

Discriminación de patologías retinianas mediante procesado avanzado de imágenes de fondo de ojo y algoritmos de Machine Learning

Cristóbal Mollá Molina

Tutor: Valeriana Naranjo Ornedo

Cotutor: Adrián Colomer Granero

Trabajo Fin de Grado presentado en la Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universitat Politècnica de València, para la obtención del Título de Graduado en Ingeniería de Tecnologías y Servicios de Telecomunicación

Curso 2016-17

Valencia, 12 de septiembre de 2017

Resumen

La retinopatía diabética (RD) es una de las principales causas actuales de ceguera. La detección precoz de ciertos signos clínicos como son los microaneurismas, los exudados o las hemorragias, es esencial para el correcto diagnóstico de la patología por parte de los especialistas.

El presente Trabajo de Fin de Grado se centra en la detección automática de lesiones vasculares en imágenes de fondo de ojo. Con este objetivo se propone un algoritmo implementado en MATLAB que utiliza un análisis de texturas a nivel local mediante descriptores de textura y forma, combinados con técnicas de aprendizaje automático o *machine learning* para la clasificación.

Los resultados de distintas simulaciones realizadas con el método de análisis local propuesto se emplean posteriormente para desarrollar un segundo algoritmo que tiene como objetivo proporcionar un diagnóstico a nivel de imagen, pudiendo así discriminar entre imágenes patológicas o sanas.

Resum

La retinopatia diabètica (RD) és una de les principals causes actuals de ceguera. La detecció precoç de certs signes clínics com son els microaneurismes, els exudats o les hemorràgies, és essencial per al correcte diagnòstic de la patologia per part dels especialistes.

El present Treball de Fi de Grau es centra en la detecció automàtica d'hemorràgies en imatges de fons d'ull. Amb aquest objectiu, es proposa un algoritme implementat en MATLAB que utilitza un anàlisi de textures a nivell local mitjançant descriptors de textura i forma, combinats amb tècniques d'aprenentatge automàtic o *machine learning*

Els resultats de distintes simulacions realitzades amb el mètode d'anàlisi local proposat s'empelen posteriorment per a desenvolupar un segon algoritme que té com objectiu proporcionar un diagnòstic a nivell d'imatge, podent així discriminar entre imatges patològiques o sanes.

Abstract

Nowadays, diabetic retinopathy (RD) is one of the leading causes of blindness in the world. Early detection of clinical signs such as microaneurysms, exudates or haemorrhages is essential for the correct diagnosis of the pathology by the specialists.

This final degree project focuses on the automatic detection of vascular lesions in fundus images. For this purpose, we propose an algorithm implemented in MATLAB based on texture and shape descriptors combined with machine learning techniques for the classification.

The results of the different experiments, using the local method proposed, are the input of a second algorithm that aims to provide an automatic diagnose of a fundus image being able to discriminate between pathological or healthy images.

Índice

| | |
|------------------------------------------------------|----|
| Capítulo 1. Introducción | 3 |
| 1.1 Guía del trabajo..... | 3 |
| 1.2 Marco del trabajo..... | 4 |
| 1.3 Motivación..... | 4 |
| 1.4 Anatomía del sistema visual humano..... | 5 |
| 1.4.1 Anatomía de la retina | 5 |
| 1.5 Imágenes de Fondo de ojo..... | 6 |
| 1.6 Diabetes Mellitus..... | 6 |
| 1.7 Retinopatía Diabética..... | 8 |
| 1.7.1 Signos clínicos..... | 8 |
| 1.8 Estado del arte en detección de hemorragias..... | 10 |
| Capítulo 2. Objetivos..... | 12 |
| Capítulo 3. Metodología de trabajo..... | 13 |
| 3.1 Base de datos ‘Diaretdb1’..... | 13 |
| 3.1.1 Ground truth | 14 |
| 3.2 Máscaras binarias | 15 |
| 3.2.1 Máscara del disco óptico..... | 15 |
| 3.2.2 Máscara externa o del FOV (Field of View)..... | 15 |
| 3.2.3 Máscara de los vasos sanguíneos | 16 |
| 3.3 Espacio de color RGB..... | 16 |
| 3.4 Reparto de Tareas y Diagrama Temporal..... | 17 |
| Capítulo 4. Detección de lesiones oscuras | 18 |
| 4.1 Introducción | 18 |
| 4.2 Local Binary Pattern (LBP)..... | 18 |
| 4.3 Granulometría | 23 |

