

# Desarrollo de software para clasificación de imágenes de retinopatía diabética

Alan David Blanco Miranda Dante Alonso Hernández Tapia, Heraclio García Cervantes,  
Miguel Ángel Santacruz Belmontes

Universidad Tecnológica de León; Blvd. Universidad tecnológica #225. Col. San Carlos.

## Resumen

La retinopatía diabética es una enfermedad que afecta a pacientes con diabetes, con un amplio número de afectados quienes son alrededor de 13 millones en México y más de 400 millones en el mundo. La consecuencia más grave de no tratar las etapas tempranas de la retinopatía diabética es la pérdida de la vista.

Para poder atacar este problema, actualmente, se utilizan modelos de inteligencia artificial usando métodos de aprendizaje profundo (deep learning) en redes neuronales convolucionales (cnn), con las cuales se pueden realizar las clasificaciones de imágenes y mediante fotografías obtenidas con cámaras de fondo de ojo, se genera el entrenamiento de un modelo de aprendizaje automático.

En el presente trabajo, para detectar las diferentes etapas de la retinopatía diabética se hace uso del modelo de clasificación de imágenes basado en el modelo VGG16, con la finalidad de clasificar e implementar dicho procedimiento en una interfaz gráfica de usuario.

## Abstract

Diabetic retinopathy is a disease that affects patients with diabetes, whom are more than 13 million in Mexico and more than 400 million in the world. The most serious consequence of not treating the early stages of diabetic retinopathy is loss of sight.

To attack this problem, artificial intelligence models are currently being developed using deep learning methods in convolutional neural networks (cnn), with which image classifications can be carried out and through photographs obtained with fundus cameras, the training of a machine learning model is generated.

In the present work, to detect the different stages of diabetic retinopathy, an image classification model based on the VGG16 network is used, to classify and integrate it into a graphical user interface.

**Palabras Clave:** Retinopatía diabética, red neuronal convolucional, inteligencia artificial.

**Keywords:** Diabetic retinopathy, convolutional neural network, artificial intelligence.

## 1. INTRODUCCIÓN

En los últimos años se ha incrementado el número de personas que padecen de algún tipo de diabetes, aunque es tratable, hay enfermedades desencadenadas que pueden incluso producir ceguera como en el caso de la retinopatía diabética (Montes, 2018).

Una red neuronal convolucional es una arquitectura o modelo de aprendizaje automático usado en la inteligencia artificial que ayuda al tratamiento de imágenes, procesándolas como matrices (Caparrini, 2019).

Hasta ahora se han desarrollado varias técnicas para identificar la retinopatía diabética usando imágenes de fondo de ojo, los cuales van desde la detección de los componentes de la retina como el disco óptico, los vasos sanguíneos y la fovea, mediante procesamiento digital de imágenes, detectando variaciones en los píxeles de las imágenes como valores de una matriz. En el 2008 Abràmoff *et al.*, realizan una secuencia de operaciones a imágenes de la retina, como el dimensionamiento a imágenes 640 x 640, determinando la probabilidad para establecer que la imagen era de mala calidad, donde se realizaba la segmentación automática de los vasos



sanguíneos, hacía la detección y enmascaramiento automático del disco óptico, incluso se detectaron microaneurismas, hemorragias y exudados, principales signos derivados de la retinopatía diabética. Estas clasificaciones si bien fueron buenas, se pueden mejorar todavía, es por eso por lo que años después se han utilizado nuevos métodos de procesamiento para las imágenes que ayudaban a mejorar la detección usando máquinas de soporte vectorial, procesamiento profundo, la transformada de Hough, entre otras técnicas (Hammad, 2016). Más adelante se generaron bases de datos open source como el Messidor 2 que recolectaron gran cantidad de imágenes de personas en diferentes etapas de la retinopatía diabética, con la intención de generar modelos predictivos más avanzados y profundos y poder encontrar mejores soluciones (Decencièrre, 2014).

La retinopatía diabética afecta sobre todo al interior de los ojos, es debido a ello que se requiere el uso de cámaras de fondo de ojo las cuales tienen la capacidad y resolución suficiente para capturar videos y fotografías a través de la pupila del paciente y que posteriormente, se pueden evaluar para detectar las diferentes anomalías que aparecen en la retina.

Actualmente en Kaggle se puede encontrar una base de datos de más de 80 GB de información que contienen imágenes de fondo de ojo, mismos que pueden ayudarnos a tener una base para un modelo que puede ser entrenado con gran cantidad de pacientes y posteriormente se puedan añadir métodos nuevos y optimizados para la detección de la retinopatía diabética en sus diferentes etapas (Tanishq, 2019).

El presente trabajo busca ser un avance en el desarrollo de hardware que pueda contraer el modelo de entrenamiento mediante una interfaz gráfica amigable y fácil de usar y que su nivel de alcance a la población sea más grande desde su bajo costo y alta eficiencia.

## 2. MARCO TEÓRICO

El **aprendizaje automático** o machine learning son la serie de algoritmos que hacen que un sistema aprenda, ya que combinan entradas de información para producir predicciones útiles sobre datos nunca vistos o registrados. Como terminologías claves se encuentran las **etiquetas**, que es el valor que se intenta predecir con estos sistemas de aprendizaje automático mediante un sistema computacional. Para cada etiqueta se debe de conocer y procesar sus **atributos**, estos son una o más variables de entrada. Como ejemplo, en un gato que tiene color, forma, textura, etc., la etiqueta sería gato y sus atributos serían color, forma, textura, etc.

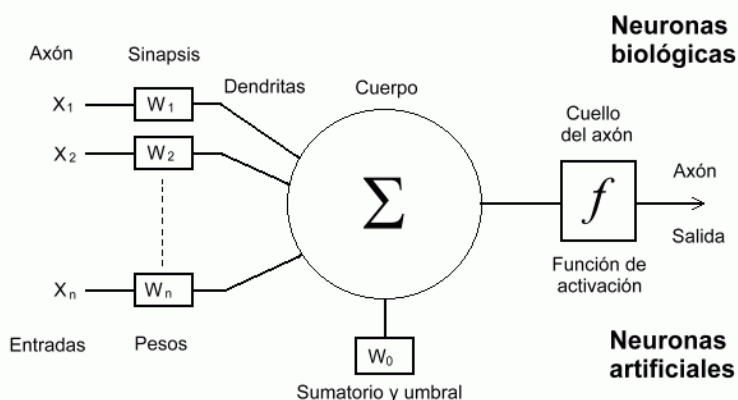
Las relaciones que existen entre los atributos y etiquetas se les conoce como **modelo** y su finalidad es predecir o detectar etiquetas con respecto a los atributos.

