

UNIVERSIDAD NACIONAL DE CÓRDOBA
Facultad de Ciencias Exactas, Físicas y Naturales

Detección Temprana de Retinopatía Diabética Mediante
Redes Neuronales Convolucionales Pre-entrenadas,
Unidas a una Capa de Clasificación de Características
en Base a Reglas

Proyecto Integrador para optar al título de grado
Ingeniero Biomédico

Autor: Marcos Ignacio Fabietti
Asesor de tesis: Elmer A. Fernández

Córdoba, 28 de Marzo de 2019

Agradecimientos

Primero, agradecer a mi familia, Constanza, Silvia y Luis, por el soporte continuo, sin ustedes esto no podría haber sido realidad.

Segundo, a mis amigos de toda la vida, por apoyarme siempre más allá de mis largas ausencias, por ser siempre mi cable a tierra.

Tercero, a mis amigos de la facultad, por hacer las horas de cursado más divertidas que las horas entremedio. En particular resalto a Jonatan Mustapic y Lucia Triñanes con quienes recorrí los últimos años, por su gran compañerismo y eterna paciencia para mis chistes malos. Además, a Manuel Robles, debido a todo lo anterior, agregado a insistirme en terminar inteligencia artificial, lo que marco el sendero por el cual recorro ahora.

Cuarto, a mi director Elmer Fernández, por apostar en mí y ser un mentor en todos los sentidos.

Y más importante a mi abuelo Cacho, que me inspiró a seguir por estos largos siete años, esto es por vos. Tomando las palabras de Luke Skywalker, "I am an Engineer, like my grandfather before me".

Muchísimas gracias a todos.

Contenido

Abstract	1
Resumen	2
Capítulo 1 - Introducción.....	3
1.1 Hipótesis.....	4
1.2 Objetivos	4
Capítulo 2 - Anatomía, Patología y Examinación.....	5
2.1 Órganos del sistema visual.....	5
2.1.1 Constitución Anatómica.....	6
2.1.2 Contenido del globo ocular	11
2.2 Retinopatía Diabética	14
2.2.1 Epidemiología	15
2.2.2 Clasificación.....	15
2.2.3 Tratamiento de la Retinopatía Diabética.....	16
2.2.4 Examen en el ser vivo	17
Capítulo 3 – Redes Neuronales	22
3.1 Introducción.....	22
3.1.1 Redes Neuronales	22
3.1.2 Redes Neuronales Convolucionales	25
3.1.2 Entrenamiento de Redes Neuronales Convolucionales	32
3.2 Métodos de Clasificación	35
3.2.1 SVM	35
3.2.2 k–Top Scoring Pairs	36
3.2.3 Análisis estadístico	37
Capítulo 4 - Materiales y métodos	40
4.1 Base de datos	41

4.1.1 Bases de datos de entrenamiento	41
4.1.2 Bases de datos de validación	43
4.2 Hardware y software	44
4.3 Pre-procesamiento.....	45
4.3.1 Enmascarado de la imagen:.....	45
4.3.2 Recorte de imagen	46
4.3.3 Mejora de contraste y dimensionado.	47
4.4 Extracción de características.....	48
4.5 Entrenamiento del clasificador.....	49
4.5.1 SVM	49
4.5.2 k-TSP	51
Capítulo 5 - Resultados y Discusión.....	52
5.1 Diferencia entre modelos	54
5.2 GUI	54
Capítulo 6 Conclusión.....	56
6.1 Trabajos futuros	58
Bibliografía	60
Anexo	66
1. CODIGO de construcción del modelo svm [Matlab]	67
2. Código de función de pre-procesado “readAndPreprocessImagedef” [Matlab]	69
3. CODIGO de construcción del modelo k-TSP [R]	71
4. Código de GUI [Matlab]	72
5. Manual de la aplicación - Readme	80

Lista de Figuras

Figura 1: <i>Geometría ocular</i>	5
Figura 2: <i>Estructura ocular</i>	6
Figura 3: <i>Vascularización de la capa interna</i>	11
Figura 4: <i>Oftalmoscopia directa</i>	18
Figura 5: <i>Oftalmoscopia indirecta</i>	18
Figura 6: <i>Oftalmoscopia con cámara de hendidura</i>	19
Figura 7: <i>Tomografía de coherencia óptica</i>	20
Figura 8: <i>Retinografía digital</i>	21
Figura 9: <i>Representación simplificada del funcionamiento de una neurona en la Red Neuronal con una neurona del cerebro humano</i>	23
Figura 10: <i>Red neuronal simple de tres capas</i>	24
Figura 11: <i>Funcionamiento de una red neuronal convolucional</i>	26
Figura 12: <i>Funcionamiento de una imagen de 32x32 con la primera capa convolucional</i>	28
Figura 13: <i>Top-1 de precisión sobre el conjunto Imagenet frente a la cantidad de operaciones requeridas para un feed-forward</i>	29
Figura 14: <i>Arquitectura de Alexnet, GoogleNet y VGG-16 Net</i>	31
Figura 15: <i>Diagrama de la transferencia de aprendizaje</i>	32
Figura 16: <i>Fine tuning</i>	33
Figura 17: <i>Diagrama de la extracción de características</i>	33
Figura 18: <i>Características de una imagen a lo largo de distintos niveles de una red convolucional</i>	35
Figura 19: <i>Metodología propuesta</i>	40
Figura 20: <i>Fondo de ojo con retinopatía diabética (izq.) y Normal (der.)</i>	44
Figura 21: <i>Imagen cruda</i>	45
Figura 22: <i>Imagen binarizada</i>	46

Figura 23: <i>Imagen estandarizada</i>	47
Figura 24: <i>Imagen realzada</i>	48
Figura 25: <i>Visualización de la optimización de los hiperparametros gamma (kernel scale) y costo (box constraint)</i>	50
Figura 26: <i>Curvas ROC obtenidas. Modelos de SVM son color rojo y k-TSP de color azul.</i>	54
Figura 27: <i>Interfaz de usuario</i>	55
Figura 28: <i>Numero de publicaciones científicas por año que incluyen el término “nueva arquitectura de red neuronal”</i>	59

Lista de Tablas

Tabla 1: <i>Etapas de la retinopatía diabética</i>	16
Tabla 2: <i>Matriz de confusión</i>	38
Tabla 3: <i>Bases de datos de fondos de ojos disponibles en internet, sin patologías o con retinopatía diabética</i>	41
Tabla 4: <i>Estructura del conjunto de datos de entrenamiento y validación</i>	43
Tabla 5: <i>Resultados de los métodos de machine learning</i>	53
Tabla 6: <i>Comparación de resultados</i>	58

Abstract

The worldwide incidence of diabetic retinopathy is growing. However, the blindness outcome could be prevented in almost 90% of cases by early detection, where automatic method may play a crucial role. Deep learning strategies emerge as a candidate solution for this problem, but building such approaches from scratch requires a high number of training data usually scarce in most eye clinics. To overcome this limitation, a hybrid strategy is proposed by concatenating a pre-trained Convolutional Neural Network (CNN) as a feature extractor layer, with another classification method. Three pre-trained CNNs and two classification methods were evaluated. They were trained and tested with 1642 and 862 retinal images respectively. It was found that features from the AlexNet CNN concatenated with a support vector machine or k-Top Scoring Pairs method achieves the best results reaching F-Score values of 86% and 83%. Furthermore, the kTSP used less than 1% of the CNN features since it simultaneously performs feature selection and classification providing an easily interpretable rule based output.

Keywords: Convolutional Neural Network, Machine Learning, k-Top Scoring Pairs, Support Vector Machine

Resumen

La incidencia mundial de retinopatía diabética está creciendo. Sin embargo, la ceguera podría prevenirse en casi el 90% de los casos mediante la detección temprana, donde métodos de detección automáticos pueden jugar un papel crucial. Las estrategias de aprendizaje profundo surgen como una posible solución para este problema, pero desarrollar estos enfoques desde cero requiere un alto número de datos de entrenamiento, generalmente escasos en la mayoría de las clínicas oftalmológicas. Para superar esta limitación, se propone una estrategia híbrida mediante la concatenación de una red neuronal convolucional (CNN) pre-entrenada como una capa extractora de características, con otro método de clasificación. Se evaluaron tres CNN pre-entrenados y dos métodos de clasificación. Fueron entrenados y validados con 1642 y 862 imágenes retinales respectivamente. Se encontró que las características de la CNN de AlexNet concatenadas con una máquina de vectores de soporte o el método k-Top Scoring Pairs (kTSP) logran los mejores resultados al alcanzar valores de F-Score de 86% y 83% respectivamente. Por otra parte, el kTSP utilizó menos del 1% de las funciones de CNN, ya que realiza simultáneamente la selección y clasificación de funciones, lo que proporciona una salida basada en reglas fácilmente interpretable.

Palabras Clave: Convolutional Neural Network, Machine Learning, k-Top Scoring Pairs, Support Vector Machine

Capítulo 1 - Introducción

La retinopatía diabética (RD) es la tercera causa de ceguera en el mundo. En Latinoamérica, 40% de los pacientes diabéticos poseen algún grado de retinopatía [1]. De acuerdo a la organización mundial de la salud, el número de personas con RD va a crecer de 126.6 millones en 2010 a 191.0 millones para el año 2030 [2]. Sin embargo, si es detectado tempranamente, la ceguera puede ser prevenida en más de 90% de los pacientes [3].

Actualmente, la principal estrategia de diagnóstico recae en oftalmólogos especializados en diagnosticar fondos de ojo digitalizados. El proceso de diagnóstico consiste en detección de sutiles detalles, una tarea que consume mucho tiempo como método de selección en una clínica de ojo atareada. Entonces, tanto el alto consumo de tiempo y las altas tasas de prevención de ceguera mediante detección temprana establecen las ventajas del desarrollo de sistemas automatizados para facilitar ambos [4].

Entre otros métodos de inteligencia artificial, la popularidad de redes neuronales artificiales de Deep Learning en los últimos años ha permitido el desarrollo de exitosas aplicaciones el campo de reconocimiento y clasificación de imágenes, evitando algunos pasos como segmentación manual y complejos pasos de pre-procesado. El Deep Learning es un conjunto de métodos de aprendizaje de representación obtenidos al componer módulos simples que transforman una representación en un nivel en una representación en un nivel más alto, ligeramente más abstracto, que ha llevado al desarrollo de redes neuronales muy complejas para abordar varios problemas por medio de una nueva representación de la estructura funcional de las unidades neuronales y que permite el entrenamiento eficiente de millones de conexiones neuronales [5]. En el campo del análisis de imágenes, existen al menos tres opciones principales para enfrentar los problemas de clasificación: a) desarrollar y entrenar una ANN desde cero (un enfoque recientemente aprobado por la FDA [6] b) usar una ANN construida para un propósito diferente y modificar (volver a entrenar) los pesos por medio de un nuevo conjunto de imágenes para el propósito específico (transferencia de la clasificación de aprendizaje) o c) utilizar ANN preformado como modelo de extracción de características generales y vincularlo a una nueva capa construida a partir de otro método de aprendizaje automático. En el primer caso, debido a que las RNA profundas de aprendizaje generalmente tienen millones de pesos de conexión, lo que hace necesario tener millones de imágenes correctamente etiquetadas, en el segundo caso, con el fin de actualizar con éxito una ANN existente, la cantidad de imágenes requeridas están

en el orden de cientos de miles [7]. Estas dos situaciones generalmente están lejos de la mayoría de las clínicas oftalmológicas. El tercer caso, aprovecha una ANN entrenada con éxito en enormes conjuntos de datos de imágenes para un propósito diferente, como aquellas Convolutional Neural Networks (CNN) entrenadas para resolver el desafío de imagen ImageNet Large Scale Visual Recognition Competition (ILSVRC) [8].

1.1 Hipótesis

Se propone que un método eficiente de clasificación de RD puede ser construido mediante un enfoque híbrido concatenando una red neuronal pre-entrenada para extracción de características con un método de machine learning (support vector machine y k-Top Scoring Pairs), usando menos información que otros métodos, creando la oportunidad de hacerlos aplicables a distintos escenarios médicos.

1.2 Objetivos

El objetivo general es construir un sistema automático clasificador de RD a partir de pocos datos.

Los objetivos específicos son:

- Explorar diversos métodos de machine learning
- Explorar las distintas estructuras de CNN
- Desarrollar un sistema eficiente, el cual:
 1. Devuelva un diagnóstico “Normal” o “Retinopatía Diabética”.
 2. Sea eficiente en tiempo de computo
 3. Cuyas métricas estén a la par del estado del arte y un oftalmólogo entrenado.
 4. Posea una interfaz (GUI) para un usuario proveniente del área de la salud.
 5. De resultados interpretables, ya sea con gráficos o probabilidad del diagnóstico.

Capítulo 2 - Anatomía, Patología y Examinación

2.1 Órganos del sistema visual

A continuación, se detallan los elementos que constituye los órganos del sistema visual y sus estructuras, como está descrito en [9].

El sentido de la vista está asegurado por un órgano receptor, el ojo, un sistema de conducción, las vías ópticas y un área de procesamiento, el área visual de la corteza cerebral correspondiente.

El ojo es un órgano par situado en la cavidad orbitaria, protegido por los párpados y por la secreción de las glándulas lagrimales. Los movilizan un grupo de músculos extrínsecos dirigidos por los nervios motores del ojo.

El globo ocular (*Figura 1*) tiene forma esférica, levemente aplanada de arriba hacia abajo. Esta forma es regular, pero en la parte anterior existe una saliente también regular formada por la córnea, que es transparente. El globo ocular, par y simétrico ocupa el tercio anterior de la órbita. Está constituido por dos segmentos de esfera de diámetros diferentes.

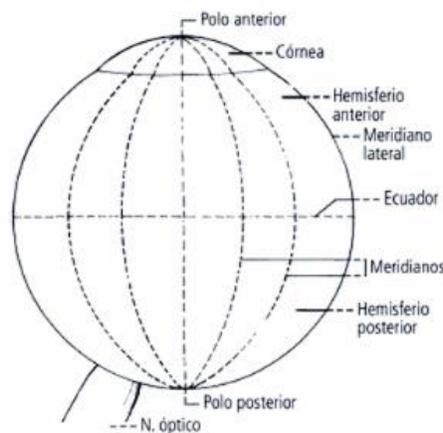


Figura 1: Geometría ocular. (Extraída de Latarjet, M. 2005).

2.1.1 Constitución Anatómica

Paredes

Corresponden a las envolturas del globo ocular (*Figura 2*). Son tres, concéntricas, que de la periferia hacia el interior se denominan capa externa, fibrosa, constituida por la esclerótica y la córnea. Capa media, vascular, comprende: la coroides, el cuerpo ciliar y el iris. Capa interna, nerviosa, la retina.

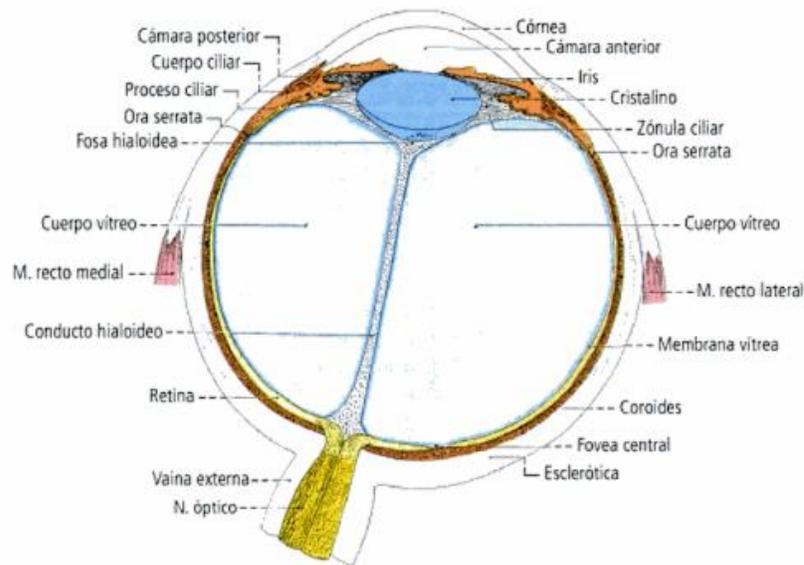


Figura 2: Estructura ocular (Extraída de Latarjet, M. 2005).

Capa fibrosa del globo ocular

Resistente e inextensible, protege las capas subyacentes y el contenido del globo. Comprende dos segmentos: un segmento posterior, la esclerótica y un segmento anterior y menos extenso y transparente, la córnea.

Esclerótica

Constituye los 5/6 posteriores de la capa periférica. Es opaca a la luz y tiene forma de calota. Se continua adelante con la córnea transparente. Su superficie externa es convexa, blanco azulada, se vuelve amarillenta con la edad. Se relaciona con la vaina del globo ocular, de

la que está separada por un espacio supra esclerótico por el que circula linfa. La esclerótica da inserción a los músculos motores del ojo. Su superficial interna, cóncava y de color oscuro, se relaciona con la coroides, a la que está unida por los vasos y nervios que la atraviesan, así como por una capa de tejido conjuntivo: la lámina fusca.

La esclerótica presenta un foramen posterior, un foramen anterior y numerosos orificios pequeños. El foramen posterior corresponde a la travesía del nervio óptico. Está tallado a bisel, con un vértice anterior. A ese nivel, el nervio está formado por fibras nerviosas libres que atraviesan una membrana fibrosa, perforada por múltiples forámenes: la lámina cribosa de la esclerótica. Alrededor del foramen posterior se hallan numerosos orificios pequeños por donde se pasan los vasos y los nervios ciliares cortos posteriores. De cada lado se encuentra, en la parte mediana, un foramen para los vasos ciliares largos posteriores. El foramen anterior corresponde a la córnea. Amplio y ovalado, su circunferencia está tallada a bisel a expensas de la capa interna. El punto de la unión se llama limbo de la córnea y está ocupada por un conducto circular, el seno venoso de la esclerótica. Este conducto tiene el valor de una vía linfática que se comunica atrás con la cámara anterior del globo ocular y delante con las venas de la esclerótica.

Cornea

Es una membrana transparente, encastrada en la abertura anterior de la esclerótica. Constituye el segmento anterior de la capa externa. Es un segmento de esfera hueca cuyo radio es menor que el de la esclerótica, lo que explica que sobresalga hacia adelante. Su diámetro transversal es algo mayor que su diámetro vertical. Se distinguen en ella: una cara anterior, convexa y lisa, humedecida por las lágrimas, que se encuentra en contacto con el aire exterior, pero esta oculta por los párpados cuando estos se ocluyen, una cara posterior, cóncava, que constituye la pared anterior de la cámara anterior del ojo, y una circunferencia, unida a la esclerótica a nivel del limbo de la córnea. La cornea está encastrada en la esclerótica mediante una superficie en bisel, oblicua hacia atrás. Este bisel avanza más arriba y abajo que a los lados. Los tejidos corneales y escleróticos están fusionados.