| | | |
|--------------|---------------------------------------------------------------------------|----|
| 4.3.1 | Operadores morfológicos básicos | 23 |
| 4.3.2 | Perfiles morfológicos | 24 |
| 4.4 | Support Vector Machine (SVM) | 27 |
| 4.4.1 | Desarrollo visual del SVM | 27 |
| 4.4.2 | Desarrollo matemático del SVM | 30 |
| 4.4.3 | Técnica de Bagging (Bootstrap Aggregating) | 31 |
| 4.4.4 | LIBSVM | 32 |
| 4.5 | Implementación del algoritmo | 33 |
| 4.5.1 | Extracción del ground truth | 33 |
| 4.5.2 | Extracción y selección de características | 35 |
| 4.5.3 | CrossValidation | 35 |
| 4.5.4 | Creación del modelo (Train) | 36 |
| 4.5.5 | Clasificación de muestras (Test) | 36 |
| 4.5.6 | Obtención de Resultados | 36 |
| 4.5.7 | Configuración del sistema | 37 |
| Capítulo 5. | Clasificación a nivel de imagen | 39 |
| 5.1 | Introducción | 39 |
| 5.2 | Implementación del algoritmo | 39 |
| 5.2.1 | Carga de resultados del algoritmo de clasificación a nivel de patch | 40 |
| 5.2.2 | Extracción del ground truth | 41 |
| 5.2.3 | Histograma de probabilidades | 42 |
| 5.2.4 | Clasificación de las imágenes y Extracción de métricas Finales | 42 |
| Capítulo 6. | Resultados | 43 |
| 6.1.1 | Visualización | 46 |
| Capítulo 7. | Conclusiones y líneas futuras | 47 |
| Bibliografía | | 48 |

Capítulo 1. Introducción

1.1 Guía del trabajo.

A continuación se expone una síntesis sobre el contenido general de los distintos capítulos de la presente memoria:

- **Capítulo 1. Introducción**
En el primer capítulo se expondrán los conceptos anatómicos más importantes del globo ocular así como todos los aspectos relacionados con la Retinopatía Diabética (RD) necesarios para comprender la motivación del proyecto que se expone.
- **Capítulo 2. Objetivos**
En este capítulo se detallarán los objetivos perseguidos por el trabajo propuesto.
- **Capítulo 3. Metodología de trabajo**
En el tercer capítulo se presentarán ciertos aspectos de gran relevancia debido a que definirán nuestro modo de trabajo. Además, en este capítulo, se adjunta una tabla con la distribución de las tareas y la carga temporal que ha conllevado cada una de ellas.
- **Capítulo 4. Detección de lesiones oscuras (hemorragias)**
En el capítulo cuarto se describirán los métodos empleados en el presente trabajo relativos a la descripción y clasificación de texturas. Asimismo, se detallará la manera en la que se han empleado estos métodos en la detección automática de lesiones oscuras que se manifiestan en el tejido retiniano debido a la Retinopatía Diabética.
- **Capítulo 5. Clasificación de imágenes patológicas**
En esta sección se detallará el algoritmo implementado para trasladar la información local, extraída mediante los descriptores detallados en el capítulo anterior, a un único vector de características que permita diagnosticar una imagen de fondo de ojo dada.
- **Capítulo 6. Resultados**
En este capítulo se expondrán los resultados obtenidos de los diferentes experimentos llevados a cabo en el trabajo que se expone. Además se realizará una comparativa con los métodos del estado del arte en detección de patrones oscuros de carácter maligno que se manifiestan en el fondo de ojo.
- **Capítulo 7. Conclusiones y líneas futuras**
En el último capítulo se extraerán las conclusiones más relevantes de nuestro proyecto y se definirán las distintas líneas a seguir para avanzar en la detección automatizada de patologías como la Retinopatía Diabética.

