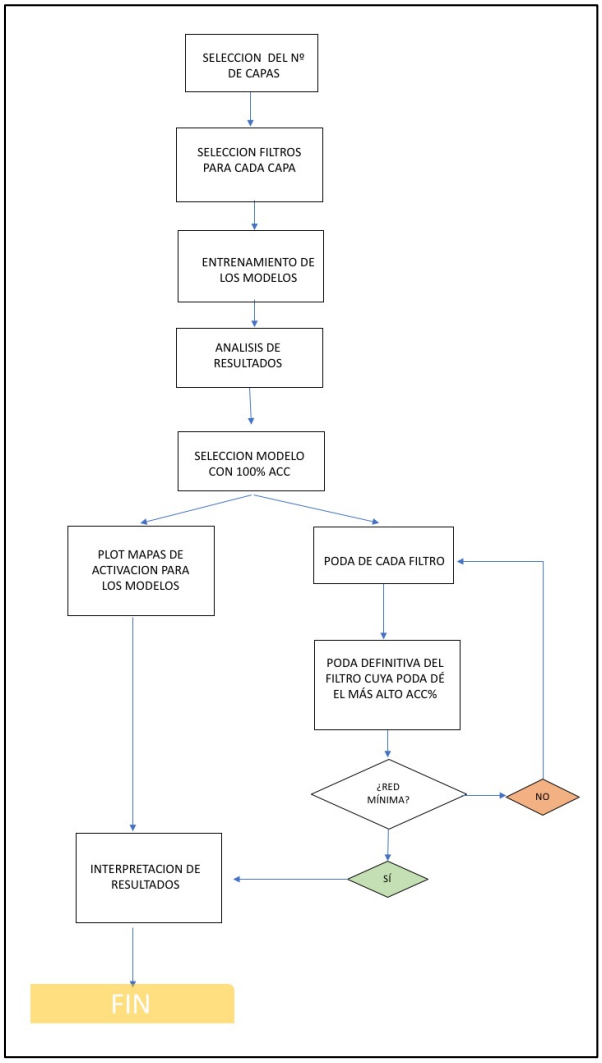


Figura 12 Procedimiento completo de análisis de redes convolucionales.

En la **Figura** 12 se encuentran todos los pasos llevados a cabo en el proceso que se ha llevado a cabo en los apartados siguientes para el análisis de los casos estudio. El procedimiento comienza con la selección de el número de capas de la red. Dado que el dataset utilizado es un dataset controlado como se explicaba anteriormente, y la simpleza de las imágenes, no es necesario utilizar un número elevado de capas como ocurren en las conocidas arquitecturas explicadas en el apartado capítulo Estado del arte. Una vez determinadas las capas del modelo, se ha. procedido a seleccionar el número de filtros adecuado para cada capa. Es entonces cuando se procede a crear el modelo y entrenarlo. Cada vez que se crea un modelo nuevo los filtros iniciales son distintos dado que los valores de dichos filtros son aleatorios. Tras estos entrenamientos, y analizando los distintos resultados para las distintas combinaciones, se han seleccionado dos casos en los cuales la red haya sido capaz de aprender de forma óptima, es decir, que las predicciones de entrenamiento y validación son altas y por tanto la clasificación es correcta. Con los modelos seleccionados se ha llevado a cabo un proceso de poda de la red hasta conocer los filtros más representativos de la red, es decir, los filtros que reconocen los patrones más resaltables de las figuras geométricas. Al mismo tiempo que se ha realizado la poda de la red, se han obtenido los mapas de activación de una selección de imágenes del dataset para cada filtro. Con los mapas de activación y los resultados de la poda se ha llevado a cabo la interpretación de resultados

Una vez seleccionado el dataset a se deben determinar las características de los modelos a utilizar. Dado que el propósito de este análisis es encontrar las redes más simples que alcancen altos niveles de correcta clasificación, se debe partir desde el menor número de capas e ir aumentando progresivamente Se ha comenzado por considerar una sola capa en la red, pero dada la gran simpleza de red, para un número suficiente de entrenamientos la red no ha sido capaz de aprender. Los resultados obtenidos aparecen en la **Tabla** 1 donde se encuentra indicado por columnas el dataset utilizado, en este caso el 1.2, del cual es posible encontrar información en el anexo *Datasets,* el numero de filtros de la capa en la columna FILTER1, el numero de épocas por cada entrenamiento en la columna NUM_EPOCH, y por últimos los resultados de acierto tras la clasificación tanto para el conjunto de entrenamiento como para el conjunto de validación en las columnas TRAIN_ACC y VALID_ACC correspondientemente.

DATASET	FILTER1	NUM_EPOCH	TRAIN_ACC	VALID_ACC
Dataset1.2	2	70	0,3420	0,3357
Dataset1.2	4	70	0,3484	0,3295
Dataset1.2	8	70	0,35	0,3348
Dataset1.2	16	70	0,3523	0,3357
Dataset1.2	32	70	0,3354	0,3357

Tabla 1 Resultados de entrenamiento de una red convolucional de una sola capa para diversos números de filtros.

Por lo tanto, se ha considerado necesario construir la red con dos capas. Tras la selección de la estructura de la red, se lleva a cabo el entrenamiento de dicha red para distintas combinaciones de filtros en las capas. Ha sido necesario crear diversos modelos y entrenar las mismas combinaciones un numero elevado de veces dado que las redes convolucionales parten de pesos aleatorios en los filtros *kernel* que se van actualizando según la red va siendo entrenada y va aprendiendo. Por lo tanto, dichos pesos iniciales, para características tan simples como las presentadas por nuestras imágenes de entrada de figuras geométricas, tienen una gran influencia en los porcentajes de acierto obtenidos.

A la hora de seleccionar el número de filtros óptimo para la red se debe tener en mente la complejidad del dataset de entrada y de la clasificación buscada, puesto que cuanto más complejo sea el patrón que diferenciar o más complejas sean las imágenes de entrada, mayor complejidad es requerida por la red.

Para cada combinación de filtros, se ha generado el modelo correspondiente que tiene la arquitectura mostradas en el ejemplo de la **Tabla** 1 Resultados de entrenamiento de una red convolucional de una sola capa para diversos números de filtros..

En la **Figura** 13 se observa una tabla representativa de un modelo de red convolucional. La primera capa corresponde a la capa convolucional, seguida de la capa *pooling*. Después de ésta se encuentra la segunda capa convolucional seguida del segundo *pooling*. Después se aplanan las salidas de la segunda capa para llevar a cabo la clasificación de manera más simple, como se comenta en el Estado de Arte anteriormente. El número de parámetros de cada capa depende del número de filtros usados

```
Model summary
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_13 (Conv2D)           (None, 28, 28, 4)         40
_________________________________________________________________
max_pooling2d_13 (MaxPooling (None, 14, 14, 4)         0
_________________________________________________________________
conv2d_14 (Conv2D)           (None, 14, 14, 4)         148
_________________________________________________________________
max_pooling2d_14 (MaxPooling (None, 7, 7, 4)           0
_________________________________________________________________
flatten_7 (Flatten)          (None, 196)               0
_________________________________________________________________
dense_7 (Dense)              (None, 3)                 591
=================================================================
Total params: 779
Trainable params: 779
Non-trainable params: 0
```

Figura 13 Ejemplo de modelo de red neuronal con dos capas

.

DATASET	N_TRAINING	FILTER1	FILTER2	NUM_EPOCH	TRAIN_ACC	VALID_ACC
1.3	1	2	2	40	0,33333	0,33278
1.3	1	2	4	40	0,33333	0,33278
1.3	1	2	8	40	0,33333	0,33278
1.3	1	2	16	40	0,33333	0,33444
1.3	2	2	2	40	1,00000	1,00000
1.3	2	2	2	40	0,99867	0,99834

Tabla 2 Entrenamiento de redes de dos capas para diversos filtros

En la **Tabla** 2 se muestra un numero reducido de ejemplos de entrenamiento de la red. La tabla tiene una gran similitud a la **Tabla** 1 explicada anteriormente, pero en. este caso hay dos columnas que indican el número de filtros puesto que esta vez la arquitectura de los modelos consiste en dos capas en vez de una. La primera columna indica el dataset utilizado, en este caso el 1.3, la columna N_TRAINING es el número de entrenamiento realizado. En las siguientes dos columnas, FILTER1 y FILTER2, se indica el numero de filtros que hay en cada capa, correspondiendo a la primera y segunda capa respectivamente. En la columna NUM_EPOCH se indica el número de épocas llevadas a cabo en cada entrenamiento, y las dos ultimas columnas, TRAIN_ACC y VALID_ACC se indican los resultados de acierto en la clasificación tanto para el conjunto de entrenamiento como para el de validación respectivamente.

En los datos mostrados podemos observar como en los cuatro primeros casos, tras un entrenamiento de 40 épocas la red no aprender, mientras que en los dos últimos sí. Además, podemos observar que, para una misma combinación de filtros, como es el caso de dos filtros en la primera capa y dos en la segunda la red puede aprender o no, como se puede observar en el caso 1, 5 y 6.

Llegados a este punto, se ha procedido a seleccionar para realizar dos casos en los cuales la red ha sido capaz de aprender para analizar los filtros uno a uno y observar, gracias a los mapas de activación, cuales son las partes activas en dichos filtros. Para ello, se ha realizado una poda a la red filtro a filtro para conocer cuales son los filtros más representativos de cada red, y cuales no aportan. Después visualmente se ha realizado una comparativa entre dichos filtros.

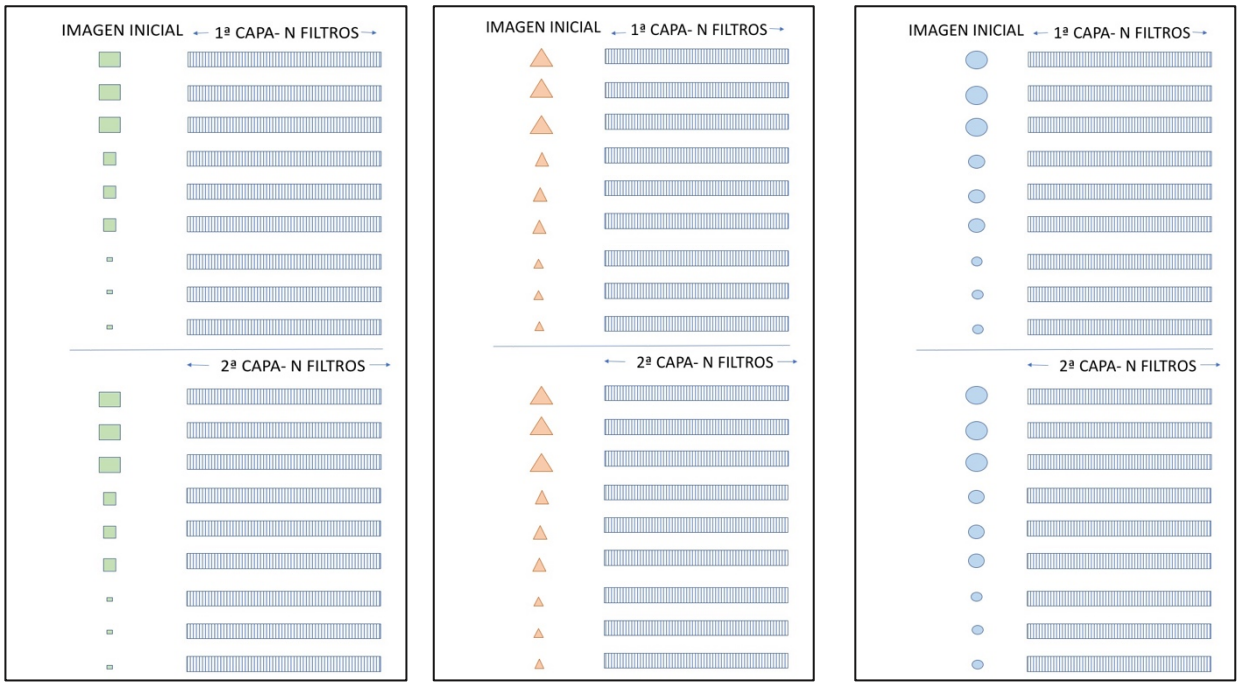


Figura 14 Esquema de la representación de mapas de activación para los casos estudio con dataset 1.3

En la 14 se muestra como son presentados los mapas de activación en los siguientes apartados donde se realiza el análisis de varios modelos. Para cada dataset la presentación de los mapas de activación será distinta debido a las distintas variaciones en los datasets, se escogerá el número de imágenes necesario para la comparación en cada caso. En este caso se ha decidido tomar tres imágenes de cada tamaño y figura.

La 14, específicamente muestra la representación de los mapas de activación para el caso del dataset 1.3, el cual consiste en variaciones de tamaño para las distintas figuras geométricas. Para evitar tomar como partes activas las que sean erróneas para cada tamaño y figura se han tomado tres imágenes distintas. Por lo tanto, se han tomado tres imágenes grandes, tres pequeñas y tres medianas para cada figura. Esto implica que el análisis de filtros para el Caso de Estudio I (apartado 8.1) consta de 27 imágenes analizadas de las cuales habrá un mapa de activación por cada filtro para cada capa.

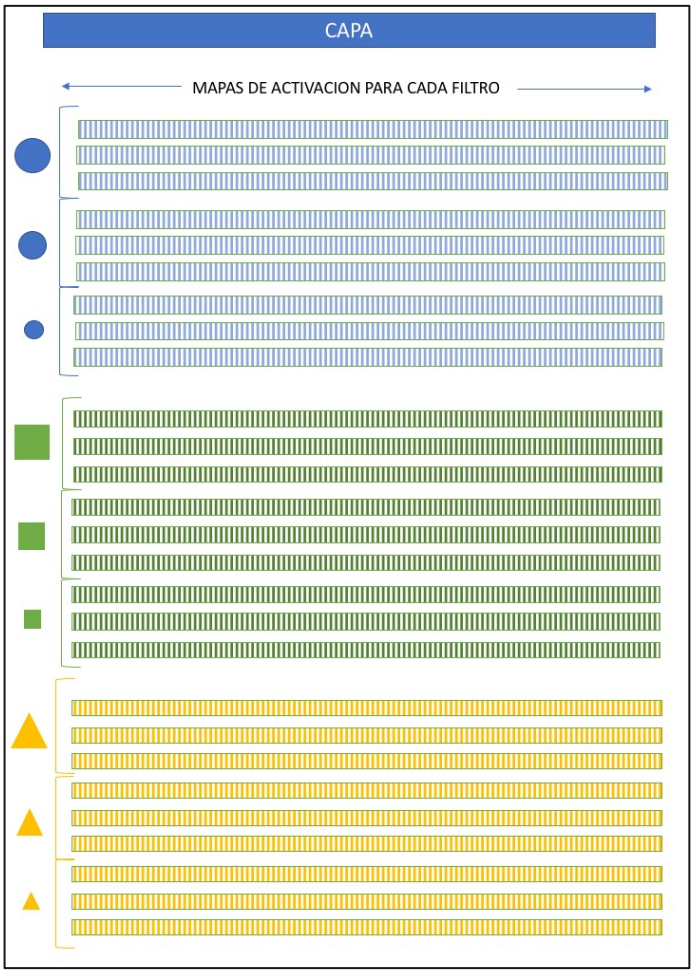


Figura 15 Esquema de la representación de todos mapas de activación para los casos estudio con dataset 1.3

Tras mostrar los mapas de activación con la imagen inicial correspondiente como se muestra en la **Figura** 14, para observar claramente los filtros obtenidos para cada imagen y poder comparar con la imagen de entrada, se ha procedido a realizar la comparación de todos los mapas de activación juntos. Con el fin comparar de forma sencilla los mapas de activación de cada filtro, que es lo que nos va a indicar lo que cada filtro extrae para cada tipo de imagen, se han colocado de forma vertical, en columna, como se ve indicado en la Figura 15.