Vasos y nervios de la capa externa

La esclerótica está irrigada por arterias que provienen de los vasos ciliares cortos anteriores y posteriores. Las venas terminan en las venas coroideas y en las venas ciliares

anteriores. Los nervios son ramas de los nervios ciliares. La cornea no tiene vasos sanguíneos ni linfáticos, pero esta ricamente innervada y su sensibilidad es muy grande.

Capa vascular del globo ocular

Esta aplicada a la cara interna de la capa fibrosa a la cual se adhiere desde el polo posterior hasta un punto situado a 1mm por detrás del limbo de la córnea. Desde allí se dirige en sentido perpendicular hacia el eje de del globo ocular, formando un tabique frontal perforado en el centro. El segmento posterior que corresponde a la esclerótica está interrumpido por una línea festoneada, situada delante del ecuador del globo ocular, la ora serrata. Esta formación divide la capa media en dos partes: una parte posterior, amplia y vascular, la coroides; la otra anterior, menos amplia y musculo vascular; el cuerpo ciliar y el iris.

Coroides

Representa los $\frac{3}{4}$ de una esfera hueca. Es una membrana más espesa atrás que adelante. Su cara externa convexa se aplica contra la cara interna de la esclerótica, a la cual está unida por la lámina fusca y por vasos y nervios que van de una a otra. La cara interna, lisa y oscura, responde a la retina, pero sin adherirse a ella. La coroides presenta un foramen posterior, donde se adhiere a la esclerótica, que es atravesado por el nervio óptico. El foramen anterior, marcado por la ora serrata, se continua con el cuerpo ciliar. Esta esencialmente constituido por una capa de vasos voluminosos, es especial venosos.

Cuerpo ciliar

El cuerpo ciliar se interpone entre la coroides y a la circunferencia del iris. Forma un anillo espeso situado entre la zona ciliar de la retina, atrás, y de la esclerótica, adelante. Es una formación musculo vascular que comprende: una parte anterior, el musculo ciliar, y una parte posterior, vascular, los procesos ciliares.

Iris

Constituye la parte más anterior de la capa vascular. Tiene forma de disco vertical perforado en su centro por la pupila. La cara anterior constituye la pared posterior de la cámara

anterior del globo ocular. Es coloreada, de forma variable, con una zona periférica más pálida y una zona central más oscura. Está marcada por estrías radiadas que corresponden a los vasos del iris. La cara posterior, de color oscuro, es cóncava. Toma contacto en su centro con el lente, separándose de él en la periferia por delante de los procesos ciliares, donde forma la pared anterior de la cámara posterior del ojo a 1 mm por detrás del limbo de la córnea. La unión iridocorneal marca el ángulo de la cámara anterior y corresponde a la circunferencia mayor del iris. La fijación está asegurada por el ligamento del ángulo iridocorneal y por los vasos que unen el iris a la zona ciliar. La circunferencia interna circunscribe la pupila. Sus dimensiones son variables, pues el iris es un órgano muscular, verdadero esfínter que estrecha o dilata la pupila. El iris estrecha o dilata la pupila, lo que aumenta o disminuye la cantidad de luz que llega al cristalino. La miosis se caracteriza por la pupila estrechada y la midriasis por la pupila dilatada. Los nervios provienen del ganglio ciliar o el nervio nasociliar, estos nervios contienen fibras motoras para el músculo ciliar y los músculos del iris, fibras sensitivas para el epitelio anterior del iris y fibras vasomotoras y tróficas.

Vasos y nervios de la capa media

Las arterias provienen de las arterias ciliares cortas posteriores, cuyo territorio alcanza la ora serrata y de las dos arterias ciliares largas anteriores lateral y medial, que atraviesan la esclerótica y llegan a la gran circunferencia del iris donde se anastomosan formando el círculo arterial mayor. Ramas de este círculo forman la circunferencia menor y rodean la pupila. Las venas siguen la disposición de las arterias, se agrupan en los procesos ciliares para formar los plexos venosos coroideos. Estos desembocan en una red venosa coroidea, tributaria de la vena oftálmica.

Capa interna del globo ocular

La retina se encuentra extendida en la cara profunda de la capa vascular, en ella está el cuerpo de la neurona ganglionar, origen del nervio óptico. Las fibras de este nervio se originan en la retina, convergen hacia el disco del nervio óptico y atraviesan sucesivamente a la coroides, luego la esclerótica, antes de constituir un cordón nervioso que emerge de la esclerótica debajo y medial al polo posterior del globo ocular. La retina constituye el órgano receptor de las impresiones luminosas. Se extiende desde el nervio óptico hasta la cara posterior del iris, pero

en contacto con la zona ciliar y el iris, pero en contacto con la zona ciliar y el iris, se adelgaza y pierde sus caracteres sensoriales. Se le describe dos partes: porción óptica y porción ciega.

Porción óptica

Se extiende desde el nervio óptico hasta la ora serrata. Su cara lateral, de color oscuro, convexa, se aplica contra la coroides, sin adherirse a ella. Su cara medial, cóncava, esta moldeada sobre el cuerpo vítreo, al que tampoco se adhiere, es rosada, vascular, lisa y presenta en su región posterior dos superficies particulares:

1. El disco óptico (papila), blanquecino, deprimido en su centro (excavación del disco). Está situado a 3 mm medialmente y a 1mm por encima del polo posterior del globo ocular. Corresponde a la expansión del nervio óptico y a la llegada de los vasos centrales de la retina. El disco óptico es el punto ciego de la retina.

2. La mácula lútea, que ocupa el polo posterior del globo ocular, es una superficie deprimida en su centro (fóvea central), bordeada por una retina espesa. La macula lútea, es el punto que percibe el máximo de rayos luminosos.

Porción ciega

La retina se halla aquí reducida a una capa de células (pigmentarias) que se adhiere a la cara posterior del musculo y de los procesos ciliares (porción ciliar) y luego a la cara posterior del iris (porción iridiana).

Vascularización de la capa interna

La arteria central de la retina (*Figura 3*), rama de la arteria oftálmica originada en la carótida interna, sigue al nervio óptico, emerge a nivel del disco óptico y se divide en dos ramas, ascendente y descendente, que se expanden en dirección a la ora serrata en ramas mediales y laterales. Esta red arterial es independiente de los vasos de la coroides. No obstante, muy finas y raras anastomosis reúnen los dos sistemas. Las venas convergen en el disco óptico para formar la vena central de la retina, que sigue al nervio óptico y va a drenar en la vena oftálmica.

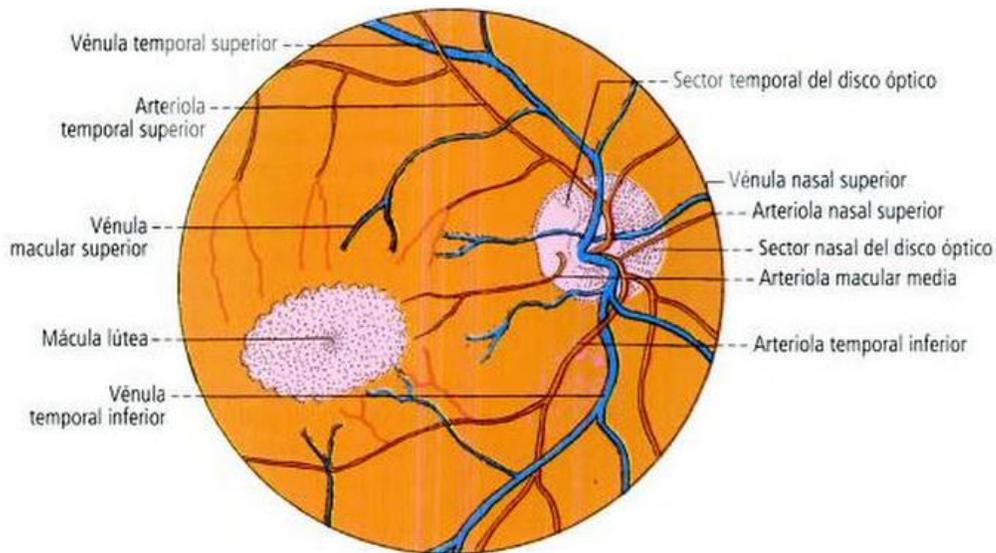


Figura 3: Vascularización de la capa interna. (Extraída de Latarjet, M. 2005).

2.1.2 Contenido del globo ocular

Está constituido por los órganos y medios transparentes o refringentes, que la luz atraviesa para llegar a la retina, los rayos se convergen en un punto exacto de esta. Estos son el lente, el cuerpo vítreo y las cámaras del ojo con el humor acuoso.

Lente (cristalino)

Es una lente biconvexa, transparente y elástica, situada verticalmente entre el iris adelante y el cuerpo vítreo atrás, se mantiene en su posición por una membrana elástica periférica, la zona ciliar. El lente tiene un diámetro de 1 cm y un espesor de 5mm en promedio. La cara anterior, convexa, se relaciona con la pupila en el centro y con la cámara posterior del globo ocular, con el iris y con los procesos ciliares en la periferia. Está separada de la córnea por una distancia de 2 a 2,5mm. La cara posterior, más convexa aun, responde a la cara anterior del cuerpo vítreo. El ecuador es redondo. Las caras se reúnen allí en un ángulo romo en relación con la zonula ciliar.

Constitución anatómica

La lente está compuesta por una envoltura delgada y por fibras dispuestas en capas concéntricas en el seno de una masa amorfa. Esta se espesa en la superficie para formar los radios de una estrella. Perfectamente transparente hasta la edad adulta, puede volverse opaco en individuos de cierta edad.

Aparato suspensor – zonula ciliar

Está formado por un conjunto de fibras transparentes que van desde la cara interna del cuerpo ciliar hasta la periferia ecuatorial del lente. Se disponen en fibras anteriores, medias y posteriores; estas últimas unen el cuerpo ciliar al vítreo. Estas fibras forman una membrana circular cuya parte periférica está interpuesta entre los procesos ciliares adelante y el cuerpo vítreo atrás. En su parte central. En contacto con el ecuador y el lente, las fibras se separan dejando entre ellas un espacio, el espacio zonular. Este es un espacio con humor acuoso que se comunica hacia adelante con la cámara posterior del globo ocular. Bajo la influencia de la contracción o de la relajación de los músculos ciliares, la zonula ciliar se halla tensa o laxa. Esta acción es transmitida al lente, cuyas curvaturas pueden ser aumentadas o disminuidas según el tipo de visión, el lente es más convexo en la visión próxima y más aplanado en la visión alejada, es el fenómeno de la acomodación. El lente no tiene vasos ni nervios propios. Lo nutren los líquidos emanados de los procesos ciliares que circulan entre las fibras del lente.

Cámara vítrea – cuerpo vítreo

La cámara vítrea o postrema se ubica por detrás del lente. Está ocupada por el cuerpo vítreo. Este es el más voluminoso de los medios transparentes del ojo. Ocupa los 2/3 posteriores del globo ocular. Está comprendido entre el lente y la zonula ciliar adelante y la retina por atrás. Su parte anterior está deprimida en su centro por el lente: es la fosa hialoidea. Otra depresión se sitúa frente al disco óptico de la retina. La cámara vítrea está limitada por una envoltura que rodea al humor vítreo.

Membrana vítrea

Resulta de la condensación de las capas periféricas del humor vítreo. no contiene ninguna célula. Se invagina en algunos puntos de la cámara vítrea.

Humor vítreo

Entra en la composición del cuerpo vítreo. Es una masa de aspecto gelatinoso bastante semejante a la clara de huevo, marcada por un sistema de hendiduras estrechas. Esta sustancia está organizada en superficie en círculos concéntricos, que reúnen fibras delgadas y refringentes, sembradas de células. La sustancia fundamental es gelatiniforme y transparente. El humor vítreo está travesado de un lado a otro por el conducto hialoideo. Este conducto atraviesa al cuerpo vítreo de adelante hacia atrás, desde el polo posterior del lente hasta el disco óptico

Cámaras del globo ocular y humor acuoso

Las cámaras del globo ocular con humor acuoso ocupan el espacio comprendido entre el lente y la córnea. Este espacio se halla dividido por el iris en dos partes, la cámara anterior y la cámara posterior.

La cámara anterior es el espacio comprendido entre la cara posterior de la córnea y la cara anterior del iris. Su diámetro anteroposterior varía entre 2 y 2,5 mm. Su circunferencia está marcada por el ángulo iridocorneal, donde el iris se adhiere al limbo esclerocorneal a través de la red trabecular que llena el ángulo iridocorneal. Este ángulo de la cámara anterior se halla próximo al seno venoso de la esclerótica. La cámara posterior está situada detrás del iris. Constituye un anillo estrecho posterior al iris y anterior al lente y la zonula. En su centro, el borde interno de la pupila se halla en contacto con el lente, mientras que, en la periferia, la circunferencia mayor corresponde a la base de los procesos ciliares. Las cámaras anterior y posterior se comunican a través de la pupila.

El humor acuoso es un líquido incoloro, perfectamente límpido, que llena las dos cámaras del globo ocular. Proviene por trasudación de los vasos de los procesos ciliares y del iris. Puede fluir hacia el espacio retrozonular a partir de la cámara posterior y participar en la nutrición del lente. Pasando a la cámara anterior por el orificio pupilar, puede concentrarse en el ángulo de la

cámara anterior y drenar en el seno venoso de la esclerótica, a partir de la cual es drenado por las venas de la esclerótica.

2.2 Retinopatía Diabética

La diabetes mellitus (DM) es una epidemia mundial con una morbilidad significativa. La retinopatía diabética (RD) es la complicación microvascular específica de la DM y afecta a 1 de cada 3 personas con DM. La RD sigue siendo una de las principales causas de pérdida de visión en las poblaciones adultas que trabajan. Se ha estudiado que los pacientes con niveles graves de RD tienen una calidad de vida más pobre y niveles reducidos de bienestar físico, emocional y social, y utilizan más recursos de atención médica. Los estudios epidemiológicos y los ensayos clínicos han demostrado que el control óptimo de la glucosa en sangre, la presión arterial y los lípidos en la sangre pueden reducir el riesgo de desarrollar retinopatía y retrasar su progresión. El tratamiento oportuno con fotocoagulación con láser y, cada vez más, el uso adecuado de la administración intraocular de los inhibidores del factor de crecimiento endotelial vascular (VEGF) puede prevenir la pérdida visual en la retinopatía que amenaza la visión, en particular el edema macular diabético (DME). Debido a que la pérdida visual puede no estar presente en las primeras etapas de la retinopatía, la evaluación regular de las personas con diabetes es esencial para permitir una intervención temprana [10].

Las siguientes descripciones de la epidemiología, clasificación y tratamiento de la patología fueron extraídas de Actualización de la Guía clínica de Retinopatía Diabética para Latinoamérica [1].

2.2.1 Epidemiología

En América Latina, hasta un 40% de los pacientes diabéticos tiene algún grado de RD y el 17% requiere tratamiento. La prevalencia de RD está relacionada con el diagnóstico de diabetes mellitus (DM). Si el diagnóstico es precoz, en caso de países con mayores recursos o con sistema de atención primaria, la prevalencia de RD será más baja, pero si el diagnóstico es tardío, en caso de países con pocos recursos, la prevalencia es más alta. En meta-análisis de estudios de gran escala se muestra que aproximadamente un tercio de las personas con DM sufrirá de RD, y aproximadamente una tercera parte de ellas o el 10% de las personas con DM sufrirá de RD que amenaza la visión, que requiere tratamiento.

Los principales factores de riesgo para el desarrollo de una retinopatía diabética son:

1. Duración de la enfermedad
2. Mal control metabólico (Hiperglicemia)
3. Hipertensión arterial
4. Hiperlipidemia
5. Desconocimiento de la enfermedad.

2.2.2 Clasificación

Existe una clasificación simplificada, elaborada por un grupo de expertos internacionales convocados por la Academia Americana de Oftalmología (AAO), que permite evaluar el estado del fondo de ojo y con ello, definir el manejo, tratamiento y controles de un paciente, considerando las sugerencias del Consejo Internacional de Oftalmología. La clasificación clínica internacional de la RD distingue las siguientes etapas:

Etapas de retinopatía diabética	Hallazgos observables en oftalmoscopia de ojos dilatados	Referencia
Sin retinopatía aparente	No hay anomalías	Revisión en 1 o 2 años
RD no proliferativa leve	Solo micro aneurismas	Revisión en 1 o 2 años si hay buen control metabólico
RD no proliferativa moderada	Más que solo micro aneurismas, pero menos que RD severa no proliferativa	Revisión en 6 meses-1 año o referir al oftalmólogo
RD no proliferativa severa	Cualquiera de los siguientes: <ul style="list-style-type: none"> • Hemorragias intraretinianas (menor que 20 en cada cuadrante) • Arrosareamiento venoso en 2 cuadrantes • Anormalidades vasculares intrarretinianas en 1 cuadrante • Sin signos de RD proliferativa 	Referir al oftalmólogo semiurgente
RD proliferativa	RD no proliferativa severa y 1 o más de: <ul style="list-style-type: none"> • Neo vascularización • Hemorragia vítrea/pre retiniana 	Referir al oftalmólogo urgente (1 mes)

Tabla 1: Etapas de la retinopatía diabética.

2.2.3 Tratamiento de la Retinopatía Diabética

Es necesario un control metabólico estricto de la hemoglobina glucosidada (HbA1c < 7%), de la hipertensión arterial y de los lípidos (reducción LDL-C) que retardan la progresión de una retinopatía diabética. Ninguno de los tratamientos médicos propuestos substituyen el tratamiento de fotocoagulación con láser.

El tratamiento de una retinopatía mediante fotocoagulación con láser está indicado en una RD proliferativa y en la fase no proliferativa severa de pacientes con diabetes tipo 2. Se estima que un 80% de los casos de RD proliferativa tratados con fotocoagulación en forma oportuna y adecuada, logran detener la progresión y en caso de alto riesgo, el tratamiento reduce 50% la pérdida de visión severa. Otros factores a considerar para un tratamiento precoz son la falta de adherencia a controles, cirugía de catarata inminente, embarazo, nefropatía y el estado del ojo contralateral, afectado de una retinopatía proliferante o que no respondió al tratamiento con láser.

La clave es la OPORTUNIDAD en la detección de una retinopatía y se debe fomentar la PREVENCIÓN PRIMARIA que es efectiva y de bajo costo.

