

MEMORIA DEL TRABAJO DE FIN DE GRADO

**SELECCIÓN DE CANALES EN
REDES NEURONALES
CONVOLUCIONALES
MEDIANTE DROPOUT**

**GRADO EN INGENIERÍA
ELECTRÓNICA INDUSTRIAL Y
AUTOMÁTICA**

Nombre del alumno: Miguel Padrón González

Tutor Académico: José Francisco Sigut Saavedra

AGRADECIMIENTOS

En primer lugar, querría agradecer a mi familia y a todos mis amigos por haberme apoyado en todo, y confiar en mí desde el principio de mi viaje en una ingeniería. Me gustaría hacer mención especial a mi pareja Claudia Sosa Gutiérrez que en los peores momentos siempre me ayudó a continuar para adelante.

Por supuesto, querría agradecer a mis dos inseparables compañeros del grado Jesús Elías Soto y Daniel Pérez Gutiérrez con los que he compartido innumerables momentos. Desde periodos enteros de exámenes estudiando juntos, hasta las mejores risas y juegos para que fuese más ameno el camino.

Por último, más en relación al proyecto final de grado querría agradecerle a mi amigo de infancia Carlos Velázquez Pérez por generar la semilla de la programación y el Deep Learning en mí. Y a mi tutor, José Francisco Sigut Saavedra que me ha ayudado en todo momento y sin él, este proyecto jamás hubiera sido posible.

RESUMEN

En la actualidad, la tecnología juega un papel fundamental en la medicina, consiguiendo grandes avances sobre todo en el diagnóstico de enfermedades gracias a la inteligencia artificial.

En este proyecto, he intentado añadir mi propia contribución a una investigación que se lleva a cabo desde hace años en la ULL y de la cual forma parte mi tutor. Dicha investigación tiene como objetivo el diagnóstico prematuro del glaucoma, una enfermedad con una prevalencia significativa que puede llegar a acabar en una ceguera permanente si no se interviene a tiempo. Para este fin, se está trabajando con redes neuronales convolucionales que constituyen el núcleo de este trabajo.

A pesar de que se han obtenido buenos resultados con estas redes, se ha comprobado que el rendimiento de las mismas disminuye considerablemente si tienen que enfrentarse a imágenes capturadas con cámaras diferentes en circunstancias diferentes.

A la vista de este problema, en este Trabajo Fin de Grado se plantea un nuevo método que permite adaptar las redes ya entrenadas a muestras diferentes de las usadas en el entrenamiento a través de un proceso de selección de canales de las capas de la red. Este método está inspirado en los métodos clásicos de selección de características en *Machine Learning*, utilizando en nuestro caso la técnica de *Dropout* para desactivar o eliminar de forma efectiva los canales en la red. Los resultados encontrados son prometedores y parecen confirmar la validez del método.

ABSTRACT

Nowadays, technology plays a fundamental role in the medicine field, progressing greatly, specially in the diagnose of diseases thanks to artificial intelligence.

In this project, I have tried to add my own contribution to an investigation that has been carried out in the ULL for many years and that my tutor takes part in. Said investigation's aim is the premature diagnose of the glaucoma, a disease with significant prevalence that can lead to permanent blindness if there is no interveneance. To this end, we are working with convolucional neural networks which constitute the nucleus of this project.

Although optimal results have been obtained with these networks, it has been proved that the efficiency of these decreases considerably if they have to confront images captured with different cameras in different circumstances.

In sight of this problem, in this Tfg we will present a new method that allows to adjust the already trained networks to different samples than the used ones in the training through a selection process of channels of the layers of the network. This method is inspired in the classical methods of characteristic selection in *Machine Learning*, using in our project the technique of *Dropout* to deactivate or remove effectively the network channels. The found results are promising and seem to confirm the validity of the method.

ÍNDICE GENERAL

CAPÍTULO 1 INTRODUCCIÓN

1.1	MOTIVACIÓN DEL TRABAJO E HIPÓTESIS DE PARTIDA	9
1.2	DEEP LEARNING Y REDES NEURONALES CONVOLUCIONALES	10
1.2.1	REDES NEURONALES	11
1.2.1.1	NEURONA	11
1.2.1.2	ARQUITECTURA DE LAS REDES	11
1.2.1.3	ENTRENAMIENTO DE LAS REDES	13
1.3	EL DIAGNÓSTICO DEL GLAUCOMA	15

CAPÍTULO 2 MÉTODO PROPUESTO

2.1	SELECCIÓN DE CARACTERÍSTICAS EN MACHINE LEARNING	17
2.2	MÉTODO WRAPPER DE SELECCIÓN DE CARACTERÍSTICAS	18
2.3	DESCRIPCIÓN DEL MÉTODO PROPUESTO	18
2.4	TRABAJO RELACIONADO	21

CAPÍTULO 3 DESARROLLO EXPERIMENTAL

3.1	MATERIALES Y MÉTODOS	22
3.1.1	BASES DE DATOS DE IMÁGENES	22
3.1.2	REDES NEURONALES CONVOLUCIONALES	23
3.1.2.1	VGG16	23
3.1.2.2	VGG19	24
3.2	EXPERIMENTOS REALIZADOS	24
3.3	DISCUSIÓN DE LOS RESULTADOS	35

CAPÍTULO 4 CONCLUSIONES Y LÍNEAS ABIERTAS

4.1	CONCLUSIONES	39
4.2	LÍNEAS ABIERTAS	39

CHAPTER 5 CONCLUSIONES Y LÍNEAS ABIERTAS

5.1	CONCLUSIONES	41
5.2	LÍNEAS ABIERTAS	41

	REFERENCIAS	43
--	-------------	----

ÍNDICE DE FIGURAS

[1] Video/ ¿Qué es una red neuronal? Parte 1: La Neurona | DotCSV Vídeo de youtube disponible en:

https://www.youtube.com/watch?v=MRlv2lwFTPg&list=PLKw_rA9Pb_ctdSYtQcHROxu3F29BU9SE&index=5&t=0s

[2] <https://link.springer.com/article/10.1007/s11042-020-08740-w/figures/7>

[3] <https://anhreynolds.com/blogs/cnn.html>

[4] <https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2#7d8a>

[5] <https://deepai.org/machine-learning-glossary-and-terms/max-pooling>

[6] https://www.researchgate.net/figure/The-structure-of-single-hidden-layer-MLP-with-Backpropagation-algorithm_fig2_234005707

[7] <https://towardsdatascience.com/machinex-image-data-augmentation-using-keras-b459ef87cd22>

[8][9][10][11] Imágenes sacadas del *database* de Rim-One.DL

[12] Imagen extraída del artículo de investigación: *Wrappers for feature subset selection* (véase en referencias).

[13] Imagen extraída del artículo de investigación: *Dropout: A Simple Way to Prevent Neural Networks from Overtting* (véase en referencias).

[14] Imagen auto creada.

[15] <https://neurohive.io/en/popular-networks/vgg16/>

[16] https://www.researchgate.net/figure/Architecture-of-the-FCN-VGG19-adapted-from-Long-et-al-2015-which-learns-to-combine_fig1_331258180

CAPÍTULO 1. INTRODUCCIÓN

1.1. MOTIVACIÓN DEL TRABAJO E HIPÓTESIS DE PARTIDA

Este trabajo se encuadra dentro de la línea de investigación de aplicación de técnicas de *Deep Learning* o aprendizaje profundo al diagnóstico del Glaucoma que está llevando a cabo un grupo de profesores del Departamento de Ingeniería Informática y de Sistemas del cual forma parte el tutor del mismo. En dicha línea de investigación participa también el Hospital Universitario de Canarias (HUC).

Las técnicas de *Deep Learning* con redes neuronales convolucionales se han convertido en los últimos años en las herramientas más potentes para abordar muchos de los problemas que se presentan en Visión por Computador, gracias a su capacidad para aprender no solo a reconocer objetos a partir de unas características dadas sino incluso a aprender dichas características a partir de datos, en este caso imágenes, con las que se entrenan estos sistemas. El *Deep Learning* utiliza modelos de redes neuronales artificiales muy complejos con muchas capas cuyo aprendizaje requiere de un número suficiente de datos para su entrenamiento. Uno de los problemas principales con la aplicación de estas técnicas es precisamente ese, la disponibilidad de un número suficientemente grande de muestras para ajustar el enorme número de parámetros que tienen estos sistemas y evitar lo que se conoce como *overfitting* o sobreajuste. En ámbitos como la Medicina, no es fácil disponer de grandes conjuntos de imágenes representativos del problema que se quiere tratar. En estos casos se suele recurrir al *transfer learning* que consiste en reentrenar modelos previamente entrenados con otro tipo de imágenes, pero estos modelos suelen ser muy complejos al estar pensados para problemas con una mayor diversidad de características y categorías por lo que, aun funcionando correctamente, estaríamos ante una clara sobredimensión.

El glaucoma es una de las patologías más importantes que afectan a la retina y constituye una de las principales causas de ceguera en el mundo. Se la suele denominar como la

“ceguera silenciosa” dada la poca sintomatología que presenta hasta que la enfermedad está bastante avanzada. Es por ello que es fundamental detectarla precozmente.

En un principio, el objetivo de este trabajo era tratar de reducir la sobredimensión a la que se hacía referencia en los párrafos anteriores, buscando modelos más adecuados al problema del diagnóstico del glaucoma con solo dos clases e imágenes que no presentan una gran variabilidad si las comparamos con imágenes naturales. Sin embargo, a medida que se avanzaba en el desarrollo del proyecto, y sin perder de vista el objetivo inicial, se vio más interesante plantearlo como la propuesta de un nuevo método para el análisis del comportamiento de las redes ya entrenadas utilizando como inspiración las técnicas *Wrapper* del *Machine Learning* tradicional para la selección de características en clasificadores. En nuestro caso, estas características son los canales de las diferentes capas de la red y la selección de los canales no supone quitarlos realmente sino desactivarlos, de alguna manera, haciendo uso de una técnica muy conocida para evitar el *overfitting* que se llama Dropout.

Teniendo en cuenta lo anterior, tomamos como hipótesis de partida la misma que se plantea en el contexto de la selección de características clásica: el rendimiento de un algoritmo de aprendizaje puede mejorarse si se selecciona un subconjunto de características adecuado del conjunto original. Esta es la premisa que hemos intentado verificar con este trabajo aplicada al Deep Learning con redes neuronales convolucionales. Los resultados obtenidos con imágenes de fondo de ojo y redes previamente entrenadas con estas imágenes permiten afirmar que esta premisa se cumple. Hay que decir que el método presentado se ha explicado de una manera no formal, intentando que se comprenda la idea intuitiva de su funcionamiento. Una formalización matemática del mismo es, por supuesto, necesaria e imprescindible para su utilización en otros contextos, pero queda fuera del nivel de un trabajo de esta naturaleza.

En el resto del capítulo se hará una breve introducción a las redes neuronales convolucionales y al problema del diagnóstico del glaucoma. El capítulo 2 de la memoria explica el método propuesto, el capítulo 3 relata el desarrollo experimental llevado a cabo y en el capítulo 4 se incluyen las conclusiones y líneas abiertas de este trabajo.

1.2. DEEP LEARNING Y REDES NEURONALES CONVOLUCIONALES

El mundo de la inteligencia artificial es muy amplio y una de sus principales ramas es el *Machine Learning*, que busca proporcionar a las máquinas la capacidad de aprender. La técnica más utilizada en la última década para tratar problemas tan complejos y grandes (*Big data*) son las redes neuronales, concretamente redes neuronales con muchas capas en las que se lleva a cabo lo que se conoce como aprendizaje profundo o *Deep learning*. En este ámbito, destacan las redes neuronales convolucionales que son las más punteras en el campo de la visión, y con las que se ha conseguido lograr una precisión de clasificación superior a la humana [1].

1.2.1. Redes neuronales

1.2.1.1. Neurona

La parte más básica de una red neuronal es la neurona. La función matemática de estas neuronas es una simple suma ponderada entre los valores de entrada y los pesos que se le asignan a las conexiones de esa entrada (Figura 1). Además, cada neurona tiene asociada una conexión de una entrada con un peso variable independiente denominado *bias* (representado como b en la Figura 1) [2].