1.2 Marco del trabajo.

El Trabajo de Fin de Grado propuesto en la presente memoria, se engloba dentro de las tareas necesarias para la realización del proyecto ACRIMA cuyo objetivo es desarrollar un sistema automático de diagnóstico de las tres patologías más significantes relacionadas con la ceguera: glaucoma, retinopatía diabética y degeneración macular asociada a la edad.

En particular, en esta memoria se abordarán aspectos necesarios para llevar a cabo la detección de ciertos patrones anómalos en el ojo humano asociados a la retinopatía diabética.

El proyecto ACRIMA se incluye en la línea de trabajo relacionada con el tratamiento de imagen del grupo de investigación CVB Lab, perteneciente al Instituto de Investigación e Innovación en Bioingeniería (I3B) integrado en la Ciudad Politécnica de la Innovación (CPI) de la Universitat Politècnica de València (UPV) y coordinado por Dr. Valery Naranjo, profesora del Departamento de Comunicaciones de la UPV y tutora del actual trabajo de fin de grado, quien ha dirigido y participado en diversos proyectos nacionales y europeos de investigación, así como conferencias internacionales y publicado numerosos artículos en diversas revistas de gran impacto.

Junto con Dr. Valery Naranjo, Adrián Colomer, ingeniero de telecomunicaciones y máster en tecnologías de telecomunicación, sistemas y redes desde 2013 y 2014 respectivamente e integrante del grupo de investigación CVB Lab, ha realizado las labores de cotutor del trabajo propuesto. Adrián Colomer es experto en diversas áreas diversas como procesado de imagen, morfología matemática, análisis de texturas o machine learning. Actualmente trabaja en el proyecto ACRIMA, habiendo publicado diversos artículos de ámbito nacional e internacional en ésta línea de investigación.

1.3 Motivación.

La Organización Mundial de la salud (OMS) calcula que 285 millones de personas tenían algún tipo de discapacidad visual en 2010. Estudios recientes demuestran que el número de casos de ceguera ha sido significativamente reducido en los últimos años (39 millones de personas en 2010) (Organización Mundial de la Salud, 2015).

A pesar de estos prometedores datos, la OMS estima que el 80% de los casos clínicos podrían ser evitados si se diagnostican y se tratan temprano. Además, la OMS señala que se podrían ahorrar 102 billones de dólares con servicios apropiados de atención oftalmológica.

El plan de acción presentado por la OMS refleja la necesidad de los servicios oftalmológicos para convertirse en una parte integral de la atención primaria de salud. El plan propuesto también destaca la importancia de las campañas de prevención de las patologías retinianas en su fase inicial (Organización Mundial de la Salud, 2015).

Sin embargo, estas propuestas requieren una gran carga de trabajo de expertos capacitados en la identificación de patrones anómalos de la retina. Este hecho, junto con el crecimiento de la población, el envejecimiento, la inactividad física, el consumo de tabaco y los crecientes niveles de obesidad hacen que estas propuestas sean inviables desde el punto de vista económico.

Por otra parte, debe señalarse que este tipo de diagnóstico de la enfermedad de la retina está altamente influenciado por la subjetividad inherente de cada experto.

De todos estos argumentos, se extrae la necesidad de desarrollar sistemas de detección automatizada que puedan ser utilizados por los profesionales en la atención primaria, y que permitan reducir el tiempo de trabajo de los especialistas al mínimo.

1.4 Anatomía del sistema visual humano.

Antes de empezar exponiendo el objeto que se tratará propiamente en este trabajo, creemos necesaria la siguiente introducción de los conceptos más importantes del sistema visual humano. Para ello, realizaremos un pequeño análisis sobre la anatomía del globo ocular, así como de sus órganos y funciones más relevantes para el estudio que nos atañe.

