



FACULTAD DE INGENIERIA Y CIENCIAS APLICADAS

RECONOCIMIENTO DE RETINAS USANDO REDES  
ATRACTORAS CONJUNTAS.

AUTOR

PAULO ALEJANDRO SANCHEZ ERAZO

AÑO

2019



FACULTAD DE INGENIERIA Y CIENCIAS APLICADAS

RECONOCIMIENTO DE RETINAS USANDO REDES  
ATRACTORAS CONJUNTAS

Trabajo de Titulación presentado en conformidad con  
los requisitos establecidos para optar por el título de  
Ingeniero en Sistemas de Computación e Informática

Profesor guía

Ing. Mario Salvador González Rodríguez

Autor

Paulo Alejandro Sanchez Erazo

AÑO

2019

## DECLARACIÓN DEL PROFESOR GUÍA

Declaro haber dirigido el trabajo, Reconocimiento de retinas usando redes atractoras conjuntas, a través de reuniones periódicas con el estudiante Paulo Alejandro Sanchez Erazo, en el semestre 201920, orientando sus conocimientos y competencias para un eficiente desarrollo del tema escogido y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación.

---

Mario Salvador González Rodríguez

Doctor en Ingeniería Informática y de Telecomunicación

CC:095837634

## DECLARACIÓN DEL PROFESOR CORRECTOR

Declaro haber revisado este trabajo, Reconocimiento de retinas usando redes atractoras conjuntas, del estudiante Paulo Alejandro Sanchez Erazo, en el semestre 201920, dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación.

---

Bernarda Cecibel Sandoval Romo  
Máster en Ciencias de la Computación  
CC: 1709974453

## DECLARACIÓN DE AUTORÍA DEL ESTUDIANTE

Declaro que este trabajo de titulación es original, de mi autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes.

---

Paulo Alejandro Sanchez Erazo

CC: 1723795348

## **DEDICATORIA**

El presente trabajo de titulación va dedicado a mis padres Paúl Santiago Sánchez Erazo y Mariuxi Judith Erazo Arboleda, quienes han sido mis guías incondicionales y me han apoyado siempre para conseguir la culminación de mi carrera profesional.

## RESUMEN

El objetivo de este documento es explicar el funcionamiento en base a resultados de un algoritmo capaz de aprender patrones binarios creados a partir de imágenes de retinas que han sido procesadas para mostrar sus vasos sanguíneos en forma de arreglo binario, el algoritmo posee las fases de aprendizaje y prueba, para lo cual se ha dividido a los 40 patrones de la base de datos en dos carpetas que sirvan para cada una de las dos fases.

El documento analiza los resultados sobre pruebas que demuestran que utilizar un número mayor de subredes es más eficiente que utilizar una única subred, se mostrarán además estos resultados enfocando el porcentaje de recuperación para varias pruebas con diferente cantidad de subredes; al aumentar la cantidad de subredes, el parámetro de dilución baja, por lo tanto se analizarán también los resultados con un mayor parámetro de dilución, es decir con una mayor cantidad de conexiones entre neuronas para la red.

Las subredes especializadas tienen el objetivo de aprender patrones con un nivel de actividad similar, además de tener las características de una red como tal; la asignación de retinas en subredes especializadas según el nivel de actividad del patrón mostrará que existen cambios en el porcentaje de recuperación de la subred según sea el promedio de actividad, por lo tanto, se mostrarán los resultados para la misma retina (que ha pasado por un proceso de esqueletización o de dilatación), pero agrupadas en subredes especializadas.

El documento mostrará resultados que buscan reconocer un mayor número de retinas, los cuales serán obtenidos a través de la rotación de los patrones iniciales, proceso que se lleva a cabo tanto para retinas de la carpeta de aprendizaje, como para retinas de la carpeta de prueba.

## **ABSTRACT**

The objective of this document is to explain the results of an algorithm capable of learning binary patterns created from images of retinas that have been processed to show their blood vessels in the form of a binary array, the algorithm has the learning and testing phases, and the 40 patterns of the database have been divided into two folders that serve for each of the two phases.

The document analyzes the results of tests that show that using a greater number of subnets is more efficient than using a single subnet, these results will also be shown focusing the recovery percentage for several tests with different subnet amounts; as the number of subnetworks increases, the dilution parameter goes down, therefore the document analyze the results with a higher dilution parameter, that is, with a greater number of connections between neurons for the network.

The assignment of retinas in specialized subnets according to the level of activity of the pattern will show that there are changes in the percentage of recovery of the subnet according to the activity average, therefore, the results for the same retina will be shown (which has gone through a process of skeletonization or dilation), but grouped into specialized sub-networks.

The document will show results that using a greater number of patterns, which will be obtained through the rotation of the initial patterns, a process that is carried out both for retinas of the learning folder, and for retinas of the test folder.



# ÍNDICE

1. INTRODUCCIÓN.....	1
1.1. Justificación del trabajo.....	2
1.2. Objetivos.....	3
1.2.1. Objetivo General del Trabajo .....	3
1.2.2. Objetivos Específicos del Trabajo.....	3
2. MARCO TEÓRICO.....	3
2.1. Aprendizaje.....	3
2.1.1. Aprendizaje supervisado.....	4
2.1.2. Aprendizaje no supervisado.....	5
2.2.3. Reconocimiento de patrones .....	5
2.2. Redes Neuronales .....	5
2.3. Modelo de Red Neuronal Atractora.....	6
3. DESARROLLO .....	9
3.1. Base de datos.....	9
3.2. Reconocimiento de patrones base de datos DRIVE .....	12
3.2.1. Reconocimiento de 20 Patrones .....	12
3.2.2. Reconocimiento de retina utilizando diferentes tipos de procesamiento de imágenes.....	12
4. RESULTADOS .....	16
4.1. Similitud de imágenes de Base de datos: .....	16
4.2. Pruebas usando 20 Patrones.....	19
4.2.1. 1 subred.....	19
4.2.2. 2 subredes .....	21
4.2.3. 4 subredes .....	23
4.2.4. 10 subredes con $k = 200$ .....	25
4.2.5. 10 subredes con $k = 300$ .....	27
4.2.6. 10 subredes con $k = 400$ .....	29
4.3. Prueba de subredes entrenadas con diferentes tipos de procesamiento de imágenes.....	32
4.3.1. 40 patrones con 0 y 90 grados de rotación .....	32
4.3.2. 60 patrones con 0, 90 y 270 grados de rotación .....	33
4.3.3. 40 patrones .....	34
4.3.4. 80 patrones con diferentes tipos de procesamiento de imágenes ...	36
5. CONCLUSIONES Y RECOMENDACIONES .....	39
5.1. Conclusiones .....	39

5.2. Recomendaciones .....	40
REFERENCIAS .....	41
ANEXOS .....	43

## 1. INTRODUCCIÓN

El proceso de escanear una retina utiliza la distribución de los vasos sanguíneos en la retina de una persona como un método de autenticación e identificación, impidiendo la suplantación de identidad debido al patrón característico y único que presenta, el cual puede ser analizado usando diversas técnicas de procesamiento de imágenes para medir los diferentes puntos singulares de los vasos sanguíneos de cada retina.

La retina presenta una estructura de vasos sanguíneos compleja que se considera única, normalmente se mantiene igual desde el nacimiento de una persona, hasta su muerte, sin embargo, factores como la diabetes, el glaucoma o diferentes trastornos degenerativos pueden alterar o modificar a la distribución de los vasos sanguíneos de una retina (Kisku, D., Gupta, P., Sing, J., 2019), estas alteraciones pueden ser analizadas y reconocidas por redes atractoras conjuntas, ya que estos patrones (retinas) tienen una estructura espacial dada por los vasos capilares y existe correlación entre ellos, por esto es interesante apreciar los límites de capacidad del modelo en diversos casos, a partir del procesamiento de patrones especialmente estructurados y con diferentes niveles de correlación, debido a posibles cambios en el tiempo por diversos factores.

La red neuronal de tipo atractor es una red que usa un tipo de distribución de conectividad métrica denominada "*small-world*", en donde existe una mayor cantidad de enlaces locales que de atajos entre las neuronas de la red, siendo capaz de mantener la existencia de registros organizados localmente. Se ha demostrado que un modelo conjunto de aprendizaje usando módulos de redes de tipo atractor mejora la capacidad de almacenamiento de la red (Gonzalez, M., Dominguez, D., Sánchez, A. & Rodriguez, F. B., 2016).

De igual manera, los módulos de la red podrían especializarse en patrones con características similares, dando mayor flexibilidad al modelo. En este trabajo se construirá una aplicación útil para el almacenamiento y recuperación de retinas

en base a la segmentación de los vasos sanguíneos; con módulos de redes de tipo atractor se va a mejorar la capacidad de almacenamiento a partir de subredes en conjunto que además tienen por objetivo optimizar puntualmente el rendimiento y la forma en que se distinguen las retinas, con lo que se plantea un modelo flexible y eficiente en el reconocimiento de retinas (Gonzalez, M., Dominguez, D., Rodriguez, F. B., & Sanchez, A., 2017).

Para la representación de la red y comprender cual es la mejor actividad para realizar la recuperación, se especializará a los tipos de retina en:

- Retina binaria obtenida de la base de datos
- Retina dilatada o engrosada en primer nivel
- Retina dilatada o engrosada en segundo nivel
- Retina esqueletada

El algoritmo usado para realizar las pruebas de aprendizaje y reconocimiento de patrones a partir de redes atractoras conjuntas será desarrollado en el lenguaje de programación "Python", el cual tiene una sintaxis legible, con una comunidad activa que aporta con una gran cantidad de bibliotecas y además de documentación sobre bibliotecas estándar; además de poseer un modo interactivo que permite analizar expresiones una a una, evaluando resultados de forma inmediata (Python, 2019).

### **1.1. Justificación del trabajo**

Se pretende usar un modelo conjunto de redes neuronales de tipo atractor, con lo cual se mejora el concepto original planteado por John Hopfield en 1982, el cual explica una red con neuronas completamente conectadas entre sí, lo que se muestra como una solución útil para la autenticación e identificación de retinas, pero muy costosa en términos de recursos computacionales, ya que una neurona tendrá una cantidad de conexiones igual a la cantidad de neuronas presentes en la red menos uno (Hopfield, 1982).

Para mejorar el modelo de John Hopfield, se ha planteado utilizar un conjunto de subredes neuronales de tipo atractor, cada una agrupando un tipo de retina, con una menor cantidad de conexiones entre las neuronas, ya que la cantidad de conexiones se divide para la cantidad de subredes que se desee implementar, reduciendo así el costo en el procesamiento de la red y mejorando el rendimiento del modelo para distinguir y reconocer retinas de manera más eficiente (Gonzalez, M., Dominguez, D., Sánchez, A. & Rodriguez, F. B., 2016).

## **1.2. Objetivos**

### **1.2.1. Objetivo General del Trabajo**

Implementar un modelo de red neuronal atractor conjunta para el reconocimiento de retinas según sus rasgos y características.

### **1.2.2. Objetivos Específicos del Trabajo**

- a) Realizar el procesamiento de imágenes de retinas de la base de datos DRIVE.
- b) Implementar el modelo de red neuronal atractor conjunta para el reconocimiento de imágenes de retinas.
- c) Realizar pruebas para identificar con qué tipo de procesamiento de imágenes la red obtiene el mejor promedio de recuperación de acuerdo a la actividad del patrón.

## **2. MARCO TEÓRICO**

### **2.1. Aprendizaje**

El aprendizaje en el campo computacional, específicamente en la inteligencia artificial, es definido como el proceso por el cual una computadora puede transformar datos del entorno en algún tipo de conocimiento; dentro de las

opciones de conocimiento que puede tener una inteligencia artificial se tiene al crecimiento, que es almacenar información en memoria, a la reestructuración que es generar conocimiento a partir de la interpretación de otro conocimiento y al ajuste, que es la generalización de conocimiento, en otras palabras, se puede decir que el aprendizaje en el campo de la inteligencia artificial es inferir conocimiento a partir del análisis de un conjunto de datos.

### 2.1.1. Aprendizaje supervisado

Este tipo de aprendizaje se caracteriza por tener un agente externo capaz de controlar el proceso de entrenamiento, ya que establece el output del sistema, es decir, la respuesta que debería ofrecer la red neuronal a partir de una determinada entrada, determinando así, las diferencias existentes entre la respuesta deseada y la obtenida.

En el aprendizaje supervisado, el proceso puede incluir un ajuste iterativo de los pesos para adaptar la salida obtenida a la salida deseada, este proceso se puede distinguir en tres tipos:

- Aprendizaje por corrección de error: aprendizaje supervisado más utilizado y la función del supervisor es ajustar los pesos de las conexiones de la red a partir de la diferencia entre valores obtenidos y los valores deseados.
- Aprendizaje por refuerzo: la función del supervisor es emitir señales de refuerzo si la salida obtenida se ajusta a la salida deseada (éxito = +1) y castigar en caso de que la salida no sea lo que se esperaba (fracaso = -1).
- Aprendizaje estocástico: la función del supervisor es añadir cambios aleatorios en los pesos de la red y evaluar el efecto a partir de los resultados, si el cambio resulta positivo, se acepta la modificación, en caso contrario, la modificación es rechazada.

### 2.1.2. Aprendizaje no supervisado

Este tipo de aprendizaje no requiere un agente externo para ajustar los pesos de las conexiones de la red, es decir, no adjunta la respuesta deseada; en este tipo de aprendizaje, se requiere de un cierto nivel de redundancia en los patrones que serán aprendidos por el sistema.

Existen diferentes formas de interpretación de resultados, debido a que no se sabe exactamente qué es lo que se debe obtener, existen casos en los que es necesario obtener una salida similar al patrón de entrada, y también otros en los que el resultado es una categorización de los patrones según la información mostrada (Flórez, R., Fernández, J., 2008).

### 2.2.3. Reconocimiento de patrones

El reconocimiento de patrones es definido como la identificación de características únicas que distinguen a un sujeto y la clasificación de patrones según características similares, y es la técnica en la cual un patrón es representado por un valor numérico con una determinada dimensión que presenta características propias; esta técnica funciona tanto en el proceso de entrenamiento como en el proceso de prueba y tiene diversas aplicaciones como reconocimiento de voz, de rostros, de huellas dactilares, de retinas, etc. (Vega, H., Cortez, A., Huayna, A., Alarcón, L., Romero, P., 2009)

## 2.2. Redes Neuronales

Las redes neuronales inspiran su nombre de la biología en la cual existe un cerebro compuesto por una cantidad  $n$  de neuronas que por sí solas pueden realizar ciertas funciones, pero al estar organizadas en redes se ve incrementada su productividad en gran escala; el aprendizaje desde el punto de vista psicológico es un proceso en el cual existen cambios que activan neuronas en el cerebro humano, concepto que es adaptado para las redes neuronales basando la idea en que un entrenamiento o experiencia inicial crearía una

organización propia ajustando las conexiones y pesos de neuronas. (Matich, 2001).