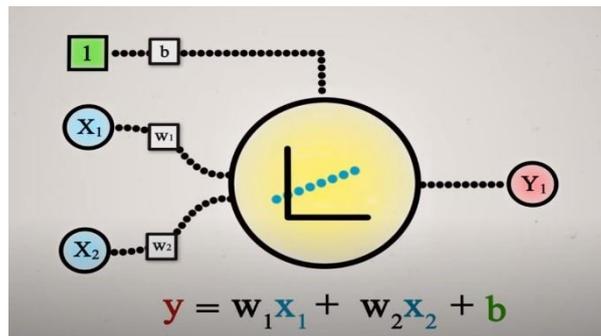


Figura 1: Esquema de una neurona artificial

Con esta fórmula matemática fácilmente podremos implementar puertas lógicas como la AND o la OR, pero en el caso de la XOR es imposible encontrar unos parámetros en los pesos y en *bias* para obtener una función perfecta. En este ejemplo empezamos a darnos cuenta de que necesitaremos más de una neurona para solucionar problemas simples y necesitaremos miles de ellas para problemas más complejos [3].

Por último, las neuronas presentan una función de activación que nos aportan la parte no lineal, las más utilizadas son la *Sigmoid*, la *Tanh* y la *Relu* (Figura 2). Tener una parte no lineal en las neuronas es fundamental para poder conectarlas entre sí y que nos ofrezcan infinitas salidas diferentes a una recta [3].

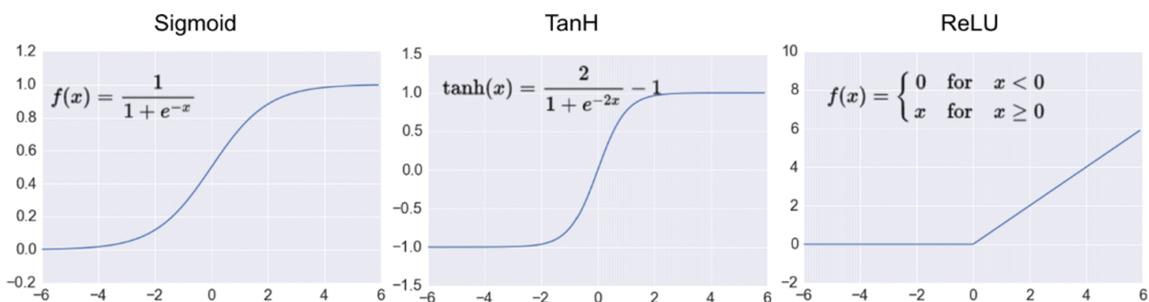


Figura 2: Sigmoid, TanH and ReLU functions

1.2.1.2. Arquitectura de las redes

Todas las redes convolucionales, siguen una misma estructura estándar. Presentan una entrada donde les llegan las imágenes, y a través de un número variable de capas

dependiendo de la red, se ejecutan una serie de operaciones de convolución. Además de ello, existen otras capas como las *Fully connected layers* [4].

- **Convolutional layers:**

Es la capa principal de este tipo de redes neuronales y su función principal es transformar la imagen de entrada en un mapa de características en sucesivas etapas en las que la entrada a cada capa es la salida de una capa o capas anteriores.

Para poder realizar esta operación necesitaremos un filtro, dicho filtro (representado en violeta en la figura 3) pasará a través de toda la imagen e irá realizando la operación para cada grupo de píxeles. Teniendo en la salida el mapa de características necesario para nuestra red [4].

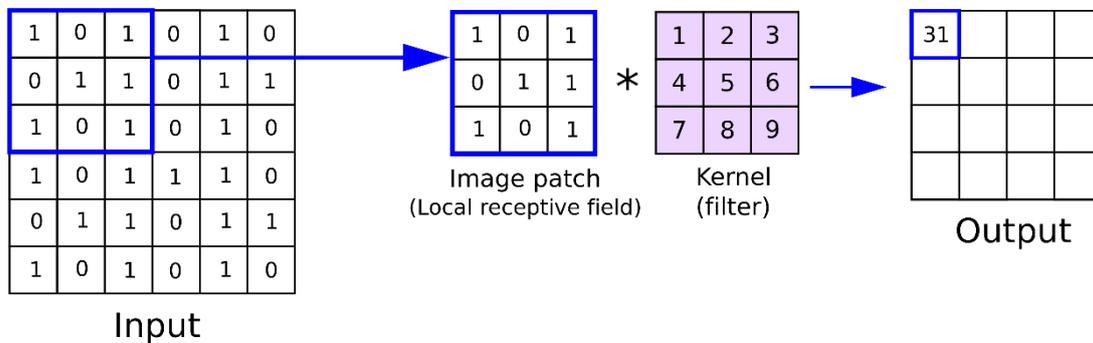


Figura 3: Convolución 2D

Sin embargo, las convoluciones se producen a una escala 3D (figura 4), puesto que las imágenes presentan una profundidad con los canales de color (RGB), en caso de la capa de entrada, y también en capas sucesivas. Para ello, entonces necesitamos que nuestras *Convolutional layers* tengan sus filtros de 3D. Además, se utilizan diferentes filtros para producir muchos mapas de características distintos que de manera conjunta serán la salida final de nuestra capa [4].

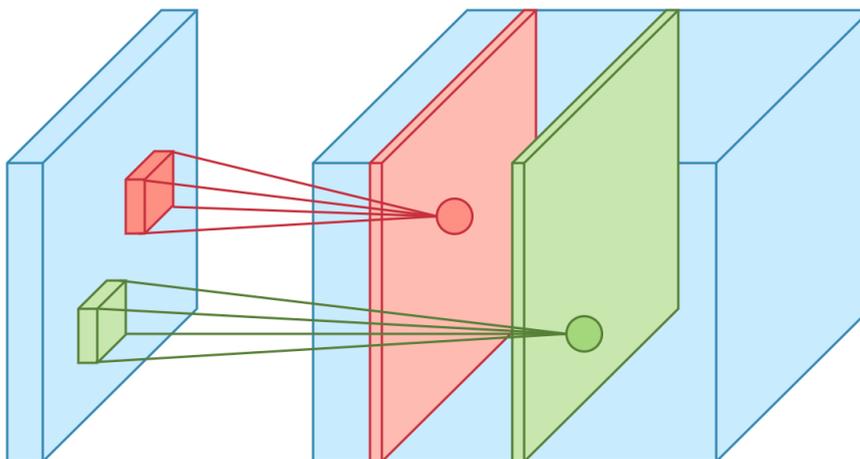


Figura 4: Convolución 3D

- **Pooling layers:**

Unidas a las *Convolutional layers* siempre encontraremos este tipo de capas, que reducen drásticamente esos mapas de características en su altura y ancho, haciendo un resumen de la información de sus mapas. En el caso más típico (las *max pooling layers*), se selecciona el valor mayor de cada sector que estemos reduciendo (figura 5). Estas capas permiten, sobre todo, que las características extraídas por las capas convolucionales se calculen sobre un dominio espacial diferente [4].

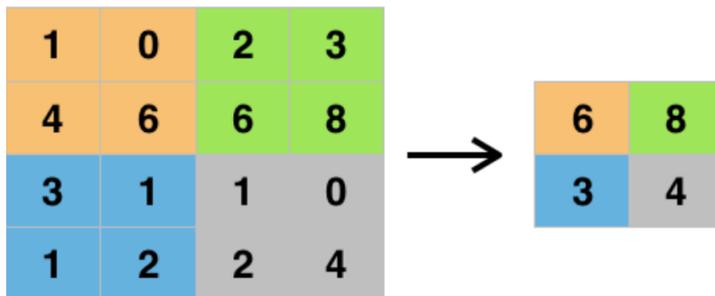


Figura 5: Max pooling

- **Fully connected layers:**

En estas capas se transforma el volumen 3D que nos llega de las *convolutional* y *pooling layers* en un vector. Es necesario tener unas capas *fully connected* al final de nuestra red para enviar la información al clasificador de forma que la pueda entender [4].

1.2.1.3. Entrenamiento de las redes

Una de las principales características de las redes neuronales, es que son ellas mismas las que aprenden a calcular los valores de los parámetros de los pesos y las *bias* a través de un proceso denominado entrenamiento. Tienen la capacidad de ir aprendiendo qué valores son los mejores para tener una mayor precisión en los resultados.

El algoritmo que se utiliza para el entrenamiento de las redes neuronales se denomina Backpropagation [5] y está basado en el cálculo del gradiente descendente y su propagación hacia atrás en la red como se ilustra en la figura 6.

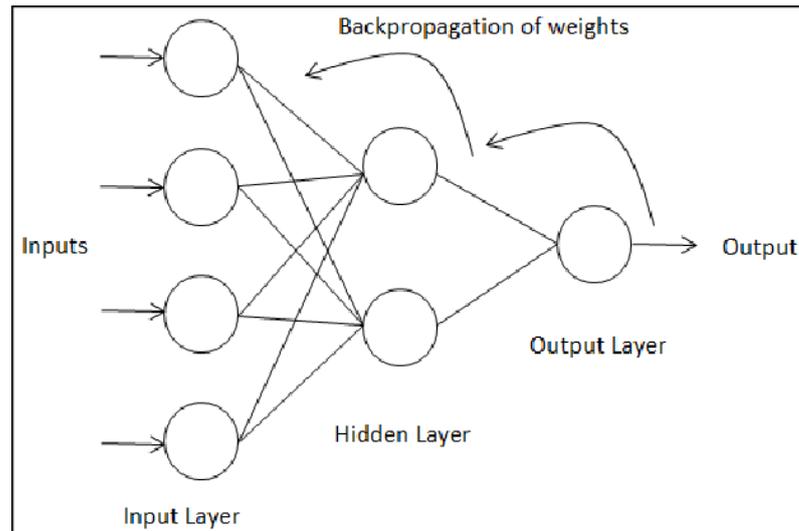


Figura 6: Ilustración del Backpropagation

Relacionado con el entrenamiento de la red y su aprendizaje, nos encontramos con uno de los mayores problemas en el *Deep Learning*, el *overfitting* [6]. Este fenómeno, se produce cuando la red aprende el ruido o detalle muy particular de la muestra de entrenamiento y no es capaz de generalizar a otras muestras independientes. Esto se produce habitualmente en la práctica por tratarse de modelos muy complejos con muchos parámetros y un número limitado de muestras. Una de las maneras habituales de intentar mitigar este fenómeno es recurrir a lo que se conoce como *Data Augmentation* [6]. Esta técnica consiste en ampliar artificialmente el número de imágenes de entrenamiento, modificando las imágenes originales por medio de rotaciones, escala, filtros, y otro tipo de transformaciones, como se muestra en la figura 7. Aunque no se trata de imágenes realmente nuevas que enriquezcan el conjunto de entrenamiento, sí que son efectivas para minimizar el *overfitting*. Otra de las técnicas que se suele utilizar para este fin es lo que se conoce como *Dropout* [6]. Nos referiremos a ella en el capítulo siguiente ya que se ha usado en este trabajo.

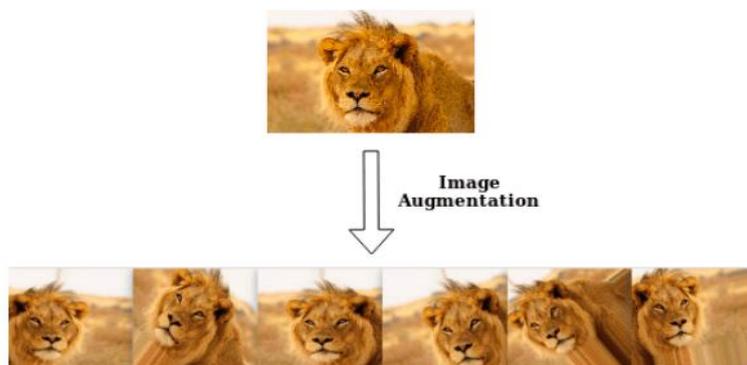


Figura 7: Ejemplo de Data augmentation

1.3. EL DIAGNÓSTICO DEL GLAUCOMA

El glaucoma se puede ver como una serie de patologías que dañan el nervio óptico produciéndole pérdidas de células ganglionares en la retina, lo que da lugar en el mayor porcentaje de los casos a un aumento en la presión intraocular [7].