El **entrenamiento** crea o aprende de ejemplos de una base de datos donde existan etiquetas y sus atributos correspondientes, el entrenamiento aprende de los patrones existentes entre la o las etiquetas y sus atributos para hacer una predicción con base en los datos contenidos dentro de la base de datos con la cual el modelo fue entrenado. La **interferencia** surge cuando se pone a prueba el modelo a fin de que entregue una predicción, en este proceso, se le agregan los atributos incluso si estos no estuviesen contenidos dentro de la base de datos con la que fue entrenado un modelo en específico, y a pesar de esto, el modelo pueda ser capaz de dar un resultado de predicción.

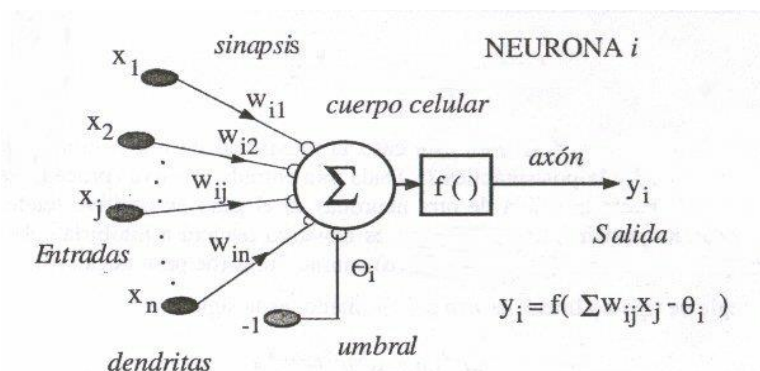


Un modelo se clasifica en dos, una regresión o clasificación, la **regresión** predice con valores continuos o flotantes, probabilidades y valores numéricos, en tanto que la **clasificación** predice valores enteros, discretos o booleanos (Google, 2018).

Las **redes neuronales artificiales** intentan representar el modelo biológico del cerebro humano mediante modelos matemáticos. La neurona artificial trata de simular el comportamiento de una neurona biológica basándose en el sistema de comunicación neuronal. Las entradas o receptores de información del modelo neuronal están representadas por  $X_1$  y  $X_2$  se pueden extender hasta  $X_n$  dependiendo de la aplicación que se requiera, estas entradas son las encargadas de recibir la información proveniente del exterior y propagarlas hacia las neuronas que se encuentran en la siguiente capa o nivel. Los pesos sinápticos están representados por  $W_1$ ,  $W_2$  hasta  $W_n$ , estos se modifican mediante un algoritmo para que la red se adapte y aprenda. En el bloque sumador se obtienen las sumas ponderadas de cada una de las entradas que luego son evaluadas en una función de activación, ésta se encarga de transformar la entrada total en la respuesta de la neurona, limitando el rango de respuesta de dicha neurona, Fig. 1 y Fig. 2.



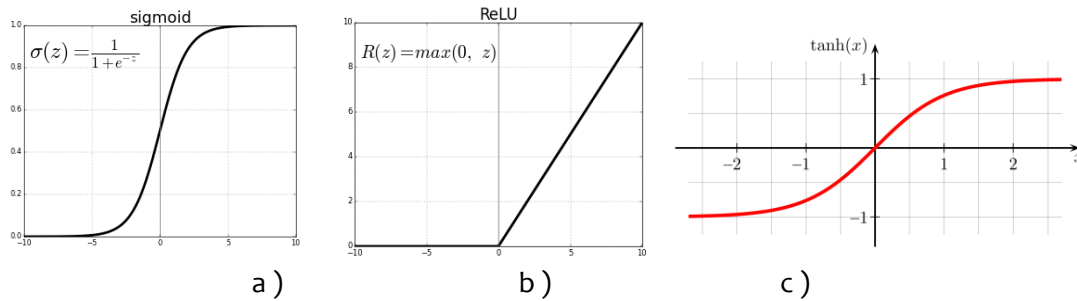
**Figura 1.** Expresión matemática del funcionamiento interno de una neurona artificial para dar una salida (Caparrini, 2019).



**Figura 2.** Diagrama matemático de la neurona artificial de cómo se asemeja el funcionamiento a una neurona biológica. (Nacelle, 2009).

La función de activación es una fase importante, ya que es la encargada de ponderar un valor numérico a la predicción realizada por un modelo, entre las más utilizadas están la función sigmoideal (Fig. 3a), cuyos valores están entre 0 y 1, Unidad lineal rectificada (ReLU Fig. 3b), con valores entre 0 y el infinito, y la tangente hiperbólica (Fig. 3c), la cual sus valores de salida pueden estar entre  $-1$  y  $1$ . Cada una tiene sus ventajas y desventajas. (Weissstein, 2020)





**Figura 3.** Funciones de activación. a) Sigmoide; b) ReLU; c) Tangente hiperbólica.

Tradicionalmente en un modelo de predicción se utiliza la función de activación sigmoidal, o incluso una función tangente hiperbólica, ReLU aparece para calcular los valores de activación en redes neuronales tradicionales o en redes neuronales profundas, la razón es debido a la capacidad de consumo en recursos computacionales que ofrece, ya que puede acelerar el entrenamiento de redes neuronales profundas comparado con las funciones de activación usadas en las redes neuronales tradicionales, como consecuencia de contar con una derivada de valor 1 para una entrada positiva y reducir errores en la fase de entrenamiento. Al conocer la predicción se evalúa el error, y con base en él, se procede a una retro-propagación o propagación hacia atrás, consiste en el método de ir calculando una variación de valores desde la predicción hasta cada uno de los pesos y bias, siguiendo el sentido contrario a como sucede el cálculo de la predicción, esto se consigue con las derivadas parciales de todos los cálculos realizados (Molina, 2019).