2.2.4 Examen en el ser vivo

El examen externo del globo ocular muestra una parte de la esclerótica, la totalidad de la córnea, a través de esta el iris y la pupila y a través de ella el lente. La pupila puede ser dilatada mediante el uso de atropina, lo que ensancha el campo de la cámara posterior. Numerosos métodos permiten examinar de manera precisa y profunda los diferentes elementos del globo ocular: oftalmoscopia, tomografía de coherencia óptica (TCO), oftalmoscopio láser de barrido y retinografía digital [9].

Oftalmoscopia

La evaluación con oftalmoscopia lleva aproximadamente de 5 a 10 minutos. Hay diferentes tipos de oftalmoscopia:

Oftalmoscopia directa

El estudio se realiza con el paciente sentado en una habitación oscura. El oftalmólogo llevará a cabo el examen proyectando un rayo de luz a través de la pupila, con el uso de un instrumento llamado oftalmoscopio. Un oftalmoscopio es aproximadamente del tamaño de una linterna. Tiene una luz y lentes diminutos y diferentes que le permiten al proveedor observar la parte posterior del globo ocular (*Figura 4*) [11].

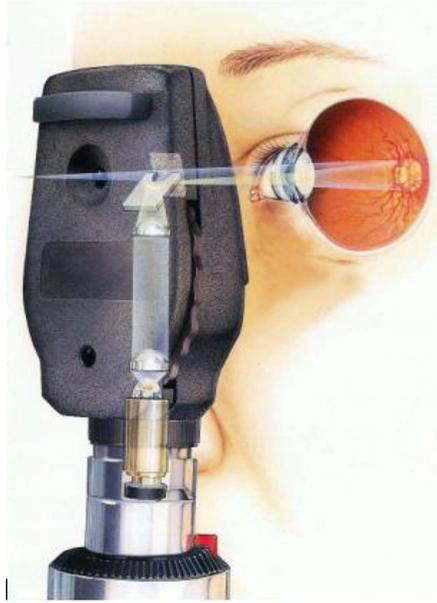


Figura 4: Oftalmoscopia directa (Extraída de http://www.medicinapreventiva.com.ve/fondo_ojo.htm).

Oftalmoscopia indirecta

El estudio se realiza con el paciente acostado o sentado en una posición semi-inclinada. El oftalmólogo le mantendrá el ojo abierto al paciente, mientras proyecta una luz muy brillante hacia el interior de este, usando un instrumento que se lleva puesto en la cabeza. El profesional visualizará la parte posterior del ojo a través de un lente manual sostenido cerca del ojo (Figura 5) [11].



Figura 5: Oftalmoscopia indirecta (Extraída de <http://infolentesperu.blogspot.com/2017/09/gran-remate-stock-limitado-03-lentes.html>).

Oftalmoscopia con lámpara de hendidura

El paciente se sentará en una silla con el instrumento colocado al frente. Se le pedirá que apoye la barbilla y la frente en un soporte para mantener la cabeza firme. El oftalmólogo usará la parte del microscopio de la lámpara de hendidura y un lente diminuto que se coloca al frente del ojo (*Figura 6*). El profesional puede ver casi lo mismo con esta técnica que con la oftalmoscopia indirecta, pero con mayor amplificación [11].

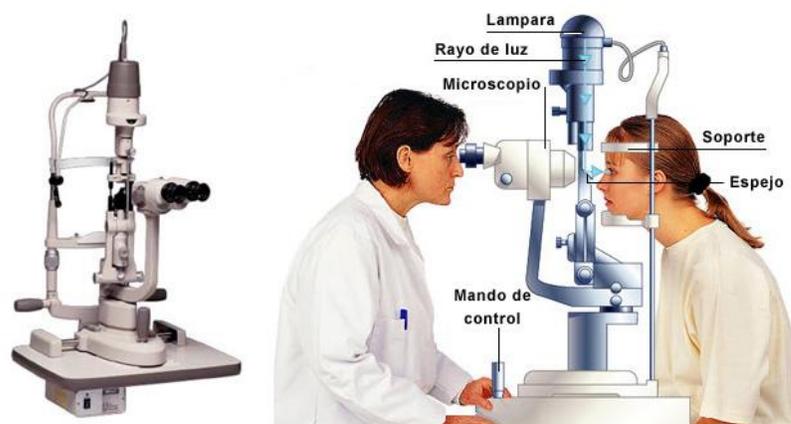


Figura 6: Oftalmoscopia con cámara de hendidura (Extraída de <http://www.pardell.es/lampara-hendidura.html>).

Tomografía de Coherencia Óptica (TCO)

La tomografía de coherencia óptica (OCT, por sus siglas en inglés) es una técnica de imágenes que utiliza luz coherente para capturar imágenes de dos y tres dimensiones con resolución micrométrica desde el medio de dispersión óptica (*Figura 7*). La tomografía de coherencia óptica se basa en la interferometría de baja coherencia, que generalmente emplea luz infrarroja cercana. El uso de una luz de longitud de onda relativamente larga le permite penetrar en el medio de dispersión. La OCT proporciona imágenes de morfología del tejido a una resolución mucho más alta (mejor que $10\ \mu\text{m}$) que otras modalidades de imagen, como la MRI o la ecografía. La OCT ocular es utilizada para obtener imágenes de alta resolución de la retina y el segmento anterior. Debido a la capacidad de la OCT para mostrar secciones transversales de capas de tejido con resolución micrométrica, la OCT proporciona un método sencillo para evaluar

la organización celular, la integridad de los fotorreceptores y el grosor axonal en enfermedades oculares o patologías sistémicas que tienen signos oculares. Más recientemente, los dispositivos de OCT oftálmicos han sido diseñados para realizar funciones similares a la angiografía con fluoresceína, pero sin la necesidad de tinte inyectable. Los beneficios del método incluyen la obtención de imágenes subsuperficiales en vivo con resolución casi microscópica, que la obtención sea instantánea y directa de la morfología tisular. Además, no requiere preparación de la muestra o el sujeto y no utiliza radiación ionizante [12].



Figura 7: Tomografía de coherencia óptica (Extraída de <https://www.epilepsysociety.org.uk/optical-coherence-tomography#.W7-fhWj0mM8>).

Retinografía Digital

La retinografía (*Figura 8*), es una técnica que se utiliza en medicina para obtener fotos en color de la retina. Las imágenes obtenidas pueden almacenarse en un sistema informático y enviarse para que sean estudiadas en un lugar diferente a donde fueron realizadas. Consta de una prueba sencilla, útil, segura y muy cómoda para el paciente. La única molestia que ocasiona es la relacionada con la necesidad de dilatar la pupila mediante gotas para conseguir imágenes de calidad. La retinografía es una documentación ocular que registran la apariencia de la retina de un paciente. La retinografía le permiten al clínico estudiar la retina de un paciente, detectar cambios en la retina y revisar los hallazgos de la retina de un paciente con otro profesional de la salud. Se utiliza para inspeccionar anomalías asociadas a enfermedades que afectan el ojo y para controlar la progresión de las enfermedades. Evalúa las irregularidades en el fondo, controla la progresión de una enfermedad, el manejo y el resultado terapéutico. Son cruciales para crear un punto de partida para comprender mejor la progresión de una enfermedad [13].

Su principal ventaja es que es una técnica rápida y sencilla de dominar. Además, se observa un campo retiniano más grande en cualquier momento en comparación con la oftalmoscopia y no se requiere dilatación, por lo que es un procedimiento menos invasivo que los métodos tradicionales. Otro aspecto positivo es el uso de diferentes filtros y tintes disponibles para permitir diferentes tipos de pruebas [12].

Por otro lado, la imagen producida es bidimensional, a diferencia de 3D en la oftalmoscopia. Esto puede producir dificultad para observar y evaluar anomalías debido a la falta de apreciación de la profundidad en las imágenes. Presenta como desventaja un menor aumento y claridad de imagen que la oftalmoscopia indirecta. Tampoco es portable y tiene un alto costo.



Figura 8: Retinografía digital. (Extraída de https://en.wikipedia.org/wiki/Fundus_photography)

A pesar de la variedad de metodologías de examinación de la retina, cada uno con sus ventajas, todas requieren un tiempo de adquisición y diagnóstico considerables. La digitalización permite el análisis automático por parte de sistemas de diagnóstico inteligentes, que en fracciones de segundos pueden diagnosticar correctamente [4]. Esto permite responder a las necesidades crecientes de evaluación sin la necesidad de sobre exigir los recursos humanos, cuya tarea deja de ser diagnóstico primario a hacer seguimiento y tratamiento de la patología.

Capítulo 3 – Redes Neuronales

3.1 Introducción

En este capítulo se explorará metodologías para el desarrollo de sistemas automáticos de diagnóstico. En esencia, realizar un diagnóstico es equivalente a una tarea de clasificación sobre conjuntos con ciertas características (síntomas, estadios, etc.). Se han desarrollado métodos de inteligencia artificial de clasificación muy robustos, que construyen un modelo en base a atributos de ejemplos que se le proveen. Estos atributos, en el caso de imágenes, solían requerir ser extraídos manualmente con pasos segmentación manual y pre-procesado complejo. Las redes neuronales aprenden dichos atributos autónomamente al componer módulos simples que transforman una representación en un nivel bajo en una representación en un nivel más alto. Esta metodología provee una clara ventaja al no depender de un equipo de especialistas para hacer dicha extracción, razón por la cual fue elegida en esta aplicación. A continuación, se desarrollará en que consiste dicha técnica, sus limitaciones y diferentes formas de aplicación, con conceptos extraídos del curso CS231n Convolutional Neural Networks for Visual Recognition [14].

3.1.1 Redes Neuronales

Las redes neuronales son algoritmos de entrenamiento de machine learning para análisis de patrones y clasificación, que funcionan mediante múltiples capas de procesamiento de información no lineal y son inspiradas en cómo trabaja el cerebro humano. Las neuronas son las unidades de computación y cada neurona lleva a cabo un producto escalar con la entrada y sus pesos, luego añade el sesgo y aplica la función de activación (no linealidad). Se puede observar en la *Figura 9*.

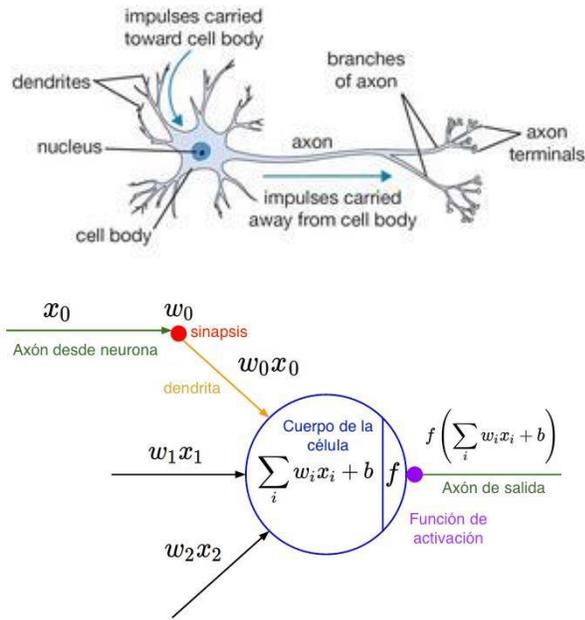


Figura 8: Representación simplificada del funcionamiento de una neurona en la Red Neuronal con una neurona del cerebro humano. El funcionamiento en la neurona es igual en las redes neuronales y en las convolucionales. (Extraída de <https://cs231n.github.io/>).

El sistema nervioso de una persona contiene alrededor de 86 billones de neuronas conectadas con 10^{15} sinapsis. De las dendritas se envía una señal de entrada a la neurona que generará señales de salida a través de su axón. Cabe la posibilidad de ramificación del axón y mediante sinapsis, que se conecte con las dendritas de otras neuronas. Como vemos en el modelo matemático de la neurona humana de la Figura 9, las señales (x_0) que viajan por el axón interactúan con las dendritas de otras neuronas ($w_0 x_0$) por sinapsis (w_0). Estos pesos que representan las fuerzas sinápticas se pueden aprender y controlan la influencia (pesos positivos) o inhibición (pesos negativos) entre neuronas. En el modelo humano, a través de las dendritas se envían señales a la neurona, donde todas se suman. Si se sobrepasa un umbral, la neurona se activa y manda un impulso a través de su axón. Matemáticamente no importan los tiempos de impulsos, aunque si la frecuencia de activación. Esta activación biológica se traduce en una función (función sigmoide, σ) que representa la frecuencia de los impulsos a lo largo del axón, que toma valores entre 0 y 1 tras recibir una señal de entrada real. Sin contar con las neuronas de la capa de entrada, cada neurona x_i de la red toma como entrada los valores de las neuronas de la capa precedente que alimentan a x_i . Por tanto, la entrada a una red neuronal calculada como la combinación lineal ponderada de sus entradas sería:

$$a_i = \sum_{i=1}^N x_i w_i + b \quad (1)$$

El término b es un sesgo asociado a la neurona. A esta (entrada a una red neuronal) se le aplica una función de activación no lineal. Esta función de activación de una neurona será:

$$z_i = f(a_i) = f(\sum_{i=1}^N x_i w_i + b) \quad (2)$$

Para aprender a través de la red y propagarse la información (forward propagation), una vez se toman los valores de entrada, la primera capa convolucional obtiene las activaciones, que solo dependen de los valores de las neuronas de las capas anteriores. Al final del proceso se obtiene un conjunto de salida y en función de las entradas x recibidas. Esta salida se usará para la clasificación de su entrada x .

Como se observa en la *Figura 10*, una red neuronal recibe una entrada (x) en forma de un único vector y lo transforma a través de una serie de capas ocultas. Cada una de estas se compone de una serie de neuronas, totalmente conectadas a todas las neuronas en la capa anterior, calculando una función dependiente de los valores de cada neurona de la capa de entrada. Las neuronas de una misma capa no están conectadas entre sí. La última capa es la capa de salida y en clasificación esta representa la puntuación de clase. La arquitectura de red es el patrón de conectividad de una red.

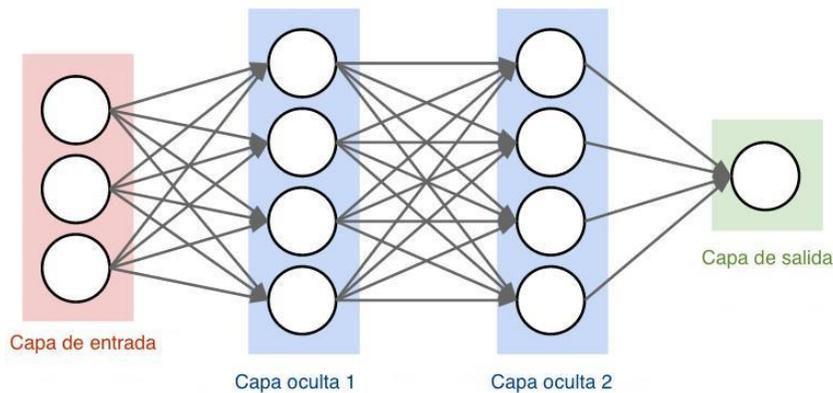


Figura 9: Red neuronal simple de tres capas. (Extraída de <https://cs231n.github.io/>).

En el caso de que las entradas sean imágenes completas, es decir, entradas de alta dimensión, las redes neuronales no funcionan bien puesto que la conectividad total de las neuronas de una capa con las de la siguiente es un desperdicio (dejando sin aprovechar la estructura de la imagen, por ejemplo, los píxeles adyacentes de una imagen tienden a estar muy correlacionados mientras que los que están separados no), y el gran número de parámetros

conduciría a un sobreajuste. En este caso se usan las redes neuronales convolucionales. Estas últimas suponen explícitamente que las entradas son imágenes, lo que permite codificar ciertas propiedades de las mismas en la arquitectura, haciendo la función hacia adelante más eficiente de implementar y reduciendo de forma considerable los parámetros de la red.

3.1.2 Redes Neuronales Convolucionales

Una red neuronal convolucional (CNN) es una poderosa técnica de deep learning. El procedimiento es similar a la red neuronal ordinaria explicado anteriormente. Básicamente se definiría como una secuencia de múltiples capas de características. Cada capa transforma un vector de entrada 3D (correspondiente a los tres canales RGB), en un vector de salida 3D, a través de funciones diferenciables que pueden tener o no tener parámetros. Las capas de la red se constituyen de neuronas dispuestas en tres dimensiones (ancho, alto y profundidad) que tienen pesos y sesgos que se pueden aprender. Cada neurona recibe entradas, lleva a cabo un producto escalar (entre la entrada y el peso) y a veces puede seguirse con una no linealidad.

En este caso las neuronas de una capa solo están conectadas a una pequeña región de la capa anterior. La extensión espacial de esta conectividad es un hiperparámetro llamado campo receptivo de la neurona, que sería lo mismo que tamaño del filtro. La extensión de la conectividad a lo largo del eje de profundidad es siempre igual a la profundidad del volumen de entrada. Las conexiones son locales en el espacio (a lo largo de la anchura y altura), pero siempre completas a lo largo de toda la profundidad del volumen de entrada.

Al final de la red se reduce la imagen completa a un único vector de puntuación de clase, que es la salida. En la *Figura 11*, La capa de entrada roja es la imagen, siendo la altura y el ancho las dimensiones de la imagen y la profundidad, los tres canales de color (rojo, verde, azul). Hay cinco capas principales que se apilan para construir arquitecturas de redes neuronales convolucionales completas, transformando los valores de los píxeles de la imagen original a una puntuación de clase finales.