El glaucoma es una enfermedad grave y una de las causas principales de ceguera total en el mundo. Se estima que en Canarias unas 60.000 personas padecen glaucoma [8]. Es una enfermedad que con tratamiento en sus fases precoces se puede combatir. Con un diagnóstico precoz y un buen tratamiento se logra evitar la ceguera en más del 90% de los casos [7][8]. Sin embargo, es muy complicada de detectar porque empieza siendo asintomática. Existen principalmente dos técnicas de diagnóstico basadas en imágenes de retina:

- Las basadas en retinografías: Son de bastante bajo coste, por lo que la mayoría de los centros de salud podrían llegar a disponer de los retinógrafos necesarios, y se obtienen las imágenes RGB del fondo de ojo a través de la pupila. La principal zona de estudio en las retinografías para el diagnóstico del glaucoma es el disco óptico. Es la parte que se daña cuando se sufre esta enfermedad y visualmente lo podemos identificar porque se va tornando a un color más blanquecino cuando la enfermedad va avanzando. En las figuras 8 y 9 se muestran unos ejemplos de un ojo sano y un ojo enfermo.



Figura 8: Recorte de una retinografía en el disco óptico de un paciente sano

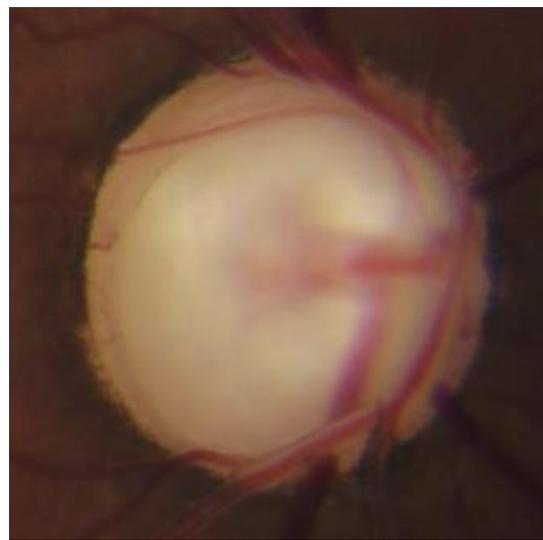


Figura 9: Recorte de una retinografía en el disco óptico de un paciente con glaucoma

- El OCT (*Optical Coherence Tomography*): Es una técnica de alto coste con mayor potencia que generalmente encuentra en hospitales. Genera unas imágenes en 3D en infrarrojo y también pueden generar imágenes 2D del fondo de ojo. En las figuras 10 y 11 se muestra la imagen de un ojo sano y de otro que padece glaucoma.

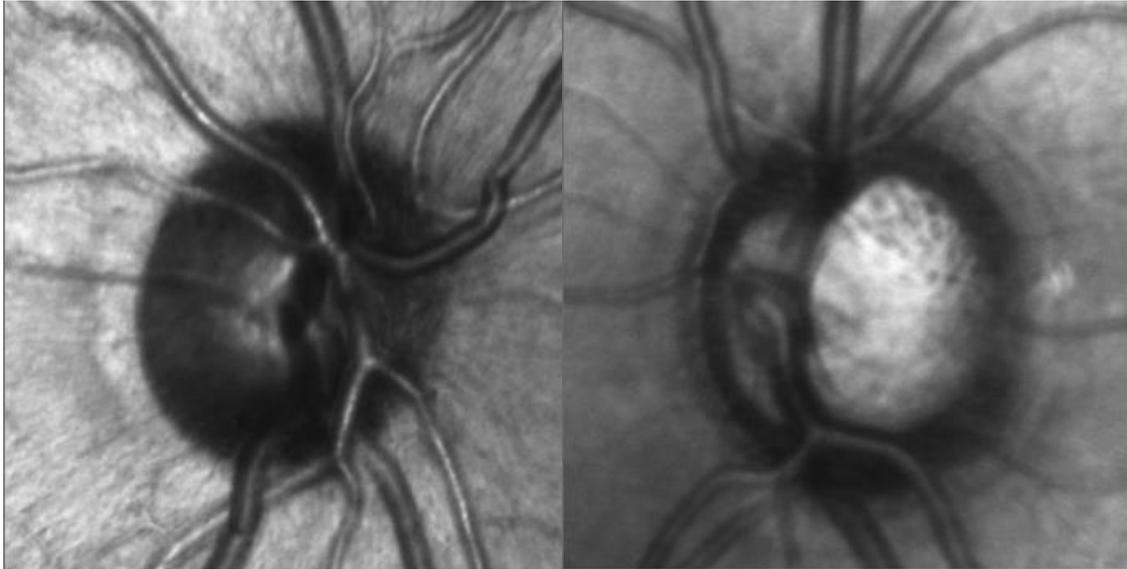


Figura 10: Recorte de un OCT en el disco óptico de un paciente sano

Figura 11: Recorte de un OCT en el disco óptico de un paciente con glaucoma

CAPÍTULO 2. MÉTODO PROPUESTO

2.1. SELECCIÓN DE CARACTERÍSTICAS EN *MACHINE LEARNING*

Los métodos de selección de características son bien conocidos en el ámbito del *machine learning* [9]. Dentro de este ámbito, en este trabajo nos hemos centrado en el aprendizaje supervisado y, más concretamente, en problemas de clasificación.

Como su propio nombre indica, dado un algoritmo de aprendizaje L y un conjunto de datos etiquetados D con características X_1, X_2, \dots, X_n , los métodos de selección de características tienen como objetivo reducir la dimensionalidad del problema de n a p , $p \leq n$, con el fin de:

- Mejorar el rendimiento del algoritmo de aprendizaje basado en esas características.
- Simplificar la estructura del modelo de aprendizaje generado.
- Mejorar la visualización y comprensión del problema de aprendizaje.

Los métodos de selección de características se suelen dividir en métodos de filtrado, métodos *Wrapper*, y métodos embebidos.

Los métodos de filtrado seleccionan las características a partir de los datos disponibles, atendiendo a las propiedades intrínsecas de dichas características sin tener en cuenta el algoritmo de aprendizaje con el que se va a trabajar. Se trata, en esencia, de establecer un ranking entre las mismas que dé cuenta de su capacidad discriminante.

Los métodos *Wrapper*, por el contrario, tiene en cuenta tanto los datos como el algoritmo de aprendizaje a la hora de elegir las mejores características. Es el método en el que se ha inspirado este trabajo y por ello le dedicaremos una atención especial.

Los métodos embebidos son similares a los *Wrapper* con la salvedad de que se integran en el propio proceso de entrenamiento del clasificador en lugar de aplicarse una vez se ha entrenado.

2.2. MÉTODO WRAPPER DE SELECCIÓN DE CARACTERÍSTICAS

El método *Wrapper* de selección de características fue introducido por Kohavi y John [10] en 1997 con la idea de mejorar el rendimiento de los clasificadores a partir de la selección de un conjunto de características del conjunto original que fuera óptimo desde ese punto de vista. En la figura 12 se muestra un esquema de dicho método.

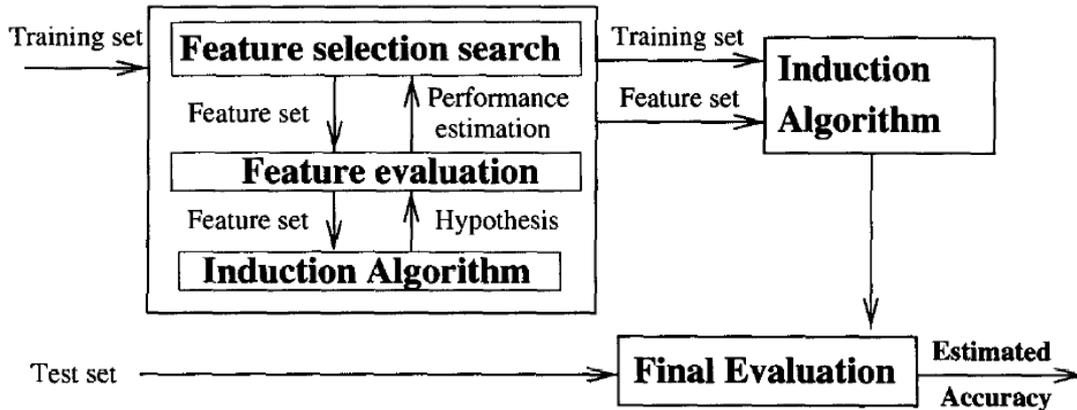


Figura 12: Esquema del método Wrapper

En el método *Wrapper*, el algoritmo de clasificación (*induction algorithm*) para el que se quiere encontrar las características óptimas se considera como una caja negra ya que solo importa su rendimiento para un conjunto de características dado. Así, se hace necesario disponer de una muestra de entrenamiento y validación para el proceso de selección, y una vez se ha encontrado el conjunto óptimo buscado, se procedería a hacer una evaluación final con una muestra de test independiente. Otra cuestión importante a destacar es la búsqueda en el espacio de posibles subconjuntos de características. Este espacio es de tamaño $O(2^n)$ para n características por lo que la búsqueda exhaustiva es inviable para la gran mayoría de problemas que se dan en la práctica, lo que lleva a plantear estrategias subóptimas. Las más habituales son la selección hacia adelante (*forward selection*) y la selección hacia atrás (*backward selection*), o una combinación de ambas. La estrategia de selección hacia adelante consiste en partir del conjunto vacío de características e ir añadiendo sucesivamente características del conjunto original hasta que se considere que se ha encontrado el subconjunto óptimo. Por el contrario, la estrategia de selección hacia atrás parte del conjunto completo de características y va eliminando progresivamente hasta llegar al subconjunto óptimo. La estrategia hacia adelante suele ser computacionalmente más eficiente pero la de selección hacia atrás parece capturar mejor las interacciones entre las características y por eso nos centraremos más en ella.

2.3. DESCRIPCIÓN DEL MÉTODO PROPUESTO

El método propuesto en este trabajo está inspirado en el método Wrapper con la idea de analizar el rendimiento de redes neuronales convolucionales previamente entrenadas y, si es posible, mejorarlo. En una red neuronal convolucional existen dos partes principales,

la parte convolucional en sí misma, que aprende las mejores características para un problema dado, y la parte de clasificación con esas características. Por lo tanto, en lo que sigue consideraremos como característica a cada uno de los canales de las diferentes capas de convolución. Parece claro que en este caso no es posible eliminar estas características, en el sentido estricto de la palabra, de una red ya entrenada. Por ese motivo, hemos recurrido a una “eliminación” tipo *dropout* siguiendo el ejemplo de la técnica bien conocida con este nombre para evitar el *overfitting* [11]. En la figura 13 se muestra el efecto de aplicar la técnica *dropout* en una red neuronal.

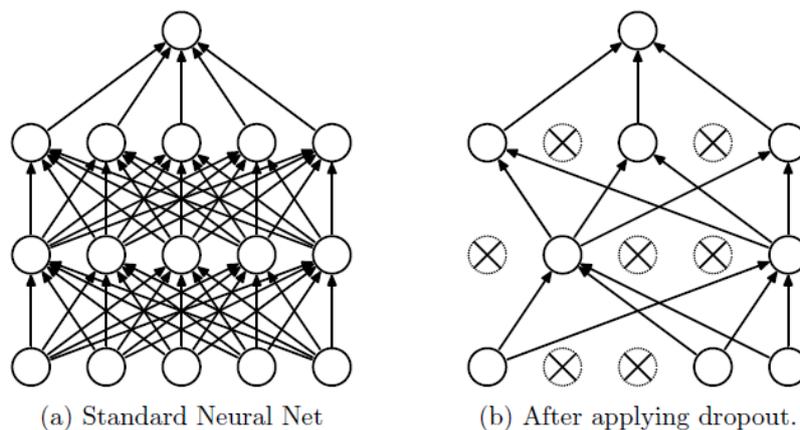


Figura 13: Esquema de la técnica Dropout

La manera habitual de aplicar esta técnica durante el entrenamiento de una red neuronal consiste en poner a cero la activación de ciertas neuronas elegidas al azar, habitualmente en la parte densamente conectada de la red donde existen muchas más conexiones y, por lo tanto, muchas más interdependencias entre las neuronas.