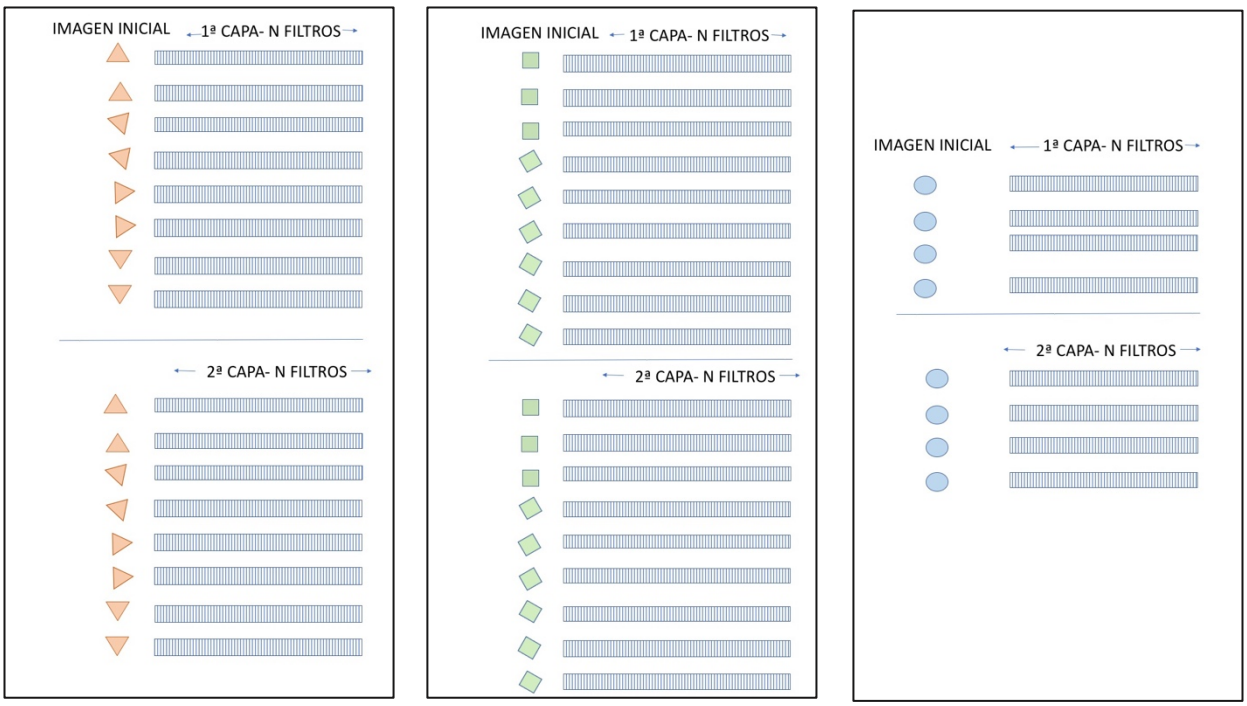


Figura 16 Esquema de la representación de mapas de activación en el caso estudio dataset 2.1.

En la **Figura** 16 se muestra un esquema de la representación de los mapas de activación para cada filtro realizado con el dataset 2.1, el cual contiene solo rotaciones de los polígonos. Es por ello por lo que el número de imágenes necesarias para el análisis es superior en el caso de triángulos y rectángulos que en el de elipses. Por esta causa se ha procedido a analizar cuatro elipses únicamente, en comparación con los nueve rectángulos analizados y los ocho triángulos. Para el caso de los rectángulos se han tomado tres imágenes giradas 0º, tres giradas 30ª y tres imágenes giradas 330ª. En el caso de los triángulos se han tomado dos imágenes para cada rotación dado que, a parte de los giros indicados en los rectángulos, también se han considerado los triángulos girados 180ª (esta rotación no tendría sentido considerarla en el análisis de rectángulos.)

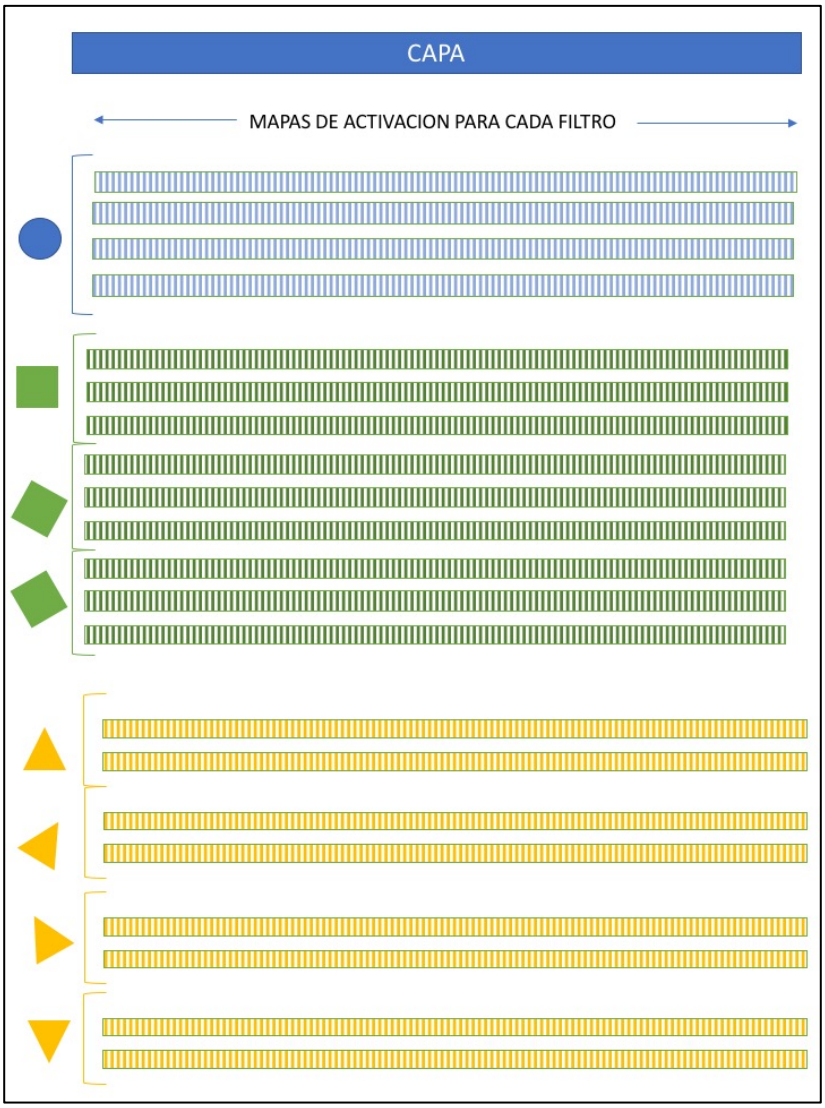


Figura 17 Esquema de la representación de todos los mapas de activación para los casos estudio con dataset 2.1

En la **Figura** 17 se presenta como se van a mostrar todas las imágenes analizadas en una misma figura para comparar de forma vertical todos los mapas de activación para cada filtro. Con el esquema mostrado en esta imagen se pretende mostrar como es la organización de las imágenes dado que la imagen de entrada no será mostrada en dicha figura. Más adelante en este apartado se puede encontrar un ejemplo de lo comentado.

Para entender el procedimiento de análisis que se lleva a cabo posteriormente en los casos estudios para el análisis visual de los filtros, y la poda de la red para conocer los filtros más relevantes para la red, a continuación, se procede a realizar el análisis de forma simplificada para una red pequeña en la que haya aprendido. Por ejemplo, de los resultados obtenidos del entrenamiento de modelos para el dataset 1.3, se ha elegido el caso mostrado en la tabla 4.

DATASET	FILTER1	FILTER2	NUM_EPOCH	TRAIN_ACC	VALID_ACC	TIMESTAMP_NAME
1.3	2	4	40	1,00000	1,00000	20190708T010213

Tabla 3 Modelo seleccionado para explicar la metodología de análisis de los casos estudio.

Seguidamente, se va a realizar el análisis de filtros observando los mapas de activación, con el fin de comprender el proceso que se llevará a cabo en los casos estudio. En las figuras que se encuentra a continuación (**Figura** 18, **Figura** 19 y **Figura** 20) se representan los mapas de activación para cada una de las figuras geométricas analizadas. La **Figura** 18 corresponde a los mapas de activación de los filtros del modelo obtenidos para las distintas elipses seleccionadas, en la **Figura** 19 los de los rectángulos, y por último, en la **Figura** 20 los de los triángulos.

Observando los resultados de las imágenes de las tres figuras, podemos observar que el segundo filtro de la primera capa en ninguna de las ocasiones activa ninguna parte de la imagen, no que podría indicar que la red no aprenda con dicho filtro. Estas características que vamos a a analizar en el último paso del análisis – en la poda de la red- se puede analizar previamente de manera visual sin tener mucha fiabilidad, pero para ir haciendo una idea de los resultados que se pueden esperar. Además,

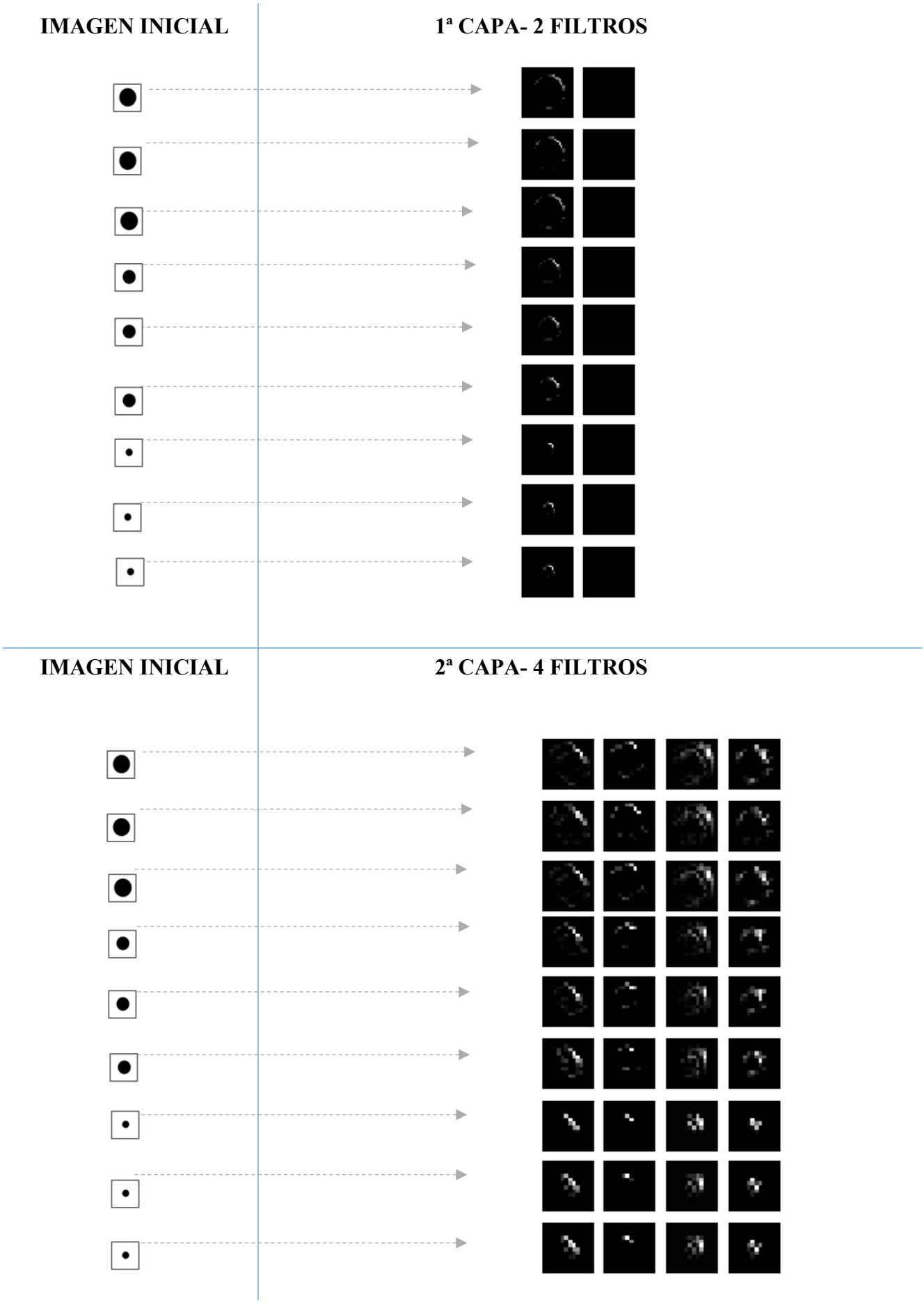


Figura 18 Mapas de activación para el modelo 2-4 filtros indicado en la tabla. 4 de las elipses

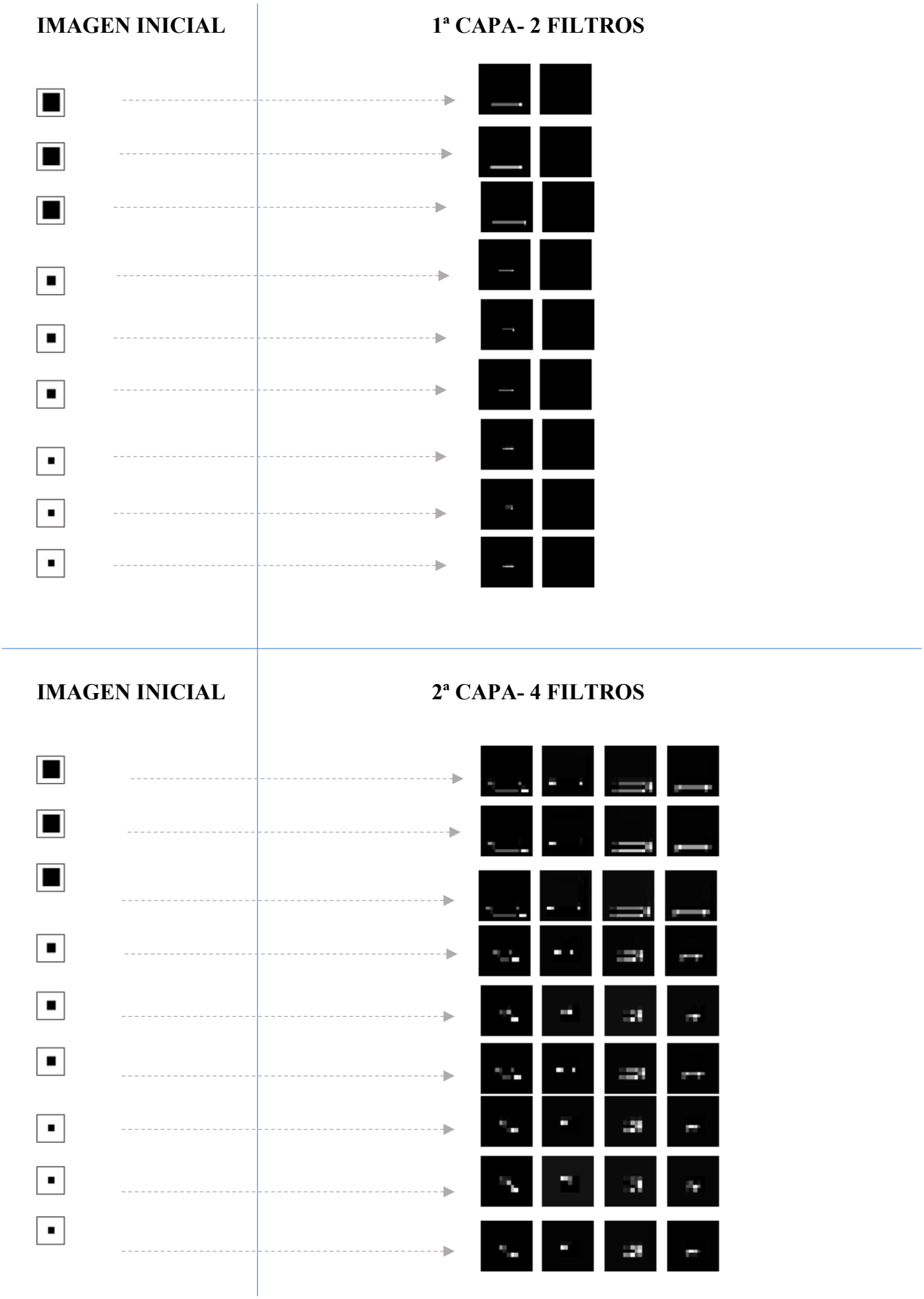


Figura 19 Mapas de activación para el modelo 2-4 filtros indicado en la tabla. 4 de los rectángulos

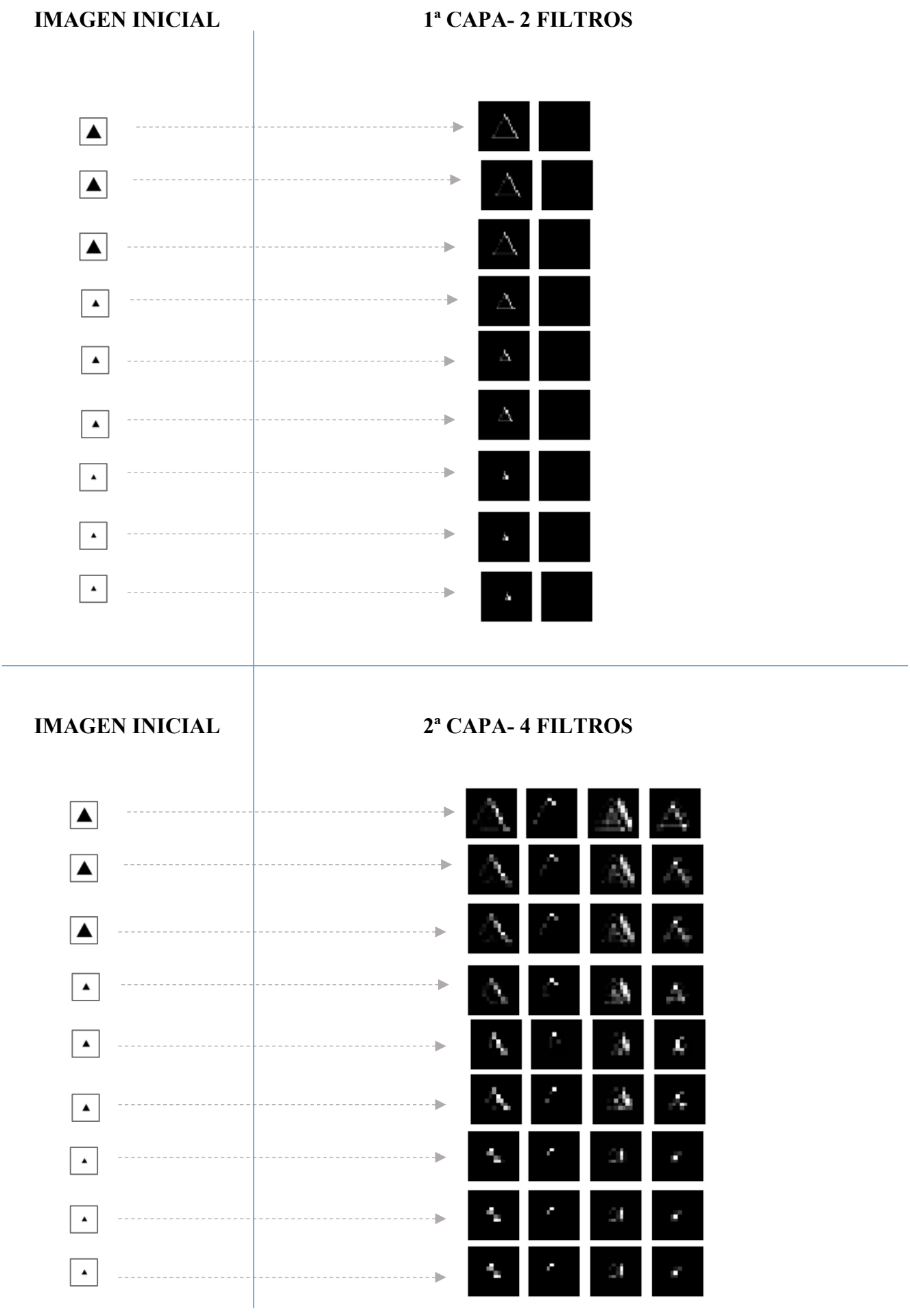


Figura 20 Mapas de activación para el modelo 2-4 filtros indicado en la tabla. 4 de los triángulos.