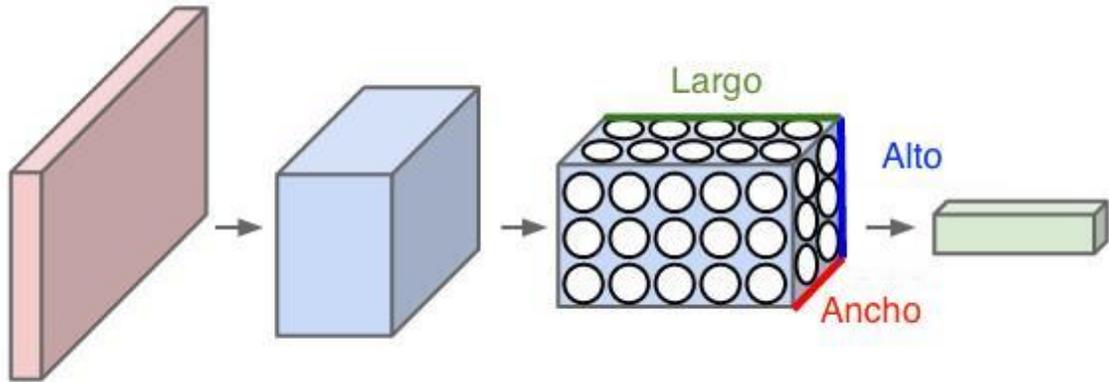


Figura 11: Funcionamiento de una red neuronal convolucional. (Extraída de <https://cs231n.github.io/>)

Las capas pueden ser:

- Entrada: Mantiene los valores de los píxeles de la imagen de entrada (el alto, el ancho y los tres canales de color).
- Convolucionales: Es el bloque central de construcción de la red. Sus parámetros consisten en un conjunto de filtros que pueden aprender. Estos filtros son pequeños espacialmente (ancho y largo) pero se extienden a lo largo de toda la profundidad del volumen de entrada. La ConvNet aprenderá los filtros que se activan cuando algún tipo de características específicas se detectan en la entrada. Esta capa calcula la salida de las neuronas conectadas a regiones locales en la entrada, siendo cada cálculo un producto escalar entre los pesos y la pequeña región a la que se conectan del volumen de entrada.
- Pooling o submuestreo: Esta capa reduce el tamaño espacial de la entrada para disminuir el número de parámetros reduciendo las respuestas convolucionales, disminuir la computación en la ConvNet y proporcionar un grado de invarianza translacional al modelo. Para ello se lleva a cabo una operación de reducción de muestreo a lo largo de las dimensiones espaciales (ancho y alto). Se toman las respuestas convolucionales y se dividen en x-y bloques. Tras aplicarle el pooling a estas respuestas nos vamos a quedar con un mapa de x-y respuestas, cada una representativa de un bloque tomando de cada uno de estos bloques se toma el mayor valor (max pooling) o el promedio (mean pooling)
- Totalmente conectadas: Neuronas entre dos capas adyacentes están todas conectadas por parejas, mientras las neuronas de la misma capa no. Pueden ser interpretadas como capas convolucionales de 1x1. Estas capas calcularán la puntuación de clase.

- ReLU (no linealidad): Esta capa calcula la función de $f(x) = \max(0, x)$, que umbraliza en 0. Esta capa mantiene el tamaño del volumen sin cambios.

Las capas convolucional y totalmente conectada llevan a cabo transformaciones que son función de la activación en la entrada y de los parámetros de estas capas: pesos y sesgos de las neuronas. Estos parámetros se entrenan con descenso de gradiente de modo que las puntuaciones de clase que la CNN calcula son consistentes con las etiquetas en el conjunto de entrenamiento para cada imagen. Las capas ReLU y pool implementan una función fija, no presentando parámetros.

En la *Figura 12* se puede observar un ejemplo de imagen (rojo) de 32x32 (ancho x alto) y tres canales de color (profundidad). En azul tenemos la primera capa convolucional compuesta por un volumen de neuronas. Cada neurona en la capa convolucional se conecta con una región local de la imagen de entrada, pero toda la profundidad de la misma. Hay 5 neuronas en la profundidad de esta capa, todas mirando hacia la misma dirección de la imagen. En las redes neuronales estándares la neurona obtenía una combinación lineal de sus entradas por los pesos. Esto se podía comprender como la evaluación de un filtro lineal sobre los valores de la entrada. Ahora, con las redes neuronales convolucionales hay varias neuronas de la misma capa convolucional que intercambian pesos lo que significaría la evaluación del mismo filtro sobre varias sub-ventanas de la imagen de entrada. De este modo la red neuronal convolucional podría entenderse como el aprendizaje más efectivo a un conjunto de filtros. Que la red pueda aprender una codificación general se podría conseguir o características de los datos de entrada se podría conseguir haciendo uso del mismo conjunto de filtros sobre la imagen entera. Con el objetivo de conseguir una generalización de la CNN se restringen los pesos a ser iguales a través de diferentes neuronas teniendo así un efecto de normalización sobre la CNN [15].

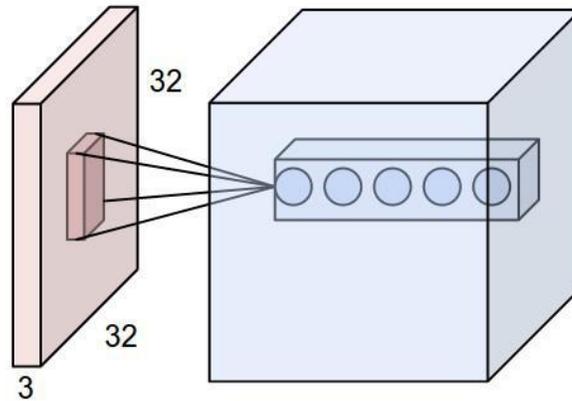


Figura 12: Funcionamiento de una imagen de 32x32 con la primera capa convolucional (Extraída de <https://cs231n.github.io/>).

Se evaluaron tres CNN pre-entrenadas: AlexNet, GoogLeNet y VGG-16, originalmente entrenadas para la clasificación de objetos multi-clase del desafío ImageNet. Se optó por estas redes sobre las debido a los siguientes criterios:

- Reconocimiento y desempeño conocido: las tres redes fueron finalistas del desafío ImageNet en sus respectivos años, lo que habla de su capacidad de clasificación y es la razón de su popularidad.
- Estructuras únicas e innovadoras:
 - Alexnet: Consiste de convoluciones de 11x11, 5x5,3x3, max pooling, dropout y activaciones ReLU. Adjuntó activaciones ReLU después de cada capa convolucional y totalmente conectada.
 - GoogLeNet: Posee un elemento novedoso al que se le llama un módulo de incepción. Este módulo se basa en varias convoluciones muy pequeñas para reducir drásticamente el número de parámetros. Además, contiene convoluciones de 1x1 en el centro de la red y pooling global al final de la red en lugar de usar capas completamente conectadas.
 - VGG-16: Consta de 16 capas convolucionales y posee una arquitectura muy uniforme. Solo realiza 3 x 33 por 33 x 3 convoluciones y 2 x 22 por 22 x 2 pooling a lo largo de toda la red.
- Tamaño: Las tres redes difieren significativamente de tamaño. De menor a mayor, GoogLeNet posee 6 millones de parámetros, AlexNet 60 millones de parámetros y finalmente VGG-16 138 millones de parámetros (Figura 13).

- Resultados exitosos obtenidos en publicaciones en las cuales se las utiliza como extractor de características: AlexNet [16], GoogLeNet [17] y VGG-16 [18].

Las características utilizadas fueron las salidas de la primera capa completamente conectada en el caso de AlexNet y VGG-16, y la última de pooling en el caso de GoogLeNet. Estas características se usaron para alimentar la capa de aprendizaje automático. Las CNN pre-entrenadas se compararon en términos de velocidad y características del rendimiento de la capa en la clasificación de las imágenes de RD.

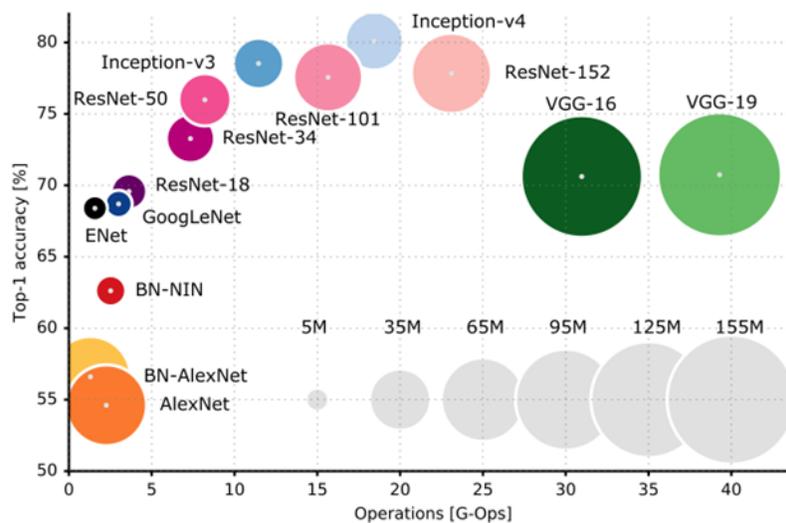


Figura 13: Top-1 de precisión sobre el conjunto Imagenet frente a la cantidad de operaciones requeridas para un feed-forward. El tamaño de las manchas es proporcional al número de parámetros de red; se reporta una escala en la esquina inferior derecha, que abarca desde 5×10^6 hasta 155×10^6 parámetros (Extraído de [19]).

1. AlexNet

AlexNet fue desarrollado en 2012 por Alex Krizhevsky [20], y es una de las CNN más famosas de la actualidad. La salida de la capa fc6 completamente conectada (cuadro remarcado en el panel izquierdo de la Figura 14) consiste en 4096 características por imagen.

2. GoogLeNet

GoogLeNet se desarrolló en Google Inc., esta CNN ganó el desafío ImageNet en 2014 [21]. La salida de la capa de agrupación fcls3_pool (capa remarcada en el panel central de la *Figura 14*) proporciona un vector de características 1024 por imagen.

3. VGG-16 Net

El tercer modelo CNN previamente entrenado fue VGG-16 Net. Fue el finalista en el desafío de ImageNet en 2014 [22]. Las características de la capa fc6 totalmente conectada (cuadro remarcado en el panel derecho de la *Figura 14*) representa 4096 características por imagen.

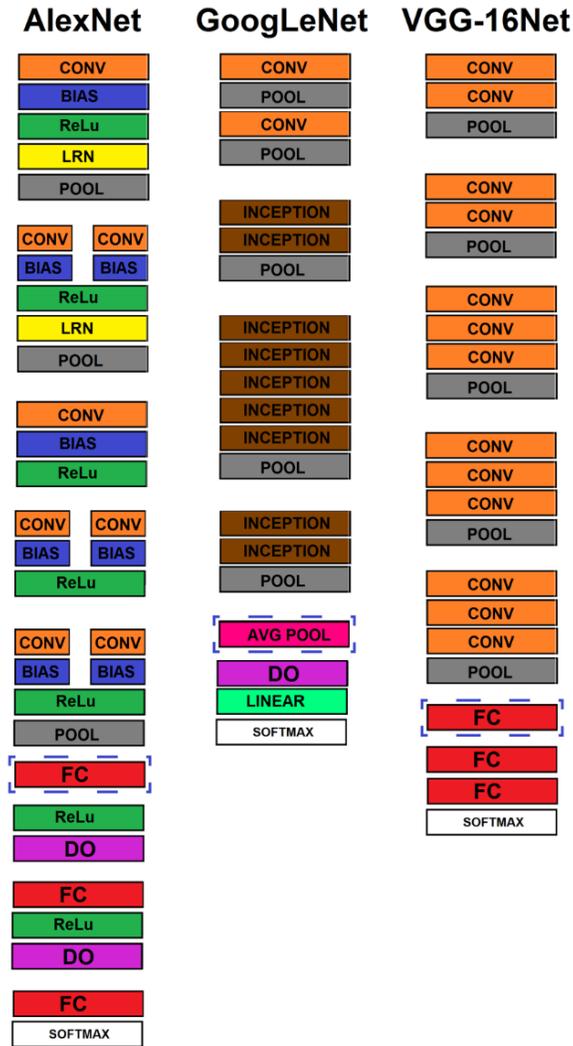


Figura 10: Arquitectura de Alexnet, GoogleNet y VGG-16 Net, con la capa de extracción de características remarcadas en azul. De izquierda a derecha, las capas son las siguientes: convolucional (CONV), bias (BIAS), unidad lineal rectificadora (ReLU), lineal (LRN), max pooling (POOL), totalmente conectada (FC), dropout (DO), softmax, modulo inception (INCEPTION) y pooling promedio (AVG POOL).

3.1.2 Entrenamiento de Redes Neuronales Convolucionales

Para el uso de una red neuronal convolucional, se debe entrenar con ejemplos supervisados del dominio del problema. A continuación, se desarrollará como son las distintas metodologías de entrenamiento y uso de redes neuronales convolucionales, con conceptos extraídos de [23]. La literatura no es exacta, pero se estima del orden 1×10^6 de ejemplos y se recomienda del orden de 5.000 imágenes por clase. Dada la escasez de datos en algunas áreas, particularmente la biomédica, surge la alternativa de transfer learning, es decir transmisión de aprendizaje (Figura 15).

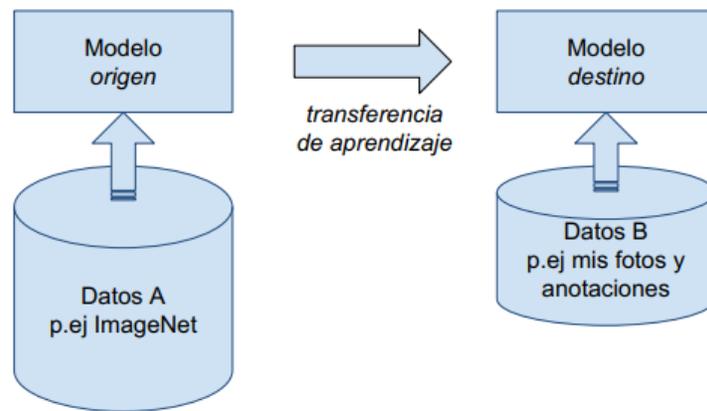


Figura 11: Diagrama de la transferencia de aprendizaje.

Esto consiste en tomar modelos ya entrenados en conjunto de datos grandes, para tareas específicas y utilizar nuestros nuevos datos para crear un nuevo modelo a partir de este. Una de las metodologías de transfer learning, es la denominada fine tuning (Figura 16).

Caffe Fine-tuning Pretrained Network

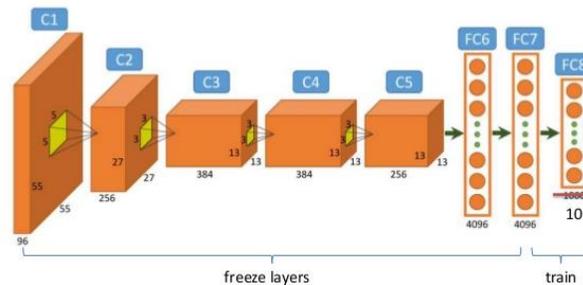


Figura 12: Fine tuning (Extraída de <https://image.slidesharecdn.com/practicaldeeplearning-160329181459/95/practical-deep-learning-16-638.jpg>).

En este caso, se “congelan” las capas de la red neuronal, salvo las últimas. Estas dependen de la estructura, pero suelen incluir una capa fully connected y la softmax. Esta última se cambia al número de clases del problema nuevo. Por ejemplo, dejan de ser las 1000 clases de Imagenet a dos, persona sana/enferma en un caso de clasificación binaria. Las características de esta metodología son:

- Se estima del orden de 1000 ejemplos por clase
- Es un método flexible y fácil de implementar
- Aprovecha días/semanas/meses de cómputo (energía)
- Como desventaja, es dependiente de la diferencia entre dominios

Este orden de magnitud de ejemplos puede ser difícil de conseguir, así que se presenta la metodología alternativa, extracción de características (Figura 17).



Figura 13: Diagrama de la extracción de características (Extraída de <https://seenu147.wordpress.com/2016/06/20/computer-vision-with-matlab/>).

Se toma una red ya entrenada, por ejemplo, en Imagenet y a diferencia del fine tuning, no se la entrena. Si no, extraemos las características aprendidas por la red neuronal y las clasificamos mediante un método de machine learning tradicional. Las ventajas de este método radican en no tener que entrenar una red (no requiere el uso de una GPU) y para machine learning hay un desarrollo grande de librerías.

Las primeras capas de la red han aprendido filtros para capturar características como manchas y ejes (*Figura 18*). Estas características primitivas son procesadas por capas más profundas de la red, que combinan las características tempranas para formar características de la imagen de más alto nivel (contornos, categorías...). Estas características de más alto nivel son más adecuadas para tareas de reconocimiento por que combinan todas las características primitivas en una imagen de representación más rica [24].

Dichas características se extraen típicamente de las ultimas capas, como puede ser fully connected o pooling, dependiendo de la arquitectura, que permiten identificar la pertenencia específica a una clase. Sin embargo, se ha demostrado que pueden describir una clase por su ausencia, lo que permite hacer distinciones sutiles en imagines similares. Por dicha especificidad, estas características pueden ser irrelevantes para la tarea nueva. Si se toma una red entrenada exclusivamente en clasificar autos y quiero clasificar células, es probable que las características aprendidas no sirvan para el nuevo dominio. Entonces, si se usan redes entrenadas en un amplio rango de imágenes (por ejemplo, Imagenet está compuesta desde caballos a montañas), al haber características más variadas, hay mayor posibilidad de éxito. En cambio, las de capas más inferiores, como las convolucionales, son más independientes de la aplicación y se pueden utilizar para problemas no supervisados [25].

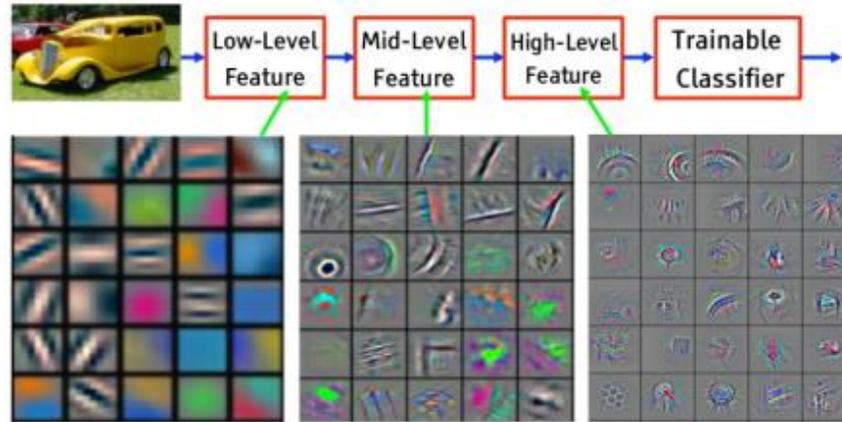


Figura 14: Características de una imagen a lo largo de distintos niveles de una red convolucional
(Extraída de <https://cs231n.github.io/>).

3.2 Métodos de Clasificación

A partir de los resultados obtenidos previamente en un trabajo exploratorio de los diversos métodos de machine learning para clasificación de retinopatía diabética [26], se optó por el método de máquinas de vectores de soporte con kernel gaussiano (SVM, por sus siglas en inglés) por su alto rendimiento y k-Top Scoring Pairs (kTSP) por su novedad como aplicación.