Nuestro método también hace uso del *dropout*, pero aplicado sobre las neuronas de los diferentes canales de la parte convolucional de la red de forma que las activaciones de las neuronas de dichos canales se ponen a cero ante cualquier entrada por lo que, de forma efectiva, es como si esa característica no jugara ningún papel en ninguna parte de la imagen. En la implementación práctica de esta idea, lo que se ha hecho es multiplicar la salida de los canales de una determinada capa por un tensor de 0s y 1s según los canales que se pretenda desactivar, como se muestra en la figura 14.

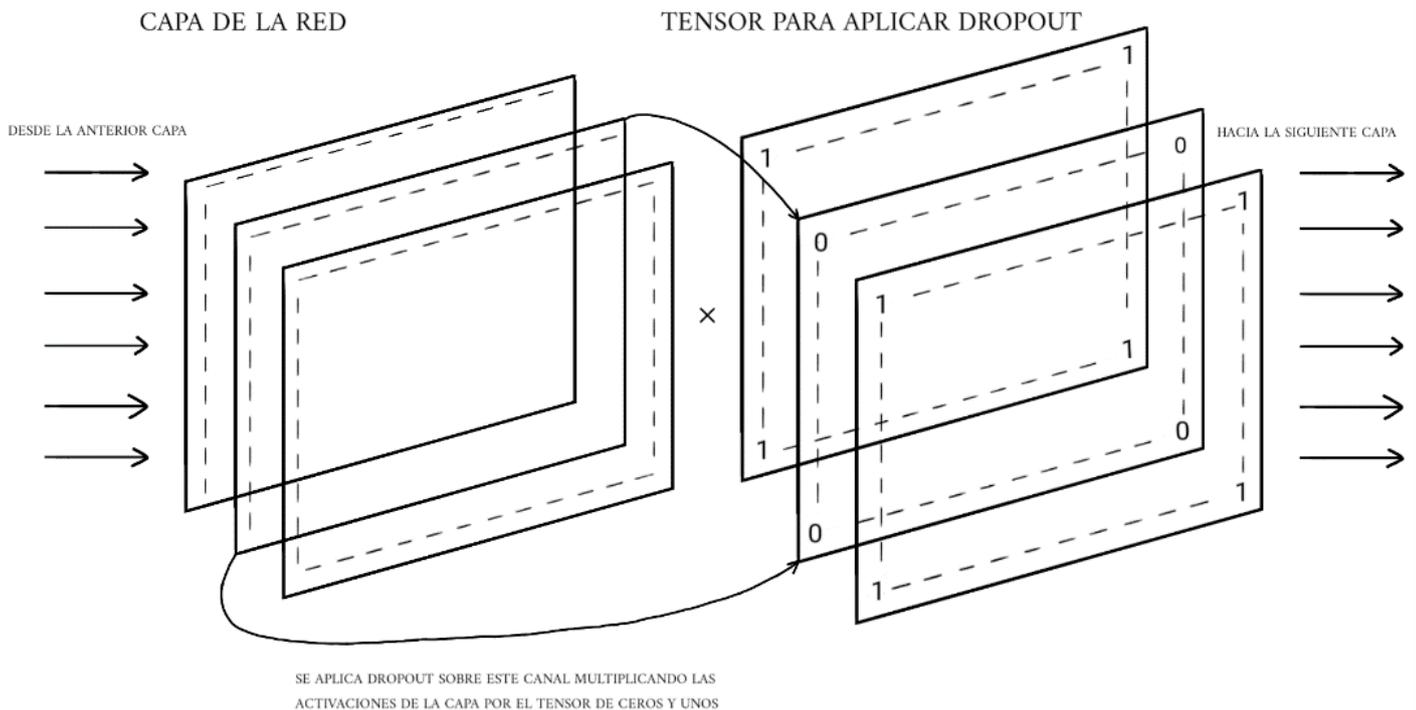


Figura 14: Método propuesto de manera práctica

Más allá de la técnica *dropout*, el resto del método se ajusta al esquema del método *Wrapper* descrito en la sección anterior, utilizando la estrategia de búsqueda hacia atrás para ir descartando canales, progresivamente. De forma un poco más precisa, los pasos que hay que dar para la implementación del método son los siguientes:

- 1) Dada una red neuronal convolucional ya entrenada y un conjunto de imágenes diferente al utilizado en el entrenamiento de dicha red.
- 2) Se divide el conjunto de imágenes en una muestra de *dropout* (equivalente a una muestra de entrenamiento), de validación y de test.
- 3) Se selecciona una capa de la red y se van eliminando iterativamente sus canales mediante *dropout*. Como hemos seguido la estrategia de búsqueda hacia atrás, se parte de considerar todos los canales y en cada iteración del método se va eliminando un canal hasta que no quede ninguno o se imponga alguna otra condición de parada.
- 4) El criterio para ir descartando los canales consiste en ver el efecto que tiene sobre el rendimiento de la red la eliminación de dichos canales. Para ello, se utiliza la muestra de *dropout* y la de validación para evaluar el rendimiento R de la red según la expresión:

$$R(c) = Acc_{Dropout}(c) + Acc_{Validacion}(c) - abs[Acc_{Dropout}(c) - Acc_{Validacion}(c)] \quad (1)$$

Donde c representa los canales que se están evaluando en cada iteración (el que se está considerando en la iteración actual junto a los que ya se han eliminado en iteraciones anteriores), $Acc_{Dropout}(c)$ es la precisión (*accuracy*) de la red calculada

sobre la muestra de *dropout*, y $Acc_{validacion}(c)$ es la precisión de la red calculada sobre la muestra de validación. De esta manera, en cada iteración del método se descartarán los canales c para los que se haga máximo $R(c)$ con la idea de que se eliminen los canales que sean lo menos útiles posible para la red.

- 5) La expresión en (1) pretende conseguir dos objetivos fundamentales. Por un lado, que en cada iteración del método se intente obtener la mayor precisión posible, y por otro que esto ocurra tanto en la muestra de *dropout* como en la de validación, tratando de evitar, en lo posible, el sesgo excesivo del método hacia una muestra concreta. Los experimentos que hemos realizado nos indican que, efectivamente, este sesgo se disminuye notablemente respecto a tener en cuenta solo la muestra de *dropout* a la hora de descartar los canales.

2.4. TRABAJO RELACIONADO

Los trabajos que hemos encontrado más relacionados con la temática de este proyecto son los que se encuadran en lo que se conoce como *channel pruning* [12][13][14]. Este tipo de técnicas tienen como objetivo la compresión de los modelos de redes neuronales para poder ser manejados más fácilmente e implementados en dispositivos cuya capacidad computacional no es demasiado grande como pueden ser los teléfonos móviles. La idea, como siempre que se hace compresión, es quitar la mayor cantidad de información sin que eso suponga una merma en su rendimiento o que esta sea lo más pequeña posible.

Este tipo de técnicas sí que quitan realmente los canales de la red por lo que se hace necesario aplicar técnicas de reconstrucción que minimicen la diferencia entre la red original y la comprimida. Algunos métodos de *pruning* hacen también selección de canales para descartar los que sean más redundantes. En cualquier caso, el objetivo principal de estos métodos, como se ha comentado, es hacer las redes más pequeñas y, por lo que hemos visto, se trata de métodos, en general, bastante más complicados que el aquí expuesto.

Otro asunto relacionado con este trabajo es la utilización de técnica de *dropout*. Ya nos hemos referido a ella en una sección anterior en su versión original. Lo cierto es que existen numerosas variantes de esta versión [15][16][17], pero todas ellas están pensadas para ser aplicadas en tiempo de entrenamiento con el fin de evitar el *overfitting* y no, como en nuestro caso, sobre una red ya entrenada para hacer selección de canales.

CAPÍTULO 3. DESARROLLO EXPERIMENTAL

3.1. MATERIALES Y MÉTODOS

En esta sección se describen los materiales y métodos utilizados para el desarrollo experimental llevado a cabo en este trabajo.

3.1.1. Bases de datos de imágenes

En este trabajo se han utilizado tres bases de datos de imágenes de fondo de ojo: RIM-ONE DL, imágenes en infrarrojo del HUC, y ACRIMA [18].

RIM-ONE DL surge como una iniciativa para aunar y revisar las tres versiones de RIM-ONE existentes y adaptarlas a los problemas de *Deep Learning*. Las imágenes fueron capturadas en el HUC, el Hospital Universitario Miguel Servet (Zaragoza) y el Hospital Clínico Universitario San Carlos (Madrid). Todas las imágenes de esta base de datos han sido recortadas alrededor de la cabeza del nervio óptico con una misma proporción. Además, encontramos una sola imagen por paciente y ojo. En total, tenemos 313 fotos de pacientes sanos y 172 fotos de pacientes con glaucoma. La responsabilidad de la categorización previa de estas imágenes corrió a cargo de tres médicos expertos en la enfermedad. En las figuras 8 y 9 del capítulo 1 se muestra un ejemplo de estas imágenes.

La muestra de imágenes de fondo de ojo de infrarrojo se ha capturado en el HUC a través de un OCT. Está compuesta por 187 imágenes de sujetos normales y 313 de sujetos con glaucoma. En las figuras 10 y 11 del capítulo 1 se muestra un ejemplo representativo de las dos categorías.

ACRIMA es una base datos publicada recientemente con 309 retinografías de sujetos sanos y 396 de sujetos con glaucoma. Están también recortadas alrededor de la cabeza del nervio óptico y han sido previamente categorizadas por expertos ajenos al grupo de investigación. Las imágenes fueron capturadas con una cámara Topcon TRC.

3.1.2. Redes neuronales convolucionales

Para este trabajo se han utilizado dos redes neuronales convolucionales muy conocidas, la VGG16 y la VGG19 [19], previamente entrenadas por el grupo de investigación utilizando las muestras de RIM-ONE DL de forma que, en el entrenamiento, validación y test, las muestras se han repartido de forma aleatoria.

Hemos elegido estos modelos porque principalmente por dos motivos. Por un lado, se trata de redes muy documentadas con una estructura muy lineal con una capa a continuación de la otra, lo cual resulta adecuado para unas primeras pruebas. Por otro lado, a pesar de esta estructura relativamente simple, se han publicado resultados muy competitivos en entrenamientos con imágenes de fondo de ojo, comparables e incluso superiores a los alcanzados por otro tipo de redes más sofisticadas.

3.1.2.1 VGG16

La VGG16 es un modelo de red neuronal creado por la Universidad de Oxford que se hizo muy famoso en 2014 al participar y ganar la ILSVRC (*ImageNet Large Scale Visual Recognition Competition*) obteniendo unos grandes resultados.

La estructura de la VGG16 consiste en 16 capas sin contar las capas de *max pooling* y *softmax* (*convolutional layers*, *activation layers* y *fully connected layers*). En la figura 15 se muestra un esquema de las capas de esta red.