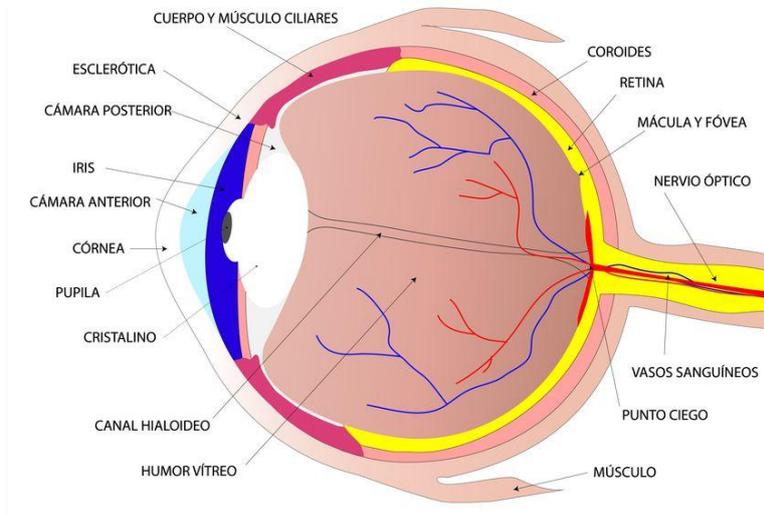


Figura 1. Anatomía del globo ocular

La función principal del ojo humano es la captación lumínica a través de una serie de estructuras transparentes para su posterior transformación en imágenes. De todas estas estructuras que conforman el globo ocular destacamos las siguientes: el **crystalino** que es una lente biconvexa que se encarga del enfoque visual; el **iris**, músculo liso que actúa como diafragma, ya que regula la cantidad de luz que atraviesa la **pupila** y que llega a la **retina**, encargada de transformar estos estímulos lumínicos en impulsos nerviosos que se transmiten a través del **nervio óptico**. [1]

1.4.1 Anatomía de la retina

La retina es la capa más interna del globo ocular. En ella se encuentran pequeños receptores que en respuesta a la luz emiten un impulso nervioso a través de células fotosensibles (**conos** y **bastones**) que permiten la formación de la imagen.

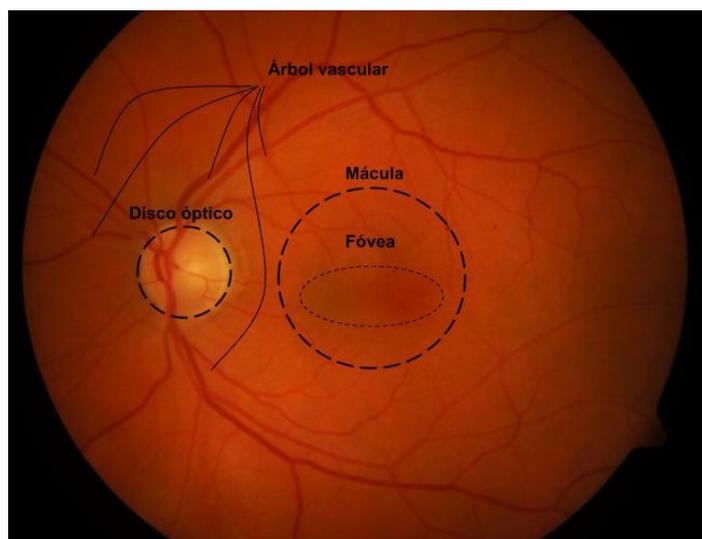


Figura 2. Imagen de Fondo de Ojo

En la retina observamos una gran cantidad de vasos sanguíneos que forman el **árbol vascular** y que confluyen en el **disco óptico**. Este último es una estructura ovalada de 1.5 mm de diámetro aproximadamente y constituye el punto de entrada del nervio óptico.

La **mácula** es la zona de la retina con la concentración de conos más alta y además se encarga de la visión central. En el centro de ella, se encuentra la **fóvea**, que es una depresión circular oscura de 1 mm de diámetro aproximadamente.