Para el cálculo del error, se utiliza la fórmula del error medio cuadrático.

$$MSE = \frac{1}{n} \sum_{j=1}^n (\hat{Y}_j - Y_j)^2$$

*MSE: (Mean Square Error), error medio cuadrático es una fórmula utilizada para calcular qué tan alejada es la predicción con respecto a los Target registrados en la base de datos.*

*N: Número de neuronas en la red neuronal artificial.*

*$Y_j$ : Target o salida correcta registrada en la base de datos con respecto a las entradas iniciales en la fase de entrenamiento.*

*$\hat{Y}_j$ : Predicción, un valor que deberá de ser muy cercano a  $Y_j$ .*

### 3. REDES NEURONALES CONVOLUCIONALES

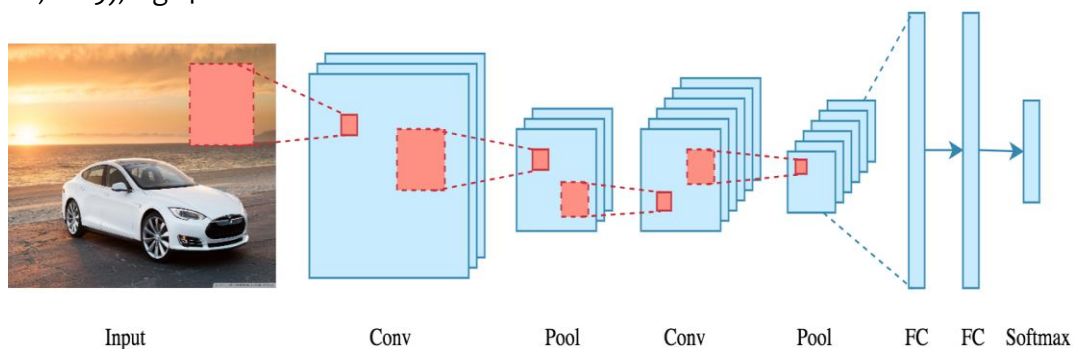
Para el reto de trabajar con imágenes, existen las **redes neuronales convolucionales**, éste es un tipo de red neuronal artificial que simulan los campos receptivos como lo hacen las neuronas en la corteza visual primaria de un cerebro biológico, son una variación del perceptrón multicapa y al redimensionamiento de matrices bidimensionales son efectivas para desarrollo de software para realizar tareas de visión artificial como en la clasificación y segmentación de imágenes, entre otras aplicaciones (Usher, 2003).

El funcionamiento de las redes neuronales convolucionales se basa en agregar pasos previos al tratado de los datos, en este caso, al considerar una clasificación de imágenes, se aplica una operación de convolución.



En las redes neuronales tradicionales como el perceptrón simple o el perceptrón multicapa, el diagrama macroscópico constaba de las capas de entrada, las capas ocultas y las capas de salida, en las redes neuronales convolucionales para tratar imágenes, es necesario aplicar una convolución a estas matrices, este proceso consta de ingresar un kernel o filtro el cual es una matriz de 3x3, 4x4 o 5x5 usualmente, y con ciertos valores pueden generar un cambio en los valores que conforman la imagen en sus 3 canales si la imagen es a color y en 1 canal, si la imagen es en blanco y negro, el kernel recorre cada parte de la imagen generando otras imágenes de menor tamaño, pero de mayor profundidad, el otro paso es ir aplicando un Max-Pooling, para resumir la matriz resultante de la convolución, en otra matriz de misma profundidad, pero de dimensiones más pequeñas, esto último se consigue cuando se pasa un filtro donde va sacando los valores máximos o saca promedio de los valores que va escaneando el proceso del Max-pooling sobre la matriz convolucionada.

El siguiente paso podría ser agregar otra convolución, con otro o el mismo filtro o kernel, que reduzca un poco las dimensiones y aumente la profundidad de la matriz resultante y volver a aplicar otro Max-Pooling que reduzca las dimensiones y mantenga la profundidad nuevamente, luego del proceso de convolución se aplica un aplanamiento sobre la matriz resultante de todo el proceso, la cual da como resultado una capa de entrada que puede conectarse a capas ocultas con respectivos pesos, bias y función de activación, para estos casos es preferible usar la función ReLU, pero también se puede usar la función sigmoideal o tangente hiperbólica y por últimos las capas de salida que serán las encargadas de dar las predicciones de clasificación correspondientes (Basaveswara, 2019), Fig. 4.



**Figura 4.** Se aprecia como una imagen de entrada va pasando por diversos filtros de convolución y reducción y finalmente sufre un aplanado, el softmax al final, es para una clasificación de múltiples etiquetas, para el ejemplo visto en la imagen de entrada, se puede especular que es para la predicción de modelos de diversos automóviles o con respecto a otros objetos (Silva, 2019).

#### 4. METODOLOGÍA

##### Creación de una base de datos estructurada

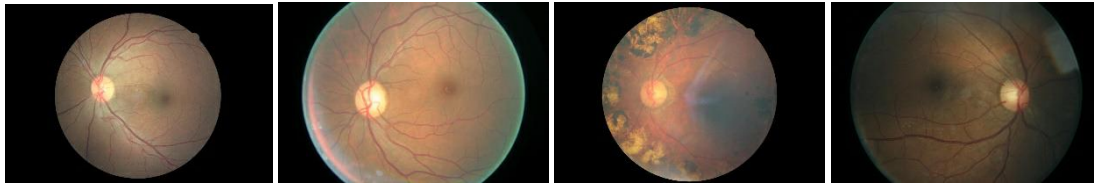
La elección de una base de datos para entrenar un modelo predictivo es el primer paso, y es muy importante para que se pueda generar un modelo confiable y que no tenga variaciones al implementarlo en una aplicación real de la que dependerán, en este caso de estudio, el diagnóstico y nivel de avance de una enfermedad.

La base de datos que se escogió para esta investigación fue obtenida por la plataforma de Kaggle, sobre imágenes de fondo de ojo y los 4 niveles de avance de la retinopatía diabética. Al ser descargada se procedió a balancearla, ya que es necesario contar con una cantidad estándar entre el número de imágenes con base en el número de etiquetas, cada etiqueta representa un nivel de la retinopatía diabética, y todas las etiquetas debe



poseer el mismo número de imágenes de entrenamiento y de validación para evitar el sobre aprendizaje (Teng, 2002).

Un problema que se encontró es que los datos no estaban del todo normalizados, ya que variaban en muchas características (Fig. 5).