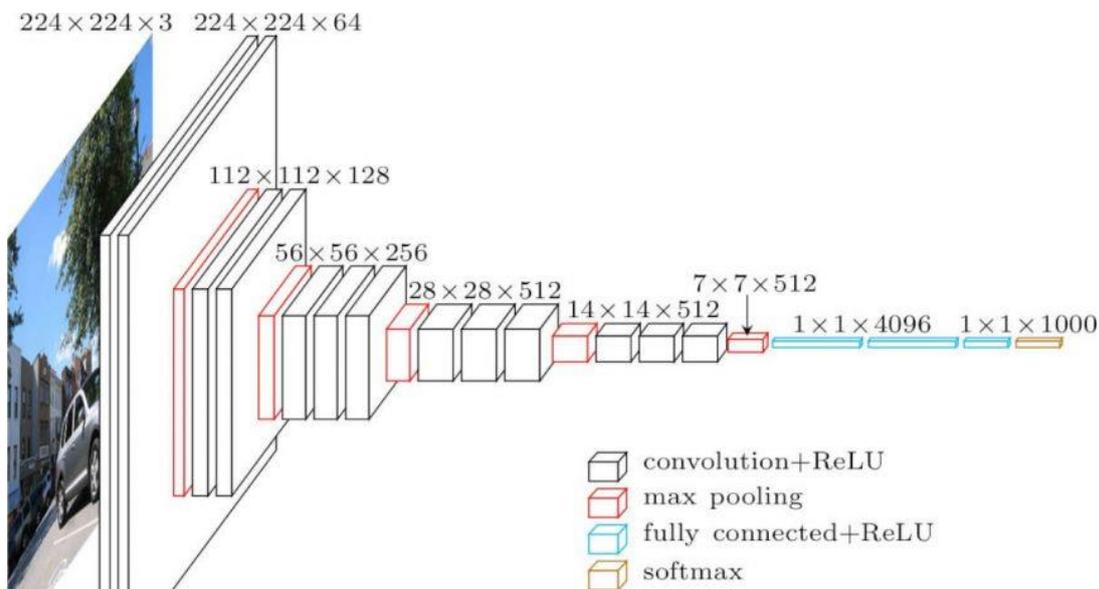


Figura 15: Estructura de la VGG16

Aparte de mostrar gráficamente las curvas que corresponden a cada una de las muestras en cada iteración de la selección de canales, también se ha incluido una tabla con los datos más relevantes. Por un lado, tenemos los valores de precisión inicial de la red para cada una de las muestras, y los valores máximos alcanzados durante el proceso de selección. Hay que tener en cuenta que en la práctica deberemos escoger una etapa en la que consideremos que el modelo de red resultante del proceso de selección es el óptimo. Para ello se pueden tomar diferentes referencias. En este trabajo hemos optado por tomar como referencia la etapa en la que $R(c)$ se hace máximo, que indica que el comportamiento de la red es el mejor posible tanto para la muestra de *dropout* como para la de validación. En esta etapa es en la que se estima el valor de precisión para la muestra de test que también aparece en las tablas. Finalmente, la columna de nº de canales seleccionados se refiere al número de canales que nos quedarían sin desactivar a partir de la etapa en la que $R(c)$ se hace máximo, lo que da una idea del número de canales efectivos que jugarían un papel relevante en el nuevo modelo con una mejor precisión.

En los casos en los que no se ha podido trabajar con las tres muestras de *dropout*, validación y test, todo lo anterior se restringe a dos muestras.

Todos los experimentos se han programado en Python con Keras y han sido ejecutados en el entorno de Google Colab.

Caso 1: redes VGG16 y VGG19 entrenadas con RIM-ONE DL y selección de canales con muestras de ACRIMA.

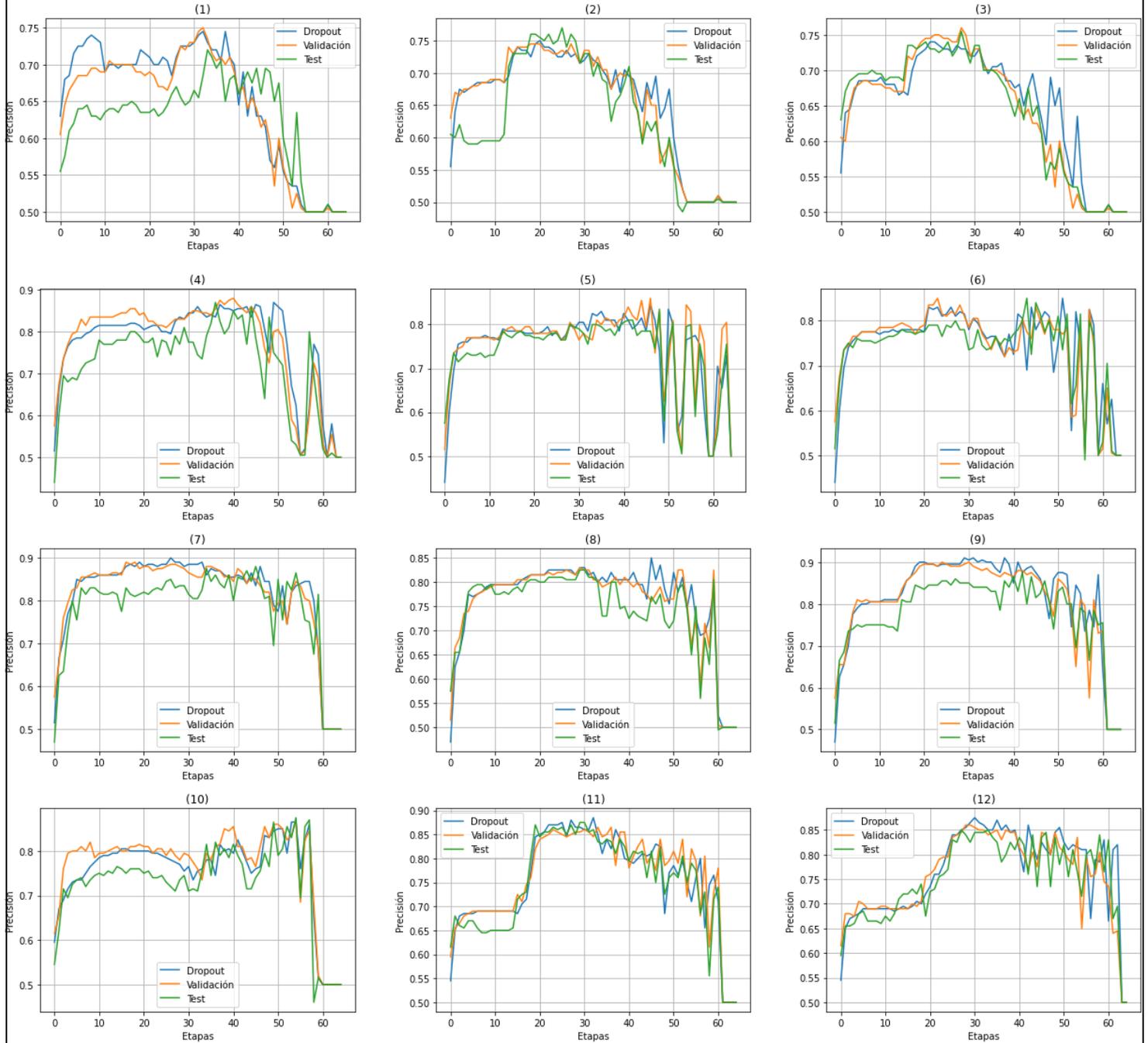


Figura 17: Gráficas para la red VGG16 en el caso 1

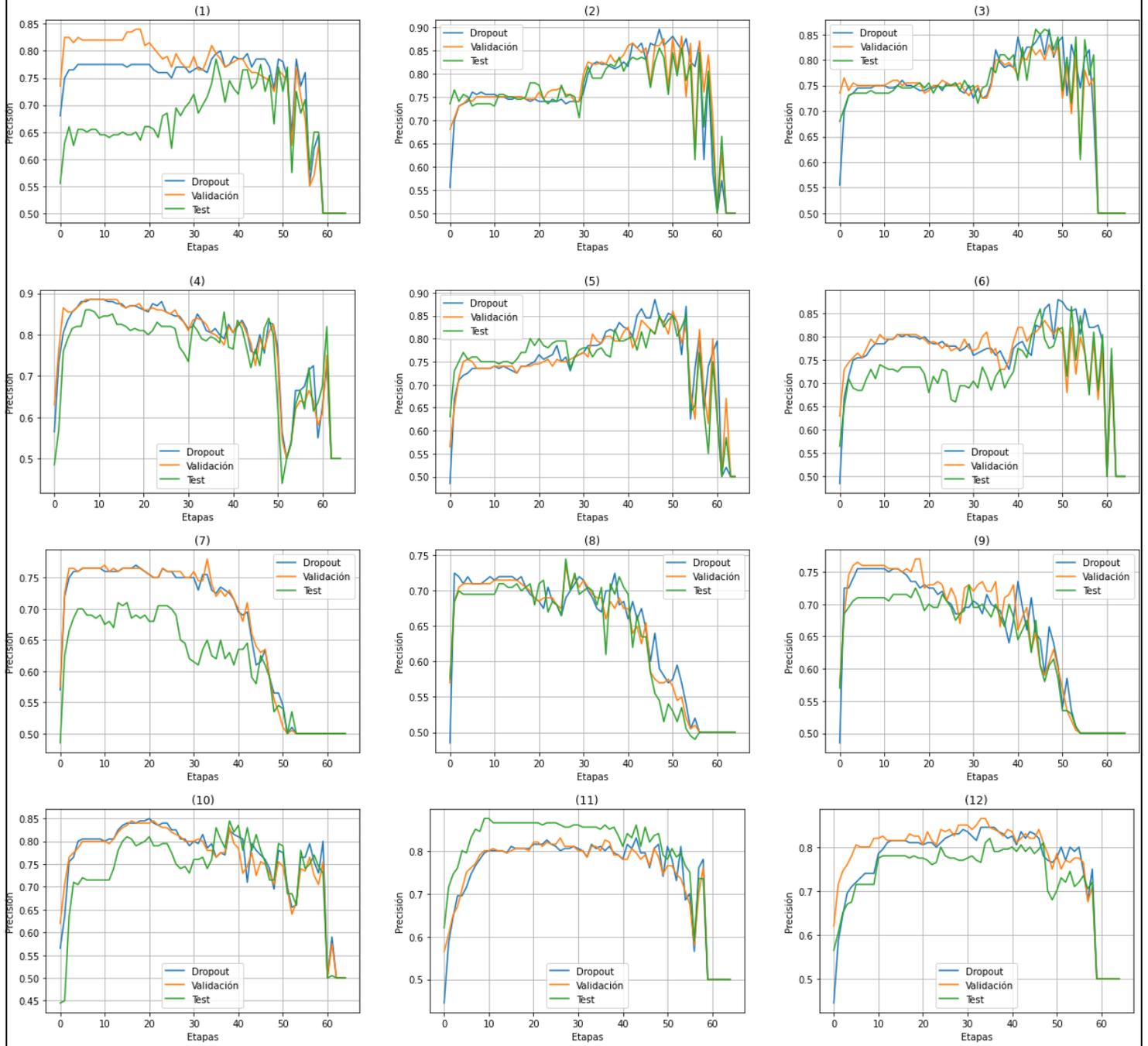


Figura 18: Gráficas para la red VGG19 en el caso 1

Tabla 1: Resultados para la red VGG16 en el caso 1

	Valor inicial Dropout	Valor Máximo Dropout	Valor inicial Validación	Valor Máximo Validación	Valor inicial Test	Máximo valor Test	Valor estimado Test	Nº canales seleccionados
1	0.630	0.745	0.605	0.750	0.550	0.720	0.685	32
2	0.555	0.750	0.630	0.745	0.605	0.770	0.760	45
3	0.555	0.740	0.605	0.760	0.630	0.755	0.730	44
4	0.515	0.870	0.575	0.880	0.440	0.870	0.825	27
5	0.440	0.845	0.515	0.860	0.575	0.835	0.785	18
6	0.440	0.850	0.575	0.850	0.515	0.850	0.790	41
7	0.515	0.900	0.575	0.890	0.470	0.880	0.815	47
8	0.470	0.850	0.515	0.830	0.575	0.825	0.825	35
9	0.470	0.910	0.575	0.900	0.515	0.875	0.850	34
10	0.595	0.865	0.615	0.870	0.545	0.875	0.875	10
11	0.545	0.885	0.595	0.865	0.615	0.875	0.865	41
12	0.545	0.875	0.515	0.860	0.595	0.850	0.825	35

Tabla 2: Resultados para la red VGG19 en el caso 1

	Valor inicial Dropout	Valor máximo Dropout	Valor inicial Validación	Valor máximo Validación	Valor inicial Test	Valor máximo Test	Valor estimado Test	Nº canales seleccionados
1	0.630	0.800	0.735	0.840	0.555	0.785	0.785	29
2	0.555	0.895	0.680	0.880	0.735	0.855	0.845	14
3	0.555	0.860	0.735	0.830	0.680	0.860	0.855	17
4	0.565	0.855	0.630	0.885	0.485	0.860	0.860	56
5	0.485	0.885	0.565	0.860	0.630	0.850	0.850	14
6	0.485	0.880	0.630	0.835	0.565	0.865	0.795	18
7	0.570	0.770	0.575	0.780	0.485	0.710	0.700	59
8	0.485	0.725	0.570	0.735	0.575	0.745	0.695	54
9	0.485	0.755	0.575	0.770	0.570	0.730	0.710	60
10	0.565	0.850	0.620	0.845	0.445	0.845	0.810	44
11	0.445	0.830	0.565	0.830	0.620	0.875	0.865	41
12	0.445	0.845	0.620	0.865	0.565	0.820	0.765	31

Caso 2: redes VGG16 y VGG19 entrenadas con muestras de ACRIMA (por otro grupo de investigación) y selección de canales con muestras de RIM-ONE DL aleatorias.