3.2.1 SVM

Las máquinas de vectores de soporte son máquinas de aprendizaje basadas en la teoría estadística de aprendizaje desarrollada por Vapnik et al. [27]. En el contexto de la clasificación, SVM proporciona una forma de encontrar el hiperplano de margen máximo que divide de manera óptima las dos clases, dando una configuración específica de la máquina. En una configuración general, el algoritmo de aprendizaje encuentra el vector de peso óptimo que resuelve el siguiente problema: dado el conjunto de muestras $\{(x_i, y_i)\}_{i=1}^N$ donde $x_i \in R^p$

es el patrón de entrada para el i -ésimo ejemplo e $y_i \in \{-1, 1\}$ la respuesta de clase deseada, el hiperplano que separa los datos debería satisfacer

$$y_i(w^T x_i + b) \geq 1 + \xi \quad (3)$$

donde b es el sesgo y el vector de peso $w \in R^p$ y la variable de margen ξ permite la incorporación de algunos errores en la solución. El vector de peso w puede obtenerse mediante la minimización de la función lagrangiana.

$$\Phi(\alpha) = \sum_{i=1}^N \alpha_i + 0.5 \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j x_i^T x_j \quad (4)$$

Sujeto a las restricciones

$$\sum_{i=1}^N \alpha_i y_i = 0 \text{ con } 0 \leq \alpha_i \leq C \text{ para } i = 1, 2, \dots, N. \quad (5)$$

Donde C , el parámetro de costo, es un valor positivo especificado por el usuario. En la práctica, si todo $i < C$, el conjunto de datos puede separarse linealmente sin error (todos $\xi = 0$) y se dice que el problema es un problema de "margen rígido" sino, un "margen blando". El puente entre la forma primaria y dual está dado por

$$w = \sum_{i=1}^N \alpha_i y_i x_i = \sum_{i \in SV} \alpha_i y_i x_i \quad (6)$$

donde SV es el conjunto de vectores de entrada para los $\alpha_i \neq 0$ (los vectores de soporte). La norma al cuadrado del vector de peso se puede calcular como

$$\|w\|^2 = \sum_{i=1}^N \alpha_i - \frac{\|\alpha\|^2}{c} \quad (7)$$

y la distancia máxima entre los patrones más cercanos en el espacio hiper-geométrico R^p está dada por

$$2\gamma = \frac{2}{\|w\|^2} \quad (8)$$

En el caso de una solución de "margen rígido", el segundo término de la Eq. 5 desaparece [28].

3.2.2 *k-Top Scoring Pairs*

k – Top Scoring Pairs [29] es un método de clasificación para datos de alta tasa de transferencia efectiva que seleccionan simultáneamente un pequeño conjunto de características mediante la creación de reglas por pares. Fue desarrollado originalmente en el campo de la bioinformática utilizando datos de expresión génica para clasificar muestras de cáncer. En esencia, el método $kTSP$ es un clasificador binario que se compone de k clasificadores

elementales. Cada clasificador elemental se basa en un par de características de entrada de "alto poder predictivo". Esto significa que la característica "i" tiene un valor predominantemente mayor que la característica "j" en una clase y viceversa en la otra. Una muestra desconocida se clasifica según los valores relativos de las características "i" y "j" en esa muestra. El método kTSP simplemente recopila los votos de los mejores clasificadores elementales separados e informa que el voto mayoritario es su decisión. Se permite que cada característica participe en un solo clasificador elemental, porque esto hace que la regla de decisión sea robusta contra los errores sistemáticos en las mediciones de una sola característica. Por otro lado, una característica indicadora que es muy alta en un grupo y muy baja en el otro es más probable que sea parte de muchos pares clasificadores de alto rendimiento.

En el caso presente, consideramos $p = (4096 \vee 1024)$ características cuyos niveles $x = (x_1, x_2, \dots, x_p)$ son provistos por la capa elegida de la CNN pre entrenada y son consideradas como variables aleatorias. Cada característica " X_i " tiene una categorización en $\{1, 2, \dots, c\}$. Por simplicidad, se asume $c=2$, aunque se puede extender a un número mayor de clases. Nos centramos en la detección de "pares de características de marcador" (i, j) para los que existe una diferencia significativa en la probabilidad de $x_i < x_j$ de la clase 1 a la clase 2. La clasificación del perfil se basa en esta colección de pares distinguidos. Aquí, las cantidades de interés son $p_{i,j}(c) = P(x_i < x_j \vee c)$, $c = 1, 2$, es decir, las probabilidades de observar $x_i < x_j$ en cada clase. Estas probabilidades se estiman por las frecuencias relativas de ocurrencias de $x_i < x_j$ dentro de los perfiles y sobre experimentos. En consecuencia, es suficiente conocer los rangos de los valores de las características dentro de los perfiles en cada imagen. La decisión kTSP se basa en k pares de características de imagen, denotadas como $\theta = \{(i_1, j_1), \dots, (i_j, j_k)\}$. La regla de decisión particular que usa las k comparaciones de $x_{i_l} < x_{j_l}$ simplemente está determinada por la estadística de voto agregada $k = \sum_{l=1}^k I(x_{i_l} < x_{j_l})$ donde I es la función de indicador lógico. La decisión de clasificación de kTSP se basa en el umbral, es decir, $\hat{y} = I\{k > \tau\}$ siempre que se incluya las clases $y \in \{-1, 1\}$. El umbral estándar es $\tau = k/2$ equivalente al voto de la mayoría.

3.2.3 Métricas Empleadas

Para poder comprender los resultados de la clasificación se definen las siguientes clases:

- TP= verdaderos positivos (True Positive) son fondos de ojos con RD correctamente detectados.

- FP= falsos positivos (False Positive) son fondos de ojos sanos detectados como RD.
- TN= verdaderos negativos (True Negative) son fondos de ojos sanos correctamente detectados.
- FN= falsos negativos (False Negative) son fondos de ojos con RD detectados como sanos.

Estas clases permiten construir una matriz de confusión (*Tabla 2*). La matriz de confusión es una tabla que se usa normalmente para describir el rendimiento de un sistema de clasificación en un conjunto de datos en el que se conocen los verdaderos valores. La matriz contiene información de las clasificaciones reales y las predichas realizadas por el sistema y se usa para calcular el desempeño del algoritmo [30].

		Diagnostico Verdadero		
		Positivo	Negativo	Total
Diagnostico Observado	Positivo	Verdadero Positivo (TP)	Falso positivo (FP)	TP+FP
	Negativo	Falso Negativo (FN)	Verdadero negativo (TN)	FN+TN
	Total	TP+FN	FP+TN	

Tabla 2: Matriz de confusión.

Además, con ellas se pueden calcular medidas del desempeño del clasificador:

- $Sensibilidad = \frac{TP}{TP+FN}$
- $Especificidad = \frac{TN}{TN+FP}$
- $Exactitud = \frac{TN+TP}{TN+TP+FN+FP}$
- $Precision = \frac{TP}{TP+FP}$
- $F - score = \frac{2}{\frac{1}{Sensibilidad} + \frac{1}{Precision}}$
- *Curva ROC*: (acrónimo de Característica Operativa del Receptor) es una representación gráfica de la sensibilidad frente a la especificidad para un sistema clasificador binario según se varía el umbral de discriminación [31].

Los mejores modelos para cada conjunto de características de CNN preentrenado se evaluaron luego sobre los conjuntos de validación por medio de Exactitud, Sensibilidad,

Especificidad y F-score, como lo sugirieron [32]. Además, se generaron las curvas ROC para comparar dichos modelos.

Capítulo 4 - Materiales y métodos

En este capítulo se describirá la metodología que se propone (Figura 19), detallando los pasos para construir los modelos predictivos.

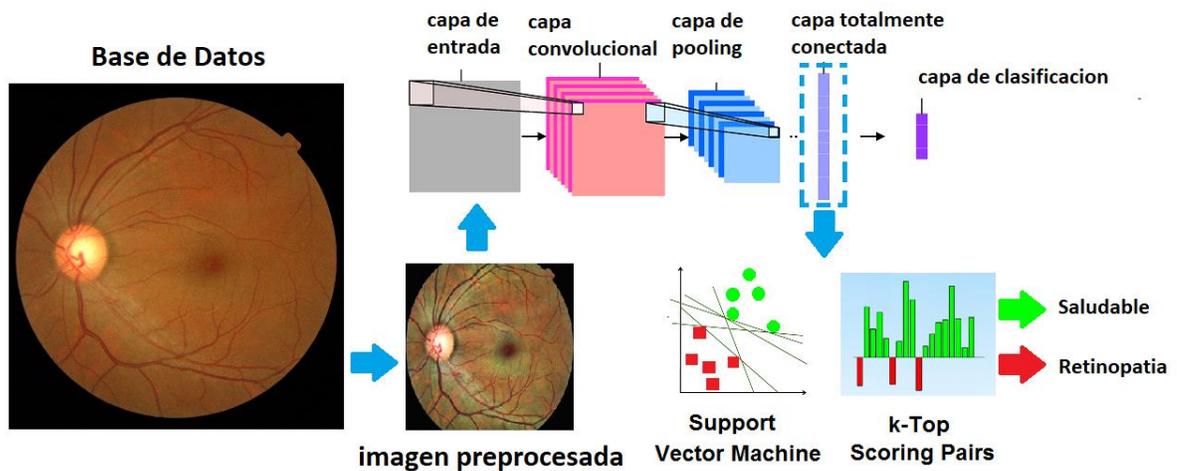


Figura 15: Metodología propuesta.

En esta tarea de clasificación, el programa debe especificar cuál de las dos categorías pertenece el input. Para resolver esta tarea, el algoritmo de aprendizaje produce una función $f: R^n \rightarrow \{1,2\}$. Cuando $y=f(x)$, el modelo asigna un input descrito por un vector x a una categoría identificada por el código numérico y . Existen otras variantes de la tarea de clasificación, cuando, por ejemplo, f devuelve la probabilidad de distribución sobre categorías. En este caso el input es una imagen, y la salida es un código identificando la imagen con dicha probabilidad.

4.1 Base de datos

Los repositorios de libre uso disponibles de fondo de ojo de retinopatía diabética son los siguientes:

Base de datos	Normal	Patológica	Detalle
VICAVR dataset [33]	58	0	Sin datos patológicos
Structured analysis of the retina dataset [34]	41	68	Pocas muestras
Kaggle (training) [35]	25810	9316	Muestras ruidosas y con diferencia substancial de calidad
Kaggle (validación)	53576		
High resolution fundus (HRF) image database [36]	15	15	Pocas muestras
Standard diabetic retinopathy database calibration level 0 [37]	20	110	Desbalance grande entre clases
Standard diabetic retinopathy database calibration level 1 [38]	5	84	Desbalance grande entre clases
RD1 (Pires et al. 2014) [39]	595	482	Lesiones de etapas avanzadas, desbalance grande entre clases
RD2 (Pires et al. 2014) [39]	300	149	Util
MESSIDOR (Decencière et al. 2014) [40]	547	653	Util
IRDID (Porwal et al. 2018) [41]	134	279	Util
E-optha (Decencière et al. 2013) [42]	268	174	Util

Tabla 3: Bases de datos de fondos de ojos disponibles en internet, sin patologías o con retinopatía diabética.

Se tomaron los siguientes criterios a la hora de elegir las bases de datos. Primero, que contuvieran como mínimo 100 ejemplos por clase. Segundo, que la base no contenga datos de muy poca resolución o mucho ruido. Tercero, como es un sistema de detección primaria, se busca que las lesiones pertenezcan solo a las etapas iniciales de la enfermedad [43]. Cuarto, que la proporción entre cantidad de muestras por clase sea lo más cercana al 50%, para no generar sesgos hacia alguna clase en el modelo de clasificación. Dichos criterios fueron definidos bajo la sugerencia del director de tesis.

Del total de bases disponibles se tomó los últimos cuatro, formando un total de 2458 imágenes. De ese total, dos bases de datos son usadas para entrenamiento (1642 imágenes) y dos para validación (816 imágenes), dando una relación entrenamiento/validación de 66%-33%. Esta relación se encuentra cercana a la sugerida por Andrew Ng (profesor asociado en el

departamento de Ciencias de la Computación y del departamento de Ingeniería Electrónica por cortesía de la Universidad de Stanford, y director del laboratorio de Inteligencia Artificial en Stanford) en el MOOC de Coursera sobre Introducción al aprendizaje automático [44], donde establece que la regla general es dividir el conjunto de datos en una proporción de 70%-30% para entrenamiento y validación respectivamente. Se decidió además no incorporar datos pertenecientes a las bases del conjunto de validación en el conjunto de entrenamiento y viceversa para observar como el modelo se desempeña sobre muestras pertenecientes a diferentes fuentes geográficas.

4.1.1 Bases de datos de entrenamiento

4.1.1.1 Messidor [40]

El conjunto de datos es de libre acceso y ha sido utilizado por otros grupos para la evaluación comparativa del rendimiento de algoritmos para RD. Consiste en 1200 imágenes a color del fondo de ojo del polo posterior. Fueron adquiridos por 3 departamentos de oftalmología usando una cámara 3CCD de video a color en un retinógrafo no midriático Topcon TRC NW6 con un campo de visión de 45 grados. Las imágenes se capturaron utilizando 8 bits por plano de color a 1440 * 960, 2240 * 1488 o 2304 * 1536 píxeles. Se adquirieron 800 imágenes con dilatación de la pupila (una gota de Tropicamide al 0.5%) y 400 sin ella. De los 1200 fondos oculares, 546 no presentaron signos de RD y 654 presentaron exudados duros considerados como clase RD.

4.1.1.2 E-ophtha [42]

Los datos Esta base de datos de imágenes de fondo de color se ha generado para la investigación científica en RD de la red telemedica OPHDIAT ©, en el marco del proyecto ANR-TECSAN-TELEOPHTA financiado por la Agencia de Investigación Francesa (ANR). Las imágenes se obtuvieron con retinografías no midriáticas: retinografías CR-DGi (Canon, Tokio) o TRC-NW6S (Topcon, Tokio). Dependiendo de la configuración de cada retinografía, se obtuvieron imágenes con tamaños variables: los tamaños de las imágenes oscilaron entre 1440 * 960 y 2544 * 1696 píxeles. Está compuesto por 268 no RD y 174 con RD (exudados duros y microaneurismas o pequeñas hemorragias).

4.1.2 Bases de datos de validación

4.1.2.1 RD2 dataset [39]

El conjunto de datos se capturó utilizando una retinografía TRC-NW8 con una cámara Nikon D90, creando imágenes de 12,2 megapíxeles, que luego se redujeron a 867 * 575 píxeles para acelerar el cálculo computacional. De las 449 imágenes, 300 sin signos de RD y 149 con RD. Las imágenes que solo tenían manchas de algodón se eliminaron, ya que pertenecen a una etapa más avanzada, dejando el conjunto con 103 imágenes RD y 300 saludables.

4.1.2.2 IRDID challenge dataset [41]

Las imágenes del fondo fueron capturadas por un especialista en retina en una clínica oftalmológica ubicada en Nanded, Maharashtra, India. Las imágenes se adquirieron utilizando una cámara de fondo digital alfa Kowa VX-10 con campo de visión de 50 grados (FOV), y todas están centradas cerca de la mácula. Las imágenes tienen una resolución de 4288 * 2848 píxeles y se almacenaron en formato de archivo jpg. El tamaño de cada imagen es de aproximadamente 800 KB. El conjunto de datos consta de 413 imágenes, 134 sin signos de RD y 279 con.

Dichas imágenes fueron almacenadas en carpetas con el nombre “Entrenamiento” y “Validación” (Tabla 4) y dentro de cada una, subcarpetas de nombre “Normal” y “Retinopatía Diabética” (ejemplo de su contenido en Figura 20).

	Base de datos	Normal	Patológico	Total
Entrenamiento	Messidor	546	654	1200
	E-ophtha	268	174	442
	Total	814	828	1642
Validación	RD2	300	103	403
	IDRID	134	279	413
	Total	434	382	816

Tabla 4: Estructura del conjunto de datos de entrenamiento y validación.

La ventaja de utilizar esta estructura de carpetas es que el imageDatstore (repositorio para colecciones de datos que son demasiado grandes para entrar en la memoria) de MATLAB será capaz de leer y administrar las ubicaciones de imágenes y etiquetas con una categoría de forma automática con el nombre de la carpeta que la aloja.

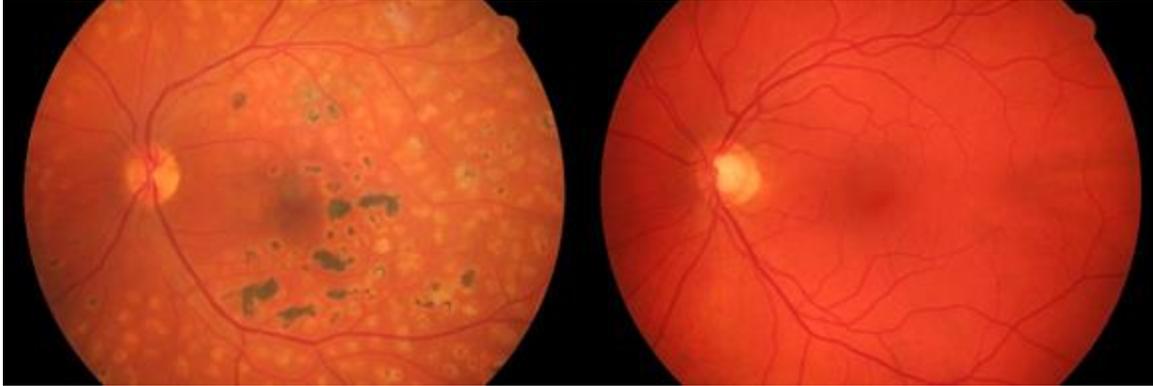


Figura 16: Fondo de ojo con retinopatía diabética (izq.) y Normal (der.)

4.2 Hardware y software

La secuencia de comandos se diseñó en Matlab 2017a, utilizando los complementos AlexNet y VGG-16 Net, que están en la biblioteca de aplicaciones disponibles para descargar desde Matlab. En el caso de GoogLeNet esta opción no está disponible, por lo que se usó la biblioteca MatCovNet [45], que requiere que el compilador se cambie a Microsoft Visual C ++ 2013 Professional.

El algoritmo k-Top Scoring Pairs se ejecutó en R, utilizando el paquete “SwitchBox” que está disponible gratuitamente desde la plataforma Bioconductor. Bioconductor es un proyecto de software de código abierto y desarrollo abierto que proporciona herramientas para el análisis y la comprensión de datos genómicos de alto rendimiento. Se basa principalmente en el lenguaje de programación R. Además, se usó la librería “readxl” que permite leer archivos .xls, dado que los vectores de características fueron exportados desde Matlab en ese formato.