Las redes neuronales se caracterizan por aprender de la experiencia, ya que modifican su comportamiento en base a las respuestas obtenidas, generalizan por lo que se vuelven hasta cierto punto insensibles a variaciones en las entradas y abstraen la esencia de las entradas por lo cual se vuelve una técnica de inteligencia artificial idónea para el reconocimiento de patrones.

Una red neuronal debe pasar por dos fases para validar que el aprendizaje ha sido exitoso, la primera fase ayudará a determinar los pesos de una red neuronal y para esto se debe pasar por un proceso de entrenamiento, en el cual se alimenta a la red con un conjunto de patrones o un conjunto de datos; y la segunda fase que será de prueba que determinará si los pesos obtenidos mediante la fase anterior son correctos y tanto el valor de salida como el valor obtenido son lo esperado (Cárdenes, 2006).

### **2.3. Modelo de Red Neuronal Atractora**

El modelo de red neuronal atractora, en inglés Attractor Neural Network (ANN) está compuesto por un conjunto con cantidad  $N$  neuronas, cada una con un estado definido en cualquier momento discreto  $t$  por un conjunto  $N$  de variables binarias  $\tau^t = \{\tau_i^t \in 0,1; i = 1, \dots, N\}$ , donde 1 y 0 representan respectivamente estados activo e inactivo.

El objetivo de la red es recuperar un conjunto de patrones que representan imágenes binarias 2D que satisfacen  $n^\mu = \{n_i^\mu \in 0,1; i = 1, \dots, N\}$ , los cuales han sido almacenadas a partir de un proceso de aprendizaje y funcionan como un punto atractor estable.

Las conexiones sinápticas entre neuronas  $i$  y  $j$  están dadas por la matriz de adyacencia  $J_{ij} \equiv C_{ij}w_{ij}$ , Donde la matriz  $C = \{C_{ij}\}$  describe la estructura de



conectividad de la red neuronal y la matriz  $W = \{W_{ij}\}$  describe los pesos aprendidos. La red aleatoria simétrica tiene un grado de red  $K$  que describe la cantidad de conexiones de cada nodo y se caracteriza por tener un parámetro de dilución  $\gamma = \frac{K}{N}$ , que representa la relación de conectividad de la red neuronal atractora.

La recuperación de un patrón se logra a través de la dinámica neuronal silenciosa:

$$\tau_i^{t+1} = \theta(h_i^t - \theta(a)), \quad i = 1, \dots, N,$$

Con función de paso o activación:

$$\theta(x) = 1, x \geq 0;$$

$$\theta(x) = 0, x < 0,$$

Donde:

$$h_i^t \equiv \frac{1}{K} \sum_j J_{ij} \frac{\tau_j^t - q_j^t}{\sqrt{Q_j^t}}$$

Denota el campo local en la neurona  $i$  y en el tiempo  $t$ ,  $\theta(a)$  es el umbral de disparo, se introduce la actividad media de los vecinos de una neurona  $i$ ,  $q_i^t = \langle \tau^t \rangle_i$ , con su varianza correspondiente  $Q_i^t = Var(\tau^t)_i = \langle (\tau^t)^2 \rangle_i - \langle \tau^t \rangle_i^2$ ; y se define el promedio de vecindad con  $\langle f^t \rangle_i \equiv \sum_j C_{ij} f_j^t / K$ .

Se utilizan variables normalizadas, la dependencia del tiempo y del sitio son implícitas:

$$\sigma \equiv \frac{\tau - q}{\sqrt{Q}}, q \equiv \langle \tau \rangle, Q \equiv Var(\tau) = q(1 - q)$$

$$\varepsilon \equiv \frac{n - a}{\sqrt{A}}, a \equiv \langle n \rangle, A \equiv Var(n) = a(1 - a),$$

Donde  $a$  es el patrón y  $q$  la actividad neuronal, el modelo neuronal binario uniforme se recupera cuando el patrón es igual a 0.5; la dinámica de las neuronas puede ser escrito cómo:

$$\sigma_i^{t+1} = g(h_i^t - \theta(a), q_i^t),$$

$$h_i^t \equiv \frac{1}{K} \sum_j J_{ij} \sigma_j^t, i = 1, \dots, N,$$

Donde la función de ganancia es dada por:

$$g(x, y) \equiv \frac{[\theta(x) - y]}{\sqrt{y(1 - y)}}$$

La matriz de pesos  $W$  es actualizada de acuerdo con la regla de aprendizaje de Hebb.

$$w_{ij} = \sum_{\mu=1}^P \varepsilon_i^{\mu} \varepsilon_j^{\mu} \text{ (Offline)}$$

$$W_{ij}^{\mu+1} = W_{ij}^{\mu} + \varepsilon_i^{\mu} \varepsilon_j^{\mu} \text{ (Online)}$$

La regla de aprendizaje “*offline*” no es iterativa, todos los patrones deben aprenderse antes del comienzo del proceso de recuperación, mientras que la regla de aprendizaje “*online*” es iterativa y el proceso de recuperación se repite para cada nuevo patrón que la red aprende.

La matriz de pesos empieza con  $W_{ij}^0 = 0$  para las reglas de aprendizaje “*online*” y “*offline*” y después de pasos de aprendizaje  $P$ , alcanzan un valor  $w_{ij} = \sum_{\mu}^P \varepsilon_i^{\mu} \varepsilon_j^{\mu}$ . La red aprende  $P = \alpha K$  patrones, donde  $\alpha$  es la relación de carga, que mide la capacidad de recuperación de la red; para mantener una actividad neuronal cercana a la de los patrones aprendidos, es necesario utilizar un umbral  $\theta(a) = \frac{1-2a}{2\sqrt{A}}$ , donde  $a$  corresponde a la actividad media del conjunto de patrones de aprendizaje y  $A = a(1 - a)$  que corresponde a la varianza de la actividad.

La capacidad de recuperación de los módulos de la red puede ser descrita a partir de la superposición:

$$m = \frac{1}{N} \sum_i^N \varepsilon_i \sigma_i$$

Que es la correlación estadística entre el patrón aprendido y el estado neural, con un valor  $m = 1$  si la recuperación del patrón por la red es perfecta,  $m = 0$  si no se logra recuperación del patrón por parte de la red y para los valores

intermedios se recupera un patrón con diferentes niveles de ruido; por lo que el valor de  $m$  es una medida de calidad de recuperación del patrón por parte de la red.

El estado inicial de la red  $\tau^{t=0}$  y su respectiva transformación  $\sigma_i^{t=0}$  es un estado con ruido cargado desde un conjunto de patrones de prueba con  $m^{t=0} < 1$ , los patrones usados en la fase de recuperación representan los mismos patrones del conjunto de aprendizaje, pero con ruido o diferencia (Erdős, P., & Rényi, A, 1959).

### 3. DESARROLLO

#### 3.1. Base de datos

Para el Desarrollo del trabajo de titulación, se escogió la base de datos DRIVE (*“Digital Retinal Images for Vessel Extraction”*), la cual ha sido establecida para permitir estudios en base a la segmentación de los vasos sanguíneos sobre las diferentes retinas que conforman esta base de datos.

Las fotografías que se encuentran en la base de datos DRIVE fueron obtenidas a través de un programa con objetivo de detección de retinopatía diabética en los Países Bajos; las fotografías se las realizaron a 400 personas diabéticas de entre 25 y 90 años, de las cuales se seleccionaron 40 fotografías al azar con un resultado de 33 sin signos de retinopatía diabética y 7 con signos de retinopatía diabética leve (Ginneken, 2002).

Las imágenes se adquirieron usando una cámara Canon CR5 con un campo de visión de 45 grados (FOV), y cada imagen usa 8 bits por plano de color a 768 x 584 pixeles, el campo de visión de cada imagen es circular con un diámetro de 540 pixeles (Ginneken, 2002).

Las 40 imágenes se dividieron en dos subconjuntos de 20 imágenes cada una

designados para entrenamiento y prueba; sin embargo, el propósito inicial de las imágenes de la base de datos era realizar diagnósticos clínicos.

La base de datos ha sido utilizada en diversos estudios en diferentes áreas, una de las aplicaciones ha sido el artículo "*Biometric retina identification based on neural network*" en donde se concluye que el reconocimiento de retinas es uno de los métodos más seguros de identificación, este reconocimiento se realizó a partir de un algoritmo desarrollado en MATLAB que implementa etapas de preprocesamiento para transformar las imágenes de las retinas en escala de grises y clasificación para el reconocimiento de los patrones por medio de redes neuronales, el resultado del algoritmo da una precisión igual a 97.5% en el reconocimiento de las retinas de la base de datos DRIVE (Sadikoglu, F., Uzelaltinbulat, S., 2016) .

Las imágenes servirán para alimentar la red neuronal, las imágenes de las retinas serán utilizadas de dos maneras distintas, en tamaño original para pruebas con una mayor importancia e imágenes de tamaño reducido para disminuir el tiempo en que demora la ejecución de las pruebas, para estas imágenes se ha seguido un proceso de adaptación, el cual conlleva los siguientes pasos:

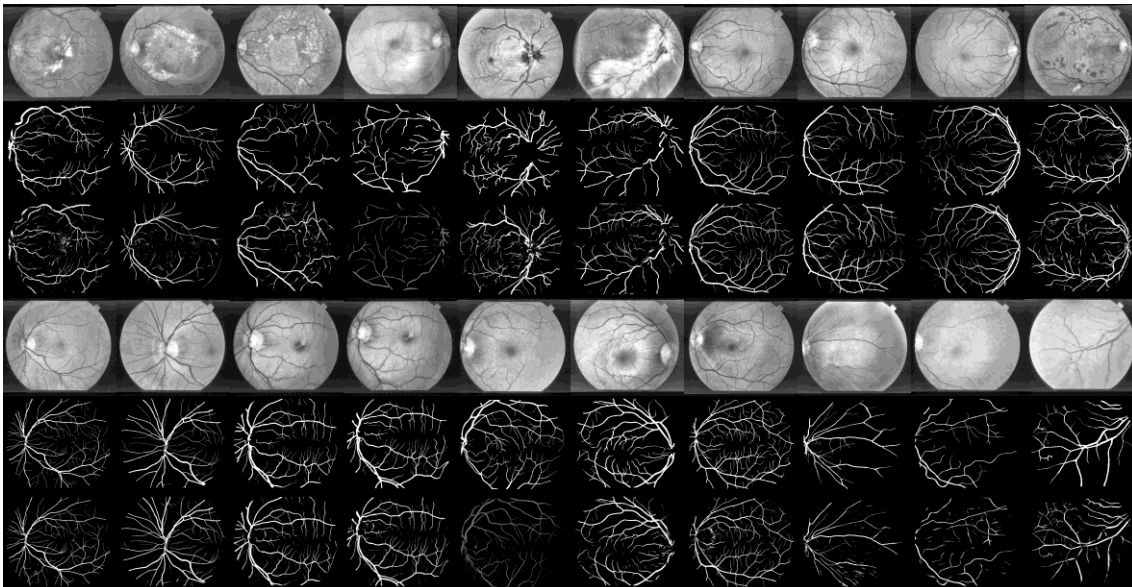
- Recortar la imagen: El proceso consiste en pasar desde la resolución original de 565 x 584 a 141 x 146 manteniendo la mayor cantidad de información de la retina, y cambiando el tamaño de la imagen de 329960 a 20586 pixeles, el proceso es llevado a cabo a partir del módulo Image de la librería gratuita Pillow y de la librería resizeimage.
- Binarización de imágenes: Al recortar la imagen binaria, se producen muchos pixeles que dejan de ser binarios, por lo que se realiza un proceso que incluye escoger un umbral de binarización, el umbral escogido es de 75, con lo cual, si un píxel es mayor a 75 se vuelve 1, caso contrario es 0.

Además, para realizar pruebas sobre el alcance de la red, se ha realizado un proceso que gira las imágenes en 90 grados y 270 grados con el objetivo de

obtener más patrones, para realizar este proceso se ha utilizado el módulo Image de la librería Pillow y la librería numpy.

Para una mejor visualización de las diferencias que tienen las imágenes de entrenamiento y prueba y recuperadas, se han creado nuevas imágenes a partir de las primeras con las diferencias que existen entre estas imágenes, para realizar este proceso se ha utilizado la librería Pillow, en donde se interpolan las imágenes de entrada y salida y se usa una máscara de transparencia para poder combinar las imágenes, el tamaño de las nuevas imágenes es de 565 x 584.

La base de datos incluye 20 fotografías de retinas y 40 imágenes binarias obtenidas a partir de las fotografías, las cuales han sido divididas en entrenamiento y prueba (sin repetir la misma retina en el subconjunto), las imágenes de la base de datos pueden ser observadas en Figura 1 mostrando las fotografías en la primera y cuarta fila, las imágenes binarias del conjunto de aprendizaje en la segunda y quinta fila y las imágenes binarias del conjunto de prueba en la tercera y sexta fila:



*Figura 1.* Base de datos de retinas DRIVE

Tomado de (Ginneken, 2002).

### 3.2. Reconocimiento de patrones base de datos DRIVE

El reconocimiento de patrones de la base de datos DRIVE utilizando redes atractoras conjuntas se da a partir de la ejecución del código en Python, el cuál lee patrones binarios ubicados en archivos de texto, los aprende y prueba con más patrones binarios, el código utilizado se encuentra en el Anexo 1.

Para el reconocimiento de patrones (imágenes binarizadas de retinas) de la base de datos se utilizarán los siguientes parámetros para las diferentes pruebas:

#### 3.2.1. Reconocimiento de 20 Patrones

El objetivo de esta sección es obtener los porcentajes de recuperación para los 20 patrones iniciales de la base de datos DRIVE con 1, 2 4 y 10 subredes.

Para 20 patrones utilizan los siguientes parámetros:

- Tamaño de la red: 329960 neuronas, valor obtenido al multiplicar 565 x 584 que es el tamaño de la imagen.
- Cantidad de conexiones en toda la red: 200 para 1, 2, 4 y 10 subredes y 300 - 400 para 10 subredes, con el objetivo de tener una dilución que oscila entre 20 y 200 según la cantidad de subredes lo que limita el aprendizaje máximo de la red.
- Cantidad de pasos para procesar: 100.
- Probabilidad de reemplazo: 100% que ayuda a que la conexión entre neuronas no sea con sus vecinos directos.