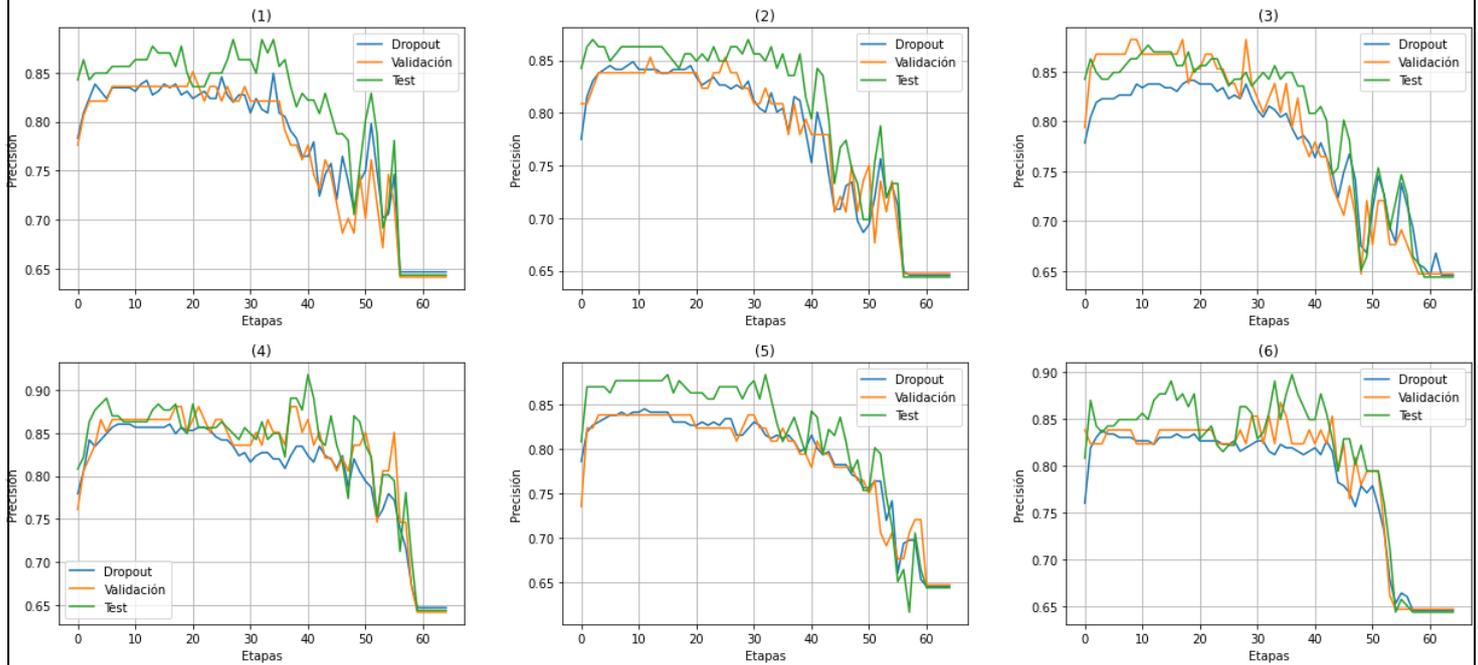


Figura 19: Gráficas para las redes VGG16 (primera fila) y VGG19 (segunda fila) en el caso 2

Tabla 3: Resultados para la VGG16 y VGG19 en el caso 2

	Valor inicial Dropout	Valor máximo Dropout	Valor inicial Validación	Valor máximo Validación	Valor inicial Test	Valor máximo Test	Valor estimado Test	Nº canales seleccionados
1	0.783	0.849	0.776	0.850	0.842	0.884	0.863	53
2	0.775	0.849	0.809	0.853	0.842	0.870	0.863	52
3	0.779	0.841	0.794	0.882	0.842	0.877	0.849	45
4	0.779	0.860	0.761	0.880	0.808	0.918	0.870	57
5	0.786	0.845	0.735	0.838	0.808	0.884	0.877	57
6	0.760	0.838	0.838	0.868	0.808	0.897	0.842	60

Caso 3: redes VGG16 y VGG19 entrenadas con muestras de RIM-ONE DL aleatorias y selección de canales con la misma muestra. Las muestras de *dropout* y validación son las muestras de test y entrenamiento, respectivamente, que se utilizaron originalmente con RIM-ONE DL.

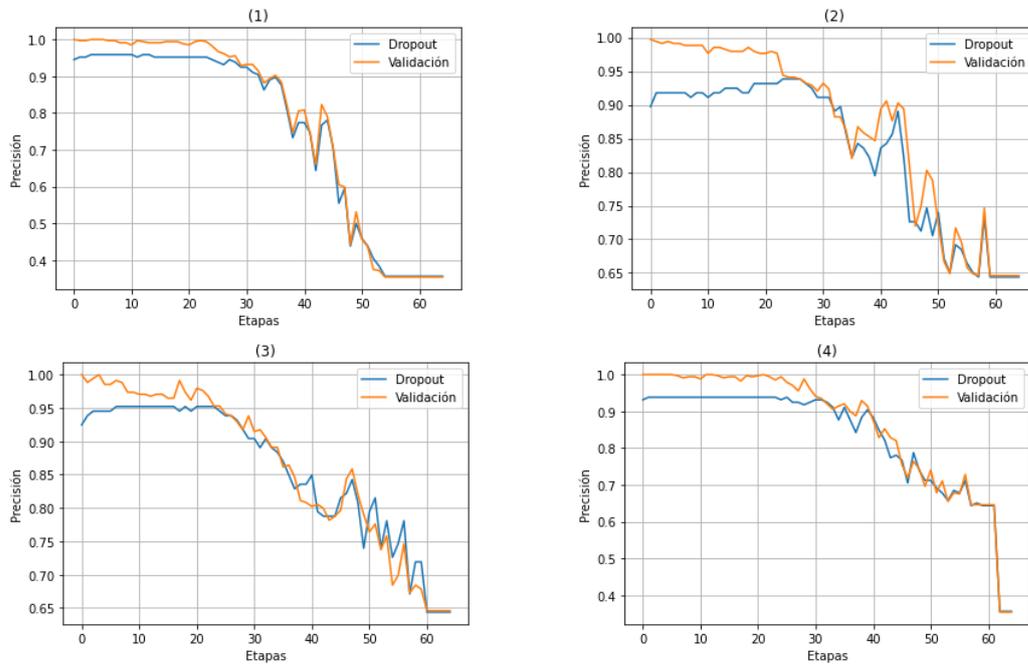


Figura 20: Gráficas para VGG16 en el caso 3

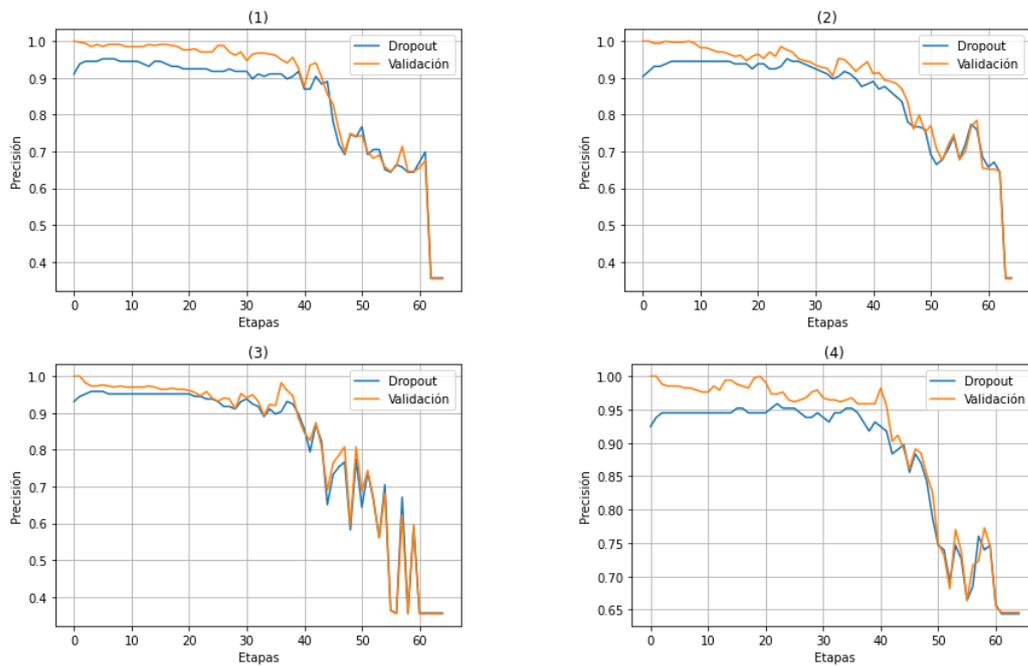


Figura 21: Gráficas para VGG19 en el caso 3

Tabla 4: Resultados para VGG16 en el caso 3

	Valor inicial Dropout	Valor máximo Dropout	Valor inicial Validación	Valor máximo Validación	Nº canales seleccionados
1	0.945	0.959	1.000	1.000	58
2	0.897	0.938	0.997	0.997	41
3	0.925	0.952	1.000	1.000	58
4	0.932	0.938	1.000	1.000	63

Tabla 5: Resultados para VGG19 en el caso 3

	Valor inicial Dropout	Valor máximo Dropout	Valor inicial Validación	Valor máximo Validación	Nº canales seleccionados
1	0.911	0.952	1.000	1.000	58
2	0.904	0.952	1.000	1.000	39
3	0.932	0.959	1.000	1.000	61
4	0.925	0.959	1.000	1.000	42

Caso 4: redes VGG16 y VGG19 entrenadas con muestras de infrarrojo y selección de canales con la misma muestra. Las muestras de *dropout* y validación son las muestras de test y entrenamiento, respectivamente, que se utilizaron originalmente con RIM-ONE DL.

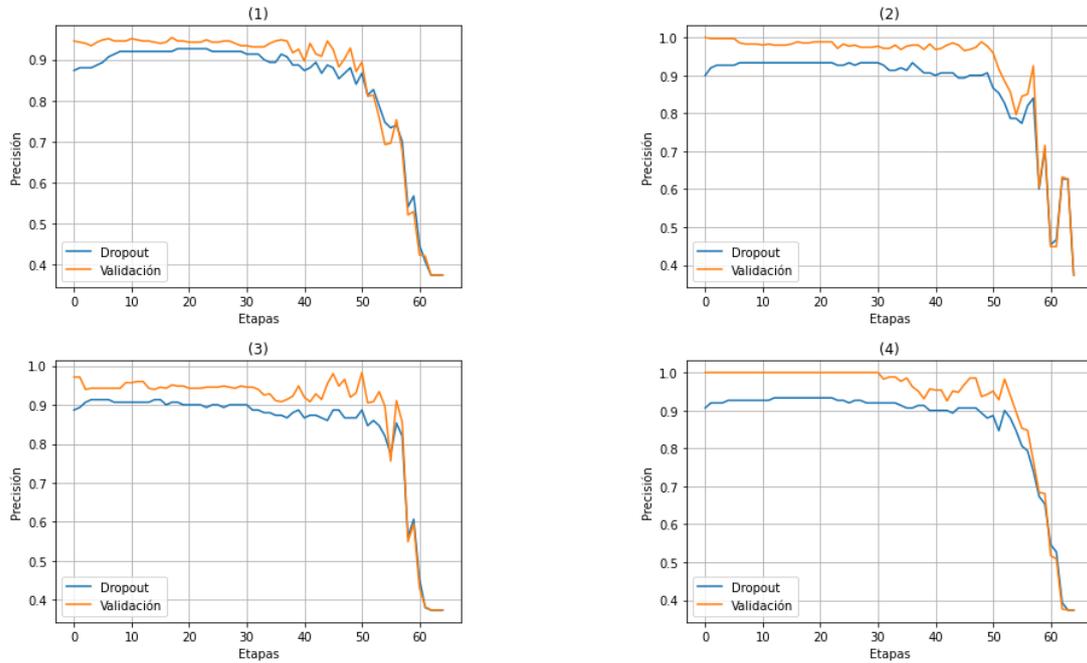
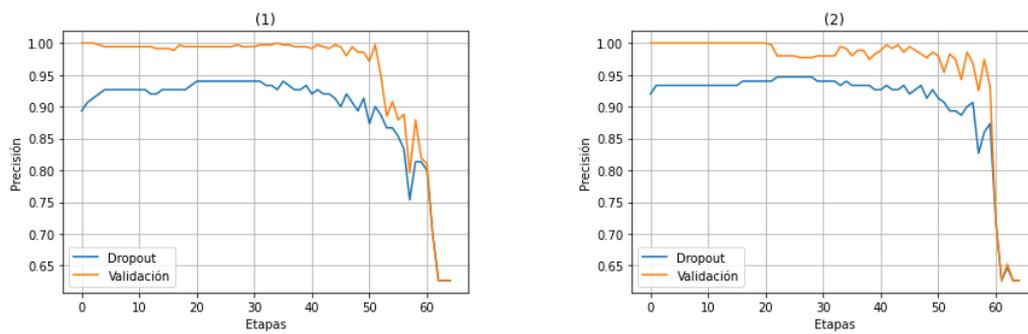