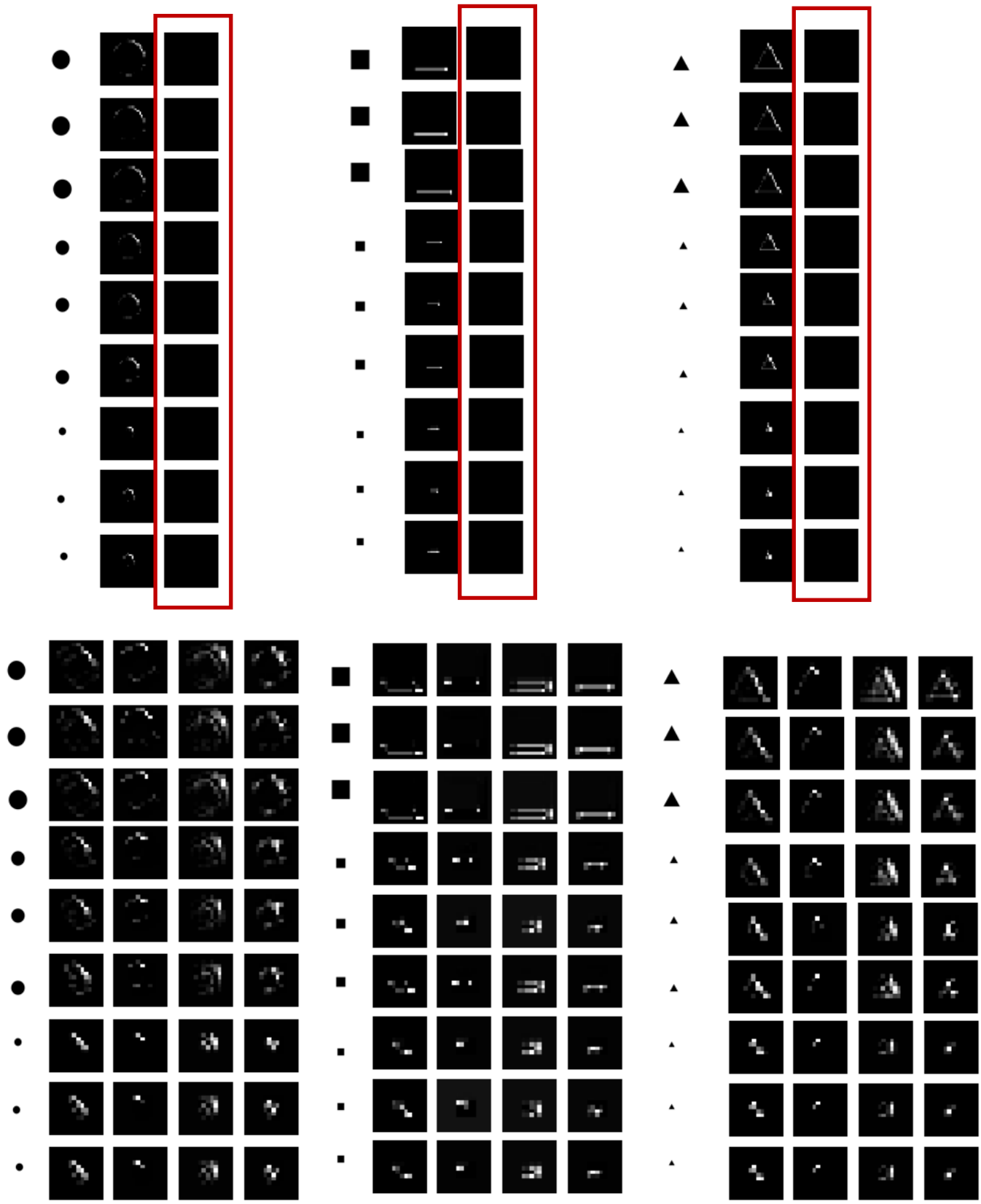


Figura 21 Mapas de activación para el modelo 2-4 20190708T010213, dataset 1.3.

La poda del caso ejemplo que se trata en este apartado, viene representada en la siguiente tabla. En la **Tabla** 4 viene representado por columnas los resultados de cada poda. En la primera columna se muestran los resultados obtenidos tras podar uno a uno cada filtro. Una vez podado el filtro que menor efecto tenga en el aprendizaje de la red, es decir, el que mayor porcentaje de acierto presente tras la poda, se procede a volver a realizar el proceso de forma reiterada hasta obtener los filtros más representativos para la red. Además, cabe resaltar que, una vez reducida cada capa a un solo filtro, no se ha continuado podando dicha capa dado que, como se explica anteriormente, una única capa no sería suficiente para la clasificación de imágenes.

Para la realización de la poda se ha utilizado la librería KerasSurgeon, la cual permite eliminar e insertar tanto capas como filtros. Es posible encontrar información sobre esta librería en las referencias [18] y [19].

	ACC% PODA 1	**ACC% PODA 2**	**ACC% PODA 3**	**ACC% PODA 4**
1ª CAPA- F1	0.3333	0.3333	0.3333	0.3333
1ª CAPA- F2	0.6116	0.66511	0.70566	-
2ª CAPA- F1	0.4583	0.563167	0.624667	0.624666
2ª CAPA- F2	0.6406	0.7056667	-	-
2ª CAPA- F3	0.6333	0.663	0.66566	0.66656
2ª CAPA -F4	0.6652	-	-	-

Tabla 4 Poda del modelo 2-4 filtros, timestamp 20190708T010213

Como podemos observar en la **Tabla** 4, en la primera iteración de podas, en el cuarto filtro de la segunda capa se obtiene en valor mas elevado- resaltado en la **Tabla** 4 en rojo, lo que implica que es el filtro menos significativo para la clasificación. Una vez podado dicho filtro, se ha procedido a repetir el proceso con los filtros restantes. Sin embargo, en este caso la reducción de porcentaje de acierto es similar en todos los casos, lo que implica que todos los filtros son significativos.

Una vez realizada la poda se ha procedido a analizar de forma visual aquellos filtros de mayor activación y aquellos que han sido podados. Como podemos observar la figura 4 donde se encuentran los mapas de activación, en primera poda las dos mayores reducciones de

porcentaje de acierto en la clasificación se dan para el primer filtro de la primera capa, y el primero de la segunda. Ambos filtros son en lo que más zonas activadas se pueden ver en los mapas, lo que implica una mayor detección de bordes y forma de las figuras geométricas.

También cabe resaltar que, para los filtros de la segunda capa , las partes activadas de la imagen son muy parecidas, lo que implica que los pesos de las matrices kernel son similares, y es por ello por lo que la poda de filtros en la segunda capa no supone una alta reducción del porcentaje de acierto en la clasificación. Sin embargo, algunos de estos filtros activan partes con más fuerza o como menos, y es esto lo que genera diferencias en los porcentajes tras la poda.

Si observamos el segundo filtro – resultado en color rojo en la **Figura** 4- de la primera capa, es fácil observar que dicho filtro no activa ninguno de los bordes o formas de las figuras geométricas que son las características que permiten diferencian las figuras geométricas para la clasificación. Sin embargo, el porcentaje de acierto inicialmente (en las primeras podas) se ve reducido de forma similar a otros filtros, un 40% prácticamente, Sin embargo, en la tercera poda, no solo la reducción de acierto es pequeña, sino que se ve aumentado este porcentaje, lo que implica que empeora la red.

CAPITULO 3

CASOS ESTUDIO

Casos estudio

A continuación, se encuentran dos casos estudio. El primero, como se explicaba anteriormente la clasificación de imágenes geométricas de las cuales se ha variado el tamaño, y el dataset consta de las tres figuras geométricas indicadas anteriormente; elipses, triángulos y rectángulos. En el segundo caso estudio se realiza el análisis de la red convolucional y su correspondiente aprendizaje con un dataset provisto de imágenes de las mismas figuras geométricas especificadas anteriormente con variación de rotaciones en vez de tamaño.

Para cada caso va a realizar la poda de dos redes en. las cuales la red haya aprendido, es decir, haya alcanzado un porcentaje de acierto del 100%.

CASO ESTUDIO I. Dataset 1.3

Primeramente, se van a analizar los entrenamientos realizados y sus resultados. Para automatizar la red, se ha creado una Pandas Dataframe en el cual se han creado para lo cual es necesario importar la librería Pandas correspondiente- *import pandas as pd* . En cada entrenamiento se añaden los resultados obtenidos en filas del dataframe, que posteriormente son guardadas como documento de Excel directamente en el drive.

De esta manera, gracias al uso de iteraciones y los dataframes se ha automatizado el proceso. Además, es importante tener en cuenta la necesidad de repetir los entrenamientos varias veces para mismas combinaciones de filtros en las dos capas, dado que la aleatoriedad inicial en los pesos del kernel tiene una gran influencia en los resultados finales.

DATASET	FILTER1	FILTER2	NUM_EPOCH	TRAIN_ACC	VALID_ACC	TIMESTAMP_NAME
1.3	2	2	40	0.33333	0.33278	20190703T161422
1.3	2	2	40	1	1	20190707T205348
1.3	2	2	40	0.99867	0.99834	20190707T211435
1.3	2	2	40	0.33333	0.33278	20190703T191444
1.3	2	2	40	1	1	20190708T065424
1.3	2	2	40	0.33333	0.33444	20190708T005401
1.3	2	4	40	0.33333	0.33278	20190703T164808
1.3	2	4	40	0.33333	0.33278	20190703T192701
1.3	2	4	40	0.33333	0.33278	20190707T210137
1.3	2	4	40	0.33333	0.33444	20190707T212225
1.3	2	4	40	1	1	20190708T010213
1.3	2	4	40	0.66667	0.66556	20190708T070304
1.3	2	8	40	0.33333	0.33278	20190703T165955
1.3	2	8	40	0.33333	0.33278	20190703T193925
1.3	2	8	40	0.33333	0.33444	20190707T213005
1.3	2	8	40	0.33333	0.33278	20190708T011031
1.3	2	8	40	0.66667	0.66722	20190708T071141
1.3	2	16	40	0.33333	0.33444	20190703T171150
1.3	2	16	40	0.33333	0.33278	20190703T195149
1.3	2	16	40	0.66667	0.66722	20190707T213748
1.3	2	16	40	1	1	20190708T011848
1.3	2	16	40	1	1	20190708T072018
1.3	2	32	40	0.66667	0.66722	20190708T012711
1.3	2	32	40	1	1	20190703T172350
1.3	2	32	40	1	1	20190703T200416
1.3	2	32	40	1	1	20190707T214541
1.3	2	32	40	1	1	20190708T072855
1.3	4	2	40	0.33333	0.33444	20190703T174743
1.3	4	2	40	0.33333	0.33278	20190703T201658
1.3	4	2	40	1	1	20190708T014350
1.3	4	2	40	1	1	20190708T074616
1.3	4	2	40	0.33333	0.33444	20190707T220134
1.3	4	16	40	0.33333	0.33278	20190703T173546
1.3	4	16	40	0.33333	0.33278	20190703T202942
1.3	4	16	40	0.33333	0.33278	20190707T215337
1.3	4	16	40	0.66667	0.66556	20190708T013528
1.3	4	16	40	0.66667	0.66556	20190708T073733
1.3	16	4	40	0.33333	0.33278	20190703T175938

1.3	16	4	40	0.33333	0.33278	20190703T204232
1.3	16	4	40	0.33333	0.33444	20190707T220930
1.3	16	4	40	1	1	20190708T015217
1.3	16	4	40	0.33333	0.33278	20190708T075503
1.3	16	8	40	0.66667	0.66667	20190703T181123
1.3	16	8	40	0.66667	0.6672	20190703T205526
1.3	16	8	40	1	1	20190707T221731
1.3	16	8	40	1	1	20190708T020046
1.3	16	8	40	0.66667	0.66722	20190708T080347
1.3	32	2	40	1	1	20190703T182316
1.3	32	2	40	1	1	20190703T210829
1.3	32	2	40	1	1	20190708T081235
1.3	32	2	40	0.66667	0.66556	20190707T222538
1.3	32	2	40	0.33333	0.33278	20190708T020925
1.3	32	4	40	0.33333	0.33278	20190703T183519
1.3	32	4	40	0.33333	0.33278	20190703T212135
1.3	32	4	40	1	1	20190707T223341
1.3	32	4	40	0.66667	0.66722	20190708T021802
1.3	32	4	40	0.33333	0.33278	20190708T082124

Tabla 5 Resultados de entrenamiento modelos de dos capas con distintas combinaciones con el dataset 1.3.

En la tabla 6 se muestran los entrenamientos realizados con el dataset 1.3. Como podemos observar, para mismas combinaciones de parámetros hay casos en los que la red aprende y casos en los que no aprende. Eso muestra la dependencia existente entre los resultados obtenidos y los valores aleatorios asignados a los kernel. Dichos valores se van actualizando con los entrenamientos pero en algunos casos se alcanzan mínimos locales de los cuales la red no puede salir y por tanto el aprendizaje no mejora. Existen métodos para evitar dichos mínimos locales, pero estos métodos no son tratados en este experimento.

Una vez entrenada la red un número elevado de veces, se ha procedido a analizar las siguientes redes en las cuales la red ha aprendido:

1. 16-4 filtros, 20190708T015217, entrenada con el dataset 1.3.
2. 2-16 filtros, 20190708T011848, entrenada con el dataset 1.3.

Ambos modelos se pueden encuentran en el recopilatorio de datos de entrenamiento adjuntados en la tabla 6.

- **Análisis modelo 16-4 filtros, 20190708T015217, entrenada con el dataset 1.3**

Para realizar la comparativa de los filtros la poda para alcanzar la red mínima se encuentra representados todos los filtros para las distintas imágenes en las figuras 5 y 6 más adelante.

Analizando visualmente los filtros, para obtener una primera idea de lo que cabe esperar en la poda se puede observar que en los filtros 5, 10 y 11 de la primera capa representados en la figura 21, la red no cambia prácticamente respecto a las imágenes iniciales, lo que implica que no esta extrayendo ningún patrón especifico.

Además, se ha observado en que entre los filtros 4 y 6, y los filtros 9 y 15 – observables en la figura 21- pare haber cierta similitud. En tal caso, puede haber un sobre aprendizaje de dichos patrones y no es necesario mantener ambos. La reducción de filtros en las redes convolucionales es de gran ayuda cuando los modelos son de gran tamaño o existen una gran cantidad de imágenes que se deben analizar, dado que la velocidad de ejecución aumenta al reducir la complejidad de la red.