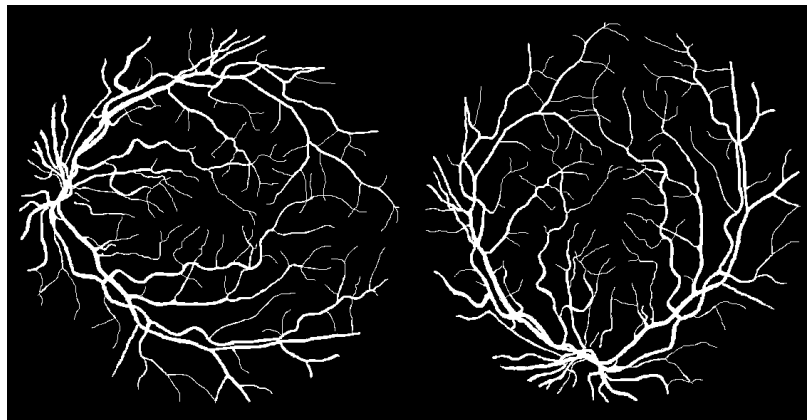
Los resultados obtenidos con los 20 patrones se encuentran en la siguiente sección.

#### 3.2.2. Reconocimiento de retina utilizando diferentes tipos de procesamiento de imágenes

El objetivo de esta sección es mostrar que un patrón con un diferente grado de inclinación es reconocido como un nuevo patrón para la red y que especializar subredes según el procesamiento de dilatación o esqueletización de imágenes

mostrará diferentes porcentajes de recuperación.

Primero, se va a obtener los porcentajes de recuperación para una cantidad mayor de patrones que la que existe en la base de datos DRIVE, para que esto suceda, se ha realizado un proceso de rotación de los 20 patrones iniciales de la carpeta de aprendizaje y de los 20 patrones iniciales de la carpeta de prueba. Para este tipo de prueba se han tomado en cuenta a los patrones de la carpeta de aprendizaje y se ha rotado a esos 20 patrones en 90 grados obteniendo así 20 patrones más que para la red simbolizan nuevos patrones para aprender, cada uno de los patrones poseen una actividad similar, una muestra de esta rotación se puede visualizar en la Figura 2, en la cual está el patrón inicial a la izquierda y el patrón rotado en 90 grados a la derecha:

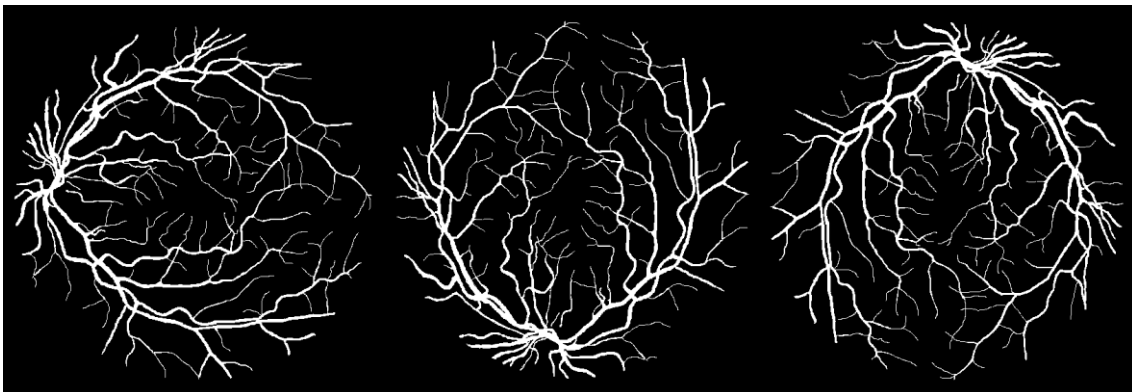


*Figura 2.* Patrón de carpeta de aprendizaje con su forma rotada a 90 grados

Para 40 patrones utilizan los siguientes parámetros:

- Tamaño de la red: 329960 neuronas, valor obtenido al multiplicar 565 x 584 que es el tamaño de la imagen.
- Cantidad de conexiones en toda la red: 600 para obtener una dilución igual a 60, ya que existen 10 subredes.
- Cantidad de pasos para procesar: 100
- Probabilidad de reemplazo: 100% que ayuda a que la conexión entre neuronas no sea con sus vecinos directos.
- Cantidad de subredes: 10

Para este tipo de prueba se han tomado en cuenta a los patrones de la carpeta de aprendizaje y se ha rotado a esos 20 patrones en 90 y 270 grados obteniendo así 40 patrones más que para la red simbolizan nuevos patrones para aprender, cada uno de los patrones poseen una actividad similar, una muestra de esta rotación se puede visualizar en Figura 3, en la cual está el patrón inicial a la izquierda, el patrón rotado en 90 grados en el centro y el patrón rotado en 270 grados a la derecha:



*Figura 3.* Patrón de carpeta de aprendizaje con su forma rotada a 90 y 270 grados

Para 60 patrones utilizan los siguientes parámetros:

- Tamaño de la red: 329960 neuronas
- Cantidad de conexiones en toda la red: 800 para obtener una dilución igual a 80, ya que existen 10 subredes.
- Cantidad de pasos para procesar: 100
- Probabilidad de reemplazo: 100% que ayuda a que la conexión entre neuronas no sea con sus vecinos directos.
- Cantidad de subredes: 10

Segundo, se va a medir el nivel de la recuperación de la red utilizando la base de datos con diferentes tipos de procesamiento de imágenes, se ha realizado el proceso de segmentación de las retinas en:

- Imagen de retina esqueletada (imagen con un grosor menor al de la retina binarizada) que se encuentra en el literal a) de la Figura 4.
- Imagen de retina binarizada (obtenida del proceso de binarizar la imagen



de la retina) que se encuentra en el literal b) de la Figura 4.

- Imagen de retina dilatada o engrosada en primer nivel que se obtiene a partir del módulo Morphology de la librería Skimage para realizar el proceso de dilatación de las imágenes que da como resultado una imagen binaria con una mayor actividad, la cual se encuentra en el literal c) de la Figura 4.
- Imagen de retina dilatada o engrosada en segundo nivel que se obtiene a partir de la repetición del proceso anterior, pero sobre la retina dilatada en primer nivel, la cual se encuentra en el literal d) de la Figura 4.

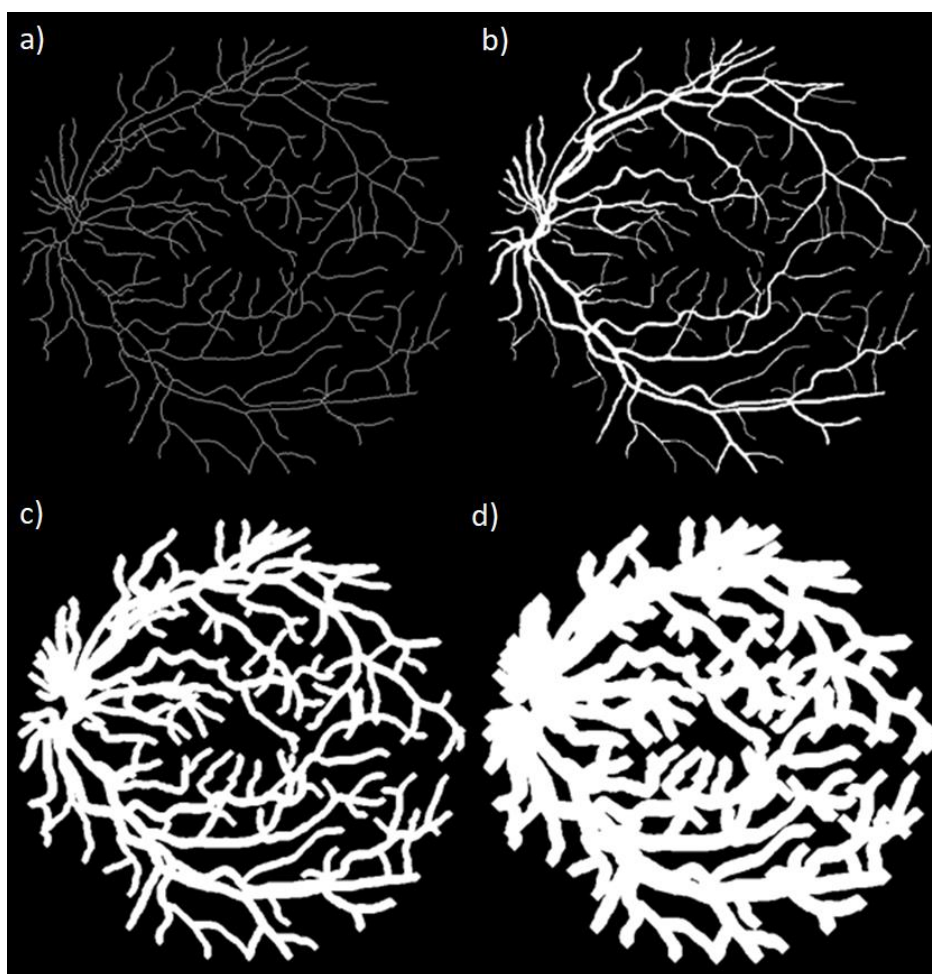


Figura 4. Imágenes de la retina número 1 con diferentes niveles de actividad

El objetivo de esta sección es obtener los porcentajes de recuperación para subredes que tienen patrones con un nivel de actividad similar.

Para 40 patrones con 4 subredes que serán entrenadas cada una con un

diferente tipo de procesamiento de imágenes, se utiliza los siguientes parámetros:

- Tamaño de la red: 329960 neuronas, valor obtenido al multiplicar 565 x 584 que es el tamaño de la imagen.
- Cantidad de conexiones en toda la red: 600 para obtener una dilución igual a 150, ya que existen 4 subredes.
- Cantidad de pasos para procesar: 100
- Probabilidad de reemplazo: 100% que ayuda a que la conexión entre neuronas no sea con sus vecinos directos.

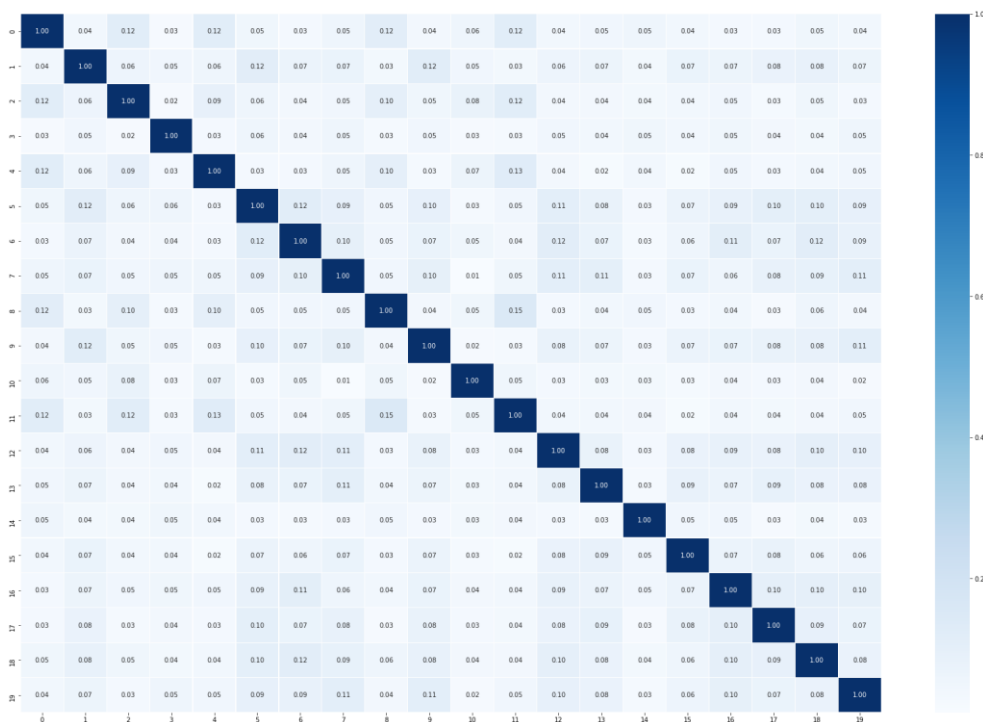
Para 80 patrones con 4 subredes que serán entrenadas cada una con un diferente tipo de procesamiento de imágenes, se utiliza los siguientes parámetros:

- Tamaño de la red: 329960 neuronas, valor obtenido al multiplicar 565 x 584 que es el tamaño de la imagen.
- Cantidad de conexiones en toda la red: 800 para obtener una dilución igual a 200, ya que existen 4 subredes.
- Cantidad de pasos para procesar: 100
- Probabilidad de reemplazo: 100% que ayuda a que la conexión entre neuronas no sea con sus vecinos directos.

## **4. RESULTADOS**

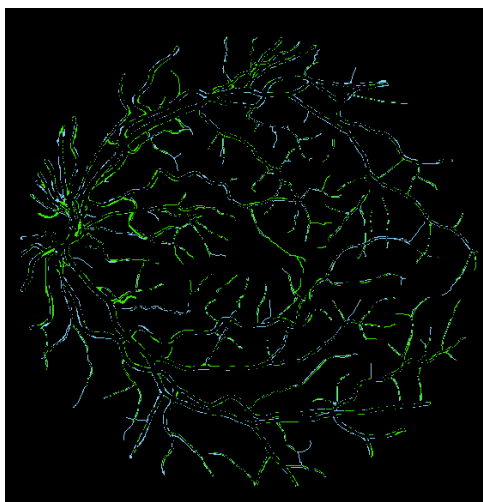
### **4.1. Similitud de imágenes de Base de datos:**

La similitud entre los patrones del conjunto de entrenamiento se puede visualizar en la Figura 5, en la cual se muestran los valores de coincidencia de cada patrón con los demás, los puntos con un color más oscuro explican un mayor nivel de semejanza, mientras que los puntos más claros explican un nivel de semejanza bajo entre los patrones del conjunto de aprendizaje:



*Figura 5.* Matriz con valores de coincidencia entre patrones de la carpeta de aprendizaje

Las imágenes del conjunto de entrenamiento y prueba tienen sus respectivas diferencias, las cuales han sido evidenciadas a partir de la creación de máscaras que muestran los puntos que el patrón de entrenamiento tiene que no tiene el patrón de prueba (color azul) y los puntos que el patrón de prueba tiene que no tiene el patrón de entrenamiento (color verde), un ejemplo de máscara se muestra en la Figura 6:



*Figura 6.* Máscara de diferencias de patrón número 1 de la carpeta de aprendizaje con patrón número 1 de carpeta de prueba

La similitud entre las imágenes del conjunto de entrenamiento y las imágenes del conjunto de prueba se puede apreciar en la Figura 7, en la cual se muestran los valores de coincidencia de cada patrón del conjunto de entrenamiento con los patrones del conjunto de prueba:

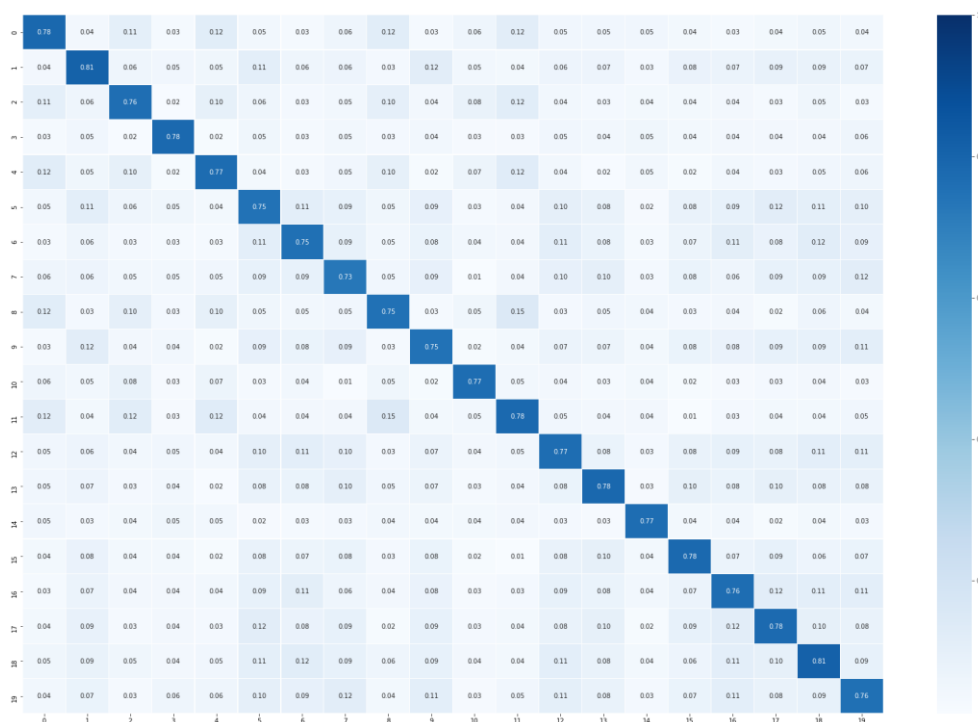
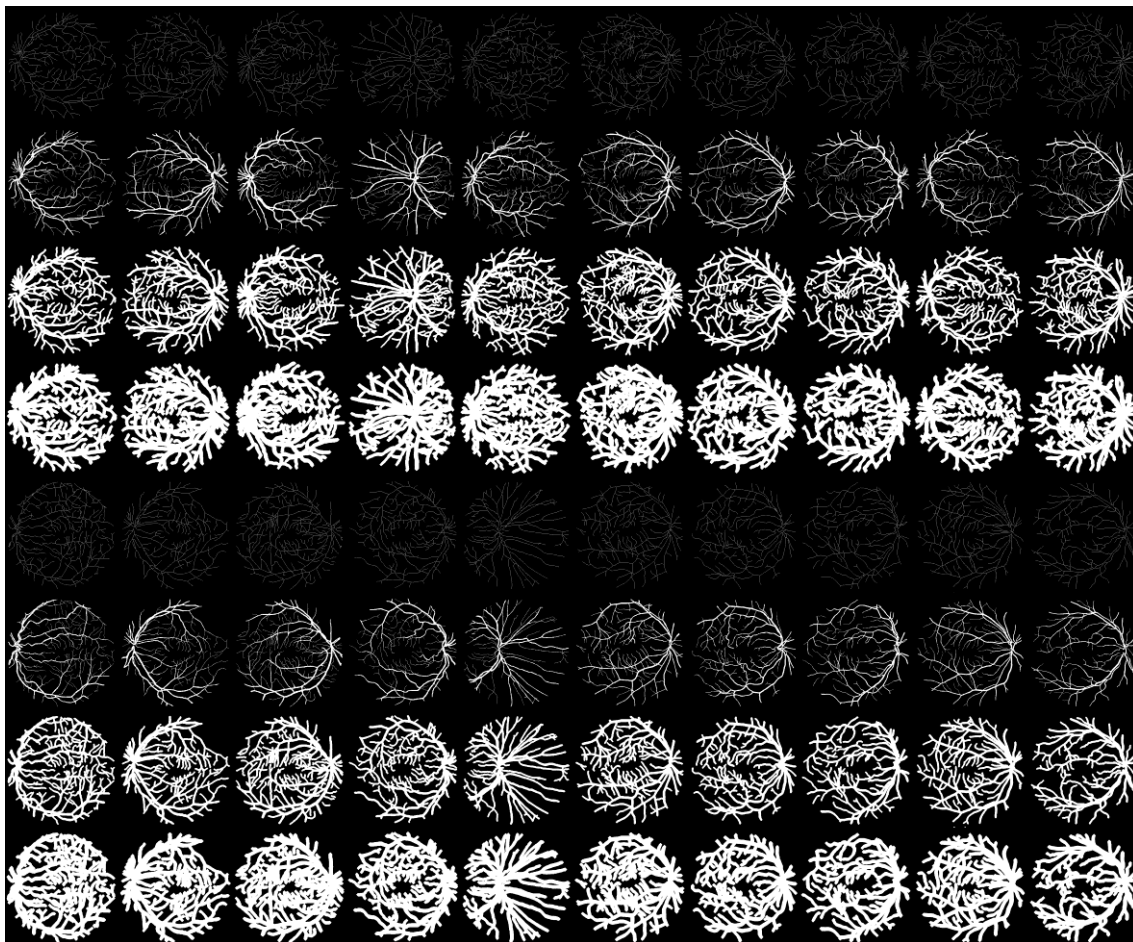


Figura 7. Matriz con valores de coincidencia entre patrones de aprendizaje y prueba

Para el uso de los patrones dentro del algoritmo, ha sido necesario crear un archivo de texto binario por patrón que contiene la misma cantidad de dígitos binarios que de píxeles de la imagen, es decir 329960, valor obtenido a partir de la multiplicación de las dimensiones de la imagen (565 x 584).

Para la segmentación de retinas y posterior prueba de estos patrones con subredes que serán entrenadas cada una con un diferente tipo de procesamiento de imágenes, se ha realizado un proceso en el cual la retina y sus rasgos se dilatan y esqueletizan para obtener distintas actividades de los patrones; los patrones son segmentados en retinas esqueletadas con una actividad promedio (relación de puntos blancos con puntos negros) igual a 0.0277, retinas de patrones obtenidas de la base de datos con una actividad promedio igual a 0.0875, retinas en primer nivel de dilatación con una actividad promedio igual a

0.277 y retinas en segundo nivel de dilatación con una actividad promedio igual a 0.428, las imágenes de los patrones con sus cuatro diferentes actividades pueden ser observadas en la Figura 8:



*Figura 8. Retinas DRIVE con diferente actividad*

## **4.2. Pruebas usando 20 Patrones**

### **4.2.1. 1 subred**

El promedio de recuperación o coincidencia entre los patrones de entrenamiento con los patrones obtenidos para una sola subred es de 0.23, los valores para cada patrón se pueden visualizar en la Figura 9:

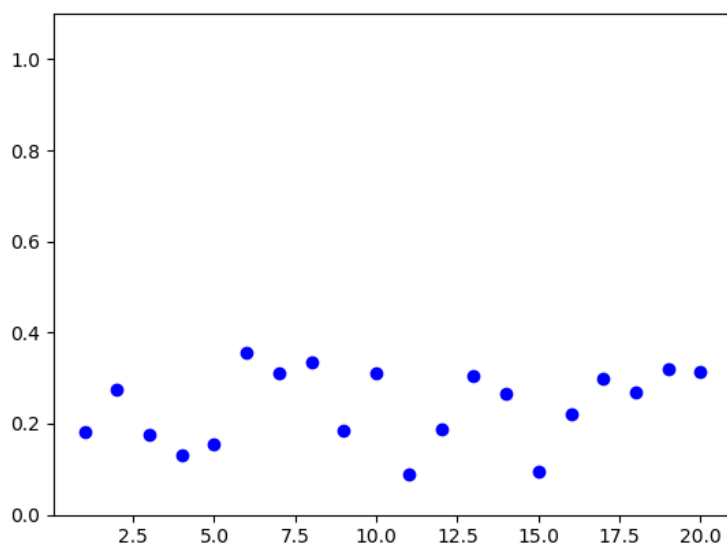


Figura 9. Valores de coincidencia de los patrones con una subred

La similitud entre las imágenes del conjunto de entrenamiento (usadas para el aprendizaje de la red) y las imágenes de los patrones resultado (patrones que devuelve la red después del proceso de prueba) se puede apreciar en la Figura 10, en la cual se muestran los valores de coincidencia de cada patrón del conjunto de entrenamiento con los patrones del conjunto de resultado para la prueba realizada con una subred:

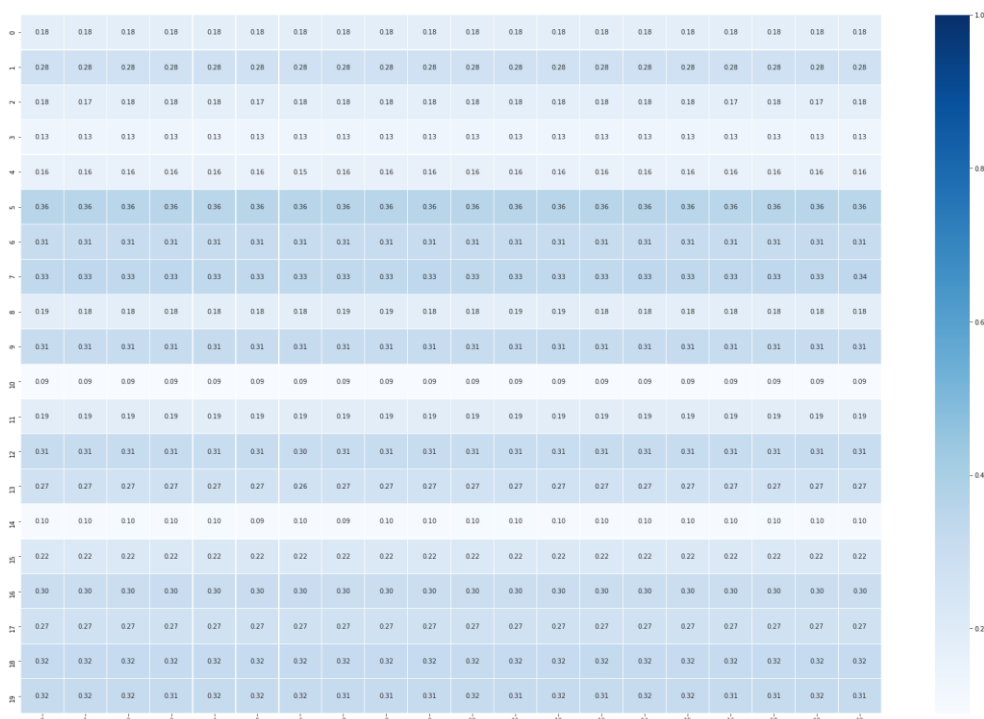


Figura 10. Matriz con valores de coincidencia entre patrones de aprendizaje y resultado con una subred

Se ha tomado como referencia el patrón que más se acerca al promedio de recuperación (patrón número 16 con recuperación = 0.2211) para mostrar la máscara que marca los puntos que el patrón resultado tiene que no tiene el patrón de entrenamiento con color verde y viceversa con color azul, en la Figura 11 se muestra el patrón de aprendizaje en la primera columna de ambas filas de la imagen, el patrón de prueba en la segunda columna de la primera fila, la máscara de diferencias entre patrón de aprendizaje y de prueba en la tercera columna de la primera fila, el patrón resultado en la segunda columna de la segunda fila, y la máscara de diferencias entre patrón de aprendizaje y patrón resultado en la tercera columna de la segunda fila, la cual muestra un nivel muy elevado de ruido explicando que el patrón resultado tiene una gran diferencia con el patrón de la carpeta de aprendizaje:

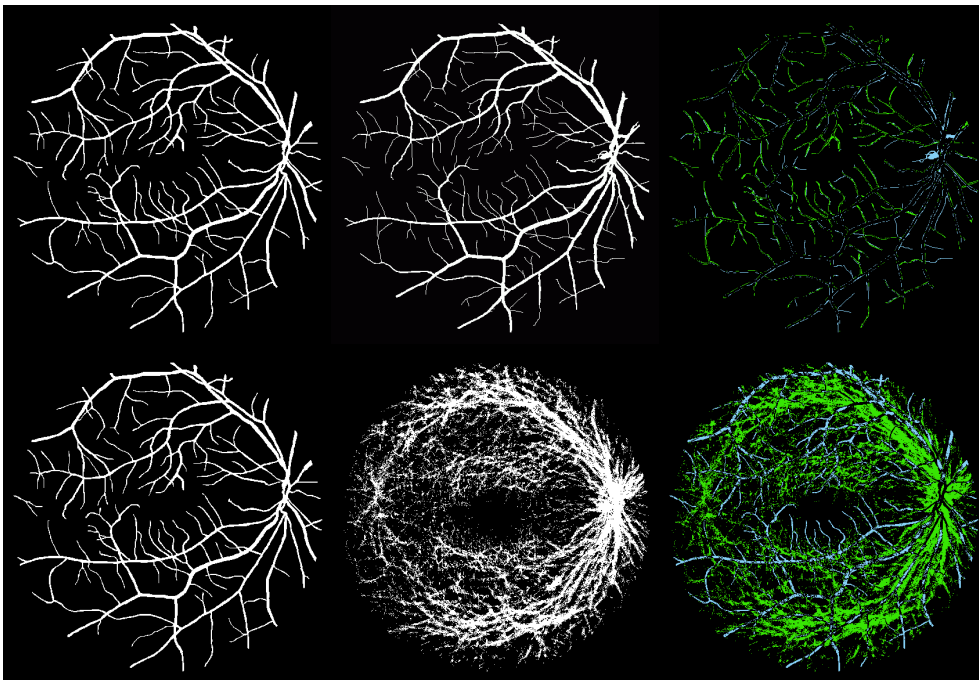


Figura 11. Máscara de patrón obtenida en prueba de 1 subred

#### 4.2.2. 2 subredes

El promedio de recuperación o coincidencia entre los patrones de entrenamiento con los patrones obtenidos para dos subredes es de 0.88, los valores para cada patrón se pueden visualizar en la Figura 12 donde se especifican los patrones con un color distinto para cada subred:

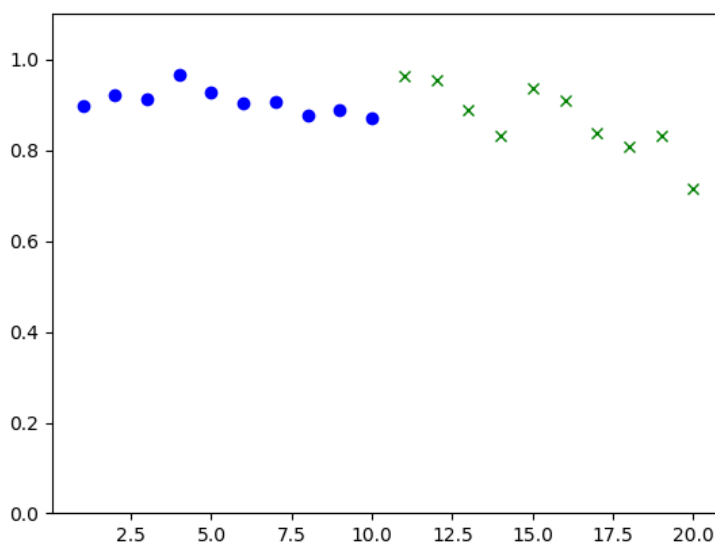


Figura 12. Valores de coincidencia de los patrones con dos subredes

La similitud entre las imágenes del conjunto de entrenamiento y las imágenes de los patrones resultado se puede apreciar en Figura 13, en la cual se muestran los valores de coincidencia de cada patrón del conjunto de entrenamiento con los patrones del conjunto de resultado para la prueba realizada con dos subredes:

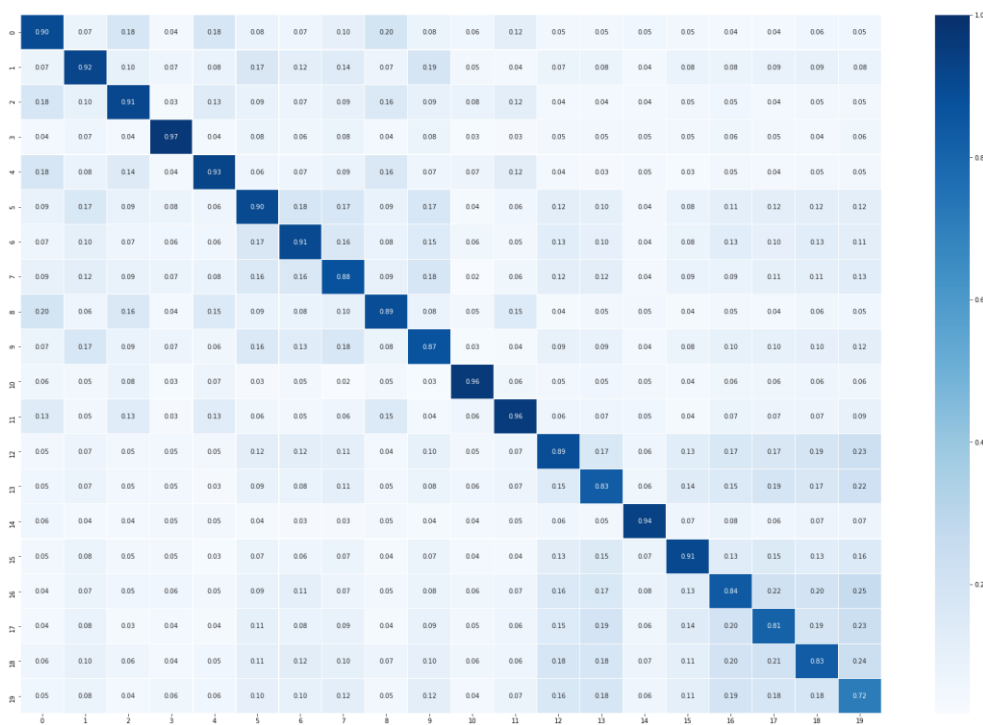
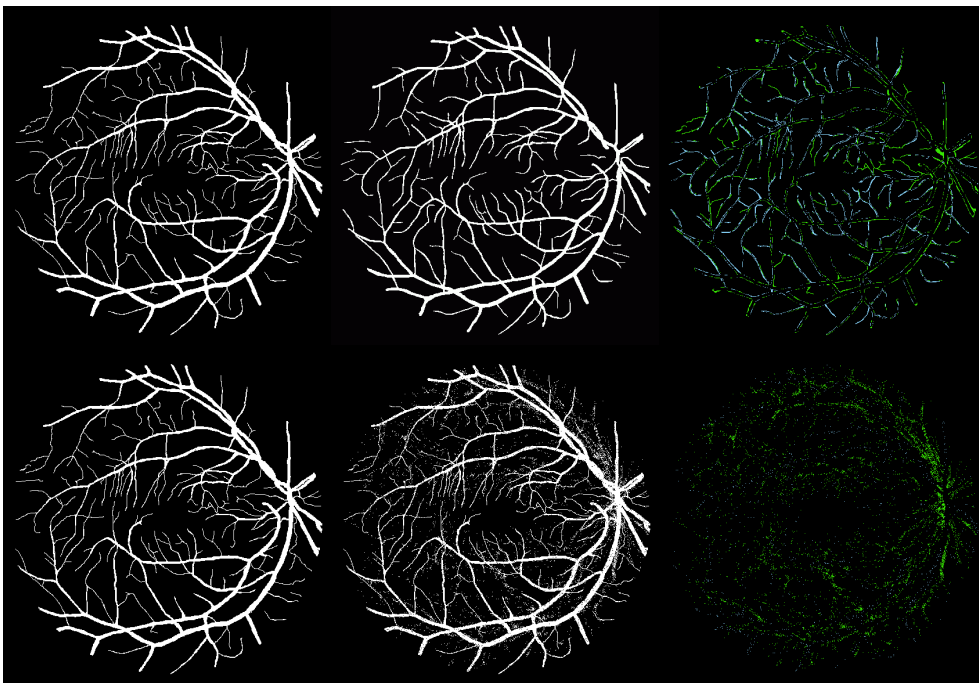


Figura 13. Matriz con valores de coincidencia entre patrones de aprendizaje y resultado con dos subredes



Se ha tomado como referencia el patrón que más se acerca al promedio de recuperación (patrón número 13 con recuperación = 0.88) para mostrar la máscara que marca los puntos que el patrón resultado tiene que no tiene el patrón de entrenamiento con color verde y viceversa con color azul, en la Figura 14 se muestra el patrón de aprendizaje en la primera columna de ambas filas de la imagen, el patrón de prueba en la segunda columna de la primera fila, la máscara de diferencias entre patrón de aprendizaje y de prueba en la tercera columna de la primera fila, el patrón resultado en la segunda columna de la segunda fila, y la máscara de diferencias entre patrón de aprendizaje y patrón resultado en la tercera columna de la segunda fila, la cual muestra un nivel de ruido mucho menor que al que se encuentra en la Figura 9, explicando que el patrón resultado es más parecido al patrón de la carpeta de aprendizaje que en la prueba con una subred:



*Figura 14.* Máscara de patrón obtenida en prueba de 2 subredes

#### 4.2.3. 4 subredes

El promedio de recuperación o coincidencia entre los patrones de entrenamiento con los patrones obtenidos para cuatro subredes es de 0.87, los valores para cada patrón se pueden visualizar en la Figura 15 donde se especifican los patrones con un color distinto para cada subred:

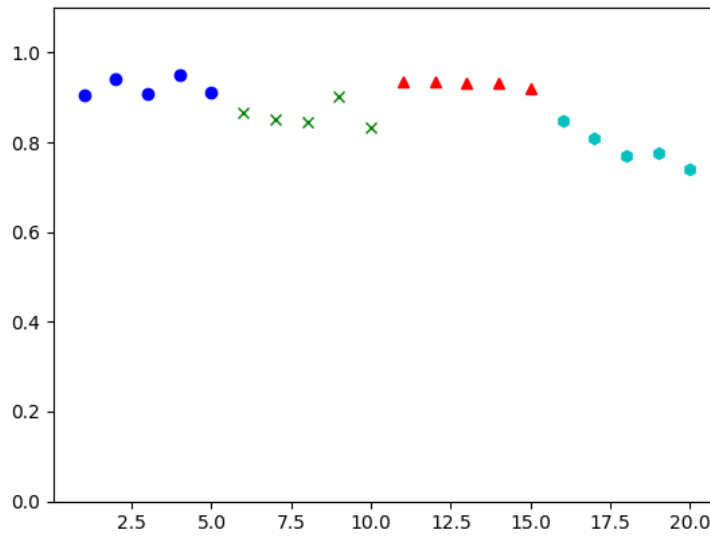


Figura 15. Valores de coincidencia de los patrones con cuatro subredes

La similitud entre las imágenes del conjunto de entrenamiento y las imágenes de los patrones resultado se puede apreciar en la Figura 16, en la cual se muestran los valores de coincidencia de cada patrón del conjunto de entrenamiento con los patrones del conjunto de resultado para la prueba realizada con cuatro subredes:

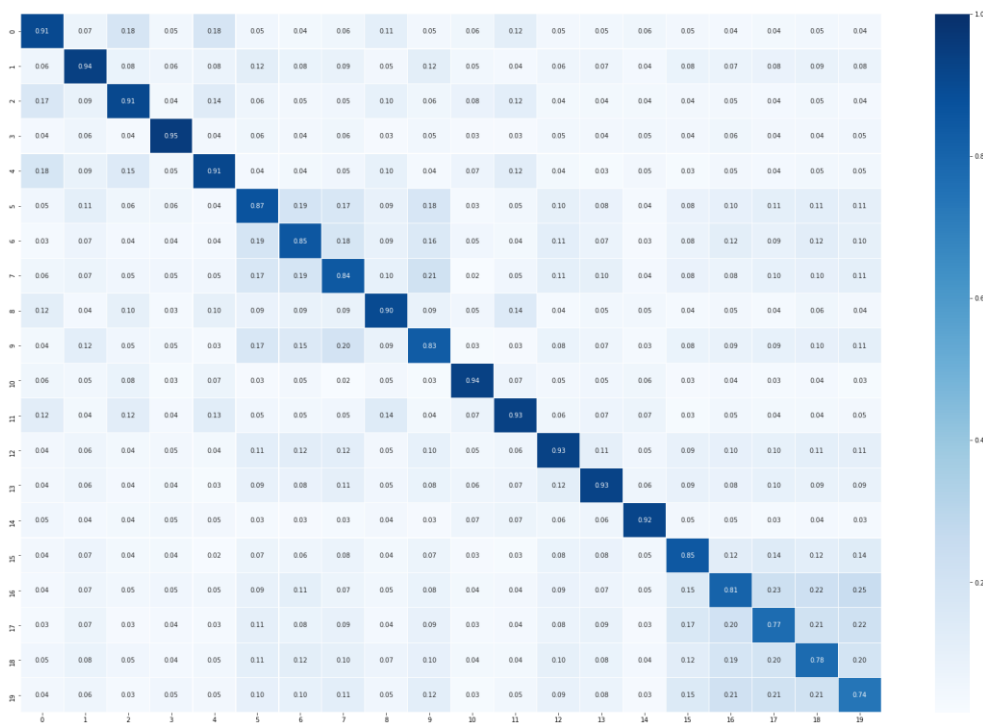
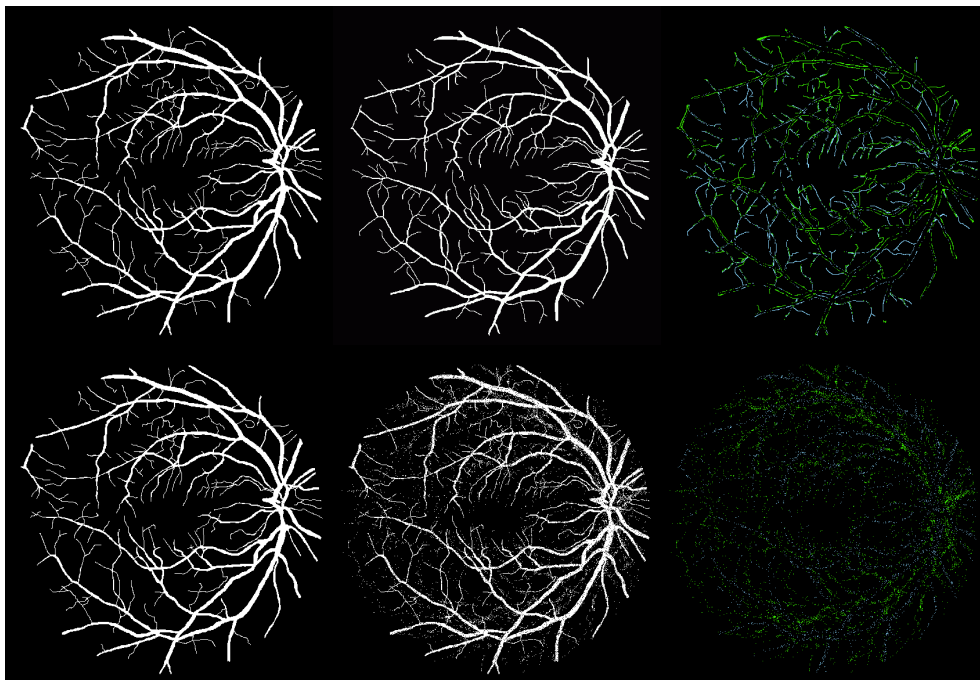


Figura 16. Matriz con valores de coincidencia entre patrones de aprendizaje y resultado con cuatro subredes

Se ha tomado como referencia el patrón que más se acerca al promedio de recuperación (patrón número 6 con recuperación = 0.86) para mostrar la máscara que marca los puntos que el patrón resultado tiene que no tiene el patrón de entrenamiento con color verde y viceversa con color azul, en la Figura 17 se muestra el patrón de aprendizaje en la primera columna de ambas filas de la imagen, el patrón de prueba en la segunda columna de la primera fila, la máscara de diferencias entre patrón de aprendizaje y de prueba en la tercera columna de la primera fila, el patrón resultado en la segunda columna de la segunda fila, y la máscara de diferencias entre patrón de aprendizaje y patrón resultado en la tercera columna de la segunda fila:



*Figura 17.* Máscara de patrón obtenida en prueba de 4 subredes

#### 4.2.4. 10 subredes con $k = 200$

El promedio de recuperación o coincidencia entre los patrones de entrenamiento con los patrones obtenidos para diez subredes con  $k=200$  es de 0.54, los valores para cada patrón se pueden visualizar en la Figura 18 donde se especifican los patrones con un color distinto para cada subred:

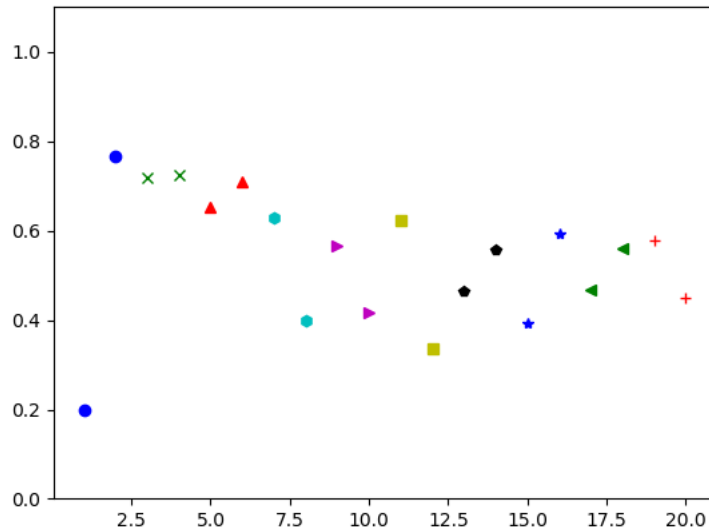


Figura 18. Valores de coincidencia de los patrones con diez subredes y  $k = 200$   
 La similitud entre las imágenes del conjunto de entrenamiento y las imágenes de los patrones resultado se puede apreciar en la Figura 19, en la cual se muestran los valores de coincidencia de cada patrón del conjunto de entrenamiento con los patrones del conjunto de resultado para la prueba realizada con diez subredes y  $k = 200$ :