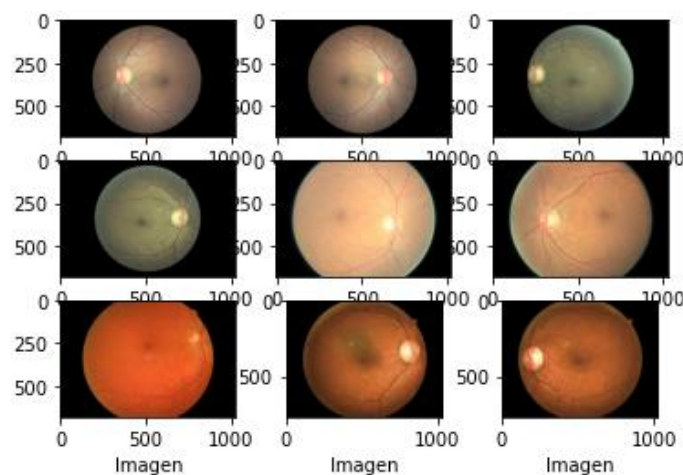


**Figura 5.** Cuatro imágenes de muestra de la base de datos de Kaggle sobre retinopatía diabética.

Las imágenes contenidas en la base de datos constan de varias irregularidades, y esto, al considerar 5 clases o etiquetas donde las imágenes tienen muchas similitudes en cuanto a los colores, además de algunos reflejos no deseados generados al momento de capturar las imágenes de fondo de ojo se traduce en una mayor dificultad de clasificación y aprendizaje para el modelo. Dadas estas condiciones, aplicar un preprocesamiento digital, así como filtros es de vital importancia para el análisis generado ya que se debe tener la precisión suficiente la cual produzca que el modelo aprecie diferencias entre pacientes de cada etapa de la retinopatía diabética.

En este caso se hará uso de la librería Open CV de Python para el procesamiento digital de imágenes utilizada en aplicaciones de visión computacional, y de Matplotlib para graficar los resultados obtenidos.

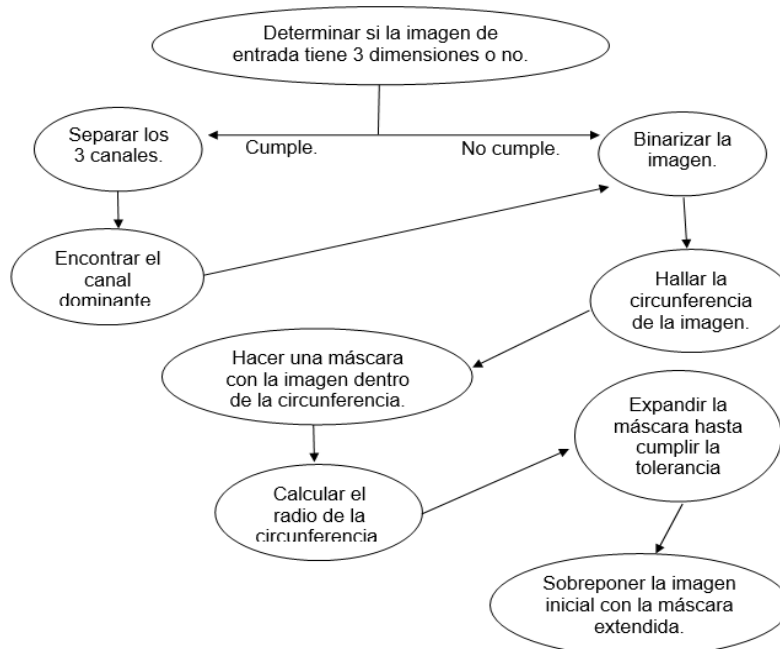
En la figura 6 se muestran 9 imágenes tomadas de manera aleatoria de la base de datos de Kaggle para mostrar el resultado antes (Fig. 6) y después (Fig.7) de realizar el preprocesamiento y aplicación de filtros.



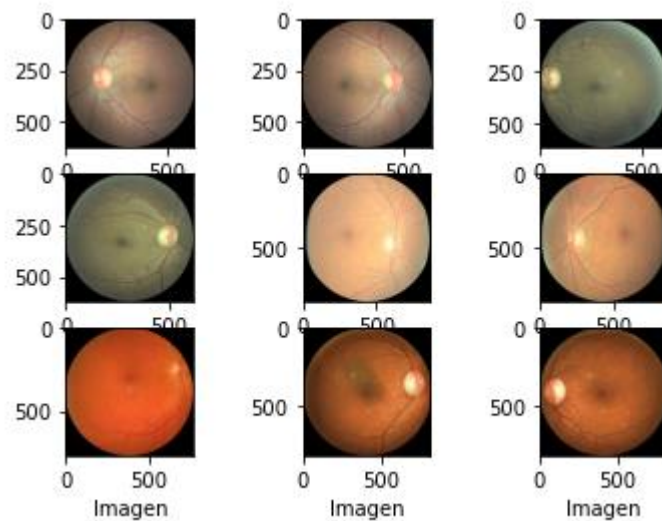
**Figura 6.** Imágenes muestra aleatorias.

Uno de los principales problemas en las imágenes es la variedad de tamaño en la circunferencia que engloba la retina. Con el objetivo de normalizar bajo cierta tolerancia este aspecto, se emplean máscaras las cuales se aplican sobre cada una de las capas de una imagen a color encapsulando la retina y reescalandola, generando imágenes uniformes, como se esquematiza en el diagrama 1.