Figura 22: Gráficas para VGG16 en el caso 4



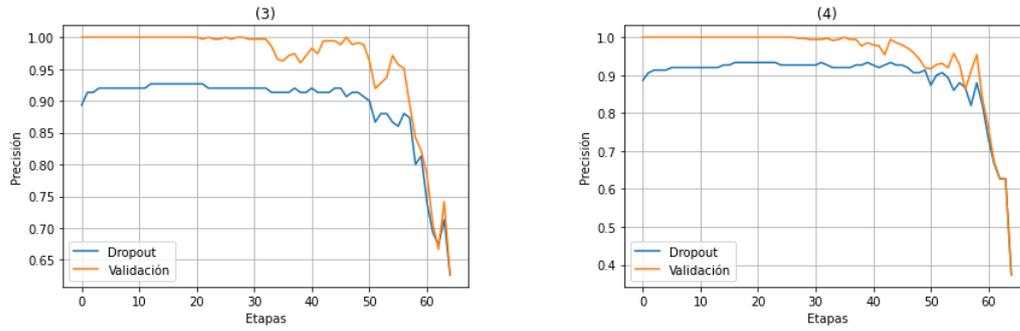


Figura 23: Gráficas para VGG19 en el caso 4

Tabla 6: Resultados para VGG16 en el caso 4

	Valor inicial Dropout	Valor máximo Dropout	Valor inicial Validación	Valor máximo Validación	Nº canales seleccionados
1	0.873	0.927	0.945	0.954	44
2	0.900	0.933	1.000	1.000	58
3	0.887	0.913	0.971	0.983	61
4	0.907	0.933	1.000	1.000	52

Tabla 7: Resultados para VGG19 en el caso 4

	Valor inicial Dropout	Valor máximo Dropout	Valor inicial Validación	Valor máximo Validación	Nº canales seleccionados
1	0.893	0.94	1.000	1.000	44
2	0.920	0.947	1.000	1.000	42
3	0.893	0.926	1.000	1.000	52
4	0.886	0.933	1.000	1.000	48

Caso 5: red VGG16 en diferentes situaciones y con selección de canales en la última capa de convolución.

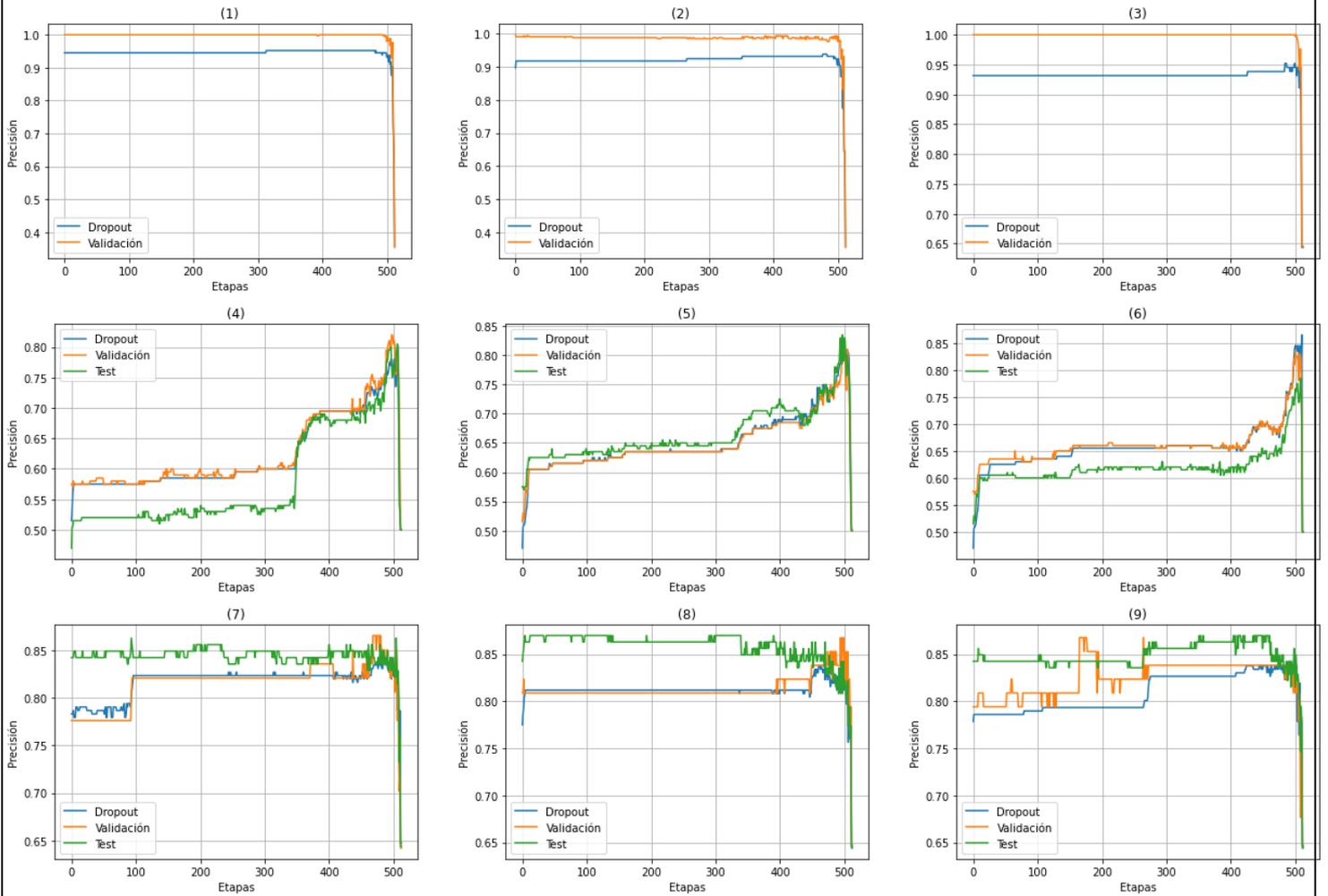


Figura 24: Gráficas de selección de características con la última capa en la red VGG16 en el caso 5. Las gráficas 1, 2, 3 corresponden al caso 3 pero con la última capa en lugar de la primera, lo mismo con las 4, 5, 6 para el caso 1 y las 7, 8, 9 para el caso 2

Tabla 8: Resultados para VGG16 en el caso 5

	Valor inicial Dropout	Valor máximo Dropout	Valor inicial Validación	Valor máximo Validación	Valor inicial Test	Valor máximo Test	Valor estimado Test	Nº canales seleccionados
1	0.945	0.952	1.000	1.000	-	-	-	-
2	0.897	0.938	0.997	0.997	-	-	-	-
3	0.931	0.952	1.000	1.000	-	-	-	-
4	0.515	0.790	0.575	0.802	0.470	0.805	0.805	6
5	0.407	0.815	0.515	0.810	0.575	0.835	0.800	8
6	0.470	0.865	0.575	0.835	0.515	0.785	0.765	12
7	0.783	0.842	0.7764	0.866	0.842	0.863	0.843	32
8	0.775	0.838	0.809	0.868	0.842	0.870	0.849	52
9	0.779	0.838	0.794	0.868	0.842	0.870	0.863	88

3.3 DISCUSIÓN DE LOS RESULTADOS

Caso 1. Este caso corresponde a las redes VGG16 y VGG19 entrenadas con muestras aleatorias de RIM-ONE DL y selección de canales con muestras de ACRIMA. La disponibilidad de 705 muestras de ACRIMA que no tienen nada que ver con el entrenamiento previo de las redes nos ha permitido generar los tres conjuntos que necesitamos, *dropout*, validación y testeo, de 200 muestras cada uno, 100 normales y 100 de sujetos con glaucoma. Las gráficas de la figura 17 muestran los resultados encontrados con cuatro redes VGG16 entrenadas con los mismos datos de RIM-ONE DL. Cada fila representa una de estas redes y cada columna corresponde a considerar como conjunto de test cada uno de los tres en que los que hemos dividido la muestra original de ACRIMA. De esta forma, podemos ver el papel que juega la red en sí y la forma de organizar los conjuntos de *dropout*, validación y testeo. Lo mismo para la red VGG19 en la figura 18.

Lo primero que se observa en todos los casos y en las dos redes es una mejora drástica en el rendimiento de las mismas que en el caso más extremo pasa de valores en torno a 0.5, sin selección de canales, a aproximadamente 0.9. El hecho de que se obtengan valores tan bajos con las redes originales sobre las muestras de ACRIMA puede deberse al *overfitting* que tienen y que hace que les resulte muy difícil reconocer algo que no sea muy parecido a las muestras con las que se entrenó. Resulta interesante ver cómo incluso en un caso como éste la desactivación de algunos canales logra resultados del orden de los que publicaron los investigadores de la Politécnica de Valencia: 0.895 para VGG16 y 0.907 para VGG19, aunque no sabemos qué muestra de test utilizaron y además hay que tener en cuenta que el etiquetado de las imágenes no lo han hecho, por lo tanto, los mismos que etiquetaron las muestras de RIM-ONE DL. En nuestro caso, con la muestra de test se llega a 0.88 – 0.875 pero con *dropout* y validación se llega incluso a 0.910.

Otra cuestión que llama la atención es la influencia que parece tener, en algunos casos, la red en sí considerada e incluso la forma de organizar los conjuntos de *dropout*, validación y testeo, que hace que haya variaciones significativas en el número de canales que se

retendrían en unos casos y en otros. Además, la propia evolución de las curvas resulta, también en algunos casos, un tanto ruidosa con algunos “saltos” muy grandes en precisión por la desactivación de un solo canal. Pensamos que puede deberse por un lado al método de búsqueda hacia atrás que no es del todo óptimo, pero también al hecho ya comentado del *overfitting* que presentan estas redes.

Caso 2. Este caso corresponde a las redes VGG16 y VGG19 entrenadas con muestras de ACRIMA por el grupo de la Politécnica de Valencia y selección de canales con muestras de RIM-ONE DL aleatorias, en concreto 271 muestras de *dropout*, 68 de validación, y 146 de testeo. La primera fila de la figura 19 corresponde a VGG16 y la segunda a VGG19. Las columnas corresponden a considerar diferentes conjuntos de *dropout* y validación coincidiendo con la división en muestras de entrenamiento y validación que se hizo en su momento para entrar las redes con RIM-ONE DL, en concreto en la primera columna se muestran los resultados para el *fold* 1, en la segunda para el *fold* 3 y en la tercera para el *fold* 5. La muestra de test coincide también con la muestra de test que se utilizó para evaluar la precisión de las redes.

De entrada, lo que se observa claramente es una gran asimetría con lo observado con el caso 1. Al intercambiar el papel de las redes y las muestras, ahora no se obtienen valores de precisión tan bajos con las muestras de RIM-ONE DL sin seleccionar ningún canal. Esto puede deberse, posiblemente, a que estas redes no presenten tanto *overfitting* y sean capaces, por lo tanto, de reconocer bastante mejor estas muestras, aunque sean de una base de datos de imágenes diferente. Aun así, se obtiene también una ganancia en precisión bastante significativa pasando de 0.808 a 0.877 en la muestra de test e incluso llegando a un valor máximo de 0.918 que sería del orden de lo alcanzado sobre la misma muestra de test por el grupo de investigación de la ULL, 0.925 para VGG16 y 0.932 para VGG19.