El sistema operativo era Windows 7 y el hardware utilizado fue una memoria RAM de 8 gigabytes y una CPU Intel® Core™ i7-6500U a 2.50 GHz en una computadora Acer TravelMate P278-MG.

4.3 Pre-procesamiento

Las imágenes de la retina, que consisten en diferentes tamaños y fueron tomadas de diferentes cámaras, se utilizan como datos de entrada del paso de pre-procesamiento. La meta del pre-procesamiento es compensar las variaciones entre imágenes mediante la normalización respecto a una imagen de referencia. Dichas diferencias pueden llevar a que sean consideradas características de una clase mediante el entrenamiento, por lo cual se creó el siguiente algoritmo.

A pesar de que el pre-procesado está presente en otros trabajos publicados en el tema, generalmente son descritos muy brevemente [47-50]. Esto no permite comparar las distintas metodologías que se tomaron y, en consecuencia, gran parte del trabajo a continuación fue exploratorio.

Dado que el fondo no es estrictamente color negro puro, que en RGB sería (0,0,0), se utiliza un enmascaramiento de imagen para etiquetar los píxeles correspondientes de la retina y el área de fondo a lo largo de la imagen. La máscara resultante se utiliza para recortar imágenes, es decir, eliminar información irrelevante de la imagen, como bordes inútiles y estandarizar las relaciones de aspecto. Las etapas de enmascaramiento y recorte de la imagen se detallan en la siguiente sección.

4.3.1 Enmascarado de la imagen

Una imagen retiniana consiste en una región circular de color (primer plano) sobre un fondo negro (*Figura 21*). Es importante distinguir el fondo del primer plano, ya que esta etapa de pre-procesamiento permite eventualmente para obtener resultados más confiables del algoritmo de evaluación de calidad de imagen.

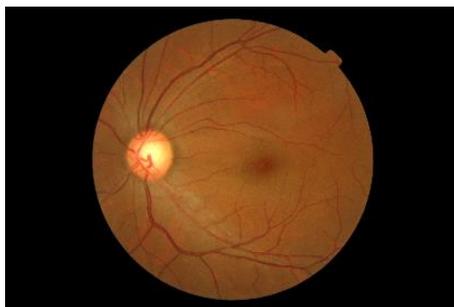


Figura 17: Imagen cruda

El algoritmo de enmascaramiento de imagen desarrollado en este estudio incluye los siguientes pasos:

- a. Se extrae la imagen del canal rojo ya que el canal rojo proporciona mayor contraste entre fondo y primer plano.
- b. Se reduce el ruido aplicando un filtro de mediana con el tamaño de máscara. El tamaño de la máscara es determinado empíricamente.
- c. Se convierte la imagen uniforme en escala de grises en una imagen binaria (*Figura 22*) basada en umbral de Otsu [51], utilizando un umbral determinado empíricamente (0.15) al haber un gran contraste.

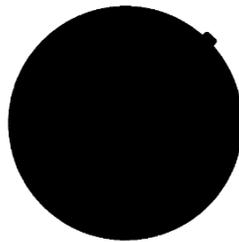


Figura 18: Imagen binarizada

- d. Se aplica la operación de apertura morfológica sobre la imagen binaria. La imagen binaria puede incluir algunos ruidos en el fondo y alrededor del límite de la retina. El procesamiento de apertura se realiza para eliminar cualquier ruido de manera que los bordes (nítidos) sin ruido sean adquiridos. El elemento estructural elegido fue 'disco' de radio 3.
- e. Luego esta imagen se invirtió y se concatenó en 3 canales, para substrair de la original.

4.3.2 Recorte de imagen

El recorte negro de las imágenes no proporciona información a la hora de clasificar, y debido a la limitada dimensionalidad de la entrada de las redes convolucionales, del orden de 224x224 píxeles, se debe eliminar. Para esta etapa, se tomaron los valores de ancho y alto de las muestras del conjunto de entrenamiento de aquellas que presentaban el disco ocular entero y, por otra parte, aquellas con el disco ocular recortado. Debido a la falta de alguna referencia en esta etapa, se calculó la media como índice de referencia en base a la sugerencia de docentes

pertenecientes a la cátedra de Procesamiento de Señales de la carrera de Ingeniería Biomédica de la Universidad Nacional de Córdoba.

El algoritmo de recorte de imagen desarrollado en este estudio incluye los siguientes pasos:

- a. La imagen obtenida del proceso anterior se transforma a escala de grises
- b. El algoritmo busca las filas y columnas cuyo valor máximo es cercano a 0 (significa que en toda la fila/columna hay pixeles negros)
- c. Se eliminan dichas filas y columnas de la imagen RGB.
- d. Se calcula la relación entre el alto y ancho, y si es mayor a 0,91 implica que es una imagen no recortada (está el disco ocular entero). Si es menor se salta los siguientes pasos hasta el punto 3.
- e. Como el promedio de la relación de alto/cho en imágenes recortadas de las bases de datos adquiridas es 0,83, se multiplica el ancho*0.5*(1-0.83). Se recorta entonces a la parte superior e inferior de la imagen este valor, estandarizando las relaciones de aspecto (*Figura 23*).

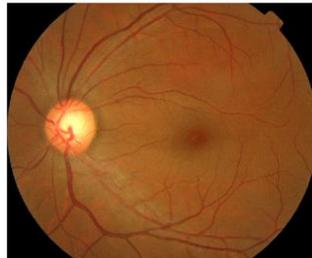


Figura 19: Imagen estandarizada

4.3.3 Mejora de contraste y dimensionado.

A continuación, se aplicó ecualización adaptativa de histogramas con contraste limitado [52] en regiones de datos pequeños (mosaicos) para mejorar los contrastes de las imágenes (*Figura 24*). Por medio de esto, el histograma de la región de salida coincide aproximadamente con el histograma plano establecido por el parámetro 'Distribución', utilizando la configuración predeterminada. Los mosaicos vecinos se combinan luego usando interpolación bilineal para eliminar los límites inducidos artificialmente, dando como resultado una limitación de contraste para evitar la amplificación del ruido que podría estar presente en la imagen. Finalmente, las

imágenes se redimensionaron de acuerdo con el CNN previamente entrenado evaluado, a 227x227 píxeles para AlexNet y 224x224 píxeles para GoogLeNet y VGG-16 Net.

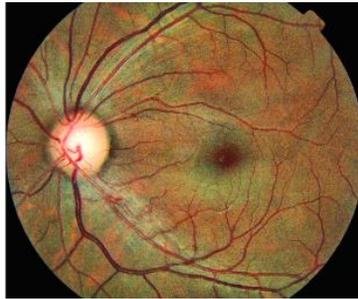


Figura 20:Imagen realzada

4.4 Extracción de características

De acuerdo a lo elaborado en el Capítulo 3.1.2, debido al número de ejemplos por clase disponibles, se recurre a la metodología de extracción de características. El método de `activations` calcula las activaciones de la capa de la red para una capa específica. Las activaciones de la red se calculan propagando hacia delante la entrada X a través de la red hasta llegara la capa específica. La capa debe tener un índice numérico o un vector de caracteres correspondiente a uno de los nombres de la red. Para una red que la capa de entrada es una imagen, X tiene que ser:

- Una imagen
- Una matriz de 4D donde las tres primeras dimensiones son el alto, el ancho y los canales de la imagen y la cuarta dimensión indica las imágenes individuales.
- Un conjunto de datos de imágenes

La Y , que es la salida, será una matriz con una fila por observación. Cada fila de la matriz es una observación y cada columna es un elemento de salida de la capa elegida. Por defecto y si no se dice lo contrario nos lo dará de esta manera, aunque habría otras opciones. El `MiniBatchSize` es el tamaño de los mini-lotes para las predicciones de computación. Los tamaños mayores de mini-lotes conducen a predicciones más rápidas, a costo de más memoria. Por defecto sería 128, pero se ajustó a 32 por la capacidad del hardware.

```
Y = activations(net, X, layer)
```

En nuestro caso se han obtenido las características de las capas fc6 de Alexnet con 4096 características por imagen, fcls3_pool de GoogLeNet con características 1024 por imagen y fc6 de VGG16 con 4096 características por imagen. No se extraen características de la última capa porque esta solo proporciona la probabilidad de la clase predicha del conjunto de entrenamiento original (no es una representación de la imagen de entrada). Para cada red, se midió y comparó el tiempo medio de las extracciones de características. Estas fueron almacenadas como un archivo .xls para su recuperación posterior y utilización en R.

4.5 Entrenamiento del clasificador

Un clasificador consta de varios parámetros que deben aprenderse de los datos, es decir, el fin del aprendizaje automático es ajustar un modelo a los datos. Este proceso de entrenamiento de un modelo con datos existentes que se ajustan a los parámetros del modelo, se denomina entrenamiento del modelo. Los hiperparámetros se refieren a otro tipo de parámetros que no pueden aprenderse directamente del proceso de capacitación y deben estar predefinidos. Estos hiperparámetros definen conceptos de nivel superior sobre el modelo, tales como complejidad, capacidad de aprendizaje, tasa de convergencia, penalización, etc. Los hiperparámetros óptimos llevan a una mejor eficiencia, una convergencia más rápida y mejores resultados en general. La optimización de dichos hiperparámetros se refiere al método para encontrar hiperparámetros óptimos para un algoritmo de aprendizaje automático. Esto es importante ya que el rendimiento de cualquier algoritmo de aprendizaje automático depende en gran medida del valor de los hiperparámetros. Dado que los hiperparámetros se establecen antes de que se ejecute cualquier algoritmo de aprendizaje automático, por lo tanto, es muy esencial establecer un valor óptimo de los hiperparámetros, ya que afecta la convergencia de cualquier algoritmo en gran medida [53].

4.5.1 SVM

Un modelo de SVM requiere dos entradas: una matriz con las características extraídas por imagen y un vector de categorías de igual longitud que el número de ejemplos, con las etiquetas de clase correspondiente. El modelo SVM se construyó utilizando la función `fitcsvm` Matlab, estandarizando los datos y utilizando un kernel gaussiano, elegido en base a los resultados de [25]. Sus hiperparámetros son el costo y gamma.

El Costo es cuánto penalizamos a la SVM por los puntos de datos dentro del margen (donde "dentro" también puede significar en el lado equivocado del hiperplano divisor). Un C grande significa penalizar mucho. Si C es grande, el SVM intentará encontrar un hiperplano y margen para que haya pocos puntos dentro del margen, lo que podría significar un modelo demasiado complejo con un pequeño margen si los puntos no son fácilmente separables. Una C más baja da un error más alto en el conjunto de entrenamiento, pero encuentra un margen más grande que podría ser más robusto.

Gamma es un parámetro del kernel, en el que ajusta la similitud de dos puntos para que se consideren "similares". Se puede pensar que el kernel gaussiano está construyendo un radio alrededor de cada punto, cuya densidad se describe mediante una función gaussiana. Gamma controla la desviación estándar de la función gaussiana, por lo que un gamma más grande significa una desviación estándar más pequeña alrededor de cada punto.

La optimización del 'box constraint' (el hiperparámetro costo) y la escala de kernel (el hiperparámetro gamma), se hizo mediante la opción 'OptimizeHyperparameters', que usa la optimización bayesiana (Figura25).

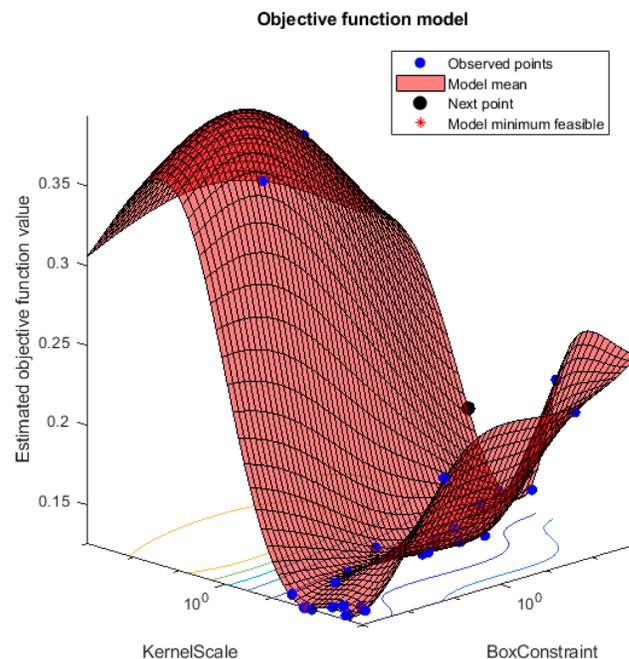


Figura 25: Visualización de la optimización de los hiperparametros gamma (kernel scale) y costo (box constraint).

4.5.2 k-TSP

Se utilizó las características de imagen extraída de los conjuntos de datos de entrenamiento para optimizar el número de reglas (funciones de pares usados) de KTSP mediante una estrategia de validación cruzada de 10 veces, con la exactitud como objetivo de optimización. Se aplicó una prueba de rango con signo de Wilcoxon para pre-filtrar las características, debido a que la reducción de $8 * 10^6$ pares a $5 * 10^3$ pares disminuye significativamente el tiempo de computo. Además, se usó un rango de 2 a 25 pares posibles, extendiendo el conjunto inicial de 2 a 10 pares debido a que los primeros modelos con este conjunto inicial presentaban un número de pares siempre cercano al límite superior.

En la validación cruzada de 10 particiones, se divide el conjunto de entrenamiento en 10 subconjuntos del mismo tamaño. Luego, 9 subconjuntos son empleados para entrenar el modelo con ciertos parámetros y el conjunto restante es utilizado para validar su desempeño. Esta operación se repite 10 veces, de modo que el modelo es secuencialmente evaluado para cada uno de los 10 subconjuntos. Se eligió $k=10$ en base a la conclusión de la publicación de Rodríguez et al. [54].

En consecuencia, el puntaje de validación cruzada representa el promedio del obtenido para cada uno de los subconjuntos evaluados. Esta técnica permite reducir el nivel de sobreajuste de los datos, debido a que el modelo se entrena sucesivamente con diferentes particiones del conjunto de entrenamiento, lo que introduce variabilidad y una mejor estimación de su desempeño en comparación con el entrenamiento con todo el conjunto en una sola vez. Sin embargo, es una técnica que tiene un costo computacional elevado, debido a que el número de iteraciones a realizar es proporcional al número de particiones elegidas.

Capítulo 5 - Resultados y Discusión

Los valores de rendimiento para cada método de clasificación CNN preentrenado se presentan en la *Tabla 5*. Se puede ver que GoogLeNet fue el más rápido en presentar las características de la imagen en la capa completamente conectada (3,87 segundos en promedio), seguido por AlexNet y VGG-16. La red es 3.6 y 74.32 veces más lenta, respectivamente, que GoogLeNet.

Para los modelos SVM, 86%, 86% y 90% de las muestras de entrenamiento fueron necesarios como vectores de soporte para AlexNet, GoogLeNet y VGG-16 Net, respectivamente. Sin embargo, para GoogLeNet el 66% de ellos eran vectores de soporte de error (variables de holgura $\xi > 0$) mientras que los otros dos rindieron menos del 32%. Esto sugirió que los últimos proporcionaron características más separables que GoogLeNet, y que la reducción del número de características en las capas completamente conectadas no resulta beneficiosa para el enfoque de extracción de características.

En el caso de la capa de clasificación kTSP, todos los métodos alcanzaron casi el mismo número de reglas óptimas, siendo 18, 17 y 20 para AlexNet, GoogLeNet y VGG-16 Net, respectivamente. Al comparar todo el sistema, es decir, el método de clasificación preentrenado - CNN + (SVM o kTSP), el que usa AlexNet, con 4 veces más características que GoogLeNet, ofrece un mejor rendimiento de clasificación para ambos clasificadores probados.

Ambos métodos de clasificación tuvieron un desempeño muy similar en la capa de clasificación, siendo el SVM solo el 2.5% en promedio más preciso con casi el mismo puntaje F (80.02 y 80.04 para SVM y kTSP respectivamente). El mejor rendimiento para ambas capas de clasificación se logró utilizando las características de AlexNet con puntajes F de 85.77% para SVM y 82.12% para kTSP. Sin embargo, este último logra dicho rendimiento mediante el uso de solo 36 características (18 reglas) contra 4096 características utilizadas por el SVM. Además, el comportamiento de las reglas kTSP se puede trazar como la diferencia entre las características involucradas en cada regla, proporcionando una interpretación visual de la función de decisión kTSP como gráficos de barras (*Figura 26*).

ML	CNN	VD	F (%)	Acc (%)	Se(%)	Sp(%)	#SV($\xi > 0$) o Reglas	IPS (sec.)
SVM	Anet	IRDID	82.37	79.98	93.55	66.42		14.19
		RD2	84.13	83.89	85.44	82.33		
		Total	85.77	84.89	91.05	78.73	1428 (69%)	
	Gnet	IRDID	71.55	72.07	70.25	73.88		3.87
		RD2	76.45	77.57	72.82	82.33		
		Total	74.2	75.33	70.94	79.72	1427 (34%)	
	Vnet	IRDID	82.84	82.4	84.95	79.85		287.65
		RD2	79.78	77.86	87.38	68.33		
		Total	80.11	78.75	85.6	71.89	1497 (66%)	
kTSP	Anet	IRDID	78.62	74.37	94.27	54.48		14.19
		RD2	82.53	80.27	93.2	67.33		
		Total	82.12	79.76	92.93	66.59	18	
	Gnet	IRDID	79.22	76.12	91.04	61.19		3.87
		RD2	81.75	79.63	91.26	68		
		Total	80.91	78.5	91.1	65.9	17	
	Vnet	IRDID	81.83	80.27	88.89	71.64		287.65
		RD2	76.82	71.59	94.17	49		
		Total	77.09	73.15	90.31	55.99	20	

Tabla 5: Resultados de los métodos de machine learning. De izquierda a derecha: machine learning (ML), support vector machine (SVM), k-Top Scoring Pairs (kTSP), red neuronal convolucional(CNN), Alexnet (Anet), GoogLeNet (Gnet), VGG-16 Net (Vnet), dataset de validacion (VD), F-score (F%), exactitud (Acc%), sensibilidad (Se%), especificidad (Sp%), numero de vectores de soporte (porcentaje con $\xi > 0$) o numero de reglas de kTSP (#SV($\xi > 0$) o Reglas), velocidad de procesamiento de imágenes (IPS).

5.1 Diferencia entre modelos

Se compararon las curvas ROC de los modelos pertenecientes a características de cada red. Como se puede observar en la *Figura 26*, los modelos de SVM y kTSP de Alexnet y GoogLeNet son bastante similares, en cambio VGG-16 Net presenta diferencias más significativas. De las tres redes, los modelos construidos a partir de Alexnet presentan un mejor poder de diagnóstico.