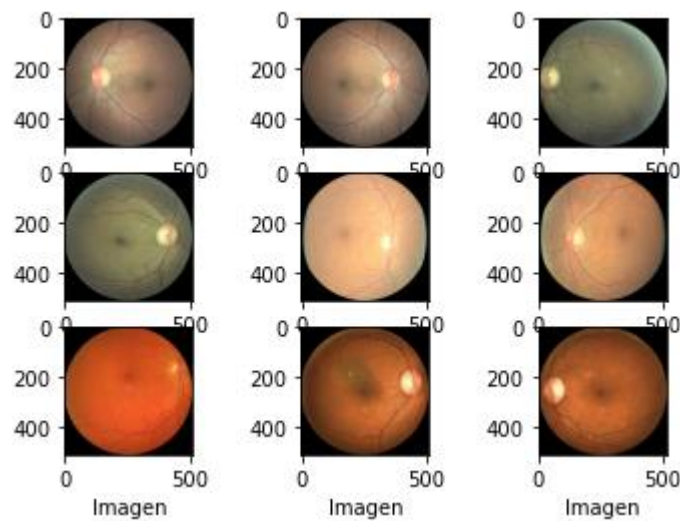
**Diagrama 1.** Diagrama de representación del flujo de secuencias que procesa la imagen entrada en la función. La figura 7 muestra el resultado de aplicar el algoritmo de reescalado a las imágenes muestra de la Fig. 6



**Figura 7.** Imágenes reescaladas desde la circunferencia que engloba la retina.

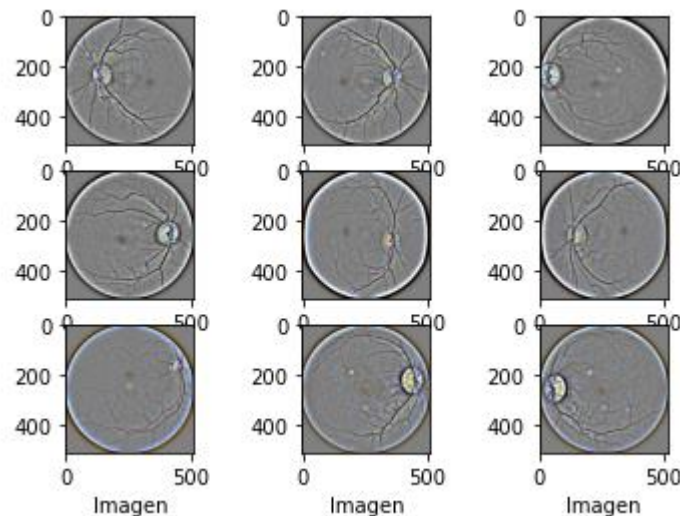
El radio de la retina ha sido normalizado como se muestra en la figura 7, sin embargo, las imágenes varían de tamaño en píxeles (largo, ancho). Debido a esto, se procedió al uso de una función de reescalamiento de imágenes (Fig. 8).





**Figura 8.** Imágenes redimensionadas en largo y ancho.

Finalmente, a las imágenes se les aplicará un filtro de media ponderada teniendo de base un filtro de difuminado gaussiano (Gimenez-Palomares, 2016), resultando en imágenes normalizadas en el color (Fig. 9)



**Figura 9.** Imágenes resultantes de aplicar un filtro de media ponderada con base en el difuminado gaussiano (tonalidad similar y contornos claros).

### Desarrollo de un modelo predictivo con redes neuronales convolucionales

Una vez establecida la base de datos con una mayor estructura, se procede a la realización del modelo predictivo.

El primer paso en la programación para crear una red neuronal convolucional de clasificación es importar las librerías necesarias para generar la arquitectura de procesamiento digital de imágenes, usando Tensor Flow



como marco de trabajo, así mismo se aprovecha y se importan las librerías de uso del sistema operativo y del análisis matemático y estadístico.

Se inicializan las variables que contendrán las direcciones donde se encuentran las imágenes de la base de datos descargada, previamente separadas por sus etiquetas y siguiendo un almacenamiento binario para dar positivo o negativo a la retinopatía diabética y también por dos lotes, uno de entrenamiento para la red neuronal convolucional y otra de validación para calificar el modelo de clasificación.

Los hiperparámetros son la parte experimental y de los cuales se tendrá el control dentro del proceso de entrenamiento y validación. Las épocas son el número de procesos sobre los cuales un lote de imágenes obtenida de la base de datos es entrenado con respecto al número de pasos. El `batch_size` es el tamaño del lote de imágenes que serán seleccionados por cada paso. Como anotación importante, los pasos de validación deben ser parecidos a los de entrenamiento, ya que será el número de veces que se calculará el nivel de efectividad del modelo predictivo.

El tamaño de los filtros serán de utilidad en los filtros de convoluciones y de max-pooling, al principio será una clasificación binaria por lo que el número de clases totales serán solamente 2 en un inicio y el learning rate o tasa de aprendizaje será un número que describa el movimiento del gradiente descendiente y las derivadas parciales que estarán retro propagando el error obtenido de la efectividad del modelo en cada paso, por lo que el número será pequeño e irá variando en la práctica con el fin de obtener una mejor efectividad en el entrenamiento y validación.

Dentro del preprocesamiento de imágenes se crearán imágenes que complementen la base de datos existente (Data augmentation (Shorten, 2019)), con el fin de obtener un modelo con mayor precisión en el entrenamiento y que pueda tener un mayor número de imágenes con las cuales se genere un aprendizaje optimizado, así mismo se darán las especificaciones para cada uno de los lotes de entrenamiento y validación.

A continuación, se crea la arquitectura de la red neuronal convolucional, la cual está compuesta por múltiples capas de convolución y de max-pooling siguiendo el orden mostrado en la siguiente tabla:



**Tabla 1.** Arquitectura utilizada para la creación del modelo predictivo, utilizando de base la arquitectura vgg16. (imagen propia)

Filtro	Tamaño
Convolución 1-1	32
Convolución 1-2	32
Max-Pooling	"
Convolución 2-1	64
Convolución 2-2	64
Max-Pooling	"
Convolución 3-1	128
Convolución 3-2	128
Convolución 3-3	128
Max-Pooling	"
Convolución 4-1	256
Convolución 4-2	256
Convolución 4-3	256
Max-Pooling	"
Convolución 5-1	512
Convolución 5-2	512
Convolución 5-3	512
Max-Pooling	"
Dropout 35%	"
Flatten	1024

Dentro de la arquitectura del modelo predictivo, visualizado en la tabla 1, se especifica el *dropout* para desactivar cierto porcentaje del total de las redes neuronales artificiales, ya que, al no incluirla, puede haber un sobre aprendizaje, y es un problema donde el modelo de clasificación solamente aprende sobre la base de datos con la que se entrenó, pero no para datos nunca vistos por el modelo.

Finalmente, se añade el optimizador Adam (Kingma, 2017) para obtener mejoras en el entrenamiento de clasificación.