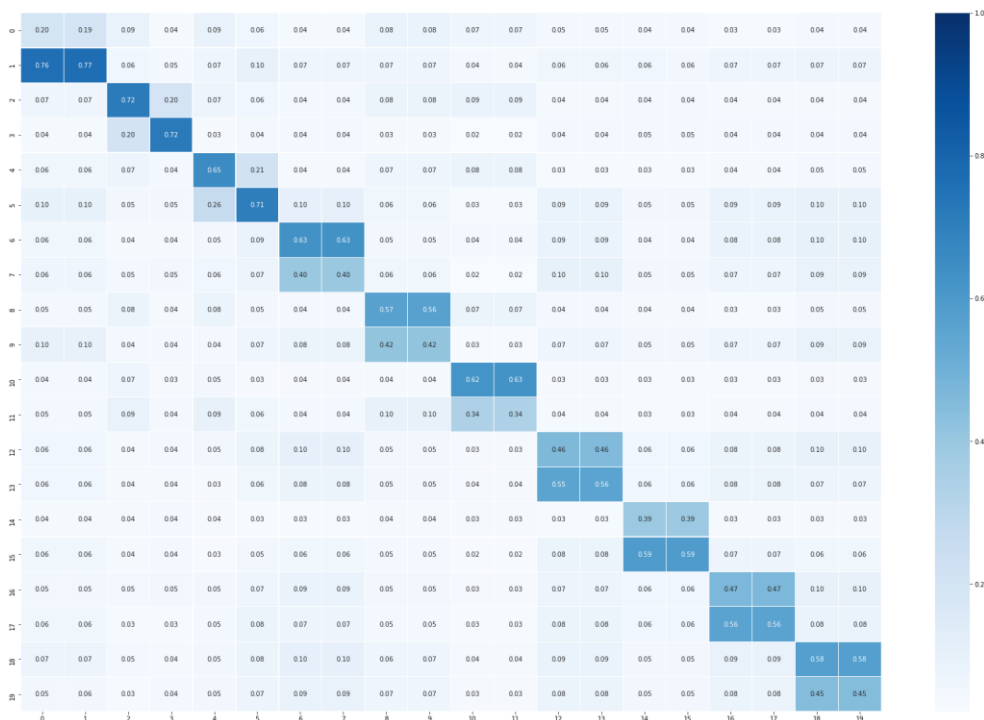
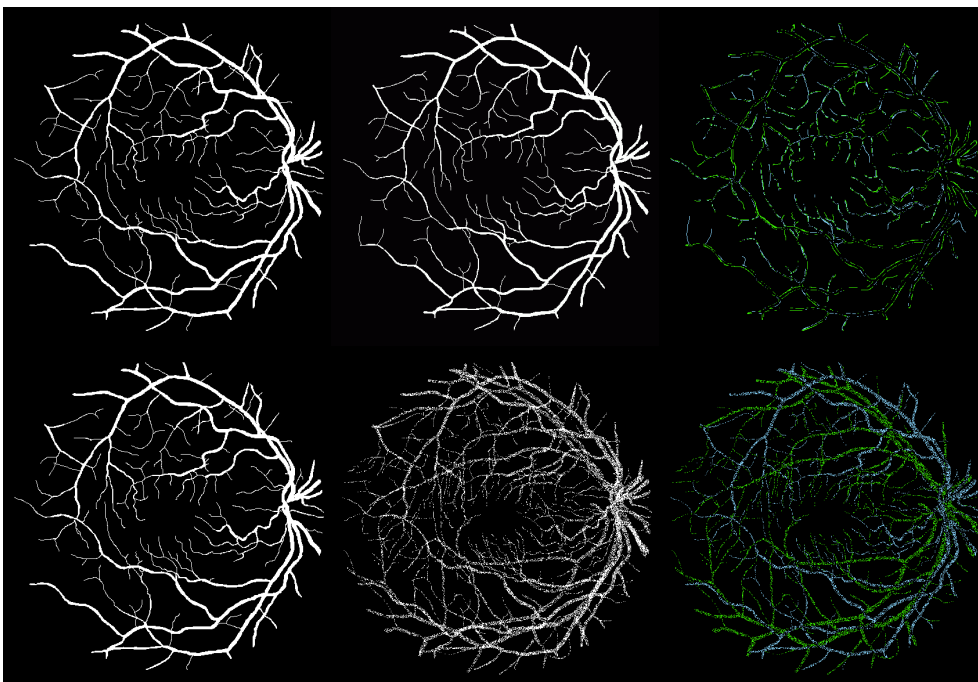


Figura 19. Matriz con valores de coincidencia entre patrones de aprendizaje y resultado con diez subredes y  $k = 200$

Se ha tomado como referencia el patrón que más se acerca al promedio de

recuperación (patrón número 14 con recuperación = 0.55) para mostrar la máscara que marca los puntos que el patrón resultado tiene que no tiene el patrón de entrenamiento con color verde y viceversa con color azul, en la Figura 20 se muestra el patrón de aprendizaje en la primera columna de ambas filas de la imagen, el patrón de prueba en la segunda columna de la primera fila, la máscara de diferencias entre patrón de aprendizaje y de prueba en la tercera columna de la primera fila, el patrón resultado en la segunda columna de la segunda fila, y la máscara de diferencias entre patrón de aprendizaje y patrón resultado en la tercera columna de la segunda fila:



*Figura 20.* Máscara de patrón obtenida en prueba de 10 subredes con  $k = 200$

Con un promedio de recuperación para diez subredes igual a 0.54, se ha optado por incrementar la cantidad de conexiones entre neuronas debido a que se encuentra muy dispersa con una cantidad actual de 200 conexiones para toda la red y 20 para cada subred, los resultados de estas pruebas con una cantidad mayor de conexiones entre neuronas se encuentran a continuación:

#### 4.2.5. 10 subredes con $k = 300$

El promedio de recuperación o coincidencia entre los patrones de entrenamiento con los patrones obtenidos para diez subredes con  $k=300$  es de 0.77, los valores para cada patrón se pueden visualizar en la Figura 21 donde se especifican los patrones con un color distinto para cada subred:

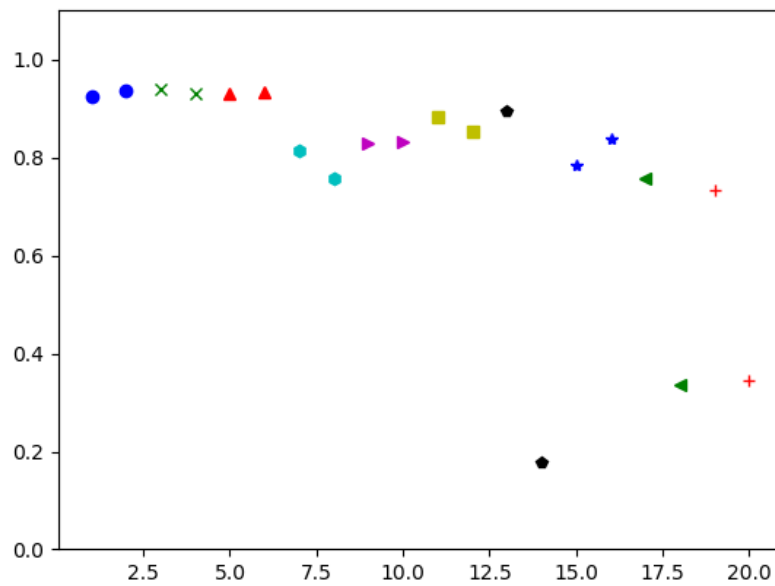


Figura 21. Valores de coincidencia de los patrones con diez subredes y  $k = 300$   
 La similitud entre las imágenes del conjunto de entrenamiento y las imágenes de los patrones resultado se puede apreciar en la Figura 22, en la cual se muestran los valores de coincidencia de cada patrón del conjunto de entrenamiento con los patrones del conjunto de resultado para la prueba realizada con diez subredes y  $k = 300$ :

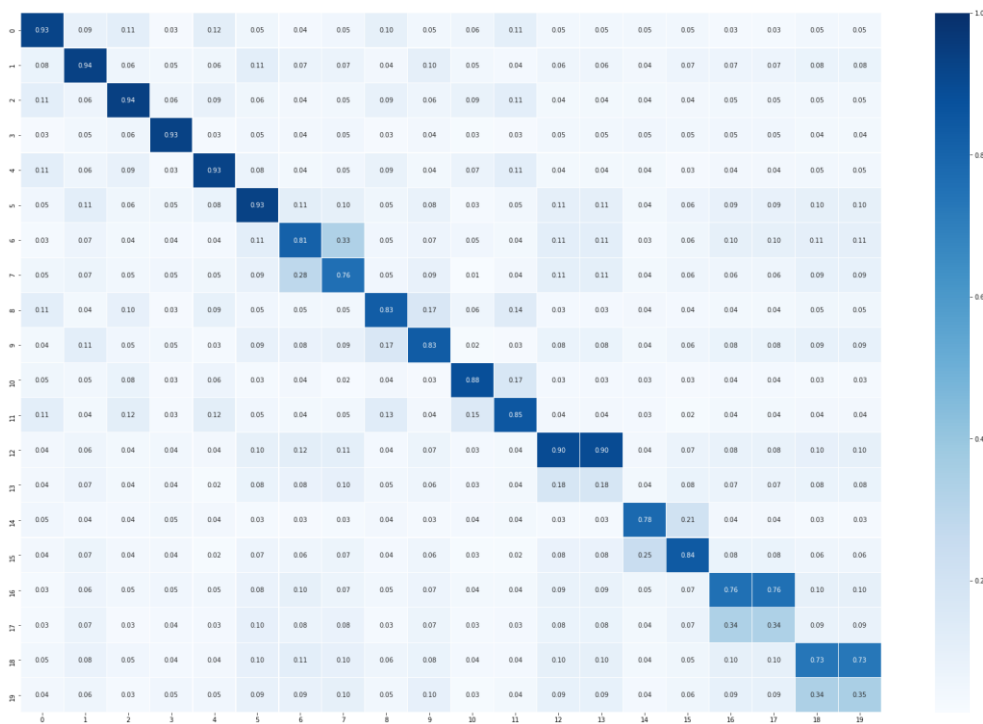
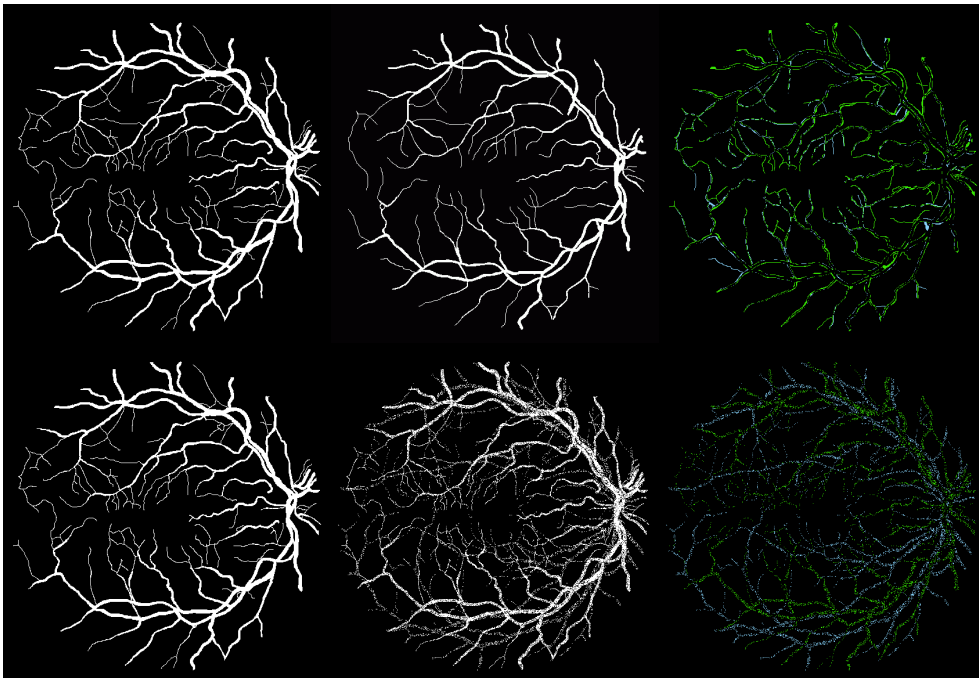


Figura 22. Matriz con valores de coincidencia entre patrones de aprendizaje y resultado con diez subredes y  $k = 300$

Se ha tomado como referencia el patrón que más se acerca al promedio de recuperación (patrón número 8 con recuperación = 0.75) para mostrar la máscara que marca los puntos que el patrón resultado tiene que no tiene el patrón de entrenamiento con color verde y viceversa con color azul, en la Figura 23 se muestra el patrón de aprendizaje en la primera columna de ambas filas de la imagen, el patrón de prueba en la segunda columna de la primera fila, la máscara de diferencias entre patrón de aprendizaje y de prueba en la tercera columna de la primera fila, el patrón resultado en la segunda columna de la segunda fila, y la máscara de diferencias entre patrón de aprendizaje y patrón resultado en la tercera columna de la segunda fila:



*Figura 23.* Máscara de patrón obtenida en prueba de 10 subredes con  $k = 300$

#### 4.2.6 .10 subredes con $k = 400$

El promedio de recuperación o coincidencia entre los patrones de entrenamiento con los patrones obtenidos para diez subredes con  $k=400$  es de 0.94, los valores para cada patrón se pueden visualizar en la Figura 24 donde se especifican los patrones con un color distinto para cada subred:

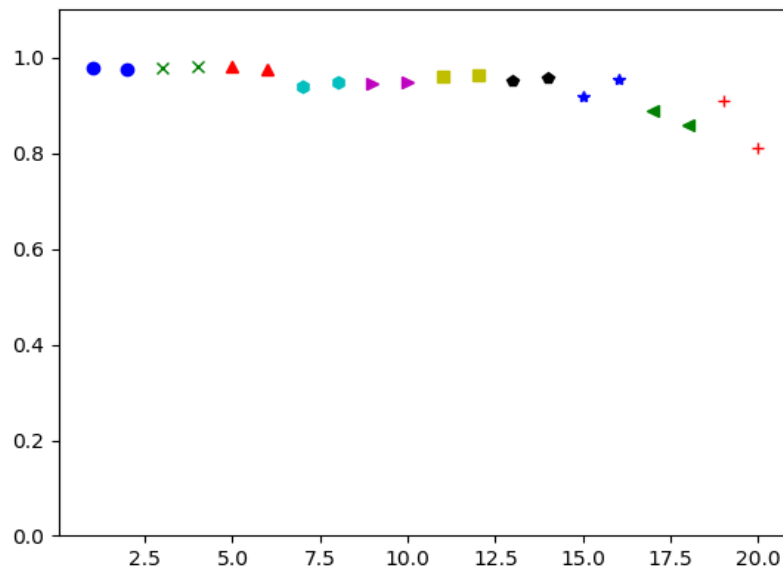


Figura 24. Valores de coincidencia de los patrones con diez subredes y  $k = 400$