A continuación, se ha analizado la segunda capa. En esta capa, tras haber pasado ya por una serie de filtros, las imágenes muestran más dificultad a la hora de ser analizadas visualmente. Sin embargo, es fácil observar que, en el segundo filtro de esta capa, no se activa ninguna parte de la imagen cuando las elipses y los rectángulos son muy pequeños, o en el caso de los triángulos no se activa ninguna parte de la imagen independientemente del tamaño de la figura geométrica.

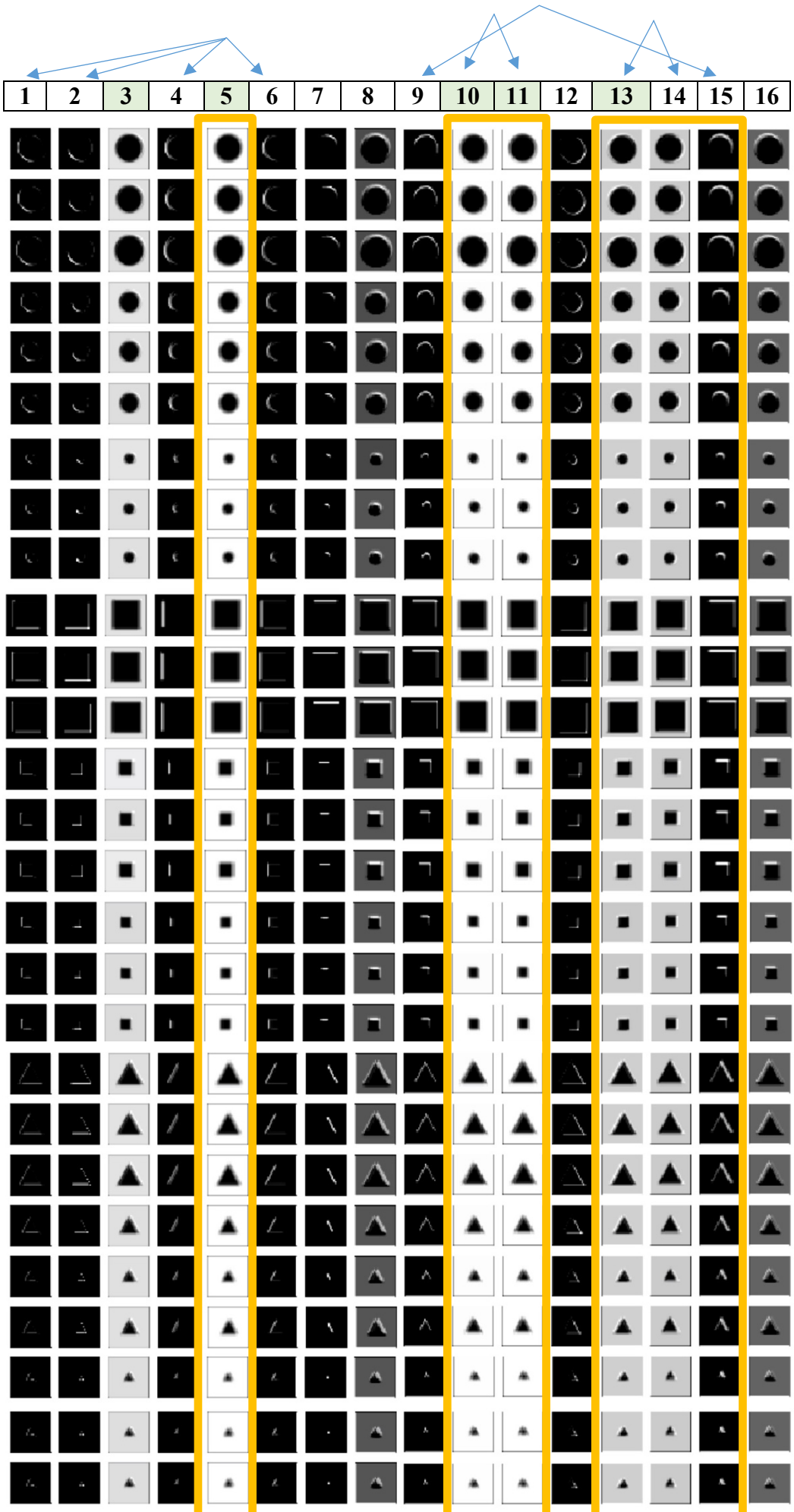


Figura 22 Mapas de activación de la primera capa de 16 filtros, para modelo 16-4

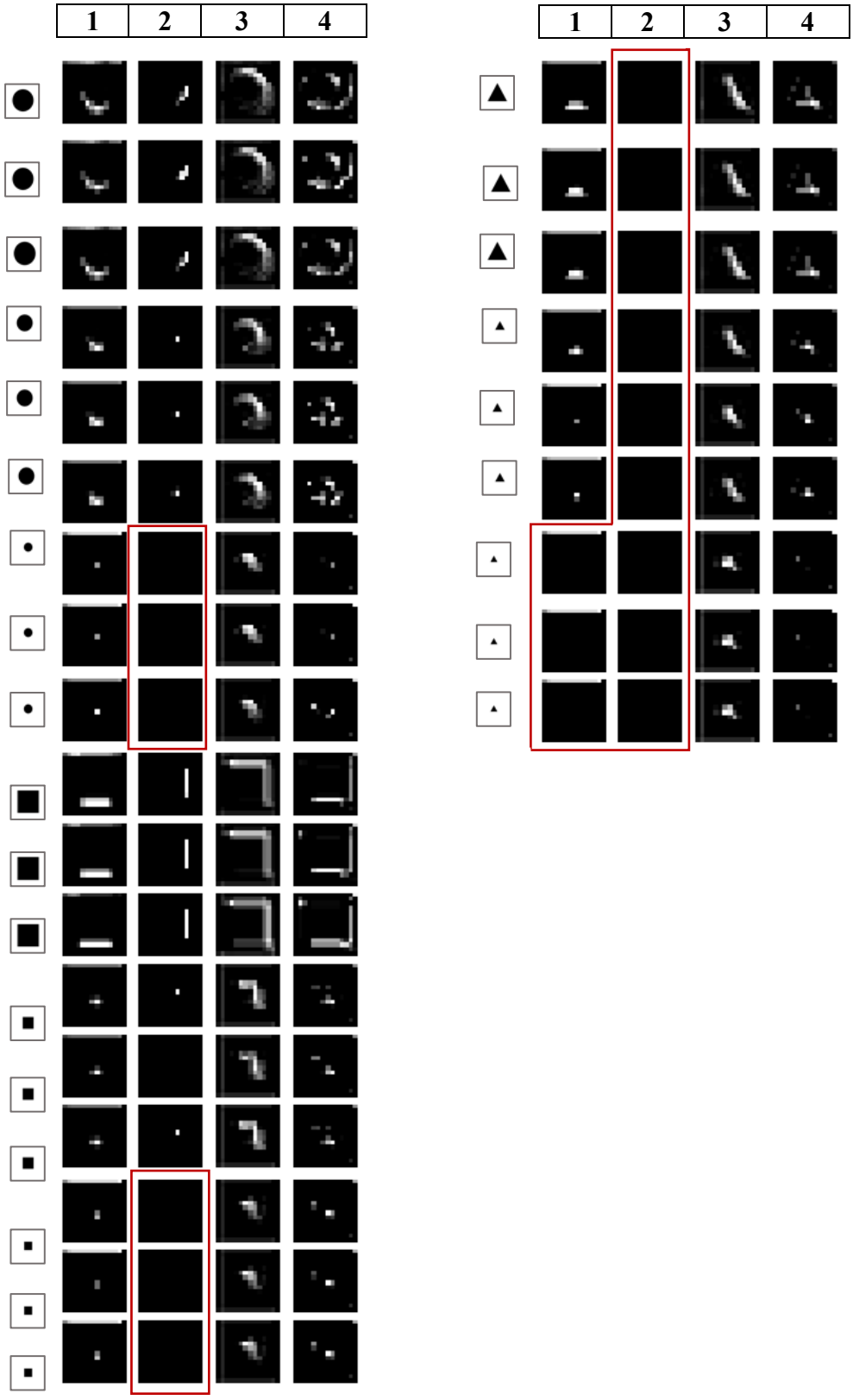


Figura 23 Mapas de activación de la segunda capa de 4 filtros, para el modelo 16-4

	ACC% 1	ACC% 2	ACC% 3	ACC% 4	ACC% 5	ACC% 6
F 1	0.659	0.840	-	-	-	-
F 2	0.673	0.809	0.827	0.858	-	-
F 3	0.333	0.333	0.333	0.333	0.333	0.389
F 4	0.676	0.811	0.823	0.826	0.825	-
F 5	0.418	0.415	0.420	0.422	0.417	0.400
F 6	0.560	0.783	0.779	0.791	0.755	0.738
F 7	0.694	0.815	0.869	-	-	-
F 8	0.400	0.599	0.588	0.593	0.571	0.541
F 9	0.617	0.783	0.768	0.445	0.745	0.693
F 10	0.583	0.553	0.559	0.558	0.593	0.303
F 11	0.667	0.333	0.333	0.333	0.333	0.333
F 12	0.624	0.822	0.830	0.847	0.828	0.679
F 13	0.482	0.519	0.516	0.515	0.511	0.358
F 14	0.333	0.626	0.630	0.624	0.617	0.727
F 15	0.601	0.740	0.782	0.803	0.774	0.776
F 16	0.465	0.639	0.622	0.632	0.603	0.751
F 1	0.823	-	-	-	-	-
F 2	0.655	0.782	0.809	0.829	0.818	0.569
F 3	0.575	0.350	0.347	0.337	0.330	0.421
F 4	0.473	0.718	0.695	0.684	0.684	0.738

	ACC% 7	ACC% 8	ACC% 9	ACC% 10	ACC% 11	ACC% 12
F 1	-	-	-	-	-	-
F 2	-	-	-	-	-	-
F 3	0.461	0.501	0.419	0.516	0.566	0.524
F 4	-	-	-	-	-	-
F 5	0.390	0.382	0.402	0.398	0.401	0.401
F 6	0.755	-	-	-	-	-
F 7	-	-	-	-	-	-
F 8	0.493	0.455	0.367	0.350	0.395	0.358
F 9	0.728	0.722	0.721	-	-	-
F 10	0.192	0.192	0.197	0.188	0.125	0.165
F 11	0.333	0.371	0.370	0.423	0.492	0.545
F 12	0.697	0.676	0.664	0.637	0.690	-
F 13	0.284	0.295	0.304	0.279	0.308	0.307
F 14	0.632	0.620	0.621	0.598	0.591	0.578
F 15	0.717	0.699	0.697	0.672	-	-
F 16	-	-	-	-	-	-
F 1	-	-	-	-	-	-
F 2	0.725	0.755	-	-	-	-
F 3	0.380	0.377	0.374	0.372	0.358	0.368
F 4	0.533	0.528	0.570	0.551	0.477	0.493

	ACC% 13	ACC% 14	ACC% 15	ACC% 16	ACC% 17	ACC% 18
F 1	-	-	-	-	-	-
F 2	-	-	-	-	-	-
F 3	0.387	0.305	0.205	0.333	0.359	-
F 4	-	-	-	-	-	-
F 5	0.406	0.408	0.408			
F 6	-	-	-	-	-	-
F 7	-	-	-	-	-	-
F 8	0.543	-	-	-	-	-
F 9	-	-	-	-	-	-
F 10	0.232	0.232	0.333	0.380	-	-
F 11	0.516	0.332	0.300	0.353	0.341	0.308
F 12	-	-	-	-	-	-
F 13	0.341	0.279	0.303	0.328	0.333	0.333
F 14	-	-	-	-	-	-
F 15	-	-	-	-	-	-
F 16	-	-	-	-	-	-
F 1	-	-	-	-	-	-
F 2	-	-	-	-	-	-
F 3	0.455	0.458	-	-	-	-
F 4	0.394	0.327	-	-	-	-

Tabla 6 Poda del modelo 16-4 entrenado con el dataset 1.3

Como se explicaba en el capítulo anterior con el ejemplo de 2-4 filtros, la poda se ha realizado eliminando los filtros de valor menos significativo y volviendo a comprobar cual eliminar hasta alcanzar los más significativos y la red mínima.

Considerando los resultados obtenidos podemos comprobar en este caso todos los filtros parte de un valor significativo considerable puesto que la primera poda supone una reducción del 25%.

La tabla 7 muestra a partir de la poda 14 los filtros más representativos, pues que es a partir de entonces donde la disminución en porcentaje de acierto es mayor del 50%. Es por tanto que los filtros más representativos son: 3,5,10,11 13 de la primera capa y los filtros 3 y 4 de la segunda. A pesar de lo que se esperaba obtener en los resultados, en las cuales se había determinado el poco aprendizaje de los filtros 5, 10, 11, se ha comprobado tras la poda dichos filtros no solo deben no ser eliminados, sino que son los filtros más representativos de la red, es decir, los filtros que más aportan a una correcta clasificación. En la figura 22 y en la tabla 7 se han señalado en color amarillo los filtros más representativos, a partir de los cuales la reducción de *accuracy* obtenidas es menor de un 65%. Podemos observar que la mayoría de ellos no transforman la imagen inicial drásticamente para la activación de partes o bordes, y eso no implica que el aprendizaje sea menor sino todo lo contrario.

Además, el filtro dos de la segunda capa, el no parece aportar aprendizaje alguno en el caso de las imágenes de triángulos, y un aprendizaje reducido en el caso de elipses y rectángulos, es eliminado en la octava iteración. Esto se debe a que la distinción que realiza de patrones a pesar de ser pequeña no debe llevar a la red a confusión. Es decir, un filtro puede extraer patrones de la imagen que conlleven a determinar una figura geométrica como otra. Pero en cambio, si el patrón es capaz de diferenciar dos de las figuras geométricas, y duda con la tercera, la posibilidad de fallo es menor.

Si observamos la tabla 7, se han resaltado en color rojo ciertos filtros. En ellos la variación de efecto de su poda es muy cambiante cuando se poda otro filtro. En el caso del dos de en la situación inicial cuando todos los filtros forman parte de la red, la predicción tras su poda es mayor que cuando el filtro 1 de la segunda capa ya ha sido podado. Estos resultados parecen mostrar cierta interacción o efecto entre filtros. Dado que los patrones extraídos de las imágenes pueden contradecirse o cuando la similitud en la activación es muy elevada no aportar

más informaciones. También pueden ocurrir ambas cosas, que en ciertas clasificaciones dichos darían la misma respuesta mientras que con otras clasificaciones los resultados de ambos son contradictorios.

Además, en la figura 23, se encuentran indicados aquellos filtros que son similares. Si analizamos los resultados obtenidos en la poda de los filtros indicados (1,2,4,6) se puede observar que en la poda de los primeros filtros no varía la caída de porcentaje de acierto. Eso puede ser debido a la similitud en la activación de partes en los mapas de activación. También cabe resaltar que el filtro que ha permanecido durante más tiempo tras la poda es el de mayor activación y no ha continuado significando la misma reducción de porcentaje de acierto tras la poda de los tres restantes.

- **Análisis modelo 2-16 filtros, entrenada con el dataset 1.3.**

PRIMERA CAPA.

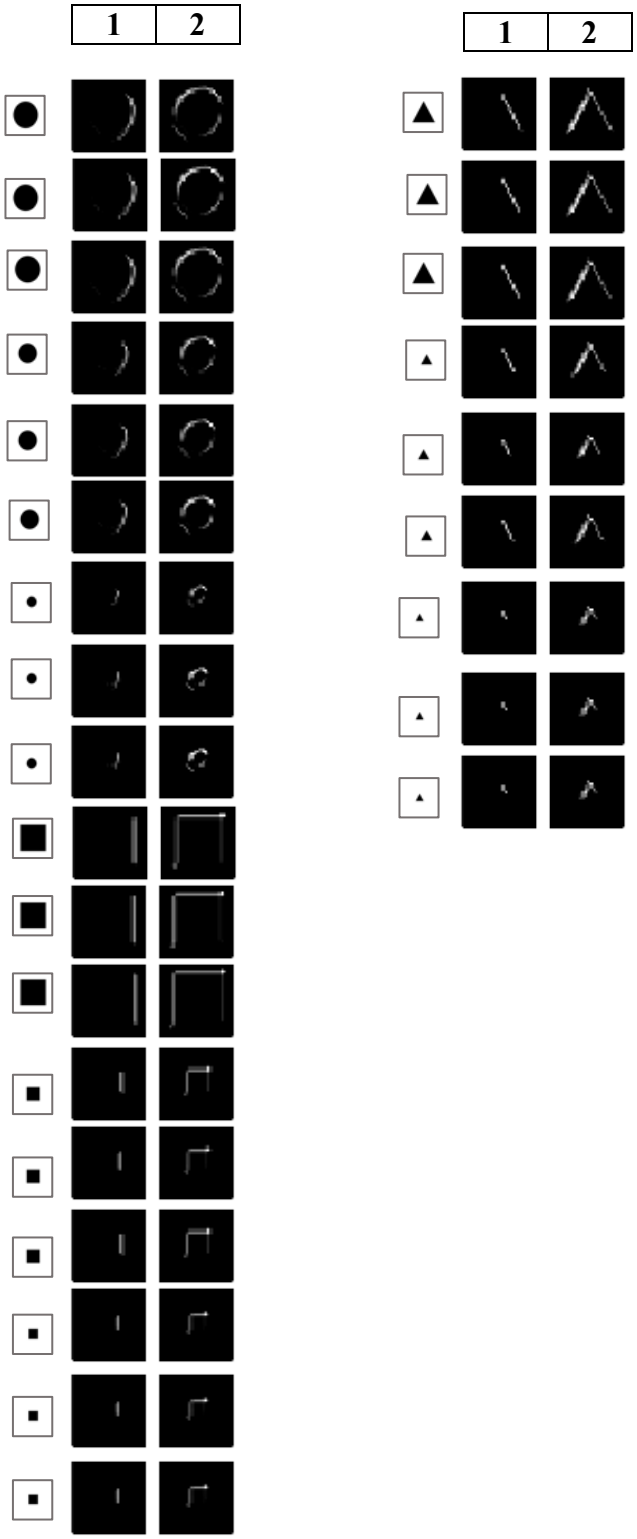


Figura 24 Mapas de activación de la segunda capa de 2 filtros, para el modelo 2-16

Figura 25 Mapas de activación de la segunda capa de 16 filtros, para el modelo 2-16.