Posterior al entrenamiento realizado, se almacena el modelo y los pesos obtenidos por el proceso. Dicho archivo puede ser llamado posteriormente con el objetivo de realizar una nueva predicción o continuar el entrenamiento en caso de requerirse.

En el caso de la clasificación por etapas, el único cambio a realizarse es de la función de activación, de una sigmoideal a un **softmax** (Nwankpa, 2018), con el fin de obtener resultados mediante probabilidades sobre cada etiqueta existente. La función de activación softmax es usada cuando el modelo de clasificación posee más de 2 etiquetas, en el caso de las etapas de la retinopatía diabética suman un total de 5 etiquetas.

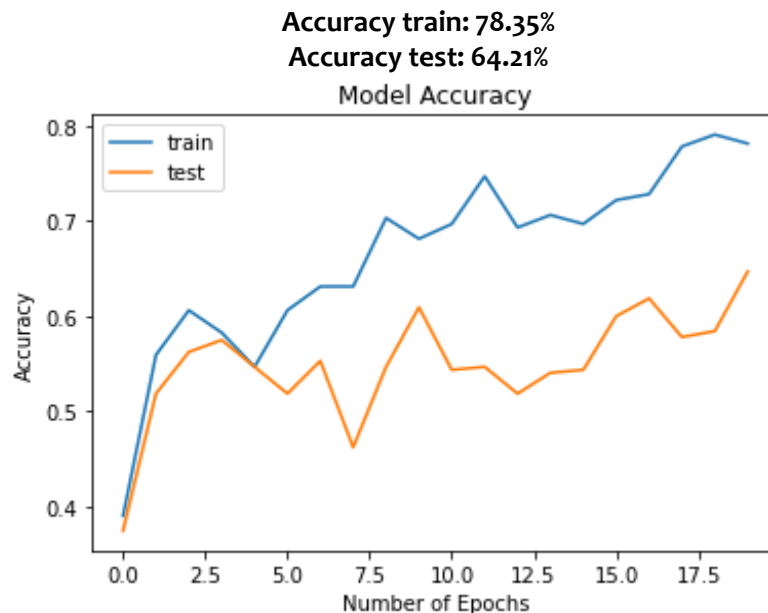


## 5. RESULTADOS

### Clasificador binario

Para el clasificador binario se obtuvieron los siguientes resultados en el entrenamiento de la red neuronal convolucional para detectar si es o no, retinopatía diabética a partir de imágenes de fondo de ojo recopiladas de una base de datos pública de la plataforma Kaggle.

El valor de la efectividad del modelo de clasificación tanto con los datos de entrenamiento (train) y/o los datos de validación (test), fueron un total de 20 épocas y 35 pasos entre cada una ellas para dar los siguientes resultados, Fig. 10:



**Figura 10.** Efectividad del modelo de la clasificación binaria (entrenamiento y validación de 20 epochs).

Para los valores del error, el cual es el cálculo de la predicción menos el error al cuadrado el cual tiene un decremento conforme va pasando las épocas, estos valores llegan a ser, Fig. 11:

**Loss train: 0.7856**  
**Loss test: 0.8027**



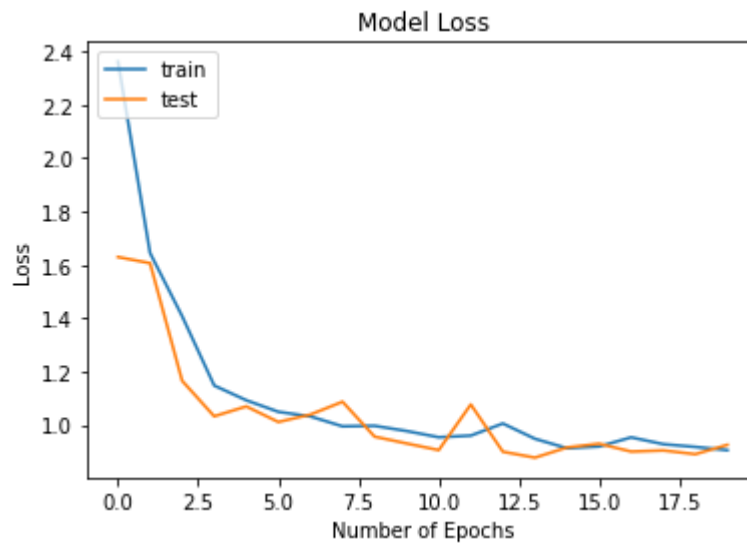


Figura 11. Cambio en el error del modelo de la clasificación binaria.

La predicción binaria era el primer resultado metodológico, en el cual se observaron buenos resultados, ya que se logró una efectividad superior al 64%, lo que equivale a tener aciertos en el 64% de las imágenes nunca vistas.

### Resultados del clasificador por etapas

La predicción por etapas de la retinopatía diabética es un trabajo de mayor grado de dificultad para el modelo de Deep learning, ya que contiene un mayor número de imágenes con características muy similares y pequeñas diferencias las cuales, para una máquina, podrían resultar de mayor dificultades encontrar diferencias clave.

Los resultados en la efectividad de validación o test son altos pese a las pequeñas diferencias encontradas entre cada una de las etapas, pues se encuentran varias similitudes al tratarse de un mismo objeto, en este caso de la retina en imágenes de fondo de ojo, Fig. 12.

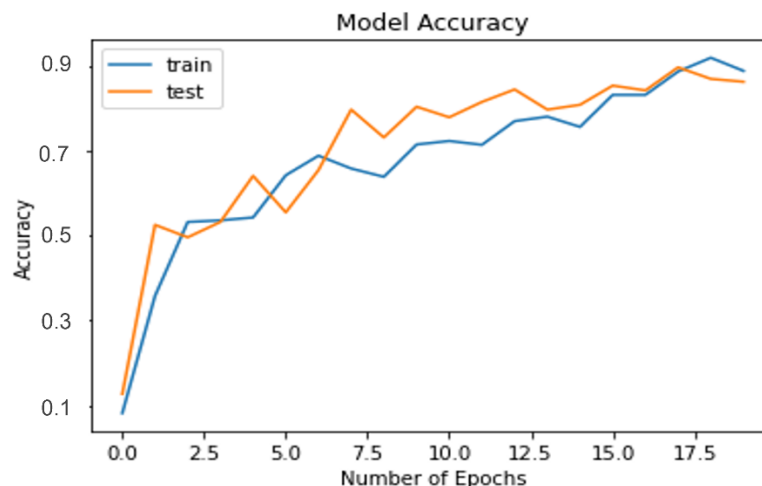


Figura 12. Efectividad del modelo para predecir la retinopatía diabética por etapas (20 epochs).