1.5 Imágenes de Fondo de ojo.

Durante el presente proyecto, se ha trabajado con imágenes de fondo de ojo como la mostrada en la Figura 2. Este tipo de imágenes, tanto para nuestro proyecto como para oftalmólogos en general, es de vital utilidad para el correcto diagnóstico de la retinopatía diabética. Para la adquisición de estas imágenes se emplean unos dispositivos conocidos como **retinógrafos**. Estos pueden ser midriáticos o no midriáticos, dependiendo de si necesitan o no de una previa dilatación de la pupila para la adquisición de la imagen. Cabe destacar que se pueden obtener imágenes centradas en la mácula o en el disco óptico dependiendo de cuál sea el protocolo de detección a seguir o cual sea la patología a detectar.



Figura 3. Retinógrafo no midriático

1.6 Diabetes Mellitus.

La *diabetes mellitus* es una enfermedad de orden metabólico que se reconoce en el paciente por tener una concentración de **glucosa** en sangre elevada. [2] [4]

El páncreas es un órgano glandular que se encuentra situado en el abdomen. Éste, entre otras muchas funciones, es el encargado de segregar la **insulina**. La insulina es una hormona que se encarga de controlar y dirigir la cantidad de glucosa a través de todo el organismo.

Las principales causas que originan el trastorno conocido como *diabetes* son: la resistencia a la insulina del propio cuerpo humano (**diabetes tipo II**) o la poca producción de ésta (**diabetes tipo I**). Pero para poder entender esta patología correctamente hemos de conocer primero como actúa la insulina en nuestro organismo.

La glucosa (coloquialmente azúcar) se obtiene en gran medida de los alimentos que uno consume y constituye la principal fuente de energía del ser humano. El torrente sanguíneo es el encargado de transportar la glucosa a las diferentes células del cuerpo. No obstante, se necesita de otro actor para la captación celular de glucosa: la insulina.

La insulina sirve de enlace entre la glucosa y las células haciendo las veces de ‘puerta de entrada’. Si este enlace no funciona correctamente, la glucosa se acumula en la sangre provocando síntomas tan diversos como el cansancio o la sed, además de otras muchas complicaciones mayores. Es por esto, que se deduce la importancia de un control glucémico.

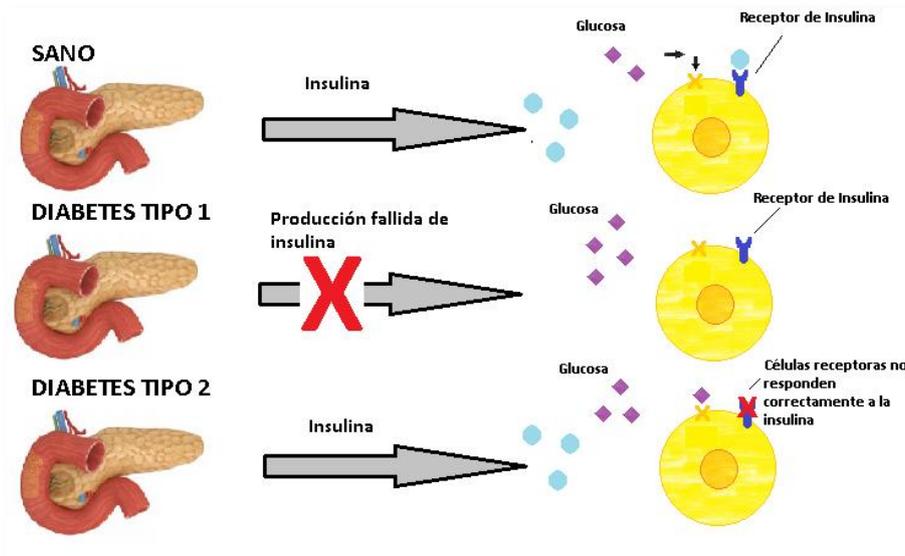


Figura 4. Esquema tipos de diabetes

La *diabetes mellitus* tipo I se puede diagnosticar a cualquier edad, no obstante, es mucho más frecuente un diagnóstico en edades tempranas que en edades adultas. Está caracterizado por un déficit de producción de insulina (las células pancreáticas no funcionan correctamente). Debido a esto, son necesarias inyecciones de insulina diarias.