	ACC% PODA 1	ACC% PODA 2	ACC% PODA 3	ACC% PODA 4	ACC% PODA 5	ACC% PODA 6	ACC% PODA 7
F 1	0.629450915	0.600332779	0.542429285	0,6294509151	0,7377703827	0,734608985	0,8204658902
F 2	0.539933444	0.485357737	0.451081531	0,5399334443	0,6287853577	0,6499168053	0,4941763727
F 1	1.0000	-	-	-	-	-	-
F 2	0.982695508	0.758569052	0,8585690516	0,9595673877	1.0000	-	-
F 3	0.974875208	0.906489185	0,9495840266	0,8637271215	0,7638935108	0,85640599	0,7760399334
F 4	0.935773711	0.778202995	0,7657237937	0,7103161398	0,8349417637	0,8103161398	0,7600665557
F 5	0.999667221	0.975374376	0,9930116473	0,9787021631	0,9252911814	0,9915141431	-
F 6	0.988186356	0.802828619	0,9161397671	0,9557404326	0,9633943428	0,8405990017	0,8755407654
F 7	0.980199667	0.955407654	0,9936772047	-	-	-	-
F 8	1.0000	-	-	-	-	-	-
F 9	0.999334443	0.943427621	0,9204658902	0,9555740433	0,9356073211	0,9787021631	0,9362728785
F 10	1.0000	-	-	-	-	-	-
F 11	0.996505824	0.983194676	0,9462562396	0,9823627288	-	-	-
F 12	0.991680532	0.85890183	0,9234608985	0,953078203	0,9309484193	0,8755407654	0,8863560732
F 13	0.971214642	0.984193012	0,8803660566	0,9169717138	0,7129783694	0,7554076539	0,7633943428
F 14	0.995341098	0.829950083	0,9134775374	0,9552412646	0,9648918469	0,9074875208	0,8968386023
F 15	0.967720466	0.9943	-	-	-	-	-
F 16	0.921630616	0.812146423	0,7973377704	0,7806988353	0,8038269551	0,7600665557	0,8369384359

	ACC% PODA 8	ACC% PODA 9	ACC% PODA 10	ACC% PODA 11	ACC% PODA 12	ACC% PODA 13	ACC% PODA 14
F 1	0,7314475874	0,8304492512	0,8239600666	0,7798668885	0,7953410982	0,640765391	0,5926788686
F 2	0,3930116473	0,5695507488	0,5151414309	0,3908485857	0,55640599	0,6783693844	0,5054908486
F 1	-	-	-	-	-	-	-
F 2	-	-	-	-	-	-	-
F 3	0,7608985025	0,8500831947	-	-	-	-	-
F 4	0,5911813644	0,6339434276	0,4672212978	0,2956738769	0,5775374376	0,6103161398	-
F 5	-	-	-	-	-	-	-
F 6	0,9116472546	-	-	-	-	-	-
F 7	-	-	-	-	-	-	-
F 8	-	-	-	-	-	-	-
F 9	-	-	-	-	-	-	-
F 10	-	-	-	-	-	-	-
F 11	-	-	-	-	-	-	-
F 12	0,8743760399	0,744093178	0,6773710483	0,6079866889	0,7104825291	0,6608985025	-
F 13	0,5633943428	0,7675540765	0,6963394343	0,9009983361	-	-	-
F 14	0,9266222962	0,7678868552	0,6227953411	0,5856905158	0,8675540765	-	-
F 15	-	-	-	-	-	-	-
F 16	0,7590682196	0,6580698835	0,8282861897	-	-	-	-

Tabla 7 Poda del modelo de 2-16 filtros hasta alcanzar la red mínima

En este caso podemos observar en la tabla 8 que hasta la hasta la poda 11 se mantiene una accurary mayor que el 90% lo que implica que hay muchos filtros que no ayudan al reconocimiento de las figuras o que los patrones que detectan son similares entre si, y por tanto la aportación de dicho filtro está siendo cubierta por otro. Es más, en la primera poda se han eliminado tres canales dado que el resultado tras su poda continuaba resultando en un 100% de porcentaje de acierto en la clasificación.

En la ultima poda los filtros que permanecen son los de la primera capa lo que implica que dicha capa es muy importante en la clasificación de imágenes, y entre los dos filtros de esta capa en más representativo es el segundo filtro. Observando los mapas de activación en la figura 25 podemos observar que el segundo filtro de la primera capa activa prácticamente los bordes completos de las imágenes, y además la activación es mayor que en el primer filtro.

En la comparación visual de la segunda capa, existe mayor dificultad puesto que la entrada de estas capas son las imágenes resultantes de pasar por la primera capa por lo que no es fácil detectar los patrones o partes activadas por estos filtros. Sin embargo, podemos observar que la activación de ciertos filtros como el 3 y el 8 para las elipses en mínima y, por otra parte, el filtro más característico de la segunda capa que según la poda mostrada en la tabla 8 es el filtro 4, y corresponde con las mayores activaciones visuales mostradas en las capas de activación.

CASO ESTUDIO II. Dataset 2.1

FILTER1	FILTER2	NUM_TRAINING	NUM_EPOCH	TRAIN_ACC	VALID_ACC	TIMESTAMP
2	2	1	50	0,6666666667	0,6666666667	20190808T194929
2	4	1	50	0,6666666667	0,6666666667	20190808T212943
2	8	1	50	0,3333333333	0,3333333333	20190808T213918
2	16	1	50	0,3333333333	0,3333333333	20190808T214845
2	32	1	50	0,3333333333	0,3333333333	20190808T215811
2	2	1	50	0,3333333333	0,3333333333	20190809T094223
2	4	1	50	0,3333333333	0,3333333333	20190809T103414
2	8	1	50	0,6666666667	0,6666666667	20190809T104855
2	16	1	50	0,3333333333	0,3333333333	20190809T110343
2	32	1	50	0,3333333333	0,3333333333	20190809T111829
2	16	1	50	1	1	20190807T200046
4	2	1	50	0,3333333333	0,3333333333	20190808T220738
4	2	1	50	0,6666666667	0,6666666667	20190809T113316
4	4	1	50	0,3333333333	0,3333333333	20190809T114806
4	8	1	50	0,3333333333	0,3333333333	20190809T120303
4	16	1	50	0,3333333333	0,3333333333	20190809T121806
4	32	1	50	0,6666666667	0,6666666667	20190809T123312
4	2	1	70	1	1	20190816T072238
4	4	1	70	0,3333333333	0,3333333333	20190816T082225
4	8	1	70	0,3333333333	0,3333333333	20190816T084212
4	16	1	70	0,6666666667	0,6666666667	20190816T090210
4	32	1	70	0,3333333333	0,3333333333	20190816T092204
4	64	1	70	0,6666666667	0,6666666667	20190816T094204
8	2	1	50	0,3333333333	0,3333333333	20190809T124814
8	4	1	50	0,3333333333	0,3333333333	20190809T130319
8	8	1	50	0,3333333333	0,3333333333	20190809T131824
8	16	1	50	0,6666666667	0,6666666667	20190809T133338
8	32	1	50	0,3333333333	0,3333333333	20190809T134847
8	2	1	70	0,6666666667	0,6666666667	20190809T192059
8	4	1	70	1	1	20190809T202124

8	8	1	70	0,3333333333	0,3333333333	20190809T203515
8	16	1	70	0,3333333333	0,3333333333	20190809T204908
8	32	1	70	0,3333333333	0,3333333333	20190809T210300
8	2	1	70	0,6666666667	0,6666666667	20190816T100202
8	4	1	70	0,3333333333	0,3333333333	20190816T102204
8	8	1	70	0,3333333333	0,3333333333	20190816T104223
8	16	1	70	0,3333333333	0,3333333333	20190816T110241
8	32	1	70	0,3333333333	0,3333333333	20190816T112308
8	64	1	70	0,3333333333	0,3333333333	20190816T114336
16	2	1	50	0,6666666667	0,6666666667	20190809T140350
16	4	1	50	0,3333333333	0,3333333333	20190809T141858
16	8	1	50	0,3333333333	0,3333333333	20190809T143410
16	16	1	50	0,3333333333	0,3333333333	20190809T144927
16	4	1	50	1	1	20190808T143225
16	2	1	70	0,3333333333	0,3333333333	20190816T120400
16	4	1	70	0,3333333333	0,3333333333	20190816T122429
16	8	1	70	0,3333333333	0,3333333333	20190816T124449
16	16	1	70	1	1	20190816T130514
16	32	1	70	0,6666666667	0,6666666667	20190816T132651
16	64	1	70	0,3333333333	0,3333333333	20190816T134845
32	2	1	70	1	1	20190813T121015
32	4	1	70	0,6666666667	0,6666666667	20190813T122403
32	8	1	70	0,6666666667	0,6666666667	20190813T123750
32	16	1	70	0,3333333333	0,3333333333	20190813T125137
32	32	1	70	0,3333333333	0,3333333333	20190813T130526
32	64	1	70	0,3333333333	0,3333333333	20190813T153822
32	64	1	70	0,3333333333	0,3333333333	20190813T184454
64	2	1	70	1	1	20190813T172213
64	4	1	70	0,3333333333	0,3333333333	20190813T173548
64	8	1	70	0,6666666667	0,6666666667	20190813T174931
64	16	1	70	0,3333333333	0,3333333333	20190813T180316
64	32	1	70	0,3333333333	0,3333333333	20190813T181702
64	64	1	70	0,6666666667	0,6666666667	20190813T183052
64	2	1	70	0,3333333333	0,3333333333	20190813T185844
64	4	1	70	1	1	20190813T191232
64	8	1	70	0,3333333333	0,3333333333	20190813T192600
64	16	1	70	0,3333333333	0,3333333333	20190813T193918
64	32	1	70	0,6666666667	0,6666666667	20190813T195239
64	64	1	70	0,3333333333	0,3333333333	20190813T200556

Tabla 8 Resultados de entrenamiento modelos de dos capas con distintas combinaciones con el dataset 2.1

- **Análisis modelo 16-16 filtros, entrenada con el dataset 2.1.**

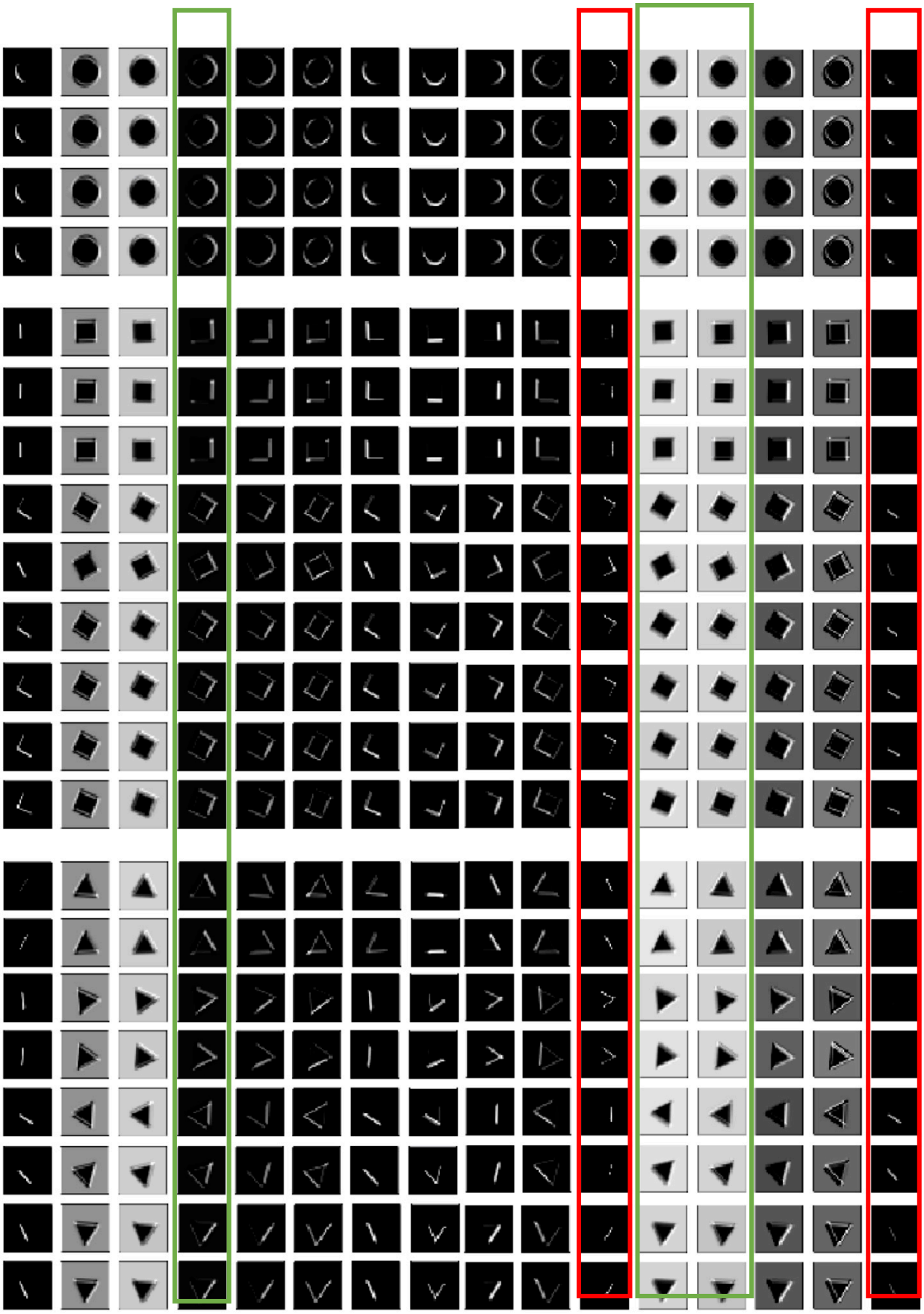


Figura 26 Mapas de activación de la primera capa de 16 filtros, para el modelo 16-16