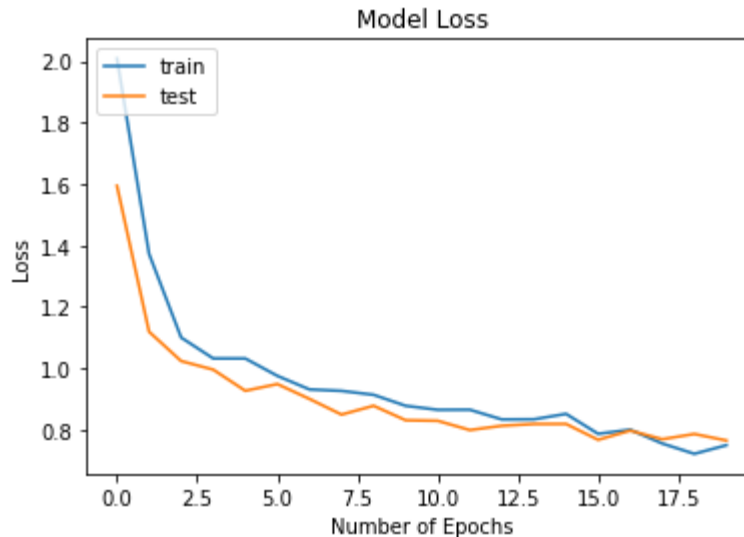
Valores finales de la efectividad del modelo predictivo por etapas de la retinopatía diabética.



**Accuracy train: 97.62%**

**Accuracy test: 86.47%**

El valor del error es bajo en ambos casos, por lo que se considera que los resultados son buenos para la clasificación por la red neuronal convolucional para identificar con buena efectividad la retinopatía diabética, Fig. 13.



**Figura 13.** Cambio del error en el modelo para predecir la retinopatía diabética por etapas.

Valores finales del error del modelo predictivo por etapas de la retinopatía diabética:

**Loss train: 0.7493**

**Loss test: 0.7645**

### Resultados de la interfaz gráfica de usuario para la clasificación de imágenes de retinopatía diabética

Con la finalidad de tener el control y almacenamiento de resultados se generó una interfaz gráfica de usuario que pueda ser implementada en un sistema físico y facilite su control para cualquier usuario.

En las primeras dos ventanas de la interfaz gráfica se muestran los campos para nombre de usuario y contraseña antes de hacer uso del software, así mismo da las opciones de registrarse (Fig. 10a). La segunda ventana (Fig. 10b) muestra el menú sobre el portal del software donde se incluye el nombre del usuario que inició sesión.



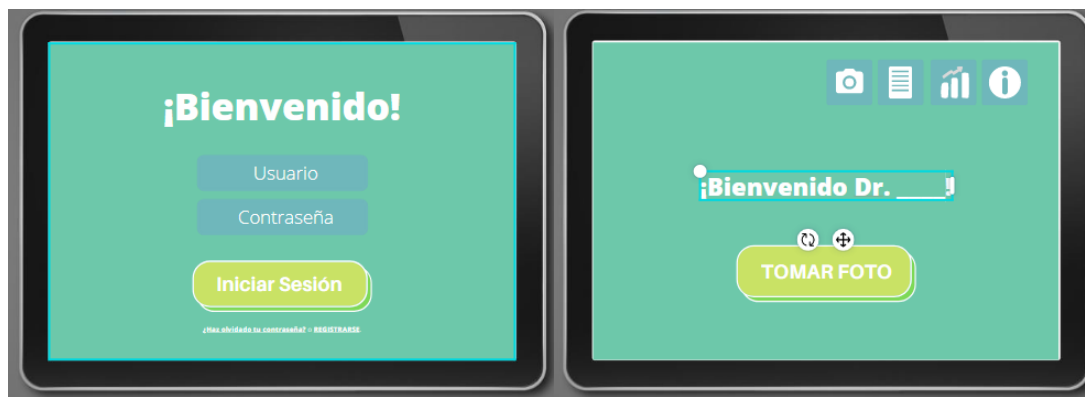


Figura 10. Ventanas de inicio en la interfaz gráfica del optoencefalómetro propuesto.

La tercera ventana (Fig. 11 a) es el resultado de hacer clic en el logo de la cámara, con lo cual preguntará si el paciente al cual se le tomará la fotografía de fondo de ojo es paciente nuevo o ya ha tenido un registro.

En la cuarta ventana (Fig. 11 b) se observa un formulario el cual se llenaría en caso de que el paciente sea nuevo y será indispensable previo a mandar la señal de que se tome una fotografía.



Figura 11. Se visualiza el proceso o cambio de estado, en requerimientos previos para el historial clínico.

En la quinta ventana (Fig. 12a) se muestra una tabla que incluye pacientes que han tenido antes un registro, por lo que será importante tener un seguimiento del avance de la enfermedad del paciente para poder generar más información que pueda ser de utilidad en el futuro. La opción de tomar fotografía (Fig. 12b) genera una previsualización de la cámara una vez que el paciente esté en posición sobre el dispositivo de captura de imagen.





a)

b)

**Figura 12.** a) base de datos de los pacientes que han sido registrados por el dispositivo de captura, b) vista previa de la cámara que se puede ir ajustando hasta tener una imagen similar a la del ejemplo.

En la séptima ventana (Fig. 13) y última parte del software se muestra la foto capturada y posteriormente se dan los resultados en probabilidades, haciendo una predicción sobre en qué etapa se encuentra la enfermedad, los porcentajes servirán para que finalmente el oftalmólogo y/o especialista a cargo decida dar un diagnóstico con mayor precisión sobre la retinopatía diabética.



**Figura 13.** Predicción realizada de acuerdo con la foto capturada, se muestran los porcentajes con los cuales el modelo estima la semejanza del paciente con las etapas de la retinopatía diabética.