Por su parte, la *diabetes mellitus* tipo II se suele diagnosticar mayormente en adultos y está asociada al estilo de vida y alimentos que consumimos mayormente. En los últimos años, debido a las altas tasas de obesidad en muchos países han producido que cada vez sea más común el diagnóstico de este trastorno. En consecuencia, las altas tasas de obesidad infantil producen que también sea ya común el diagnóstico en edades tempranas. En estos casos, el organismo patológico es resistente a la insulina debido a que los receptores celulares de insulina no funcionan adecuadamente.

Además de estos dos tipos de diabetes, existen muchos casos clínicos que no pueden ser incluidos en ninguno de los aquí mencionados, como por ejemplo, la diabetes gestacional que es un tipo de diabetes que se desarrolla solo durante el periodo de embarazo.

Para entender la magnitud del problema que nos atañe, hemos de conocer que la diabetes ha sido elegida como una de las 4 enfermedades no transmisibles (ENT) a tratar con carácter prioritario. La carga mundial de adultos diagnosticados con esta patología alcanzaba ya en el 2014 los 422 millones de personas, y su crecimiento es continuado. [3]

Asimismo, debido a que la esperanza de vida, sobretodo en países desarrollados ha ido en los últimos años en aumento, han aparecido últimamente un gran número de casos de complicaciones asociadas a la diabetes como son daños renales, neuropáticos, infecciones o complicaciones oculares (diabetes ocular). En el caso del presente trabajo, trataremos estos últimos, en concreto, hablaremos de la **Retinopatía Diabética**.

1.7 Retinopatía Diabética.

La **retinopatía diabética** (RD) es la manifestación clínica de la diabetes en la retina. Esta patología afecta precapilares, capilares y vénulas de la retina y es una de las mayores complicaciones de la diabetes considerada en la actualidad como una de las principales causas de ceguera.[5]

Algunos factores que pueden favorecer la aparición de esta enfermedad son la duración de la diabetes, un mal control metabólico u otros factores genéticos.

La respuesta tan específica de la diabetes en el ojo humano es debida a que este órgano tiene una tremenda capacidad metabólica y es la estructura que, en relación unidad/peso, mayor consumo de oxígeno y glucosa requiere.

En las figuras 5a y 5b se presenta la visión de una escena por un sujeto sano y un sujeto que padece retinopatía diabética:



Figura 5a. Fotografía vista por un ojo sano



Figura 5b. Fotografía vista por un ojo con RD

1.7.1 Signos clínicos

A la hora de evaluar la posibilidad de sufrir la patología en cualquier paciente, es de vital importancia el reconocimiento de una serie de signos que, por su naturaleza, son los más característicos de esta enfermedad (Figura 6).

En primer lugar, debido a la hiperglucemia característica de la diabetes, son propicios cambios a nivel vascular (alteraciones de la pared y un aumento del flujo sanguíneo). Como consecuencia de estas variaciones se produce isquemia en los capilares que generará la hipoxia (estado de deficiencia de oxígeno) de la retina. Debido a este proceso, diversas lesiones se manifiestan en la retina. Estas lesiones permiten llevar a cabo un diagnóstico certero por parte de los especialistas a partir de una instantánea del fondo de ojo y será posible distinguir las lesiones que permitirán diagnosticar de manera correcta la retinopatía.

Uno de los primeros signos oftálmicos detectables y por ello que sea de gran importancia su detección, es la aparición de **microaneurismas**. Los microaneurismas son dilataciones de pequeños vasos que son reconocibles ya que su forma es similar a unas manchas redondas de color oscuro con bordes afilados. Miden entre 15 y 125 micras y se forman con mayor frecuencia en los capilares arteriales, aunque en los venosos también lo hacen.