SEGUNDA CAPA

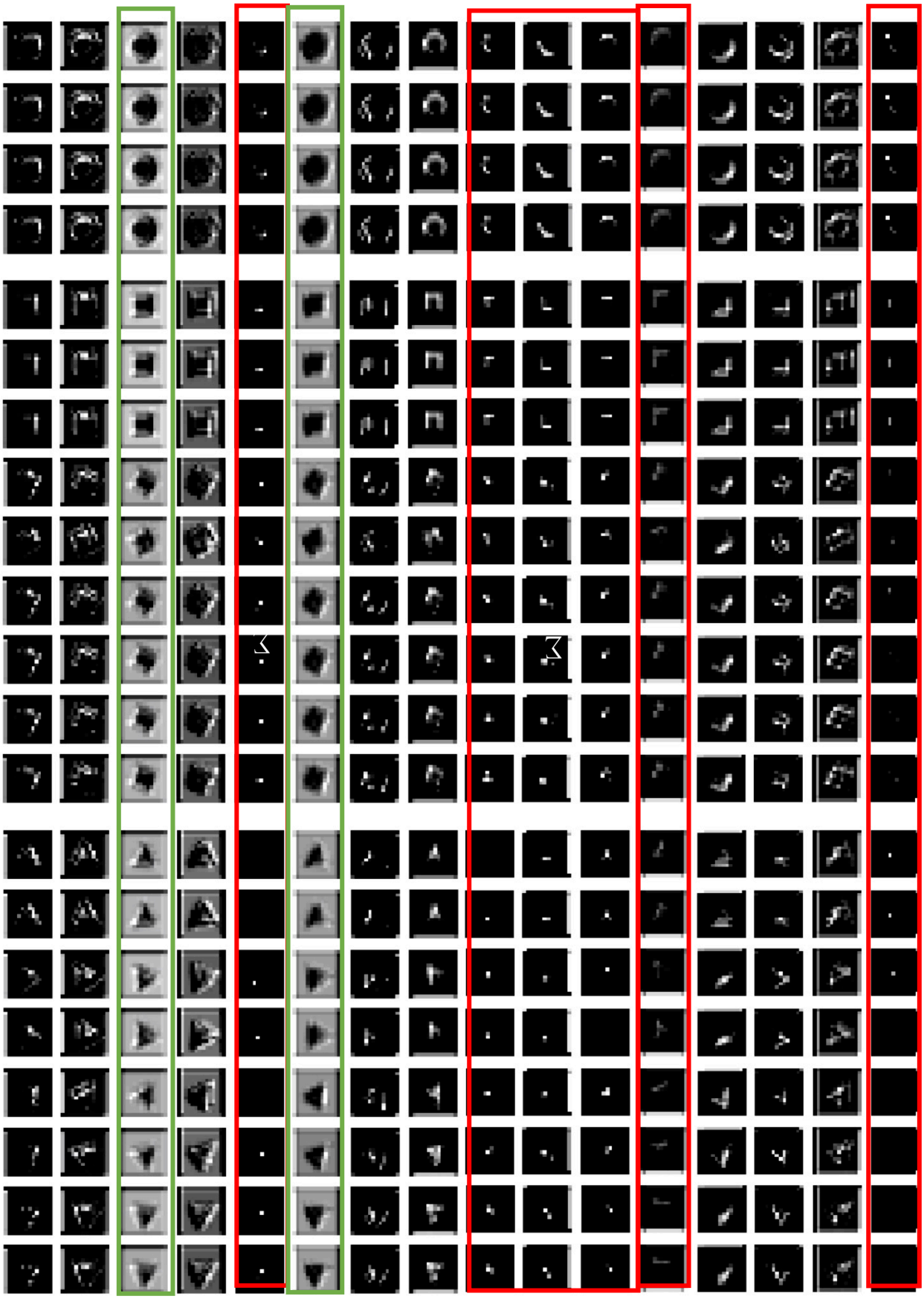


Figura 27 Mapa activación de la segunda capa de 16 filtros, para el modelo 16-16

	ACC% 1	ACC% 2	ACC% 3	ACC% 4	ACC% 5	ACC% 6	ACC% 7	ACC% 8
F 1	0.990333333	0.6185	1.0000	-	-	-	-	-
F 2	0.801	0.947833333	0.8615	0.803	0.697666667	0.794666667	0,6991666667	0,719
F 3	0.881833333	0.790666667	0.750166667	0.765666667	0.777	0.745	0,789	0,7816666667
F 4	0.813	0.443166667	0.333333333	0.333333333	0.333333333	0.333333333	0,3333333333	0,3333333333
F 5	0.982	0.617	0.9655	0.935333333	0.9395	0.8935	0,897	0,8761666667
F 6	0.9935	0.948166667	0.978833333	0.9555	0.959833333	0.945	0,9233333333	-
F 7	0.829333333	0.954	0.859166667	0.887	0.846	0.900166667	0,8898333333	0,8075
F 8	0.953333333	0.985166667	0.9495	0.960833333	0.9375	0.941	0,9185	0,8906666667
F 9	0.937333333	0.9968	-	-	-	-	-	-
F 10	0.924833333	0.654166667	0.9225	0.928333333	0.921666667	0.933833333	0,9265	0,9021666667
F 11	1.0000	-	-	-	-	-	-	-
F 12	0.636333333	0.588833333	0.3625	0.4595	0.3505	0.333333333	0,3556666667	0,348
F 13	0.805	0.771833333	0.714666667	0.722666667	0.722666667	0.712666667	0,7081666667	0,728
F 14	1.0000	-	-	-	-	-	-	-
F 15	0.929833333	0.577	0.5795	0.586333333	0.923333333	0.836666667	0,7825	0,8085
F 16	1.0000	-	-	-	-	-	-	-
F 1	0.848333333	0.525666667	0.990166667	0.9785	0.9838	-	-	-
F 2	0.536333333	0.453	0.473666667	0.489833333	0.4945	0.410166667	0.398333333	0.430333333
F 3	0.354166667	0.407	0.365166667	0.333333333	0.333333333	0.333333333	0.333333333	0.333333333
F 4	0.943	0.6235	0.6635	0.604166667	0.605166667	0.5955	0.59	0.580833333
F 5	1.0000	-	-	-	-	-	-	-
F 6	0.333333333	0.333333333	0.333333333	0.333333333	0.333333333	0.333333333	0.333333333	0.333333333
F 7	0.905833333	0.969	0.952166667	0.982333333	0.962333333	0.9675	-	-
F 8	0.864666667	0.81	0.760833333	0.786	0.782833333	0.725	0.678166667	0.6555
F 9	0.982166667	0.987833333	0.987833333	0.9888	-	-	-	-
F 10	0.956	0.9945	0.937666667	0.9475	0.916	0.943833333	0.9175	0.9167
F 11	1.0000	-	-	-	-	-	-	-
F 12	0.666666667	0.666666667	0.666666667	0.666666667	0.666666667	0.666666667	0.666666667	0.666666667
F 13	0.666666667	0.666666667	0.666666667	0.666666667	0.666666667	0.666666667	0.666666667	0.666666667
F 14	0.966166667	0.573833333	0.609	0.5955	0.602833333	0.5775	0.572166667	0.528
F 15	0.8265	0.524166667	0.744166667	0.788666667	0.777666667	0.707	0.702833333	0.687166667
F 16	1.0000	-	-	-	-	-	-	-

	ACC% 9	ACC% 10	ACC% 11	ACC% 12	ACC% 13	ACC% 14	ACC% 15	ACC% 16
F 1	-	-	-	-	-	-	-	-
F 2	0,6875	0,7231666667	0,6918333333	0,6801666667	0,6983333333	0,6666666667	0,6666666667	0,6865
F 3	0,8493333333	0,886	0,8913333333	0,9146666667	-	-	-	-
F 4	0,6681666667	0,3956666667	0,3918333333	0,399	0,781	0,4523333333	0,4421666667	0,3333333333
F 5	0,9071666667	0,8766666667	0,9431666667	-	-	-	-	-
F 6	-	-	-	-	-	-	-	-
F 7	0,7318333333	0,9411666667	0,9331666667	0,9093333333	0,6438333333	0,7078333333	0,731	0,5373333333
F 8	0,8433333333	0,9683333333	-	-	-	-	-	-
F 9	-	-	-	-	-	-	-	-
F 10	0,8493333333	0,9578333333	0,9138333333	0,8785	0,8803333333	0,546	0,5883333333	0,8071666667
F 11	-	-	-	-	-	-	-	-
F 12	0,3415	0,4145	0,386	0,3495	0,365	0,1165	0,1303333333	0,3333333333
F 13	0,8311666667	0,8673333333	0,8516666667	0,8838333333	0,3333333333	0,3333333333	0,3858333333	0,3333333333
F 14	-	-	-	-	-	-	-	-
F 15	0,8341666667	0,5791666667	0,5841666667	0,526	0,8843333333	0,9268333333	-	-
F 16	-	-	-	-	-	-	-	-
F 1	-	-	-	-	-	-	-	-
F 2	0.461166667	0.500833333	0.510666667	0.483	0.9277	-	-	-
F 3	0.333333333	0.333333333	0.333333333	0.333333333	0.333333333	0.333333333	0.333333333	0.333333333
F 4	0.9282	-	-	-	-	-	-	-
F 5	-	-	-	-	-	-	-	-
F 6	0.333333333	0.333333333	0.333333333	0.333333333	0.333333333	0.333333333	0.333333333	0.333333333
F 7	-	-	-	-	-	-	-	-
F 8	0.616333333	0.751833333	0.7	0.666666667	0.737166667	0.806833333	0.8533	-
F 9	-	-	-	-	-	-	-	-
F 10	-	-	-	-	-	-	-	-
F 11	-	-	-	-	-	-	-	-
F 12	0.666666667	0.666666667	0.666666667	0.666666667	0.666666667	0.666666667	0.666666667	0.666666667
F 13	0.666666667	0.666666667	0.666666667	0.666666667	0.666666667	0.666666667	0.673166667	0.666666667
F 14	0.865833333	0.559666667	0.5795	0.536333333	0.832833333	0.5285	0.51	0.406
F 15	0.728666667	0.4775	0.463666667	0.793666667	0.3795	0.396666667	0.386666667	0.333333333
F 16	-	-	-	-	-	-	-	-

Tabla 9 Poda del modelo 20190816T130514, de filtros 16-16. 16/25 podas.

	ACC% 17	ACC% 18	ACC% 19	ACC% 20	ACC% 21	ACC% 22	ACC% 23	ACC% 24	ACC% 25
F 1	-	-	-	-	-	-	-	-	-
F 2	0,6666666667	-	-	-	-	-	-	-	-
F 3	-	-	-	-	-	-	-	-	-
F 4	0,3333333333	0,569	0,6943333333	0,8868333333	-	-	-	-	-
F 5	-	-	-	-	-	-	-	-	-
F 6	-	-	-	-	-	-	-	-	-
F 7	0,4655	0,6666666667	-	-	-				
F 8	-	-	-	-	-	-	-	-	-
F 9	-	-	-	-	-	-	-	-	-
F 10	-	-	-	-	-	-	-	-	-
F 11	-	-	-	-	-	-	-	-	-
F 12	0,3333333333	0,328	0,5591666667	0,6235	0,5006666667	0,4516666667	0,4516666667	-	-
F 13	0,3333333333	0,5638333333	0,4698333333	0,378	0,4071666667	0,6666666667	0,3333333333		
F 14	-	-	-	-	-	-	-	-	-
F 15	-	-	-	-	-	-	-	-	-
F 16	-	-	-	-	-	-	-	-	-
F 1	-	-	-	-	-	-	-	-	-
F 2	-	-	-	-	-	-	-	-	-
F 3	0.333333333	0.333333333	0.333333333	0.333333333	0.333333333	0.333333333	0.381166667	0.33333333	0.621333
F 4	-	-	-	-	-	-	-	-	-
F 5	-	-	-	-	-	-	-	-	-
F 6	0.333333333	0.333333333	0.333333333	0.333333333	0.333333333	0.558666667	0.333333333	0.7473	-
F 7	-	-	-	-	-	-	-	-	-
F 8	-	-	-	-	-	-	-	-	-
F 9	-	-	-	-	-	-	-	-	-
F 10	-	-	-	-	-	-	-	-	-
F 11	-	-	-	-	-	-	-	-	-
F 12	0.666666667	0.666666667	0.333333333	0.666666667	0.6667	-	-	-	-
F 13	0.666666667	0.666666667	0.666666667	0.666666667	0.666666667	0.6667	-	-	-
F 14	0.4505	0.4515	0.666666667	0.666666667	0.348833333	0.666666667	0.333333333	0.4475	0.7390
F 15	0.333333333	0.424833333	0.8708		-	-	-	-	-
F 16	-	-	-	-	-	-	-	-	-

Tabla 10 Continuación poda del modelo 20190816T130514 de 16-16 filtros. 9/25 podas restantes.

Dados los resultados de la poda para el modelo de 16 filtros en cada capa entrenado con el dataset 2.1 (rotación de imágenes) cabe destacar que, tras 19/25 podas, el porcentaje de acierto resultante es de un 88.63% lo que implica que existe un gran número de filtros que no son necesarios para la clasificación de imágenes. Como ocurrió en el caso anterior en la primera poda existen seis casos en los cuales el resultado de porcentaje de acierto obtenido es del 100%.

Podemos observar en la tabla 11, que en la ultimas podas el filtro que permanecería los filtros 3 y 6 de la segunda capa, el cual durante todas las podas muestra una gran caída de *accuracy* y en la capa 1 el filtro 13. En el caso de los filtros más característicos de la segunda capa, se puede observar que son los filtros que menos transforman la imagen inicial, al contrario de lo que se había pensado inicialmente.

En este caso de estudio, como ocurría en el caso estudio 2.1 entre las podas 18 y 19, la poda realizada en el filtro 15 supone un aumento considerable del porcentaje de acierto. Esto puede ser debido a la detección de filtros que se contradicen en ciertos casos.

- Análisis modelo **8-4** filtros, entrenada con el dataset 2.1

PRIMERA CAPA

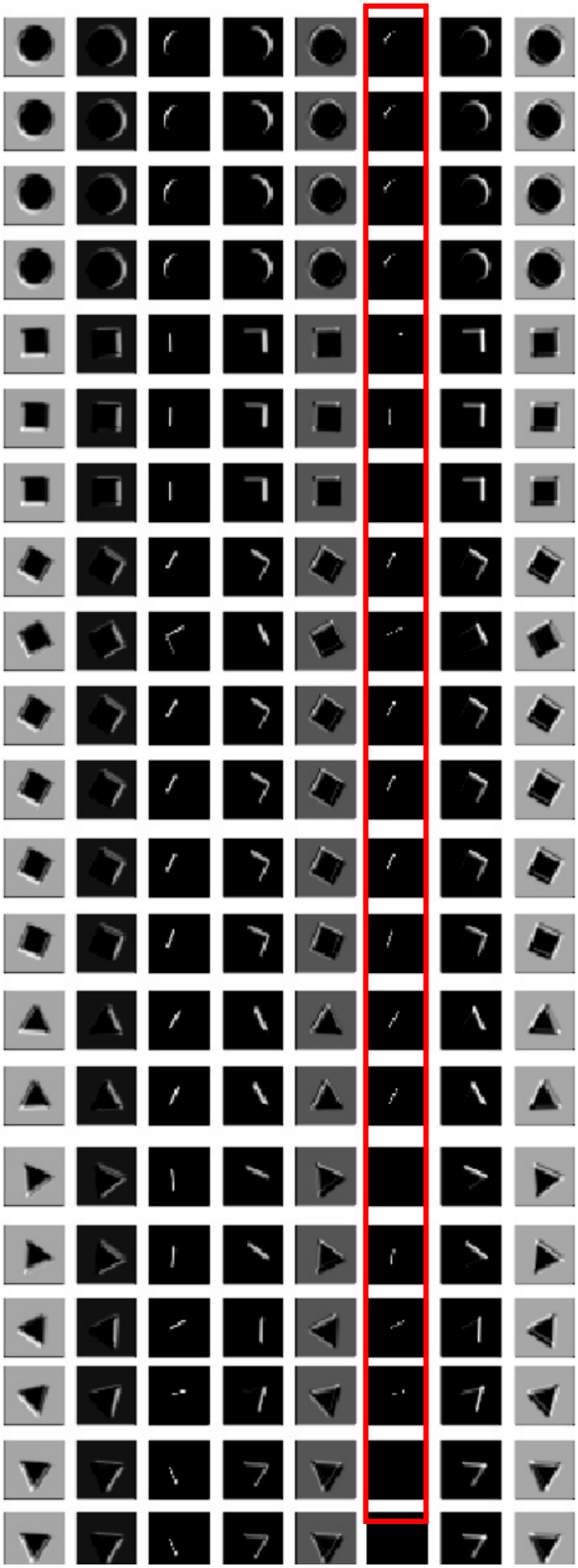


Figura 28 Mapa de activación de la primera capa de 8 filtros del modelo 8-4.

SEGUNDA CAPA

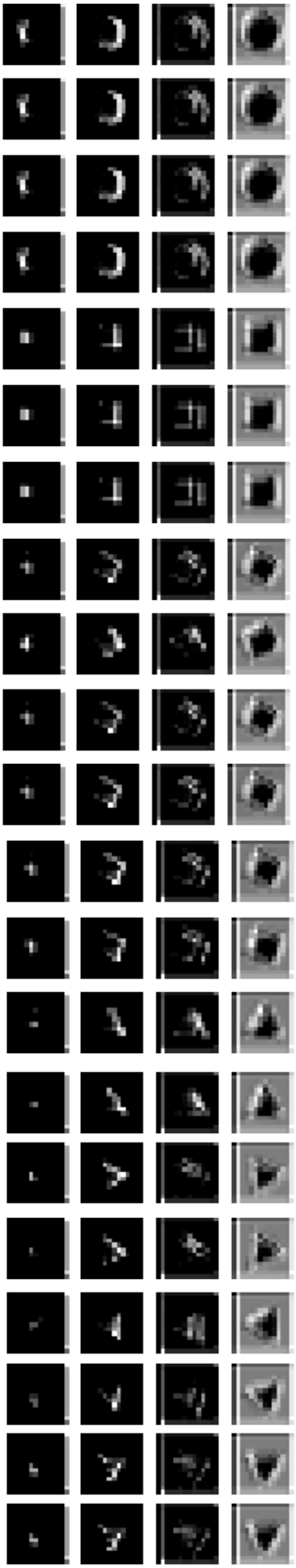


Figura 29 Mapas de activación de la segunda capa de 4 filtros del modelo 8-4

	ACC% 1	ACC% 2	ACC% 3	ACC% 4	ACC% 5	ACC% 6	ACC% 7	ACC% 8	ACC% 9	ACC% 10
F 1	0,4213333333	0,4166666667	0,4973333333	0,4556666667	0,5958333333	0,6035	0,434	-	-	-
F 2	0,3333333333	0,3333333333	0,3333333333	0,4443333333	0,6085	0,6883333333	-	-	-	-
F 3	0,6146666667	0,9495	0,6663333333	0,6666666667	0,6666666667	-	-	-	-	-
F 4	0,6666666667	0,6666666667	0,6666666667	0,3333333333	0,3333333333	0,3333333333	0,3855	0,7436666667	-	-
F 5	0,3735	0,3716666667	0,3716666667	0,6666666667	-	-	-	-	-	-
F 6	0,9973333333	-	-	-	-	-	-	-	-	-
F 7	0,4748333333	0,4743333333	0,4878333333	0,3333333333	0,3333333333	0,3333333333	0,3333333333	0,1348333333	0,3468333333	0,3405
F 8	0,4106666667	0,4101666667	0,7753333333	0,368	0,3458333333	0,3506666667	0,4048333333	0,3333333333	0,4031666667	0,4296666667
F 1	0,6666666667	0,6666666667	-	-	-	-	-	-	-	-
F 2	0,3828333333	0,3828333333	0,3908333333	0,3333333333	0,3333333333	0,3333333333	0,3333333333	0,3333333333	0,01766666667	-
F 3	0,4648333333	0,4791666667	0,5006666667	0,4248333333	0,422	0,4216666667	0,4406666667	0,5325	0,4098333333	-
F 4	0,3333333333	0,3333333333	0,6666666667	-	-	-	-	-	-	-

Tabla 11 Poda del modelo 20190809T202124 de 8-4 filtros en las capas.