## 6. CONCLUSIONES

En el planteamiento del desarrollo de un software que funcione como base para tener la capacidad y el control de un modelo de clasificación con Deep Learning para predecir si un paciente padece de retinopatía diabética o no, y si la padece, predecir en qué fase se encuentra la enfermedad los resultados han sido bastante satisfactorios, de acuerdo al estudio de Google sobre la retinopatía diabética, se encontró que el oftalmólogo promedio tiene una certeza del 65% para determinar la etapa de la enfermedad, con el modelo de clasificación



se logró obtener más de 86%, por lo que es posible poseer una efectividad alta con sólo un software, el cual puede ser replicado.

La interfaz gráfica resulta ser amigable para que una persona (sin conocimientos previos) pueda usarlo con pocos y sencillos pasos.

Respecto a los resultados obtenidos en cuanto al porcentaje de certeza en el entrenamiento y observación de información de prueba se realizará un estudio comparativo con otras arquitecturas de redes neuronales existentes hoy en día.

## REFERENCIAS

- [1] Abràmoff MD, Niemeijer M, Suttorp-Schulten MS, Viergever MA, Russell SR, van Ginneken B. (2008). Evaluation of a system for automatic detection of diabetic retinopathy from color fundus photographs in a large population of patients with diabetes. *Diabetes Care*. Feb;31(2):193-8. doi: 10.2337/dco7-1312. Epub 2007 Nov 16. PMID: 18024852; PMCID: PMC2494619.
- [2] Basaveswara, S. (2019). CNN Architectures, a Deep-dive. Towards data science. Recuperado de: <https://towardsdatascience.com/cnn-architectures-a-deep-dive-a99441d18049>
- [3] Caparrini, F. (2019). Redes Neuronales: una visión superficial. Fernando Sancho Caparrini Recuperado de: <http://www.cs.us.es/~fsancho/?e=72>
- [4] Decencièrre, E., Zhang, X., Cazuguel, G., Laÿ, B., Cochener, B., Trone, C., Gain, P., Ordonez, R., Massin, P., Erginay, A., et al., (2014), "Feedback on a publicly distributed image database: the messidor database," *Image Analysis & Stereology*, vol. 33, no. 3, pp. 231-234.
- [5] Giménez-Palomares, F.; Monsoriu, J.; Alemany, E. (2016). Aplicación de la convolución de matrices al filtrado de imágenes. *Modelling in Science Education and Learning*, [S.l.], v. 9, n. 1, p. 97-108. ISSN 1988-3145. Disponible en: <<https://polipapers.upv.es/index.php/MSEL/article/view/4524>>. Fecha de acceso: 18 jun. 2020 doi: <https://doi.org/10.4995/msel.2016.4524>.
- [6] Google. (2018). Marco: Terminología clave de AA. Aprendizaje automático Google. Recuperado de: <https://developers.google.com/machine-learning/crash-course/framing/ml-terminology?hl=es>
- [7] Hammad, M., UzZaman, A. & Kawnain, S. (2016). Diabetic retinopathy detection using image-processing. Bangladesh: BRAC University.
- [8] Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
- [9] Montes, S. (2018). Estudio de métodos para identificar signos de retinopatía diabética en imágenes de fondo de ojo. México: Centro de Investigación científica y de educación superior de Ensenada, Baja California.
- [10] Molina, F. (2019). Desarrollo de un perceptrón multicapa en Raspberry Pi. Pontificia Universidad Católica de Valparaíso. Recuperado de: [http://opac.pucv.cl/pucv\\_txt/txt-8500/UCC8513\\_01.pdf](http://opac.pucv.cl/pucv_txt/txt-8500/UCC8513_01.pdf)
- [11] Nacelle, A. (2009). Redes neuronales artificiales, Núcleo de Ingeniería Biomédica - Facultades de Medicina e Ingeniería - Universidad de la República, Uruguay.
- [12] Nilsson, N. J. (2014). Principles of artificial intelligence. Stanford.edu. Recuperado de: <https://books.google.es/books?hl=es&lr=&id=mT-jBQAAQBAJ&oi=fnd&pg=PP1&dq=artificial+intelligence&ots=hL1ogK5l9l&sig=-cEk6cJHcligjFq2mMaoxyT9CM#v=onepage&q&f=false>
- [13] Nwankpa C, Ijomah W, Gachagan A, Marshall S., (2018) Activation functions: comparison of trends in practice and research for deep learning. Preprint arXiv:1811.03378
- [14] Palacios-Burgos, F. J., (2003). La neurona artificial. Herramientas e GNU/Linux para estudiantes universitarios. Recuperado de: [https://www.ibiblio.org/pub/linux/docs/LuCaS/Presentaciones/200304curso-glisa/redes\\_neuronales/curso-glisa-redes\\_neuronales.html/x38.html](https://www.ibiblio.org/pub/linux/docs/LuCaS/Presentaciones/200304curso-glisa/redes_neuronales/curso-glisa-redes_neuronales.html/x38.html)
- [15] Shorten, C., Khoshgoftaar, T.M. A survey on Image Data Augmentation for Deep Learning. *J Big Data* 6, 60 (2019). <https://doi.org/10.1186/s40537-019-0197-0>.
- [16] Silva, S. & Freire, E., (2019). Intro a las redes neuronales convolucionales. Medium. Recuperado de: <https://medium.com/@bootcampai/redes-neuronales-convolucionales-5e0ce960caf8>
- [17] Tanishq M., Ataman, R., (2019). Diabetic Retinopathy (resized), Version 7. Recuperado el 20 de junio de 2020 de: <https://www.kaggle.com/tanlikesmath/diabetic-retinopathy-resized/code>.



- [18] Teng, T., Lefley, M. & Claremont, D., (2002). Progress towards automated diabetic ocular screening: a review of image analysis and intelligent systems for diabetic retinopathy. Ucrania: Academic Biomedical Engineering Research Group, School of Desing, Engineering & Computing, Bournemouth University.
- [19] Usher, D., Dumskyj, M., Himaga, M., Williamson, T., Nussey, S. & Boyce, J. (2003)., Automated detection of diabetic retinopathy in digital retinal images: a tool for diabetic retinopathy screening. Reino Unido: King's College London-Department of Physics.
- [20] Weisstein, Eric W. "Hyperbolic Tangent." From *MathWorld*--A Wolfram Web Resource. <https://mathworld.wolfram.com/HyperbolicTangent.html>

Correo autor: ***ablanco@utleon.edu.mx***