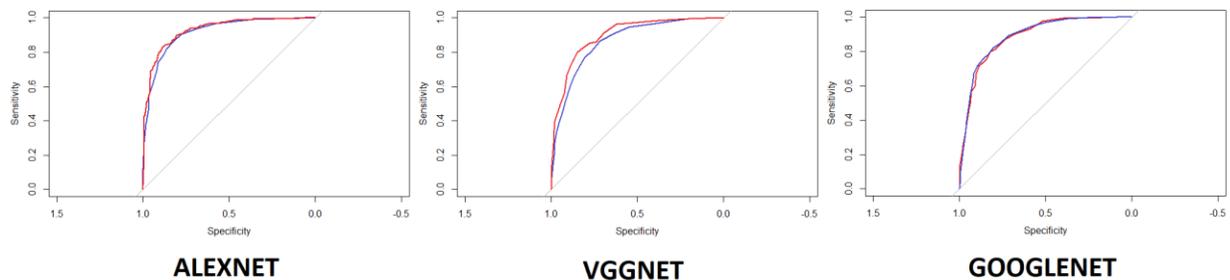


Figura 21: Curvas ROC obtenidas. Modelos de SVM son color rojo y k-TSP de color azul.

5.2 GUI

Se desarrolló una herramienta GUI de MATLAB (*Figura 27*) implementando AlexNet y ambas capas de clasificación. El GUI se desarrolló con la herramienta GUIDE, que es un entorno de programación visual disponible en MATLAB para realizar y ejecutar programas que necesiten ingreso continuo de datos. Tiene las características básicas de todos los programas visuales como Visual Basic o Visual C++ [55]. Para empezar, se diseñó la parte gráfica de la interfaz. Se tiene que definir el uso que se le quiere dar para que contenga todos los elementos relevantes en la interacción entre la aplicación y el clínico. Se pueden añadir botones push, de radio o de activación o grupos de botones, texto estático o editable, deslizadores de opciones, casillas de verificación, menús emergentes, cuadros de listas, tablas, paneles, ejes, gráficas, dependiendo de la finalidad del sistema. En este caso se desea cargar una imagen de fondo de ojo de cualquier computadora, mostrándose por pantalla. Esta imagen se envía como nueva tarea de clasificación a los clasificadores previamente entrenados y almacenados dentro de la aplicación. Se eligió un diseño minimalista con un solo botón push para que la interacción con el profesional de la salud sea de una forma más fácil, cómoda e intuitiva.

Para usar la aplicación debemos pulsar en “load image” y cargar una imagen de fondo de ojo. Automáticamente se muestra en el panel principal y luego es procesada por AlexNet y por ambas capas de clasificación. Al obtener los resultados de clasificación, se muestran en la pantalla las probabilidades posteriores de clase positiva dada la SVM y la representación gráfica de kTSP, con la relación positivo / negativas de las reglas.

Fue nombrado DiRekTS (Evaluación de la Retinopatía Diabética con KTSP y SVM) y la aplicación está disponible libremente en <https://doi.org/10.5281/zenodo.1296971>. La aplicación incluye un manual llamado README.txt (Anexos, punto 5), que explican paso a paso como utilizarla y una imagen de fondo de ojo de prueba.

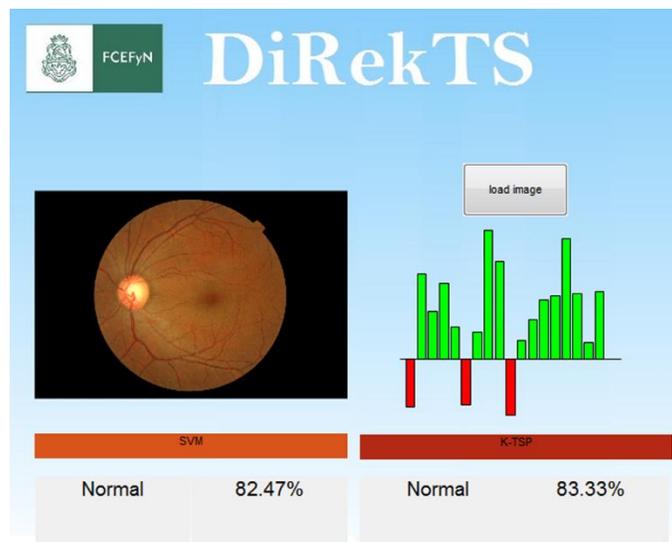


Figura 27: interfaz de usuario

Capítulo 6 - Conclusión

Se mostró un enfoque versátil de clasificación basada en el aprendizaje profundo para la baja disponibilidad de muestras, que se encuentra comúnmente en la mayoría de las clínicas oftalmológicas. El método tiene un rendimiento superior en comparación con otros ya disponibles en la literatura, a pesar de utilizar un conjunto de datos de entrenamiento sustancialmente más pequeño. Se arribó a dos modelos competitivos propuestos con una sensibilidad superior al 91% y puntuaciones F superiores al 82%. El enfoque propuesto resultó ser muy competitivo (*Tabla 6*) frente a otros enfoques de última generación que construyen CNN desde cero utilizando grandes conjuntos de datos, así como la re-anotación de imágenes por parte de los oftalmólogos. A pesar de los conjuntos de datos de hecho con varios miles de imágenes disponibles, como las del repositorio Kaggle [35], su uso no muestra un buen rendimiento (*ver Tabla 6*) debido al alto desequilibrio, tamaño y diferencias de resolución, así como a altos niveles de ruido que requiere una anotación artesanal profesional del estado de la enfermedad para mejorar el rendimiento del modelo. A partir de estos resultados, se puede concluir que la metodología elegida para el pre-procesado de las imágenes fue exitosa, a pesar de haber sido llevada a cabo con pocas referencias. Además, este método se desempeñó bien sobre diferentes fuentes geográficas de los datos, lo que sugiere su idoneidad y robustez. En este sentido, el enfoque propuesto brinda la posibilidad de que las clínicas de ojos con menos recursos ajusten el método propuesto a su población particular o construyan sus propios modelos utilizando bases de datos de imágenes pequeñas.

El haber elegido redes de distintas características permitió observar que no hay una red que para todo problema diferente extraiga mejores características para clasificar [16-18]. Sin embargo, el número de parámetros de una CNN es proporcional al tiempo que se tarda en propagar una entrada hasta la capa de extracción, por lo que se debe analizar el costo-beneficio utilizar aquellas de mayor tamaño. En el caso de este trabajo, donde se busca reducir al menor tiempo posible el proceso de diagnóstico, se hace inviable la utilización de la red VGG-16. Por otra parte, los modelos construidos con características de GoogLeNet presentaron menor poder diagnóstico comparado con AlexNet, por lo que también se descartaron.

Vale la pena mencionar que un método de clasificación de RD debería ser altamente sensible y su predicción también debería ser altamente confiable, es decir, alta predictibilidad positiva [32]. Por lo tanto, el puntaje F, que mide tanto la sensibilidad como la predictibilidad

positiva, es un puntaje de mejor desempeño que incorpora estas dos características positivas para los sistemas de apoyo a la toma de decisiones médicas como los propuestos aquí. Además, considerar modelos con un pequeño número de parámetros como el modelo propuesto de KTSP que utiliza menos del 1% de todas las características de imagen disponibles también es una buena característica bajo el concepto de parsimonia de la teoría de la información [56], con respecto a modelos con menos parámetros, pero igual poder de clasificación. El método kTSP, derivado de la clasificación bioinformática de muestras de cáncer basadas en la expresión génica, resultó ser altamente competitivo en este escenario. Su idoneidad se basa en el hecho de que las características extraídas previamente de la CNN comparten características de distribución similares que hacen plausible la comparación por pares. Este es un nuevo enfoque para este campo. Además, la representación del gráfico de barras de las reglas de KTSP proporciona una guía visual que permite la evaluación de las reglas de decisión, una característica deseable de los sistemas de imágenes de diagnóstico médico a diferencia de otros enfoques de aprendizaje profundo. En conclusión, se lograron los objetivos planteados con resultados mejores de los esperados.

Autor	TDS	VD	Se%	Sp%	Acc%	F%	Oph
trained ophthalmologist [57]	14	Privado	67	84	75	-	
Peng et al. [46]	128175	Eyepacs	97.50	93.40	-	-	yes
		Messidor	96.10	93.90	-	-	
Abràmoff et al. [47]	66680	Private	96.80	87.00	-	-	yes
AlexNet + SVM	1642	DR2/IDRID	91.05	78.73	84.89	85.77	no
AlexNet + kTSP	1642		92.93	66.59	79.79	82.12	
Lopez et. al. [48]	80000	Kaggle	54.47	93.65	83.68	-	no
Pratt et al. [49]	80000	Kaggle	30.00	95.00	75.00	-	no
Rakhlin [50]	81670	Kaggle	92.00	72.00	-		no
		Messidor	99.00	71.00	-		no

Tabla 6: Comparación de resultados. De izquierda a derecho: autor, tamaño de dataset de entrenamiento (TDS), support vector machine (SVM), k-Top Scoring Pairs (kTSP), dataset de validación (VD), sensibilidad (Se%), especificidad (Sp%), exactitud (Acc%), F-score (F%), re-evaluación por un oftalmólogo (Oph).

6.1 Trabajos futuros

En la época actual, hay constante adquisición y armado de bases de datos nuevas. Esto permitiría construir modelos más robustos con una mayor cantidad de datos. No se descarta la posibilidad de desarrollar una base de datos de retinopatía diabética endémica de Argentina, con la que se podría construir un sistema de detección automático más específico a las características poblacionales locales.

Para su implementación o comercialización requeriría llevar a cabo un ensayo en población para ser aprobado por ANMAT. La FDA recién está empezando a certificar este tipo de tecnologías, por lo que hay mucho camino por recorrer, tanto en normativas como legalmente.

Otro punto a consideración es el desarrollo constante de nuevas redes neuronales. La *Figura 28* fue obtenida a partir de estadísticas de Google Scholar de publicaciones en con el término “nueva arquitectura de redes neuronales” en los años 2015 a 2018. Es posible observar un crecimiento exponencial en el desarrollo de nuevas redes neuronales. Habría que explorar si hay cambios significativos en las características extraídas por estas nuevas redes y construir mejores modelos.



Figura 22: Número de publicaciones científicas por año que incluyen el término “nueva arquitectura de red neuronal”

Bibliografía

[1] Terradella, J., Martínez Castro, F., Barría von-Bischho-shausen, F. (2016). Asociación Panamericana de Oftalmología PAAO. Actualización de la Guía clínica de Retinopatía Diabética para Latinoamérica.

[2] Zheng, Y., He, M., & Congdon, N. (2012). The worldwide epidemic of diabetic retinopathy. *Indian journal of ophthalmology*, vol. 60 (5), 428-31. DOI: 10.4103/0301-4738.100542.

[3] Ferris, FL (1993). How effective are treatments for diabetic retinopathy? *JAMA*. 269: 1290–1291. DOI:10.1001/jama.1993.03500100088034.

[4] Abràmoff M.D., Folk JC., Han DP. et al (2013). Automated Analysis of Retinal Images for Detection of Referable Diabetic Retinopathy. *JAMA Ophthalmol*, vol. 131(3), 351–357. DOI: 10.1001/jamaophthalmol.2013.1743.

[5] Srivastava, R. K., Greff, K., & Schmidhuber, J. (2015). Training very deep networks. *Advances in neural information processing systems*, 2377-2385.

[6] FDA (2019). FDA permits marketing of artificial intelligence-based device to detect certain diabetes-related eye problems. [En línea] Disponible en: <https://www.fda.gov/NewsEvents/Newsroom/PressAnnouncements/ucm604357.htm> [Accedido: Mar. 2019].

[7] Soekhoe D., van der Putten P., Plaat A. (2016). On the Impact of Data Set Size in Transfer Learning Using Deep Neural Networks. *IDA*, 50-60. DOI: 10.1007/978-3-319-46349-0_5.

[8] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, vol. 115(3), 211-252. DOI: 10.1007/s11263-015-0816-y.

[9] Latarjet, M. and Ruiz Liard, A. (2005). *Anatomía Humana*. 4ta ed. Buenos Aires: Panamericana, vol. 1, 403-411.

[10] Consejo Internacional de Oftalmología (2017). Guías Clínicas del ICO para el manejo de la patología ocular del diabético.[En línea] Disponible en:

http://www.icoph.org/enhancing_eyecare/international_clinical_guidelines.html [Accedido: Mar. 2019].

[11] Lusby, F., Zieve, D. (2019). Ophthalmoscopy - A.D.A.M. Interactive Anatomy - Encyclopedia. [En línea] Disponible en: <http://aia5.adam.com/content.aspx?productId=117&pid=1&gid=htm> [Accedido: Mar. 2019].

[12] Abramoff M.D., Garvin, M. K., Sonka, M. (2010). Retinal Imaging and Image Analysis. *IEEE Reviews in Biomedical Engineering*, vol. 3, 169-208. DOI: 10.1109/RBME.2010.2084567.

[13] Piniés, J.A. (2005). Retinografía con Cámara no Midriática. *Avances en Diabetología*, vol. 21, 217-221.

[14] CS231n Convolutional Neural Networks for Visual Recognition. [En línea] Disponible en: <https://cs231n.github.io> [Accedido: Mar. 2019].

[15] Herrera A. (2015). Detección de Texto Utilizando Redes Neuronales Convolucionales. Barcelona: Universitat Politècnica de Catalunya.

[16] Alaslani, M., Elrefaei, L. (2018). Convolutional Neural Network Based Feature Extraction for IRIS Recognition. *International Journal of Computer Science & Information Technology (IJCSIT)*, vol. 10(2), 65–78. DOI: 10.5281/zenodo.1288473.

[17] Lopes, U.K., Valiati, J.F. (2017). Pre-trained Convolutional Neural Networks as Feature Extractors for Tuberculosis Detection. *Computers in Biology and Medicine*, vol. 89, 135-143. DOI: 10.1016/j.compbimed.2017.08.001.

[18] Minaee, S., Abdolrashidiy, A., Wang, Y. (2016). An Experimental Study of Deep Convolutional Features for Iris Recognition. *IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, 1-6. DOI: 10.1109/SPMB.2016.7846859.

[19] Canziani, A., Paszke, A., Cukurciello, E. (2016). An Analysis of Deep Neural Network Models for Practical Applications. arXiv preprint arXiv:1605.07678.

[20] Krizhevsky, A., Sutskever, I., Hinton, G. E. (2012). Imagenet Classification with Deep Convolutional Neural Networks. *Advances in neural information processing systems*, 1097-1105. DOI: 10.1145/3065386.

[21] Szegedy, C. et al. (2015). Going Deeper with Convolutions. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1-9. DOI: 10.1109/CVPR.2015.7298594.

[22] Simonyan, K., Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv preprint arXiv: 1409.1556.

[23] Sánchez, J. A. (Septiembre de 2017), Aprendizaje con Pocos Ejemplos, Conferencia realizada en la XLIII Conferencia Latinoamericana de Informática y las 46^o Jornadas Argentinas de Informática, Córdoba, Argentina. [En línea] Disponible en: <https://sites.google.com/view/eaia2017/charlas-y-tutoriales> [Accedido: Mar. 2019].

[24] Deng L., et al. (2013). Deep Learning for Signal and Information Processing. Microsoft Research Monograph.

[25] Garcia-Gasulla, D., Parés, F., Vilalta, A., Moreno, J., Ayguadé, E., Labarta, J., Suzumura, T. (2018). On the Behavior of Convolutional Nets for Feature Extraction. *Journal of Artificial Intelligence Research*, vol. 61, 563-592. DOI: 10.1613/jair.5756.

[26] Fabietti, M.I., Fernández, E.A., Diaz, L. (Octubre de 2017). Machine learning applied to detection of diabetic retinopathy. Proceedings of XXI Congreso Argentino de Bioingeniería y X Seminario de Ingeniería Clínica, SABI 2017, 91-92. Córdoba. [En línea] Disponible en: <http://sabi2017.cim.unc.edu.ar/es/node/26> [Accedido: Mar. 2019].

[27] Vapnik, V. N. (1995). The Nature of Statistical Learning Theory. DOI: 10.1007/978-1-4757-2440-0.

[28] Cohen, D.A., Fernández, E.A. (2012). SVMTOCP: A Binary Tree Base SVM Approach through Optimal Multi-class Binarization. *Lecture Notes in Computer Science*, vol. 7441, 472-478. DOI:10.1007/978-3-642-33275-3_58.

[29] Bahman, A., Fertig, E., Donald, G., Luigi, M. (2015). SwitchBox: An R Package for k-Top Scoring Pairs Classifier Development. *Bioinformatics*, vol. 31, 273–274. DOI: 10.1093/bioinformatics/btu622.

[30] Visa S., Ramsay B., Ralescu A., et al. (2011). Confusion Matrix-based Feature Selection. *Proceedings of The 22nd Midwest Artificial Intelligence and Cognitive Science Conference*, 120-127.

[31] Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, vol. 27, 861-874. DOI: 10.1016/j.patrec.2005.10.010.

[32] Fernández, E.A., Valtuille, R., Presedo, J., Willshaw, P. (2005). Comparison of Different Methods for Hemodialysis Evaluations by Means of ROC Curves: From Artificial

Intelligence to Current Methods. *Clinical Nephrology*, vol. 64(3), 205-213. DOI: 10.5414/CNP64205.

[33] [Base de Datos] VICAVER dataset, VARPA Group, Ophthalmology. [En línea] Disponible en: <http://www.varpa.es/research/ophtalmology.html> [Accedido: Mar. 2019].

[34] [Base de Datos] Structured Analysis of the Retina Dataset. [En línea] Disponible en <http://www.ces.clemson.edu/~ahoover/stare/> [Accedido: Mar. 2019].

[35] [Base de Datos] Kaggle, Inc. (2016). Diabetic Retinopathy Detection. [En línea] Disponible en: <https://www.kaggle.com/c/diabetic-retinopathy-detection> [Accedido: Mar. 2019].

[36] [Base de Datos] High Resolution Fundus (HRF) Image Database. [En línea] Disponible en: <https://www5.cs.fau.de/research/data/fundus-images/> [Accedido: Mar. 2019].

[37] [Base de Datos] Standard Diabetic Retinopathy Database Calibration Level 0 (DIARETDB0). [En línea] Disponible en: <http://www.it.lut.fi/project/imageret/diaretdb0/> [Accedido: Mar. 2019].