En este último caso de estudio para el dataset 2.1, el número de filtros totales es menor y es por ello por lo que los resultados obtenidos muestran que hay un mayor número de filtros representativos. En la tabla 12 se puede observar que en la primera poda se puede observar que en la primera capa el filtro 6 aporta al aprendizaje de la red. Efectivamente podemos comprobar en la figura 28 que este filtro prácticamente no activa la forma de las figuras geométricas.

Si embargo, cabe resaltar al mismo tiempo el incremento de *accuracy* que ocurre en la poda del filtro 4 tras la poda de filtro 1 de la primera capa. Como ha ocurrido en otras ocasiones, parece existir cierta interacción entre dichos filtros puesto que la aportación que este filtro hacía de importancia alta desaparece al podar el filtro 1.

Como podemos observar en la tabla 12 los cuatro últimos filtros que forman segunda capa, desde el principio todos los filtros presentan una gran caída de porcentaje de acierto en la clasificación tras su poda. Sin embargo, cabe resaltar en considerable incremento de caída en el filtro 3 tras la poda de filtro 4 de la primera capa.

CAPITULO 4

INTERPRETACIÓN DE RESULTADOS.

Interpretación de resultados.

• Caso 1. Análisis de los mapas de activación del caso 16-4, entrenado con dataset 1.3.

Es posible observar varias similitudes en el análisis de la primera capa de este modelo.

Partiendo por el primer filtro, es posible observar grandes similitudes los filtros dos, cuatro y seis, puesto que todos ellos activan los bordes de la zona izquierda de las imágenes. Sin embargo, es posible observar diferencias relevantes en la activación de éstos. La activación del contorno de la figura en el caso de las elipses es prácticamente la misma todos estos filtros. Sin embargo, en el caso de los triángulos existen diferentes activaciones. El primer y segundo filtro activan prácticamente la forma total del contorno de esta figura geométrica, mientras que los filtros 4 y 6 activan el lado izquierdo del triángulo equilátero. Además, considerando las activaciones de los rectángulos se puede observar que los filtros cuatro y seis activan únicamente el lado lateral izquierdo del rectángulo al igual que ocurría con los triángulos. En el caso de los filtros 1 y 2 la activación de los rectángulos parece resaltar principalmente una de las esquinas de la figura. En el primer filtro se activa la esquina inferior izquierda mientras que en el segundo caso activa la esquina inferior derecha con ciertas dificultades según decrece el tamaño de los rectángulos.

Si se observa el comportamiento de los filtros 8, 9, y 15 la parte activada de las imágenes en estos casos es la superior. Las elipses, al igual que en todos los filtros, muestran una activación similar en los tres casos, siendo la parte activada la parte superior de la figura. En estos casos la parte activada de los rectángulos coincide en todos los casos y es la parte superior derecha, activando una de las esquinas de la figura, lo que puede ser una clara característica para diferenciar los rectángulos de las otras figuras. Es más, si observamos los resultados ofrecidos por la poda de estos filtros se puede observar que principalmente el filtro 8 es muy representativo en la óptima clasificación del modelo, dado que la reducción de *accuracy* es muy elevada. La poda de este filtro se realiza en la 13/18 podas, lo que implica que pertenece a los 6 filtros más representativos de la red. El hecho de que sea el filtro 8 en vez de uno de los otros tres filtros, dados los resultados obtenidos en los filtros 3,5,10,11,13 y 14, los cuales no han cambiado a negativo la imagen y muestran una gran relevancia en. los resultados de la red, puede ser ya que este filtro no llega a pasar a negativo de forma completa la imagen.

Considerando la dificultad añadida de tratar de analizar de forma visual los filtros de la segunda capa, cabe resaltar las activaciones nulas en el primer, segundo y cuarto filtro. Sin embargo, las activaciones en el filtro 3 parecen ser claras para una optima diferenciación de las figuras geométricas. Observando la **tabla 7**, los resultados obtenidos a lo largo de todo el proceso de poda en el filtro 3 de la segunda capa, se puede observar en la mayoría de casos una reducción de porcentaje de acierto del 70%, y es finalmente, acorde con lo observado, el filtro más representativo de la capa.

- **Caso 2. Análisis de los mapas de activación del caso 2-16, entrenado con dataset 1.3.**

Primeramente, se ha procedido a observar los resultados obtenidos con el método de la poda ya que los resultados en este caso son muy llamativos. Se han realizado un total de 16 podas en 14 pasos ya que tres podas han sido realizadas al mismo tiempo en el primer paso al no mostrar ningún aprendizaje. Esto se puede observar en la **tabla 8.** Además, lo más resaltante de los resultados obtenidos es el hecho de que hasta el paso 13 del método, en el cual 15 filtros habían sido podados, se continúa manteniendo un valor de *accuracy* superior al noventa por ciento. Esto implica que el aprendizaje de ciertos filtros es nulo, o redundante por el hecho de tener filtros que alcanzan un nivel de diferenciación prácticamente perfecto.

Una vez tomada conciencia de los resultados obtenido en la poda de filtros se ha procedido a analizar de forma visual los mapas de activación, que se encuentran en las figuras 24, y 25.

En la primera capa se puede observar las claras activaciones de las partes características de las figuras. Pero es llamativa la activación generada por el segundo filtro en las figuras, ya que por primera vez todo el contorno de las figuras o prácticamente todo esta siendo activado. Efectivamente coincide con el último filtro podado seguido del primero, lo que implica una clara dependencia de la primera capa para la clasificación.

El análisis visual en la segunda capa resulta más complicado de realizar dado que los patrones extraídos por el filtro se activan de forma confusa, ya que la imagen de entrada de esta capa es la salida de la primera capa. Lo que si se puede observar en esta capa son los distintos grados de activación, otra característica muy relevante para que el filtro sea representativo.

Cuanto mayor sea la activación, más sencillo será para la red realizar la clasificación gracias a dicho filtro.

- **Caso 3. Análisis de los mapas de activación del caso 16-16, entrenado con dataset 2.1.**

De nuevo, se ha procedido a observar los resultados obtenidos con el método de la poda en primer lugar. En este caso se han llevado a cabo 25 pasos de poda, y un total de 30 podas. Hasta el paso 16 de la poda con un total de 21 podas se mantiene un decrecimiento constante de la *accuracy* manteniendo su valor superior al 80%. Esto implica que estos 21 filtros no parecen ser representativos para una óptima clasificación. Sin embargo, hay un factor claramente llamativo en los resultados. En el paso 18 de la poda existe un repentino incremento de la *accuracy* pasando de un 66.67% a un 87.08%. Esto implica que existe cierta interacción entre filtros y puede implicar cierta contradicción entre los patrones detectados por ambos.

Analizando los mapas de activación de la primera capa representado en la **figura 26**, se puede observar como existen en los cuales la activación de la figura es prácticamente total mientras que en otros casos solo partes de las imágenes están activadas o ninguna.

En el **filtro 8** es llamativa la comparación entre la activación de los rectángulos y los triángulos ya que en algunos casos solos activa un lado de las figuras. Dado que estas figuras están rotadas, dichos lados horizontales pueden llevar a la red a confusión entre estas dos figuras.

Dado que la variación en este dataset es la rotación parece necesario para red la activación de las esquinas de las figuras geométricas. Otro caso que podría llevar a confusión entre rectángulos y triángulos se da en los filtros 1,7, 9 y 11.

El último filtro de la primera capa muestra una activación prácticamente nula y es efectivamente uno de los filtros podados en el primer paso en el método de podas por suponer ninguna reducción en el porcentaje de acierto en la predicción de figuras.

En la segunda capa existen un gran número de filtros que muestran una activación insuficiente para la diferenciación de figuras como ocurre en los filtros 1, 5,9, 10, 11, 12 y 16. El filtro 13 parece activar los lados de la parte derecha de la imagen. Esto supone en los rectángulos y triángulos la activación de solo un lado por lo que puede llevar a la confusión

explicada en la primera capa. Observando la tabla 10, en los primeros 10 pasos del método de podas se lleva a cabo la poda de los filtros especificados por lo que el análisis llevado a cabo a partir de los mapas de activación parece ser coherente con los resultados de la poda.

- **Caso 4. Análisis de los mapas de activación del caso 16-16, entrenado con dataset 2.1.**

Dados los resultados obtenidos de la en el caso anterior, se ha concluido que para la correcta clasificación de las figuras rotadas, la red necesita de la activación de un mínimo de dos lados en el caso de triángulos y rectángulos para no caer en ciertas confusiones.

Si observamos los mapas de activación de la primera capa representada en la figura 29, podemos observar que existen varios filtros en los que la activación de rectángulos y triángulos consiste en un solo lado como es el caso del filtro 3,4 y 6. En cambio, el resto de los filtros activa un mínimo de dos lados. Si observamos los resultados de la poda observamos que a pesar de ser prácticamente la mayoría de los filtros necesarios para mantener una *accuracy* mayor del 80%, las reducciones de porcentaje de acierto menores se dan para los filtros indicados, lo que implica que son los menos representativos en la clasificación.

Respecto a la segunda capa, se puede observar la coherencia entre los resultados de la poda y la observación de los mapas de activación en la figura 29, ya que en el caso del primer filtro las activaciones son prácticamente la mismas en todas las figuras, mientras que el resto de los filtros activan esquinas o incluso más partes de las figuras, lo que hace que sean muy representativos.

CAPITULO 5

CONCLUSIONES Y FUTUROS DESARROLLOS.

Conclusiones y futuros desarrollos

Conclusiones

En vista global de los cuatro casos estudio desarrollados, y realizando un análisis del aprendizaje realizado por la red, cabe resaltar ciertas conclusiones a las que se han llegado.

Primeramente, se ha observado cierta similitud en las matrices *kernel* obtenidas en las redes tras el entrenamiento de esta. Esto se debe a que la aleatoriedad de valores adjudicados a dichas matrices puede darse de tal que los valores sean similares, y por tanto, tras el entrenamiento las activaciones y los valores de dichas matrices se actualicen tal que activen más dichas partes. Y esto resulta en que los patrones detectados para la posterior clasificación son similares. Como hemos podido comprobar en los casos estudio, principalmente en el caso de estudio 8.1.1, la poda de un filtro que tiene similares, no afecta mucho a la red dado que el patrón detectado por dicho filtro ya forma parte de la red.

Además, cabe resaltar lo que se ha comentado a lo largo de todos los casos estudio, y es la posible interacción entre filtros. En ocasiones, la poda de un filtro afecta drásticamente a como el patrón detectado por otro filtro afecta en la clasificación. Eso implica que para un análisis profundo de una red, no solo habría que realizar la poda filtro a filtro y eliminar el que muestre menor *accuracy*, sino que habría que analizar al mismo tiempo el efecto que dicho filtro tiene sobre otros. En el caso en que la reducción de *accuracy* quiere decir que el filtro podado anteriormente era una confusión para este, pues que dichos filtros pueden contradecirse en la predicción de una imagen.

Por último, hemos comprobado que tanto el método de podar la red filtro a filtro y observar el efecto en el porcentaje de acierto en la clasificación y el método de visualización de los mapas de activación sugieren acciones similares. Cuando se observa un filtro que no activa partes representativas de las figuras, o directamente no activa nada para ciertas imágenes, dicho filtro mostrará en su poda una reducción mínima de *accuracy* por lo que es posible realizar la poda de una red convolucional de forma visual en las capas altas. En

el caso de la segunda capa de las redes, este método sería más complicado de utilizar puesto que la visualización de patrones en estas capas es mucho más confusa, y sería difícil determinar las partes activadas. Solo en determinados casos que hemos visto ha sido clara la visualización de filtros de que aporten aprendizaje mínimo o nulo en la segunda capa.

- ## Futuros desarrollos

Una parte del proyecto consiste en el análisis de la sensibilidad de la red realizando la poda de la red para obtener aquellos filtros más significativos, pero este no es el único método que se ha estudiado para alcanzar este fin. Otra opción es realizar el análisis con el método L1-norm. Este método consiste en sumar el valor absoluto de los valores de los pesos de los filtros kernel puesto que cuantos mayores sean los pesos, mayor es la activación en los mapas de activación. [23] El resultado de utilizar este método es conocer que filtros deben ser podados para alcanzar una red más rápida y eficiente. Este proceso ha sido comenzado a lo largo de este proyecto, pero no de forma tan profunda como el método basado en la poda de cada filtro como se ha mostrado anteriormente. La dificultad encontrada a la hora de utilizar este método es que la forma en que el valor de los pesos de la matriz kernel era devuelta no era como vector, sino como *string* por lo que hubo que las opciones ofrecidas por la librería pandas para los *numpy arrays* . Finalmente se encontró la forma de tener los valores como vector de otra manera.

Método inicial:

```
layer1=np.asarray(layer1_filters[0])

layer2=np.asarray(layer2_filters[0])

weights=np.array([])
for i in range(len(layer1)):
for j in range(len(layer1[i])):
string= str(layer1[i][j])
string= remove_char(string)
string= string.split()
myarray=np.asarray(string)
print(len(myarray))
weights= np.append(weights,[[myarray]])
print(weights)

  #print(weights)
  #print(len(myarray))
  #print(len(weights))
rango=len(weights)/len(myarray)
  #print(rango)
```

```
sumfil=([])
suma=np.array([])
for i in range(len(myarray)):
  for j in range(9):
    weight= weights[len(myarray)*j+i]
    suma=np.append(suma,weight)
    #print(suma)
    #print(weights[len(myarray)*j])
  print(suma)


  #L_1=np.sum(suma, dtype=np.float64)
  #sumfil=np.concatenate((sumfil, suma), axis=0)
  suma=[]
```

Y el mejor método para obtener los pesos que facilita el proceso es:

```
layer1_filters=cnn.plot_filters(layer_name='conv2d_47')
layer2_filters=cnn.plot_filters (layer_name='conv2d_48')
print(layer1_filters[0][:,:,:,0])
```

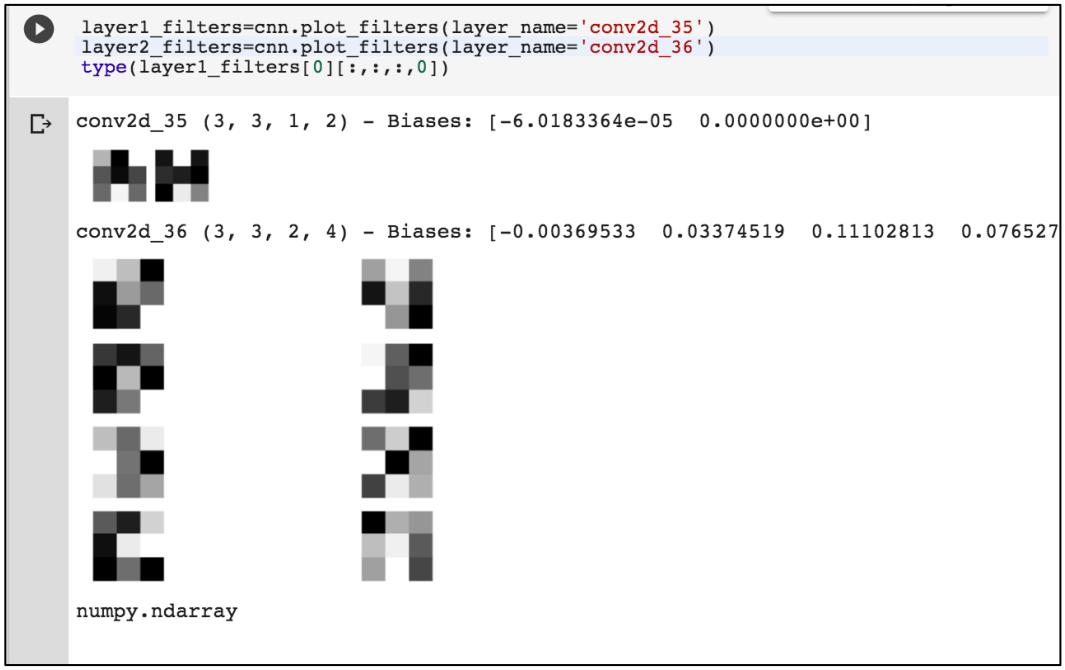


Figura 30 Obtención de los pesos para la aplicación del método L1-norm

Como podemos observar en la figura 30, esta segunda forma de obtener los pesos los devuelve directamente como un *np.ndarray* lo cual facilita todo el proceso mostrado anteriormente, puesto que el paquete Numpy de Python permite operar matemáticamente los valores de los vectores de este tipo de forma sencilla.