Por otro lado, la evolución de las curvas resulta mucho más estable que en el caso 1 lo que se refleja también en el número de características que se seleccionarían que está en un promedio de 54 por lo que quitando unos pocos canales (partimos de 64), conseguimos una notable mejora del rendimiento de la red. En este comportamiento más estable, posiblemente tenga que ver lo ya comentado de que estas redes parecen no tener tanto *overfitting*, pero habría que hacer más pruebas para estar seguros.

Caso 3: Este caso corresponde a las redes VGG16 y VGG19 entrenadas con muestras de RIM-ONE DL aleatorias y selección de canales con la misma muestra. Las muestras de *dropout* y validación son las muestras de test y entrenamiento, respectivamente, que se utilizaron originalmente con RIM-ONE DL.

Lo que queremos comprobar es qué ocurre con el comportamiento de las redes si intercambiamos el papel de las muestras de entrenamiento y testeo y ver si la selección de canales a partir de la muestra de test original puede conseguir igualar las precisiones

que da la red para ambas muestras y disminuir así el *overfitting* sin que esto suponga una merma en su rendimiento.

Se observa que todas las curvas siguen un comportamiento similar, partiendo de un desfase inicial de hasta unos 10 ptos. y tendiendo a igualarse bastante a partir de cierta etapa de la selección, y quizás el caso más claro sea el de la gráfica 2 de la figura 20 en la que se parte del valor más bajo encontrado de test, 0.897 y se llega a 0.938 después de 23 iteraciones de la búsqueda hacia atrás. Tanto en VGG16 como en VGG19 se llega a un valor máximo de precisión de 0.959 que es superior a cualquiera de los que ya se tenían.

Hay que decir que por el hecho de intentar maximizar el rendimiento de la red $R(c)$, de acuerdo a la expresión ya vista, no tenemos una muestra de test independiente en la que evaluar los resultados. Sin embargo, aunque no se ha incluido en esta memoria, se hicieron experimentos similares en los que en lugar de maximizar $R(c)$ según la expresión ya vista, se maximizó solamente la precisión de la red sobre la muestra de *dropout*, que se utilizó en exclusiva para la selección de los canales, haciendo la evaluación de forma independiente sobre la muestra de validación y se obtuvieron casi idénticos resultados.

Caso 4. Este caso corresponde a las redes VGG16 y VGG19 entrenadas con muestras de infrarrojo y selección de canales con la misma muestra. Las muestras de *dropout* y validación son las muestras de test y entrenamiento, respectivamente, que se utilizaron originalmente con RIM-ONE DL.

Se trata de un caso similar al anterior, pero con las imágenes de infrarrojo. En general, se observa el mismo fenómeno, pero, aunque también se mejoran los valores iniciales de precisión llegando a un máximo de 0.947, cuesta bastante más que las curvas se igualen y si lo hacen, en algunos casos, es para valores de precisión más bajos. Sin un análisis más profundo, es difícil explicar este comportamiento.

Caso 5. Este caso corresponde a la red VGG16 en diferentes situaciones y con selección de canales en la última capa de convolución. Las gráficas 1, 2, 3 corresponden al caso 3 pero con la última capa en lugar de la primera, lo mismo con las 4, 5, 6 para el caso 1 y las 7, 8, 9 para el caso 2.

Hemos incluido este caso como representativo de lo que puede ocurrir si en lugar de la primera capa convolucional de la red se considera la última. Los resultados en las gráficas 1, 2, 3 son similares a los encontrados en el caso análogo en la primera capa. Sin embargo, en las otras gráficas se observa que los valores de precisión alcanzados son un poco inferiores a los que se obtuvieron en la primera capa de la situación equivalente. Esto, quizás, podría explicarse porque los canales en la primera capa constituyen las características más básicas a partir de las cuales se construyen los canales de las sucesivas capas de la red, por lo que en este punto es donde parece que la selección de canales puede ser más efectiva que sobre canales que representan características de un nivel mucho más

alto. En cualquier caso, como en todo lo demás, harían falta más estudios para verificar esta suposición.

Por último, es destacable, también, el número de canales seleccionados que no llega ni a 100 en el mayor de los casos, lo que lleva a pensar que la gran mayoría de los 512 canales de esta capa podrían considerarse como poco útiles y ser descartados.

CAPÍTULO 4. CONCLUSIONES Y LÍNEAS ABIERTAS

4.1. CONCLUSIONES

En este trabajo se ha expuesto un nuevo método de selección de canales en redes neuronales convolucionales utilizando *dropout* para desactivar de forma efectiva algunos de estos canales. La principal conclusión que se puede sacar a partir de las pruebas experimentales realizadas es que se cumple la hipótesis de partida de que es posible mejorar el rendimiento de la red seleccionando un cierto número de canales de algunas de sus capas, aunque nos hemos centrado, básicamente, en la primera capa de las redes VGG16 y VGG19. En todos los casos se ha observado dicha mejora y en algunos de forma muy notable, incluso con muestras diferentes a las que se utilizaron para entrenar las redes. Por lo tanto, parece que este método podría considerarse como una forma de *transfer learning*, pero sin requerir el reentrenamiento de la red como suele ser habitual, lo que además también exige la disponibilidad de muestras de mayor tamaño. Por otro lado, aunque no se ha hablado de tiempos de cómputo en la memoria, los scripts ejecutados en Google Colab tardan en torno a una hora con versiones poco optimizadas para este fin, lo que hace suponer que este método puede llegar a ser, también, bastante más rápido que un entrenamiento convencional. En cualquier caso, se trata de una primera aproximación a esta manera de hacer las cosas y, sin duda, habrá que hacer un estudio en mayor profundidad de todos sus aspectos.

4.2. LÍNEAS ABIERTAS

En este trabajo quedan muchas más líneas abiertas que las que están cerradas. Algunas de ellas podrían ser:

- Utilizar otro tipo de redes más sofisticadas y actuales que las VGG16 y VGG19.
- Utilizar otros conjuntos de datos para poder probar de forma más clara el *transfer learning*.

- Comprobar si estos resultados son generalizables a otro tipo de imágenes y no solo de fondo de ojo.
- Utilizar alguna otra estrategia para la selección de los canales alternativa a la búsqueda hacia atrás.
- Aprovechar la selección de canales para, de alguna manera, obtener una red más reducida de forma similar a lo que se consigue con los métodos de *channel pruning*.

CHAPTER 5. CONCLUSIONS AND FUTURE LINES OF INVESTIGATION

4.1. CONCLUSIONS

In this project, a new method of channel selection in neural convolutional networks using dropout to effectively deactivate some of these channels has been exposed. The principal conclusion we can draw from the carried out experimental tests is that it is possible to improve the efficiency of the network selecting a certain number of channels from one of its layers although we have focused, basically, on the first layer of the network VGG16 and VGG19. In all of the samples, said improvements has been observed and in some of them notably, even with different samples to those used to train the networks. Thus, it seems that this method could be considered as a sort of *transfer learning* but without requiring the network training as it is usual, which also requires the disponibility of samples of bigger size. On the other side, although we have not talked about computing times in the memory, the scripts executed in Google Colab take about one hour with poorly optimized versions to this end, which makes us suppose that this method can also be quicker than conventional trainings. In any case, it is a first aproximation to this way of doing things and, without doubt, a deeper study on all of its aspects will be required in the future.

4.1. FUTURE LINES OF INVESTIGATION

In this project we end up with a higher number of open lines rather than closed ones for further investigation. Some of them could be

- Use other types of networks more refined and current than VGG16 nd VGG19
- Use other groups of data to test more clearly the transfer learning
- Verify if these results are generazible to other types of images and not only the bottom of the eye

- Use another strategy for the selection of the channels as an alternative to backwards search
- Take advantage of the channel selection to, somehow, obtain a network more reduced similarly to those obtained with channel pruning methods.

REFERENCIAS

- [1] Video/Deep Learning In 5 Minutes | What Is Deep Learning? | Deep Learning Explained Simply | Simplilearn Vídeo de youtube disponible en:
<https://www.youtube.com/watch?v=6M5VXKlf4D4&t=15s>
- [2] Video/ ¿Qué es una red neuronal? Parte 1: La Neurona | DotCSV Vídeo de youtube disponible en:
https://www.youtube.com/watch?v=MRlv2lwFTPg&list=PLKw_rA9Pb-ctdSYtQcHROxu3F29BU9SE&index=5&t=0s
- [3] Video/ ¿Qué es una Red Neuronal? Parte 2 : La Red | DotCSV Vídeo de youtube disponible en:
<https://www.youtube.com/watch?v=uwbHOpp9xkc>
- [4]<https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2#7d8a>
- [5] Video/ ¿Qué es una Red Neuronal? Parte 3 : Backpropagation | DotCSV Vídeo de youtube disponible en:
https://www.youtube.com/watch?v=eNlqz_noix8
- [6] François Chollet (2017). *Deep Learning with Python* -
- [7] <https://www.eurocanariosoftalmologica.com/patologias/glaucoma/>
- [8]<https://www.laprovincia.es/sociedad/2019/03/11/canarias-60000-personas-padecen-glaucoma/1155349.html>
- [9] Benyamin Ghojogh, Maria N. Samad, Sayema Asif Mashhadi, Tania Kapoor, Wahab Ali, Fakhri Karray, Mark Crowle (7 May 2019). *Feature Selection and Feature Extraction in Pattern Analysis: A Literature Review*. Disponible en:
<https://arxiv.org/abs/1905.02845>
- [10] Ron Kohavi, George H. John (20 May 1997). *Wrappers for feature subset selection*. Disponible en:
https://www.researchgate.net/profile/Ron_Kohavi/publication/243768287_Wrappers_for_feature_selection/links/55a6b4d108ae51639c573eef/Wrappers-for-feature-selection.pdf
- [11] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov (June 2014). *Dropout: A Simple Way to Prevent Neural Networks from Overfitting*. Disponible en:
http://www.jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf?source=post_page-----
- [12] Yihui He, Xiangyu Zhang, Jian Sun (2017) *Channel Pruning for Accelerating Very Deep Neural Networks*. Disponible en:

http://openaccess.thecvf.com/content_iccv_2017/html/He_Channel_Pruning_for_ICCV_2017_paper.html

[13] Zhuangwei Zhuang, Mingkui Tan, Bohan Zhuang, Bohan Zhuang, Bohan Zhuang, Jing Liu, Yong Guo, Qingyao Wu, Junzhou Huang, Jinhui Zhu (2018). *Discrimination-aware Channel Pruning for Deep Neural Networks*. Disponible en:

<http://papers.nips.cc/paper/7367-discrimination-aware-channel-pruning-for-deep-neural-networks>

[14] Jinyang Guo, Wanli Ouyang, Dong Xu (15 Mar 2020). *Channel Pruning Guided by Classification Loss and Feature Importance*. Disponible en:

<https://arxiv.org/abs/2003.06757>

[15] Alex Labach, Hojjat Salehinejad, Shahrokh Valaee (25 Apr 2019). *Survey of Dropout Methods for Deep Neural Networks*. Disponible en:

<https://arxiv.org/abs/1904.13310>

[16] HaibingWu, Xiaodong Gu (November 2015). *Towards dropout training for convolutional neural networks*. Disponible en:

<https://www.sciencedirect.com/science/article/abs/pii/S0893608015001446>

[17] Yarin Gal, Zoubin Ghahramani (2016). *A theoretically grounded application of dropout in recurrent neural networks*. Disponible en:

<http://papers.nips.cc/paper/6241-a-theoretically-grounded-application-of-dropout-in-recurren>

[18] Andres Diaz-Pinto, Sandra Morales, Valery Naranjo, Thomas Köhler, Jose M. Mossi, Amparo Navea (20 Mar 2019). *CNNs for automatic glaucoma assessment using fundus images: an extensive validation*. Disponible en:

<https://link.springer.com/article/10.1186/s12938-019-0649-y>

[19] Karen Simonyan, Andrew Zisserman (10 Apr 2015). *Very Deep Convolutional Networks for Large-Scale Image Recognition*. Disponible en:

<https://arxiv.org/abs/1409.1556>