La similitud entre las imágenes del conjunto de entrenamiento y las imágenes de los patrones resultado se puede apreciar en la Figura 25, en la cual se muestran los valores de coincidencia de cada patrón del conjunto de entrenamiento con los patrones del conjunto de resultado para la prueba realizada con diez subredes y  $k = 400$ :

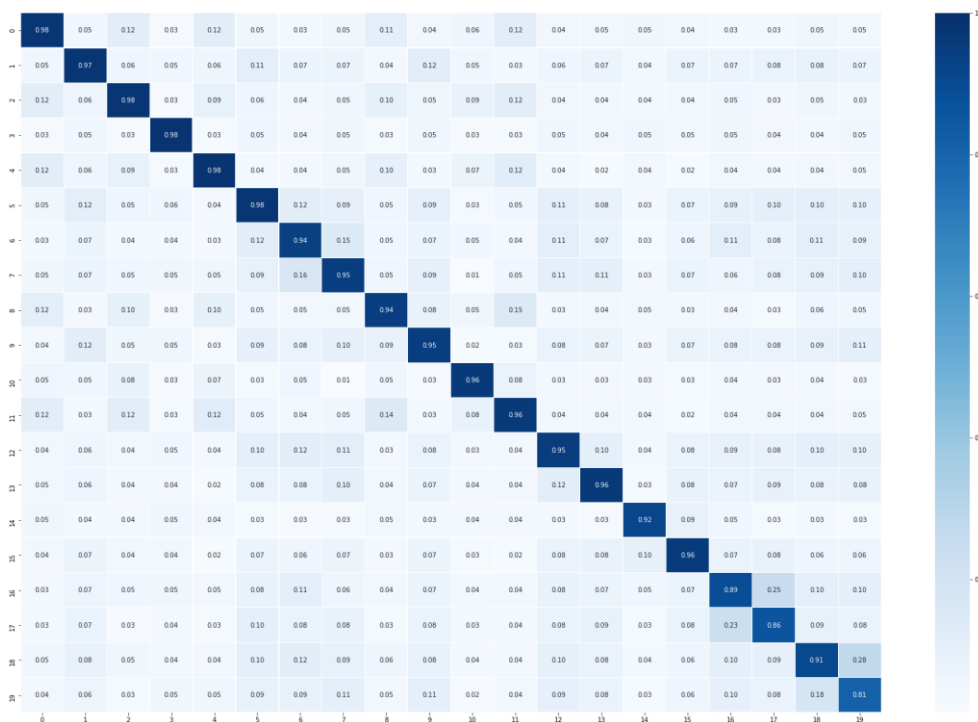


Figura 25. Matriz con valores de coincidencia entre patrones de aprendizaje y resultado con diez subredes y  $k = 400$



Se ha tomado como referencia el patrón que más se acerca al promedio de recuperación (patrón número 7 con recuperación = 0.94) para mostrar la máscara que muestran los puntos que el patrón resultado tiene que no tiene el patrón de entrenamiento (color verde) y los puntos que el patrón de entrenamiento tiene que no tiene el patrón resultado (color azul), la máscara se muestra en la Figura 26 mostrando el patrón de entrenamiento en el lado izquierdo, el patrón resultado en el centro y la máscara en la derecha:

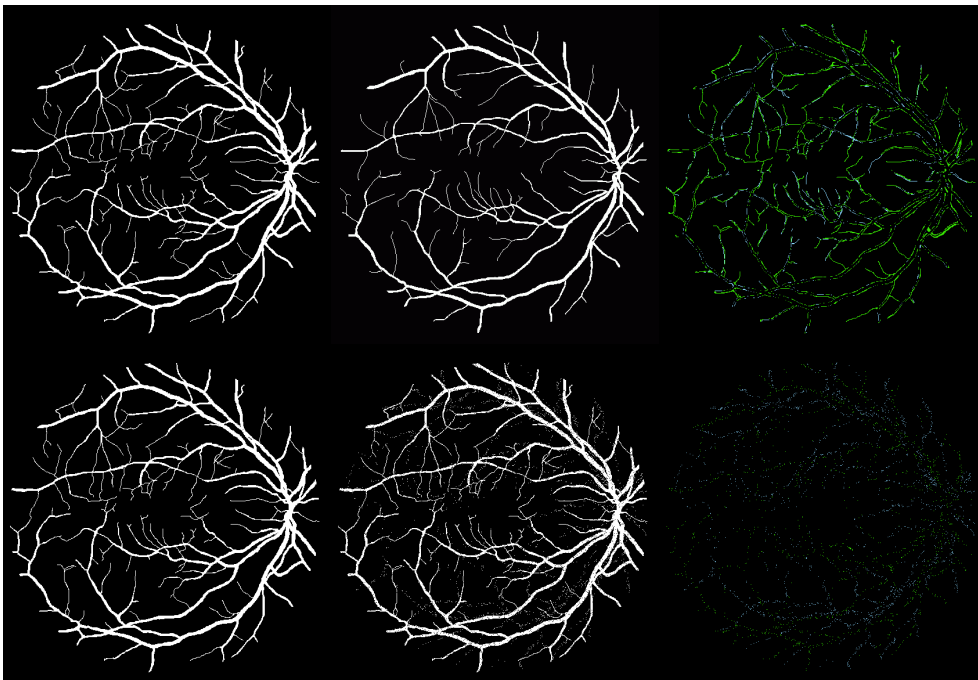


Figura 26. Máscara de patrón obtenida en prueba de 10 subredes con  $k = 400$

En la Tabla 1 se muestra el valor de recuperación para las pruebas de diferentes modelos de redes neuronales atractoras, el peor rendimiento en la recuperación es la que tiene una sola subred, en este caso el promedio de recuperación es igual a 0.23; el mejor rendimiento en la recuperación es la que tiene 2 subredes con un parámetro de dilución suficiente evitando limitar el aprendizaje de cada subred, en este caso usando 2 subredes con un promedio de recuperación igual a 0.88.

Tabla 1.

*Resultados del aprendizaje de 20 patrones con diferentes parámetros*

Cantidad de subredes	Cantidad de Patrones	Cantidad de conexiones en toda la red	Valor de recuperación
----------------------	----------------------	---------------------------------------	-----------------------

1	20	200	0.23
2	20	200	0.88
4	20	200	0.87
10	20	200	0.54

En la Tabla 2 se muestra el valor de recuperación para las pruebas con 10 subredes y un número variable en la cantidad de conexiones en toda la red, el peor rendimiento en la recuperación es la que tiene una menor cantidad de conexiones, en este caso con un promedio de recuperación igual a 0.54 y el mejor resultado muestra una mayor cantidad de conexiones con un parámetro de dilución suficiente que evita limitar el aprendizaje de cada subred, en este caso con un promedio de recuperación igual a 0.94.

Tabla 2.

*Resultados del aprendizaje de 20 patrones con diferentes parámetros y 10 subredes*

<b>Cantidad de subredes</b>	<b>Cantidad de Patrones</b>	<b>Cantidad de conexiones en toda la red</b>	<b>Valor de recuperación</b>
10	20	200	0.54
10	20	300	0.77
10	20	400	0.94

### **4.3. Prueba de subredes entrenadas con diferentes tipos de procesamiento de imágenes**

#### **4.3.1. 40 patrones con 0 y 90 grados de rotación**

El promedio de recuperación o coincidencia entre los patrones de entrenamiento con los patrones obtenidos para los 40 patrones con diferentes grados de rotación es de 0.951, los valores para cada patrón se pueden visualizar en la Figura 27 donde se especifican los patrones con un color distinto para cada subred:

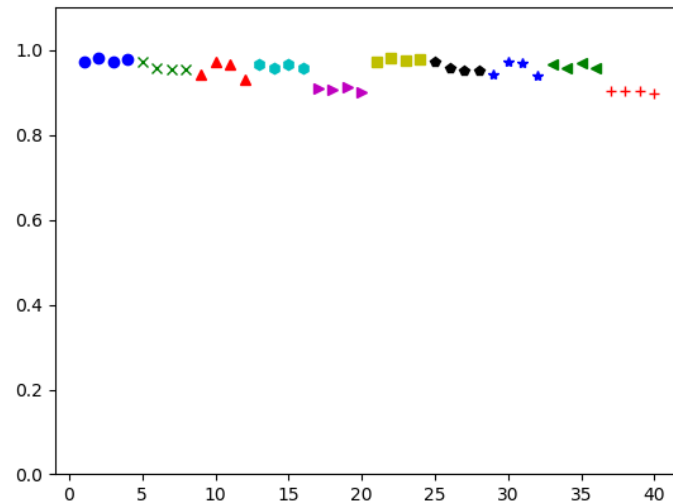


Figura 27. Valores de coincidencia de los 40 patrones

En la Tabla 3 se muestra los resultados para las pruebas con diferentes grados de rotación (0 y 90 grados), en este caso se observa que, al aumentar patrones, la red reconoce a los nuevos patrones como si fueran únicos y tiene un promedio de recuperación promedio igual a 0.951 usando 40 patrones y un  $k=600$ .

Tabla 3.

*Resultados del aprendizaje de 40 patrones con diferentes grados de rotación*

Cantidad de subredes	Cantidad de patrones	Cantidad de conexiones en toda la red	Valor de recuperación
10	40	600	0.951

#### 4.3.2. 60 patrones con 0, 90 y 270 grados de rotación

El promedio de recuperación o coincidencia entre los patrones de entrenamiento con los patrones obtenidos para los 60 patrones es de 0.957, los valores para cada patrón se pueden visualizar en la Figura 28 donde se especifican los patrones con un color distinto para cada subred:

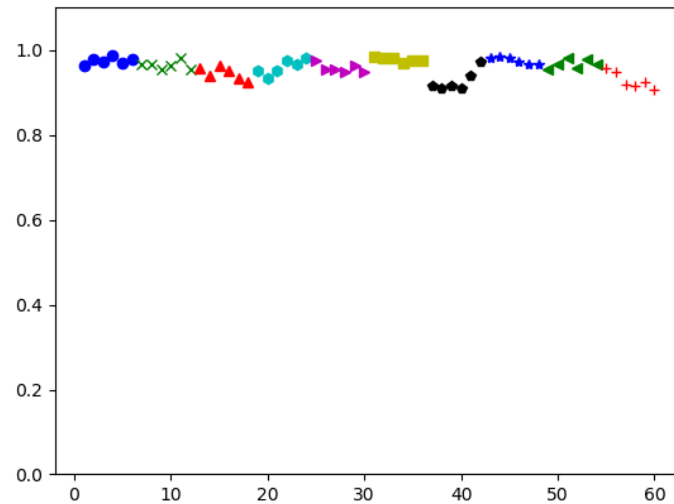


Figura 28. Valores de coincidencia de los 60 patrones

En la Tabla 4 se muestra los resultados para las pruebas con diferentes grados de rotación (0, 90 y 270 grados), en este caso se observa que, al aumentar patrones, la red reconoce a los nuevos patrones como si fueran únicos y tiene un promedio de recuperación promedio igual a 0.957 usando 60 patrones y un  $k=800$ .

Tabla 4.

*Resultados del aprendizaje de 60 patrones con diferentes grados de rotación*

Cantidad de subredes	Cantidad de patrones	Cantidad de conexiones en toda la red	Valor de recuperación
10	60	800	0.957

#### 4.3.3. 40 patrones

Para este tipo de prueba se han tomado en cuenta a los patrones de la carpeta de aprendizaje, a los cuales se les ha realizado el proceso de dilatación y se han utilizado cuatro subredes, cada una con patrones con una actividad similar, es decir especializadas según la relación de caracteres binarios iguales a 1 con caracteres binarios iguales a 0, valor obtenido a partir de las diferentes técnicas de procesamiento de imágenes, las imágenes de retinas esqueletadas se encuentran en la primera subred (10 primeros patrones), las imágenes binarizadas se encuentran en la segunda subred (patrones 11 al 20), las

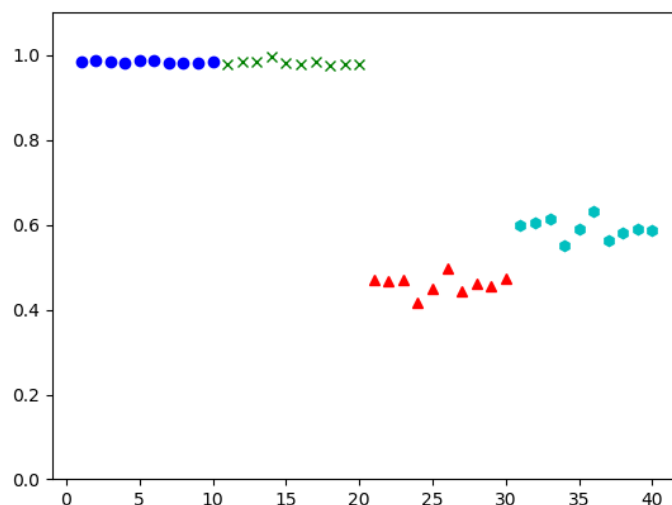
imágenes dilatadas en primer nivel se encuentran en la tercera subred (patrones 21 al 30) y las imágenes dilatadas en segundo nivel se encuentran en la última subred (patrones 31 al 40) de la Figura 29.

Los resultados de la Figura 29 con color azul muestran el valor de coincidencia de los diez patrones de la subred para retinas esqueletadas (retinas con actividad promedio igual a 0.0277), en la cual el promedio de recuperación de los diez patrones es de 0.983.

Los resultados con color verde muestran el valor de coincidencia de los diez patrones de la subred para retinas obtenidas de la base de datos (retinas con actividad promedio igual a 0.0875), en la cual el promedio de recuperación de los diez patrones es de 0.981.

Los resultados con color rojo muestran el valor de coincidencia de los diez patrones de la subred para retinas con primer nivel de dilatación (retinas con actividad promedio igual a 0.277), en la cual el promedio de recuperación de los diez patrones es de 0.46.

Los resultados con color rojo muestran el valor de coincidencia de los diez patrones de la subred para retinas con segundo nivel de dilatación (retinas con actividad promedio igual a 0.428), en la cual el promedio de recuperación de los diez patrones es de 0.59.



*Figura 29. Valores de coincidencia de los 40 patrones con diferentes tipos de procesamiento de imágenes*

En la Tabla 5 se muestra los resultados para las pruebas de subredes con diferentes tipos de procesamiento de imágenes, en este caso se observa que, existen diferentes valores en el promedio de recuperación según la técnica de procesamiento de imágenes, los mejores resultados se presentan para retinas esqueletadas en la cual se tiene un promedio de recuperación igual a 0.983 con 10 patrones.