Por otra parte, un problema que se ha tratado desde el surgimiento de las redes neuronales es que la red puede dejar de aprender por el hecho de haber caído en un mínimo

local. En aquellos casos en los que la red consta de muchas capas, el número de parámetros es tan elevado que solventa por si solo este problema. Pero en el caso de nuestro proyecto, en el cual la clasificación es muy simple por el hecho de tratarse de figuras geométricas, sería necesario tratar este problema, o al menos comprobar si esta afectando. El hecho de que siempre deje de aprender para unos mismos valores de *accuracy* puede significar que se ha quedado en un mínimo local. Conocer bien la razón de que la red deje de aprender sería de gran ayuda para la continuación del análisis de redes neuronales.

Otra observación que se ha podido obtener de este proyecto es el hecho de que, debido a la aleatoriedad de valores en los pesos de los filtros, y la posterior realimentación de los valores con el entrenamiento, hace que en muchas ocasiones existan varios filtros en una misma red que activen las mismas partes, es decir que los filtros son parecidos. Sin embargo, unos activan más las partes diferenciantes de las imágenes y otros menos. Un posible método para evitar la similitud entre filtros sería realizar una comparación de las matrices de los kernel de forma automática mediante métodos matemáticos con el fin de eliminar aquellas en las que los valores de las matrices tengan un mínimo de diferencia.

Por otra parte, tras haber conocido en mayor profundidad las posibles influencias de los factores comentados anteriormente, una posible vía para continuar este proyecto e incrementar el conocimiento de las CNN sería continuar realizando variaciones en los datasets, como incluir deformaciones en las imágenes, en las proporcionalidades, etc.

Bibliografía

[1] D. H. Hubel and T. N. Wiesel, "Receptive fields of single neurones in the cat's striate cortex," *The Journal of Physiology,* vol. 148, no. 3, p. 574–591, Oct 1959.

[2] Y. LeCun, L. Bottou, Y. Bengio and P. and Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE,* vol. 86, no. 11, p. 2278–2324, 1998.

[3] A. Krizhevsky, l. Sutskever and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *NIPS'12 Proceedings of the 25th International Conference on Neural Information Processing Systems*, 2012.

[4] "Keras: The Python Deep Learning library," [Online]. Available: https://keras.io/. [Accessed 04 05 2019].

[5] Google, "TensorFlow," [Online]. Available: https://www.tensorflow.org/. [Accessed 04 05 2019].

[6] Google, "Google Colab," [Online]. Available: https://colab.research.google.com/. [Accessed 04 05 2019].

[7] Briega, R. E. (2016, August 2). Redes neuronales convolucionales con TensorFlow. "https://relopezbriega.github.io/blog/2016/08/02/redes-neuronales-convolucionales-con-tensorflow"

[8] Das, S., & Das, S. (2017, November 16). CNN Architectures: LeNet, AlexNet, VGG, GoogLeNet, ResNet and more .... Retrieved from https://medium.com/@sidereal/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5

[9] Na, P. (2019, May 16). Qué es overfitting y underfitting y cómo solucionarlo. Retrieved from http://www.aprendemachinelearning.com/que-es-overfitting-y-underfitting-y-como-solucionarlo/

[10] A. K., I. S., & G. E. (2015, May 18). ImageNet Classification with Deep Convolutional Neural Networks. Retrieved May 22, 2019, from http://vision.stanford.edu/teaching/cs231b_spring1415/slides/alexnet_tugce_kyunghee.pdf

[11] Zeiler, M. and Fergus, R. (2019). Visualizing and Understanding Convolutional Networks. [online] arXiv.org. Available at: https://arxiv.org/abs/1311.2901 [Accessed 24 Jun. 2019].

[12] He, K., Zhang, X., Ren, S. and Sun, J. (2019). Deep Residual Learning for Image Recognition. [online] arXiv.org. Available at: https://arxiv.org/abs/1512.03385 [Accessed 24 Jun. 2019].

[13] Olaode, Abass & Naghdy, Golshah & Todd, Catherine. (2014). Unsupervised Classification of Images: A Review. International Journal of Image Processing. 8. 2014-325.

[14] Bau, D., Zhou, B., Khosla, A., Oliva, A. and Torralba, A. (2019). Network Dissection: Quantifying Interpretability of Deep Visual Representations. [online] arXiv.org. Available at: HYPERLINK "https://arxiv.org/abs/1704.05796" https://arxiv.org/abs/1704.05796

[15] Zhang, Q., Wu, Y. and Zhu, S. (2019). Interpretable CNNs. [online] Arxiv.org. Available at: https://arxiv.org/pdf/1901.02413.pdf

[16] J. B. (2019, April 06). ¿Cómo funcionan las Convolutional Neural Networks? Visión por Ordenador. Retrieved from http://www.aprendemachinelearning.com/como-funcionan-las-convolutional-neural-networks-vision-por-ordenador/

[17] Agarwal, M., & Agarwal, M. (2017, December 14). Back Propagation in Convolutional Neural Networks - Intuition and Code. Retrieved from https://becominghuman.ai/back-propagation-in-convolutional-neural-networks-intuition-and-code-714ef1c38199

[18] Shah, A. (2019). Model Pruning in Keras with Keras-Surgeon. https://medium.com/@anuj_shah/model-pruning-in-keras-with-keras-surgeon-e8e6ff439c07

[19] BenWhetton/keras-surgeon. (2019). https://github.com/BenWhetton/keras-surgeon

[20] Fuat. (2018). Google Colab Free GPU Tutorial. https://medium.com/deep-learning-turkey/google- colab-free-gpu-tutorial-e113627b9f5d

[21] https://www.flir.es/discover/iis/machine-vision/collect-image-data-train-a-neural-network-and-deploy-on-an-embedded-system/

[22] S. Wu et al., "$L1$ -Norm Batch Normalization for Efficient Training of Deep Neural Networks," in IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 7, pp. 2043-2051, July 2019. doi: 10.1109/TNNLS.2018.2876179

[23] Li, Hao & Kadav, Asim & Durdanovic, Igor & Samet, Hanan & Graf, H.P.. (2016). Pruning Filters for Efficient ConvNets

[24] https://www.geeksforgeeks.org/numpy-array-creation/

[25] Calvo, D. Red Neuronal Convolucional. CNN. http://www.diegocalvo.es/red-neuronal-convolucional/

[26] https://engmrk.com/lenet-5-a-classic-cnn-architecture/

[27] AlexNet - ImageNet Classification with Convolutional Neural Networks. (2019). from https://neurohive.io/en/popular-networks/alexnet-imagenet-classification-with-deep-convolutional-neural-networks/

[28] Review: ZFNet — Winner of ILSVRC 2013 (Image Classification). (2019). Retrieved 25 August 2019, from https://medium.com/coinmonks/paper-review-of-zfnet-the-winner-of-ilsvlc-2013-image-classification-d1a5a0c45103

[29] Review: GoogLeNet (Inception v1)— Winner of ILSVRC 2014 (Image Classification). (2019). https://medium.com/coinmonks/paper-review-of-googlenet-inception-v1-winner-of-ilsvlc-2014-image-classification-c2b3565a64e7

[30] Deep Learning básico con Keras (Parte 3): VGG. (2019). Retrieved 25 August 2019, from https://enmilocalfunciona.io/deep-learning-basico-con-keras-parte-3-vgg/

[31] Sun, L. (2019). ResNet on Tiny ImageNet, Stanford University https://pdfs.semanticscholar.org/8bb8/136b453163fa99a8fba86a640b694db55184.pdf

[32] K., Trainor, E., & Doe, J. (2019). Kernel sizes for multiple convolutional layer neural networks.. https://stats.stackexchange.com/questions/340258/kernel-sizes-for-multiple-convolutional-layer-neural-networks

[33] Discriminative Unsupervised Feature Learning with Exemplar Convolutional Neural Networks - IEEE Journals & Magazine. (2019). https://ieeexplore.ieee.org/abstract/document/7312476/

[34] Erhan, Dumitru & Bengio, Y & Courville, Aaron & Vincent, Pascal. (2009). Visualizing Higher-Layer Features of a Deep Network. Technical Report, Univeristé de Montréal.

[35] Brownlee, J. (2019). A Gentle Introduction to the Rectified Linear Unit (ReLU). Retrieved 28 August 2019, from https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/

[36] Dosovitskiy, Alexey & Brox, Thomas. (2016). Inverting Visual Representations with Convolutional Networks. 4829-4837. 10.1109/CVPR.2016.522.

CAPITULO 6

ANEXOS.

Anexos

Datasets

En este proyecto se hace referencia a los datasets cuando se habla de las imágenes de entrada que se introducen en la red convolucional para entrenar la red y proceder a la clasificación de las imágenes. Los datasets utilizados constan de dos partes, una llamada *training* y otra llamada *validation*. Hay el doble de imágenes de *training* que de *validation.* Las imágenes de la parte de *training* son las utilizadas para el entrenamiento de la red. La red realiza todos los entrenamientos con dichas imágenes y va actualizando el valor de las matrices de los kernel de cada filtro. Una vez el entrenamiento ha finalizado se hace uso de las imágenes de *validation* para comprobar con unas imágenes nuevas, es decir, no conocidas por la red, si la clasificación se realiza con el mismo acierto que con las imágenes de entrenamiento. Es por ello que en algunos casos el porcentaje de acierto obtenido de las imágenes de *validation* es menor que el obtenido de las imágenes de *training*.

En este proyecto han entrado en juego tres datasets distintos, el 1.2, el 1.3, y el 2.1 (así son los nombres que reciben a lo largo del proyecto). Todas imágenes tienen un tamaño de 28x28 y muestran una figura geométrica y las distintas figuras que podemos encontrar son triángulos, rectángulos y triángulos.

Los datasets 1.2 y. 1.3 constan de variaciones de tamaño, mientras que el dataset 2.1 esta formado por las figuras geométricas de tamaño constante, rotadas.

En el inicio del proyecto, los entrenamientos fueron realizados con el dataset 1.2, y analizando las imágenes de fallo en aquellos casos en los que la red había aprendido, pero tenia un mínimo porcentaje de fallo observamos la red siempre fallaba en el reconocimiento de las mismas imágenes, lo que indicaba que los menores tamaños de las figuras contenidos en el datasets eran irreconocibles para la red. El análisis llevado a cabo se puede observar más adelante en la tabla 13.

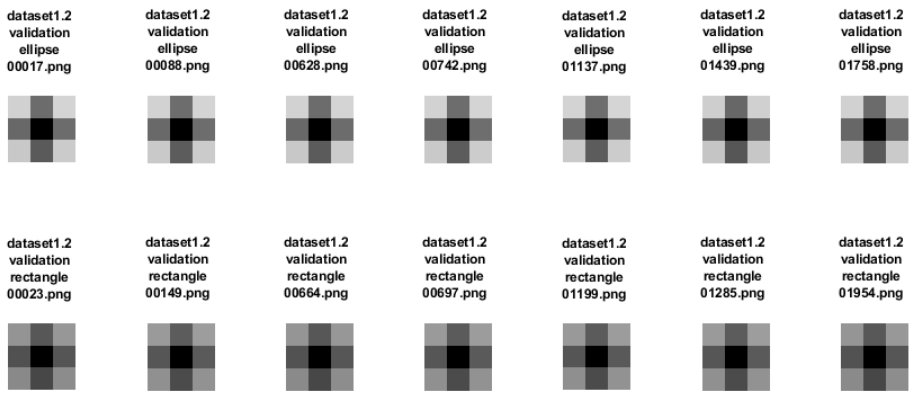


Figura 31 Imágenes irreconocibles para la red del dataset 1.2

En la figura 30 se puede comprobar que las imágenes generadas de menor tamaño pierden incluso la forma de la figura geométrica, y son por lo tanto irreconocibles incluso para el ojo humano. Es por ello por lo que se aumenta el tamaño mínimo de las imágenes y se crea el dataset 1.3. Los limites del radio de las elipses, que anteriormente oscilaban entre 0.1 a 0.9, este nuevo dataset oscilan entre 0.3 y 0.9.

CASO	IMÁGENES FALLO	OBSERVACION DE IMÁGENES
En dos casos 2-32 (30 épocas),2-8 (30 épocas), 16-4 (10 épocas) fallo en la predicción de ellipses. En el caso de 2-32 la predicción es de rectángulos mientras que en los otros dos casos la predicción son triángulos. Las imagenes de fallo son coincidentes en los tres casos, con alguna imagen más en el caso de 16-4.	17,88, 628, 742, 1137, 1439, 1758.	Las imágenes coinciden en ser todas elipses del tamaño más reducido
Hay dos casos tres casos con predicción erronea de rectangulos. En los casos 16-4 (10 épocas), 16-4 (10 épocas, tercer entrenamiento), 4-2 (10 épocas, quinto entrenamiento). Es por ello que vamos a analizar solo las imagenes del casos 16-4 entrenada repetidas veces, y el modelo 4-2. Las imagenes son coincidentes como en el caso anterior. Las imágenes de fallo principalmente son	23, 149, 664,697, 1199,1285, 1954	Las imágenes coinciden en ser todas rectángulos del tamaño más reducido

Tabla 12 Análisis de imágenes fallidas del dataset 1.2

El dataset 2.1 formado por figuras centradas de tamaño radio=0.7 como en el dataset 1.2 y con una rotación al azar. La rotación se realiza respecto al centroide de la figura. A continuación pu

Google Colaboratory

Google Colaboratory es un servicio gratuito para trabajar en la nube basado en los Jypiter Notebooks. Permite el acceso a tiempo de ejecución configurado para aprendizaje profundo y *machine learning* y acceso gratuito a una GPU de mayor tamaño que la de ordenadores, lo que reduce los tiempos de ejecución. Los Jypiter Notebooks son servicios abiertos basados en la búsqueda de herramientas, que permiten a partir de diversas librerías que han sido creadas y diversos lenguajes trabajar tanto de forma local como en la nuevo.

Google Colab es una función creada con el fin de ser una herramienta para el estudio e investigación de la inteligencia arficial y *machine learning.* Ofrece dos opciones a la hora de trabajar en los Notebooks, Python 2 o Python 3 *runtimes*, con una preconfiguración previa que contiene las librerías más utilizadas en inteligencia artificial como TensorFlow, Matplotlib o Keras.

El uso de la GPU a través de Google Colaboratory es de uso personal hasta que la maquina virtual este desactivada durante un periodo de tiempo donde la sesión queda suspendida y solo se conserva el. Notebook. Sin embargo, es fácil guardar los datos desde los notebooks en una cuenta de Google Drive.

Respecto a la literatura que se puede encontrar sobre el uso de Google Colaboratory esta formada principalmente por tutoriales online basados en los documentos oficiales de Google., sin embargo, el tutorial más conocido es el creado por Faut [20].

Tras estudios experimentales y análisis del hardware de Google Colaboratory se ha demostrado que la velocidad que ofrece ejecutar programas no solo de *Deep learning* sino cualquier otra aplicación que requiera el uso principal de una CPU, es 20 veces mayor que la de una CPU core física. La capacidad que Google Colaboratory ofrece de forma gratuita es más que suficiente para poder trabajar en proyectos de gran escala.

Además, en caso de querer trabajar en local, la transferencia de todos de un Colab Notebook a Jupyter es instantánea. Un Colab Notebook se puede correr sin ningún problema.

Respecto a la infraestructura de la pagina web de Google Colaboratory es sencilla y rápida, especialmente cuando los datos de trabajo están almacenados en servicios pertenecientes a Google. Por ejemplo, trabajar con datasets contenidos en un Google Drive resulta realmente sencillo y veloz.

Otra aplicación que podría tener Google Colaboratory es la enseñanza dado que para el aprendizaje de aplicaciones especificas como puede ser redes neuronales en *Deep learning* el uso de los servicios de Google Colaboratory evita la necesidad que la instalar drivers, compiladores y las diversas configuraciones necesarias para preparar el entorno de trabajo. Estos pasos suelen ser complicado para personas inexpertas en el área.