[38] [Base de Datos] Standard Diabetic Retinopathy Database Calibration Level 1 (DIARETDB1). [En línea] Disponible en: <http://www.it.lut.fi/project/imageret/diaretdb1/> [Accedido: Mar. 2019].

[39] [Base de Datos] Pires, R., Jelinek, H. F., Wainer, J., Valle, E., and Rocha, A. (2014). Advancing Bag-of-Visual-Words Representations for Lesion Classification in Retinal Images. *PLoS One*, vol. 9(6), e96814. DOI: 10.1371/journal.pone.0096814.

[40] [Base de Datos] Decencièere, E. et al. (2014). Feedback on a Publicly Distributed Database: the Messidor Database. *Image Analysis & Stereology*, vol. 33 (3), p. 231-234. DOI:10.5566/ias.1155.

[41] [Base de Datos] Porwal, P., Pachade, S., Kamble, R., Kokare, M., Girish, D., Sahasrabuddhe, V. and Meriaudeau, F. (2018). Indian Diabetic Retinopathy Image Dataset (IDRiD): A Database for Diabetic Retinopathy Screening Research. *Data*, vol. 3, 25. DOI: 10.3390/data3030025.

[42] [Base de Datos] Decencièere, E. et al. (2013). TeleOphta: Machine Learning and Image Processing Methods for Teleophthalmology. *IRBM*, vol. 34(2), 196-203. DOI: 10.1016/j.irbm.2013.01.010.

[43] Watkins, P.J. (2003). Retinopathy. *BMJ: British Medical Journal*, vol. 326, 924-926. DOI: 10.1136/bmj.326.7395.924.

[44] Ng, A. (2019). Train / Dev / Test sets - Practical aspects of Deep Learning | Coursera. [En línea] Disponible en: <https://es.coursera.org/lecture/deep-neural-network/train-dev-test-sets-cxG1s> [Accedido: Mar. 2019].

[45] Vedaldi, A., Lenc, K. (2015). Matconvnet: Convolutional Neural Networks for Matlab. *Proceedings of the 23rd ACM international conference on Multimedia*, 689-692. DOI: 10.1145/2733373.2807412.

[46] Peng, L., Gulshan, V., Coram, M. et al (2016). Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs. *JAMA*, vol. 316 (22), 2402–2410. DOI: 10.1001/jama.2016.17216

[47] Abràmoff M.D., Lou, Y., Erginay, A., Clarida, W., Amelon, R., Folk, J.C., Niemeijer, M. (2016). Improved Automated Detection of Diabetic Retinopathy on a Publicly Available Dataset Through Integration of Deep Learning. *Invest. Ophthalmol. Vis. Sci.*, vol. 57(13), 5200-5206. DOI: 10.1167/iovs.16-19964

[48] López, J., García, G., Gallardo, J., Mauricio, A., Carpio, C.D. (2017). Detection of Diabetic Retinopathy Based on a Convolutional Neural Network Using Retinal Fundus Images. *ICANN*, 635-642. DOI: 10.1007/978-3-319-68612-7_72.

[49] Pratt, H., Coenen, F., Broadbent, D.M., Harding, S.P., & Zheng, Y. (2016). Convolutional Neural Networks for Diabetic Retinopathy. *Procedia Computer Science*, vol. 90, 200-205. DOI: 10.1016/j.procs.2016.07.014

[50] Rakhlin, A. (2017). Diabetic Retinopathy detection through integration of Deep Learning classification framework. DOI: 10.1101/225508

[51] Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Trans. Sys., Man., Cyber.*, vol. 9 (1), 62–66. DOI: 10.1109/TSMC.1979.4310076

[52] Zuiderveld, K. (1994). Contrast Limited Adaptive Histogram Equalization. *Graphics Gems IV, Academic Press*, 474-485. DOI: 10.1016/B978-0-12-336156-1.50061-6.

[53] Palacios Cuesta, A. (2018). Hyperparameter Optimization for Large-scale Machine Learning. DOI: 10.13140/RG.2.2.33876.65927.

[54] Rodríguez, J., Pérez, A., Lozano, J.A. (2010). Sensitivity Analysis of k-Fold Cross Validation in Prediction Error Estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32 (3), 569-575. DOI: 10.1109/TPAMI.2009.187

[55] Barragán, D. (2007). Manual de Interfaz Gráfica de Usuario en Matlab. [En línea] Disponible en: https://www.dspace.espol.edu.ec/bitstream/123456789/10740/19/%255Bmatlab%255D_MATLAB_GUIDE.pdf [Accedido: Mar. 2019].

[56] MacKay, D. J. C. (2002). *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, New York, NY, USA.

[57] Sundling, V., Gulbrandsen, P., Straand, J. (2013). Sundling, V., Gulbrandsen, P., & Straand, J. (2013). Sensitivity and Specificity of Norwegian Optometrists' Evaluation of Diabetic Retinopathy in Single-Field Retinal Images - a Cross-Sectional Experimental Study. *BMC health services research*, vol. 13, 17. DOI:10.1186/1472-6963-13-17

Anexo

1. CODIGO de construcción del modelo svm [Matlab]

```
% se arma el imagedatastore en referencia a un directorio
imds=imageDatastore('C:\','LabelSource',
'folderNames','IncludeSubfolders',1);

% imprimimos en pantalla en número de imágenes por categoria
countEachLabel(imds)

% asociamos la función de pre-procesado a la llamada de una imagen
imds.ReadFcn = @(filename)readAndPreprocessImagedef(filename);

% cargamos una red neuronal ya obtenida de toolbox, ej:alexnet
net=alexnet;

% declaramos la capa de extracción de característica
featureLayer = 'fc6';

% extraccion de características
trainingFeatures= activations(net, imds, featureLayer,'MiniBatchSize',
32, 'OutputAs', 'columns');

% extraccion de etiquetas del datastore
trainingLabels =imds.Labels;

% restamos la media de cada característica por columna
trainingFeaturesEQU=
bsxfun(@minus,trainingFeatures,median(trainingFeatures));

% optimizamos hiperparametros

load ionosphere

rng default

MdlT=
fitcsvm(trainingFeaturesEQU,trainingLabels,'Standardize',true,'KernelF
unction','gaussian', ...
```

```
'OptimizeHyperparameters','auto','HyperparameterOptimizationOptions',s  
truct('AcquisitionFunctionName','expected-improvement-plus'));  
  
% hacemos cross-validacion del modelo  
  
CMdl2 = crossval(MdlT);  
  
L1 = kfoldLoss(CMdl2);  
  
% vemos la precisión del modelo  
  
validationAccuracy = 1 - L1
```

2. Código de función de pre-procesado “readAndPreprocessImagedef”

[Matlab]

```
I=imread(strcat(Path,FileName));
rojo=I(:,:,1);
H=fspecial('average');
GS=imfilter(rojo,H);
umbral=0.15;
fondo1=im2bw(rojo,umbral);
str=strel('disk',3,0);
final=imopen(fondo1,str);
finalb=1-final;
I2 = im2uint8(finalb);
fondo = cat(3,I2,I2,I2);
g=imsubtract(I,fondo);
saco=rgb2gray(g);
columna=max(saco);
RemoveC=find(columna<10);
g(:,RemoveC,:)=[];
fila= max(saco,[],2);
RemoveF=find(fila<10);
g(RemoveF,:,:)=[];
tamaño=size(g);
alto=tamaño(1);
ancho=tamaño(2);
taza=alto/ancho;
if (taza>0.91)
borro1=ancho*0.5*(1-0.83);
borro2=round(borro1);
limiteinf=borro2;
limitesup=alto-borro2;
final=g(limiteinf:limitesup,:,:);
g=final;
end
```

```
R=g(:, :, 1);  
G=g(:, :, 2);  
B=g(:, :, 3);  
AD=adapthisteq(R);  
AD2=adapthisteq(G);  
AD3=adapthisteq(B);  
Nadpt=cat(3,AD,AD2,AD3);  
Iout = imresize(Nadpt, [227 227]);  
End
```

3. CODIGO de construcción del modelo k-TSP [R]

```
% instalamos las librerías a usar
install.packages("readxl")
source("http://bioconductor.org/biocLite.R")
biocLite("switchBox")

% cargamos las librerías a usar
require(switchBox)

library(readxl)
% cargamos las características extraídas
valormessidor<-read_excel("C:/ ")
etiquetasmessidor<-read_excel("C:/ ")

% estructuramos las matrices para el modelo
etiquetasmessidor<-t(etiquetasmessidor)
labels<-factor(etiquetasmessidor)
value<-as.matrix(valormessidor)
rownames(value)<-rownames(valormessidor)

% construimos el modelo de ktsp
classifiercomp<- SWAP.Train.KTSP(value,labels, krange=2:25, FilterFunc
= SWAP.Filter.Wilcoxon,verbose = TRUE)

% vemos que pares arman las reglas
classifiercomp$TSPs
```

4. Código de GUI [Matlab]

```
function varargout = DIREKTS(varargin)
% DIREKTS MATLAB code for DIREKTS.fig
%     DIREKTS, by itself, creates a new DIREKTS or raises the existing
%     singleton*.
%
%     H = DIREKTS returns the handle to a new DIREKTS or the handle to
%     the existing singleton*.
%
%     DIREKTS('CALLBACK',hObject,eventData,handles,...) calls the local
%     function named CALLBACK in DIREKTS.M with the given input
arguments.
%
%     DIREKTS('Property','Value',...) creates a new DIREKTS or raises
the
%     existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before DIREKTS_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to DIREKTS_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only
one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES
%
% Edit the above text to modify the response to help DIREKTS
%
% Last Modified by GUIDE v2.5 03-Jun-2018 15:38:57
%
% Begin initialization code - DO NOT EDIT
```

```

gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @DIREKTS_OpeningFcn, ...
                  'gui_OutputFcn',  @DIREKTS_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);

if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before DIREKTS is made visible.
function DIREKTS_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to Figura
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to DIREKTS (see VARARGIN)
%ORGANIZA LA PARTE GRAFICA (BOTONES/FONDO)
axes(handles.axes3)
axis off;
axes(handles.axes4)
axis off;
axes(handles.axes1)
axis off;
background = imread('fondo.jpg');
imshow(background);

```

```

set(handles.text2, 'Visible', 'off');
set(handles.text6, 'Visible', 'off');
set(handles.text8, 'Visible', 'off');
set(handles.text9, 'Visible', 'off');
svm='SVM';
set(handles.text4, 'String', svm)
set(handles.text4, 'BackgroundColor', [0.85 0.33 0.1])
ktps='K-TSP';
set(handles.text5, 'String', ktps)
set(handles.text5, 'BackgroundColor', [0.7 0.16 0.08])

% Choose default command line output for DIREKTS
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes DIREKTS wait for user response (see UIRESUME)
% uiwait(handles.Figural);

% --- Outputs from this function are returned to the command line.
function varargout = DIREKTS_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to Figura
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

function pushbutton2_Callback(hObject, eventdata, handles)
cla(handles.axes4, 'reset')
axes(handles.axes4)

```

```

axis off;
[FileName Path]=uigetfile({'*.jpg;*.bmp;*.tif;*.png'},'Abrir Imagen');
if isequal(FileName,0)
return
else
I=imread(strcat(Path,FileName));
axes(handles.axes3)
axis off;
imshow(I);
% PREPROCESADO
rojo=I(:,:,1);
H=fspecial('average');
GS=imfilter(rojo,H);
umbral=0.15;
fondo1=im2bw(rojo,umbral);
str=strel('disk',3,0);
final=imopen(fondo1,str);
finalb=1-final;
I2 = im2uint8(finalb);
fondo = cat(3,I2,I2,I2);
g=imsubtract(I,fondo);
saco=rgb2gray(g);
columna=max(saco);
RemoveC=find(columna<10);
g(:,RemoveC,:)=[];
fila= max(saco,[],2);
RemoveF=find(fila<10);
g(RemoveF,:,:)=[];
tamaño=size(g);
alto=tamaño(1);
ancho=tamaño(2);
taza=alto/ancho;
if (taza>0.91)
borro1=ancho*0.5*(1-0.83);
borro2=round(borro1);

```

```

limiteinf=borro2;
limitesup=alto-borro2;
final=g(limiteinf:limitesup, :, :);
g=final;
end
R=g(:, :, 1);
G=g(:, :, 2);
B=g(:, :, 3);
AD=adapthisteq(R);
AD2=adapthisteq(G);
AD3=adapthisteq(B);
Nadpt=cat(3,AD,AD2,AD3);
Iout = imresize(Nadpt, [227 227]);
end
handles.direccion=strcat(Path,FileName);
handles.Iout=Iout;
load 'variables.mat'
IM=handles.Iout;
% OBTENCION DE CARACTERISTICAS
net=alexnet;
featureLayer = 'fc6';
trainingFeaturesB= activations(net, IM, featureLayer, 'MiniBatchSize',
32, 'OutputAs', 'columns');
% SVM
trainingFeaturesB=trainingFeaturesB';
trainingFeaturesEQUB =
bsxfun(@minus, trainingFeaturesB, median(trainingFeatures50));
[label, scores] = predict(svm, trainingFeaturesEQUB);
label=char(label);
color=strcmp(label, 'Background Diabetic Retinopathy');
if color==1
set(handles.text2, 'BackgroundColor', [0.3 0.75 0.93])
set(handles.text8, 'BackgroundColor', [0.3 0.75 0.93])
label=char(label);
probsvm=num2str(round(scores(1,1)*100,2));

```

```

probsvm=strcat(probsvm,'%');
else
set(handles.text2,'BackgroundColor',[0.94 0.94 0.94])
set(handles.text8,'BackgroundColor',[0.94 0.94 0.94])
probsvm=num2str(round(scores(1,2)*100,2));
probsvm=strcat(probsvm,'%');
end
set(handles.text2, 'Visible', 'on');
set(handles.text2,'String',label)
set(handles.text8, 'Visible', 'on');
set(handles.text8,'String',probsvm)
% KTSP
k=0;
x=zeros(1,18);
x(1)=trainingFeaturesEQUB(1,3860)-trainingFeaturesEQUB(1,2020);
x(2)=trainingFeaturesEQUB(1,2671)-trainingFeaturesEQUB(1,2236);
x(3)=trainingFeaturesEQUB(1,2724)-trainingFeaturesEQUB(1,3397);
x(4)=trainingFeaturesEQUB(1,2204)-trainingFeaturesEQUB(1,3157);
x(5)=trainingFeaturesEQUB(1,3537)-trainingFeaturesEQUB(1,1221);
x(6)=trainingFeaturesEQUB(1,1097)-trainingFeaturesEQUB(1,890);
x(7)=trainingFeaturesEQUB(1,3672)-trainingFeaturesEQUB(1,1010);
x(8)=trainingFeaturesEQUB(1,2822)-trainingFeaturesEQUB(1,263);
x(9)=trainingFeaturesEQUB(1,2200)-trainingFeaturesEQUB(1,3817);
x(10)=trainingFeaturesEQUB(1,1539)-trainingFeaturesEQUB(1,1632);
x(11)=trainingFeaturesEQUB(1,604)-trainingFeaturesEQUB(1,2152);
x(12)=trainingFeaturesEQUB(1,3409)-trainingFeaturesEQUB(1,3638);
x(13)=trainingFeaturesEQUB(1,1916)-trainingFeaturesEQUB(1,236);
x(14)=trainingFeaturesEQUB(1,937)-trainingFeaturesEQUB(1,1218);
x(15)=trainingFeaturesEQUB(1,676)-trainingFeaturesEQUB(1,95);
x(16)=trainingFeaturesEQUB(1,248)-trainingFeaturesEQUB(1,3651);
x(17)=trainingFeaturesEQUB(1,3795)-trainingFeaturesEQUB(1,823);
x(18)=trainingFeaturesEQUB(1,3747)-trainingFeaturesEQUB(1,1140);
for i=1:18
    if x(i)>0
        k=k+1;
    end
end

```

```

        end
end
if k<10
diag='Background Diabetic Retinopathy';
diag2=num2str(round((18-k)*100/18,2));
diag2=strcat(diag2,'%');
else
diag='Normal';
diag2=num2str(round(k*100/18,2));
diag2=strcat(diag2,'%');
end
color2=strcmp(diag,'Background Diabetic Retinopathy');
if color2==1
set(handles.text6,'BackgroundColor',[0.3 0.75 0.93])
set(handles.text9,'BackgroundColor',[0.3 0.75 0.93])
else
set(handles.text6,'BackgroundColor',[0.94 0.94 0.94])
set(handles.text9,'BackgroundColor',[0.94 0.94 0.94])
end
set(handles.text6,'Visible','on');
set(handles.text6,'String',diag)
set(handles.text9,'Visible','on');
set(handles.text9,'String',diag2)
axes(handles.axes4)
hold on
for i = 1:18
    if x(i)>0
        bar(i, x(i),'FaceColor','g');
    else
        bar(i, x(i),'FaceColor','r');
    end
end
hold off
guidata(hObject,handles)
%hObject    handle to pushbutton1 (see GCBO)

```

```
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% 67.8202 81.2470 62.1610

%9.0408e-04 9.3269e-04 8.3542e-04
```

5. Manual de la aplicación - Readme

DiRekTS (Diabetic Retinopathy Evaluation with kTSP and SVM) for MATLAB

DiRekTS is a MATLAB GUI tool that classifies fundus images in whether it presents diabetic retinopathy or is healthy.

The algorithm was developed implementing the pret-CNN and k-Top Scoring Pairs and Support Vector Machine, for AlexNet extracted-features.

The Software allows loading a fundus image, which is displayed on the main panel which is processed by the AlexNet and by both classification layers providing the positive class posterior probabilities given the SVM and the graphical representation of kTSP appears on screen, with the positive/negative rule ratio.

An example fundus image is included in the zip.

Requirements

You will need a sufficiently recent MATLAB version (MATLAB 9.1 (R2016b) or newer). Depending on the specific MATLAB version, further constraints may apply.

1. Start MATLAB
2. Select the 'APPS' section, and click on 'Get More Apps'
3. On the Search bar, type: 'Neural Network Toolbox Model for AlexNet Network'
4. Select 'Add' and then 'Add to Matlab'
5. Check it is installed by typing

```
> alexnet
```

Using DiRekTS

In order to use, follows these steps:

1. Download and unpack the library source code into a directory of your choice.
2. Start MATLAB and search for the directory containing the folder. Then, type:

```
> cd DiRekTS
```

3. Open the code by typing

```
> open 'DIREKTS.m'
```

4. Now, click on the 'EDITOR' section, followed by 'Run'

You are now able to use the app.

As an example, click 'load image' on the app, and select 'test.tiff', after a few seconds you will be able to see its diagnosis.