Tabla 5.

*Resultados del aprendizaje de pruebas de subredes con diferentes tipos de procesamiento de imágenes y 10 patrones por subred*

<b>Tipo de imagen de retina</b>	<b>Cantidad de patrones</b>	<b>Cantidad de conexiones en toda la red</b>	<b>Valor de recuperación</b>
Esqueletada	10	600	0.983
Binarizada	10	600	0.981
Dilatada Primer Nivel	10	600	0.46
Dilatada Segundo Nivel	10	600	0.59

#### 4.3.4. 80 patrones con diferentes tipos de procesamiento de imágenes

Para este tipo de prueba se han tomado en cuenta a los patrones de la carpeta de aprendizaje, a los cuales se les ha realizado el proceso de dilatación y se han

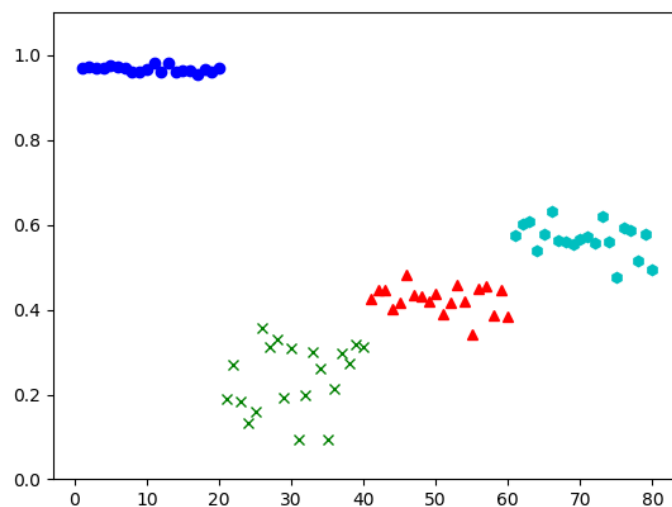
utilizado cuatro subredes, cada una con patrones con una actividad similar, es decir especializadas según la relación de caracteres binarios iguales a 1 con caracteres binarios iguales a 0, valores obtenidos a partir de las diferentes técnicas de procesamiento de imágenes.

Los resultados de la Figura 30 con color azul muestran el valor de coincidencia de los veinte patrones de la subred para retinas esqueletadas (retinas con actividad promedio igual a 0.0277), en la cual el promedio de recuperación de los veinte patrones es de 0.967.

Los resultados con color verde muestran el valor de coincidencia de los veinte patrones de la subred para retinas obtenidas de la base de datos (retinas con actividad promedio igual a 0.0875), en la cual el promedio de recuperación de los veinte patrones es de 0.24.

Los resultados con color rojo muestran el valor de coincidencia de los veinte patrones de la subred para retinas con primer nivel de dilatación (retinas con actividad promedio igual a 0.277), en la cual el promedio de recuperación de los veinte patrones es de 0.424.

Los resultados con color rojo muestran el valor de coincidencia de los diez patrones de la subred para retinas con segundo nivel de dilatación (retinas con actividad promedio igual a 0.428), en la cual el promedio de recuperación de los veinte patrones es de 0.566.



*Figura 30.* Valores de coincidencia de los 80 patrones con diferentes tipos de procesamiento de imágenes

En la Tabla 6 se muestra los resultados para las pruebas de subredes con diferentes tipos de procesamiento de imágenes, en este caso se observa que, existen diferentes valores en el promedio de recuperación según la técnica de procesamiento de imágenes, los mejores resultados se presentan para retinas esqueletadas en la cual se tiene un promedio de recuperación igual a y 0.967 con 20 patrones.

Tabla 6.

*Resultados del aprendizaje de pruebas de subredes con diferentes tipos de procesamiento de imágenes y 20 patrones por subred*

<b>Tipo de imagen de retina</b>	<b>Cantidad de patrones</b>	<b>Cantidad de conexiones en toda la red</b>	<b>Valor de recuperación</b>
Esqueletada	20	800	0.967
Binarizada	20	800	0.24
Dilatada Primer Nivel	20	800	0.424
Dilatada Segundo Nivel	20	800	0.566



## 5. CONCLUSIONES Y RECOMENDACIONES

### 5.1. Conclusiones

El rendimiento del modelo para el aprendizaje de patrones con una subred para 20 patrones tiene un promedio de recuperación igual a 0.23, valor obtenido debido a que la red combinó a todos los patrones de la carpeta de aprendizaje en uno solo y cada uno de los patrones de la carpeta de prueba, fueron atraídos por el único patrón aprendido, es por esto que los valores de recuperación son muy bajos en comparación a las pruebas realizadas para más subredes.

El rendimiento del modelo para el aprendizaje de patrones con más de una subred utilizando la misma cantidad de conexiones para 20 patrones, tiene un nivel mayor de recuperación, lo que muestra que, al añadir subredes al modelo, cada subred se concentrará en recuperar a los patrones dedicados para dicha subred y así los patrones recuperados tendrán una similitud mayor a la del patrón de aprendizaje.

La cantidad de conexiones de la red y la cantidad de subredes, modifican el parámetro de dilución  $\gamma = \frac{K}{N}$ , por lo tanto, cuando se usa  $k = 200$ , y 2 subredes, cada una tendrá un parámetro de dilución igual a 100, pero si se utilizan 10 subredes, el parámetro de dilución será igual a 20, limitando el aprendizaje máximo de la red, este factor se puede visualizar en el aprendizaje obtenido en pruebas para dos y diez subredes con  $k = 200$ , en donde el aprendizaje promedio es mucho mayor para dos subredes, sin embargo, al aumentar la cantidad de conexiones de la red a 300 y 400 para 10 subredes, el rendimiento de la red es mejor alcanzando un promedio de recuperación de 0.94.

Realizar un proceso de dilatación para conseguir una actividad mayor mostrará diferentes porcentajes de recuperación para la subred, y realizar un proceso de esqueletización para conseguir una actividad menor de las imágenes binarias de retinas según corresponda, funciona para tener un mejor nivel de recuperación, ya que los rasgos de la segmentación de los vasos sanguíneos serán más reconocibles para la red tanto en la fase de aprendizaje, como en la de prueba. El rendimiento de la red atractora conjunta mejora su eficiencia cuando se realiza la agrupación de patrones a partir de la actividad, el nivel de recuperación

“*offline*” varía en cada subred según su actividad ya que cada una tendrá un valor de umbral distinto, por lo que, al agruparlas y utilizar una actividad con un buen nivel de recuperación se obtendrán mejores resultados en la autenticación de retinas.

En los resultados se muestra que, para los 40 y 60 patrones, usando 10 subredes y un  $k$  igual a 600 y 800 en orden correspondiente, se obtuvo una recuperación promedio de 0.95, demostrando que es posible recuperar más patrones y que para la red una retina que ha pasado por un proceso de rotación es diferente a la retina original.

## **5.2. Recomendaciones**

Se recomienda realizar el proceso de aprendizaje y reconocimiento de imágenes de retinas con una actividad acorde que permita ahorrar recursos computacionales.

Se recomienda utilizar librerías existentes en Python que sean eficientes con respecto al tiempo en que se obtienen los resultados para diferentes procesos.

## REFERENCIAS

- Cárdenes, R. (2006). *Inteligencia Artificial*. Las Palmas de Gran Canaria: Universidad de Las Palmas de Gran Canaria. Recuperado el 3 de Mayo de 2019, de [https://www2.ulpgc.es/hege/almacen/download/38/38584/practica\\_ia\\_2.pdf](https://www2.ulpgc.es/hege/almacen/download/38/38584/practica_ia_2.pdf).
- Erdős, P., & Rényi, A. (1959). *On random graphs I. Publicationes Mathematicae* Debrecen. Recuperado el 15 de Enero de 2019, de <http://snap.stanford.edu/class/cs224w-readings/erdos59random.pdf>.
- Flórez, R., Fernández, J. (2008). *Las Redes Neuronales Artificiales*. La Coruña, España: Netbiblo.
- Ginneken, M. N. (2002). *DRIVE: Digital Retinal Images for Vessel Extraction*. Recuperado el 12 de Diciembre de 2018, de <https://www.isi.uu.nl/Research/Databases/DRIVE>.
- Gonzalez, M., Dominguez, D., Rodriguez, F. B., & Sanchez, A. (2017). *Capacity and Retrieval of a Modular Set of Diluted Attractor Networks with Respect to the Global Number of Neurons. Advances in Computational Intelligence*. IWANN 2017. 10305, 497-506.
- Gonzalez, M., Dominguez, D., Sánchez, A. & Rodriguez, F. B. (2016). *Increase attractor capacity using an ensembled neural network*. *Expert Syst. Appl.* 71, 206–215.
- Hopfield, J. (1982). *Neural networks and physical systems with emergent collective computational abilities. Proc. Natl. Acad. Sci.* 79(8), 2554–2558.
- Kisku, D., Gupta, P., Sing, J. (2019). *Design and Implementation of Healthcare Biometric Systems*. Durgapur, India: *Medical Information Science Reference*.
- Matich, D. (2001). *Redes Neuronales: Conceptos Básicos y Aplicaciones*. Recuperado el 15 de Mayo de 2019, de [https://www.frro.utn.edu.ar/repositorio/catedras/quimica/5\\_anio/orientador/monograias/matich-redesneuronales.pdf](https://www.frro.utn.edu.ar/repositorio/catedras/quimica/5_anio/orientador/monograias/matich-redesneuronales.pdf).
- Python. (2019). *Welcome to Python.org*. Recuperado el 28 de Noviembre de

2018, de Python.org: <https://www.python.org/doc/>.

Sadikoglu, F., Uzelaltinbulat, S. (2016). *Biometric retina identification based on neural network*. *Procedia Computer Science*. 102, 26-33.

Vega, H., Cortez, A., Huayna, A., Alarcón, L., Romero, P. (2009).

Reconocimiento de patrones mediante redes neuronales artificiales.

Revista de Ingeniería de Sistemas e Informática. 6(2), 17-26.

# **ANEXOS**

## Anexo 1: Código Python

```
global overlap_Values, local_overlap_Values, initialNetworkState,
initialNetworkStateToProcess
os.chdir(os.getcwd()+ '/' +patternsFolderIn)
initRandomNetwork()
sn = 0
i = 0
hasNext = True
for filename in os.listdir():
    if hasNext:
        initialNetworkState[sn][i] = ReadFile(filename,sn,i)
        i+=1
        if(i>=N_patterns_per_subnet):
            i = 0
            sn += 1
            if(sn==N_subnetworks):
                hasNext = False
                break
calculateConstraints()
for subn in range(N_subnetworks):
    for patt in range(N_patterns_per_subnet):
        calculateWeights(subn,patt)
os.chdir('..')
os.chdir(os.getcwd()+ '/' +patternsFolderOut)
sn = 0
i = 0
hasNext = True
for filename in os.listdir():
    if hasNext:
        initialNetworkStateToProcess[sn][i] = ReadFile(filename,sn,i)
        i+=1
        if(i>=N_patterns_per_subnet):
            i = 0
            sn += 1
            if(sn==N_subnetworks):
                hasNext = False
                break
for subn in range(N_subnetworks):
    local_overlap_Values = []
    for patt in range(N_patterns_per_subnet):
        processPatronSteps(subn,patt,0)
    overlap_Values.extend(local_overlap_Values)

def initRandomNetwork():
    global connections, weights
```

```

    connections = [[[[] for i in range(N_neurons)] for i in
range(N_subnetworks)]
    weights = [[[[] for i in range(N_neurons)] for i in
range(N_subnetworks)]
    for sn in range(N_subnetworks):
        sample = np.arange(N_neurons)
        for i in range(0, N_neurons):
            np.random.shuffle(sample)
            while i in sample[:N_neighbors_per_subnet]:
                np.random.shuffle(sample)
            neuronConnection = []
            for j in sample[:N_neighbors_per_subnet]:
                neuronConnection += [j]
            connections[sn][i] = np.array(neuronConnection)
            weights[sn][i] = np.zeros(N_neighbors_per_subnet)

def ReadFile(filename,sn,i):
    global initialNetworkState
    with open(filename) as f:
        rawArray = np.fromstring(f.read(), dtype=int, sep=' ')
    return np.array(rawArray)

def calculateConstraints():
    global umbral, A, a
    a = np.mean(np.mean(initialNetworkState))
    A = a*(1-a)
    umbral = (1-2*a)/(2*np.sqrt(A))

def calculateWeights(subnet, i):
    global weights, Xi_zeroValues
    aPattern = np.mean(np.mean(initialNetworkState[subnet][i]))
    APattern = aPattern*(1-aPattern)
    Xi_zeroValues = (initialNetworkState[subnet][i]-
aPattern)/np.sqrt(APattern)
    for n in range(N_neurons):
        Aux = Xi_zeroValues[n]*Xi_zeroValues[connections[subnet][n]].T
        weights[subnet][n] = np.add(weights[subnet][n], Aux)

def processPatronSteps(subnet,i,flag=0):
    global local_overlap_Values, Tau_Values, nomArchivo
    nomArchivo += 1
    aPattern = np.mean(np.mean(initialNetworkState[subnet][i]))
    APattern = aPattern*(1-aPattern)
    Xi_zeroValues = (initialNetworkState[subnet][i]-
aPattern)/np.sqrt(APattern)
    m_flag = True

```

```

m = 0
if(flag==0):
    Tau_Values = np.array(initialNetworkState[subnet][i])
else:
    Tau_Values = np.array(initialNetworkStateToProcess[subnet][i])
for it in range(N_steps):
    calculateVariables()
    if(m_flag):
        tau_plusValues = [] # Reset variable
        for n in range(N_neurons):
            tau_plusValues +=
[HeavisideFuntion(calculateH(subnet,n))]

        mAux = abs(np.mean(sigma_Values*Xi_zeroValues))
        if(mAux == m):
            m_flag = False

        m = mAux
        Tau_Values = np.array(tau_plusValues)
        if(it == N_steps-1):
            local_overlap_Values.append(m)
            break
    else:
        local_overlap_Values.append(m)
        break

def calculateVariables():
    global sigma_Values
    q = np.mean(Tau_Values)
    Q = q*(1-q)
    sigma_Values = (Tau_Values-q)/np.sqrt(Q)

def HeavisideFuntion(h): # Test and create the new neuron's state /
known as Xi(t)
    if h >= 0:
        return 1
    else:
        return 0

def calculateH(subnet,neuronNumber):
    tau_local = Tau_Values[connections[subnet][neuronNumber]]
    qL = np.mean(tau_local)
    QL = qL*(1-qL)
    sigma_Values_Local = (tau_local-qL)/np.sqrt(QL)
    Aux = weights[subnet][neuronNumber]*sigma_Values_Local
    return np.mean(Aux)-umbral

```