Cuando un microaneurisma, un capilar o una vénula se rompen, aparecen las **hemorragias intrarretinales**. Existen dos tipos: las profundas, localizadas en capas medias de la retina, de forma redondeada y pequeño tamaño, y las superficiales que son de forma más alargada. Son por tanto, una amplia acumulación de sangre en la retina. Las hemorragias empiezan a ser visibles cuando la retinopatía diabética progresa.

Estos dos signos explicados anteriormente se engloban en lo que conocemos como **lesiones oscuras**, ya que en las imágenes de fondo de ojo aparecen de un tono más oscuro que el fondo retiniano. El otro gran grupo de lesiones son las **lesiones brillantes** en los cuales se consideran los exudados duros y blandos.

Los **exudados duros** son depósitos de lípidos y proteínas en la retina. Son de color blanco (o amarillento) con límites irregulares pero precisos y de tamaño variable. Se pueden ver de forma aislada o agrupados formando anillos o estrellas y tomando como punto focal un microaneurisma.

Los **exudados blandos** son depósitos algodonosos de forma redondeada y color amarillento con bordes imprecisos causados por oclusión capilar. Son en realidad microinfartos.

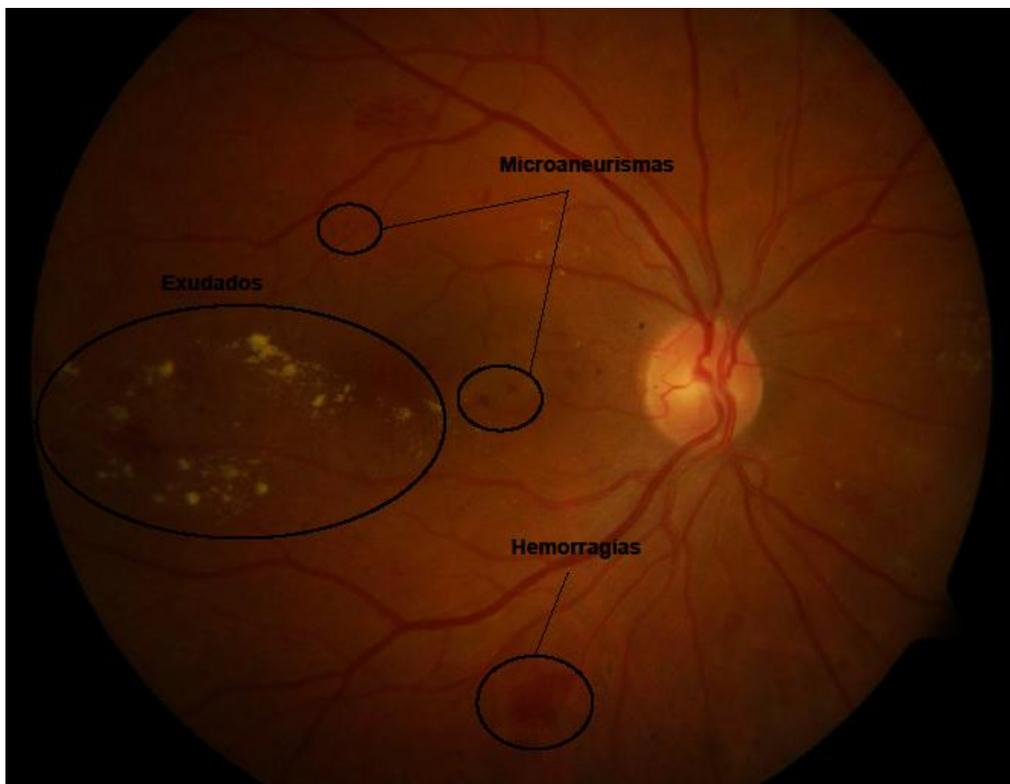


Figura 6. Patrones anómalos presentes en la RD

Además de estas lesiones, son apreciables otras distintas a las aquí mencionadas, pero en nuestro trabajo y en los casos clínicos con los que tratamos únicamente se han tenido estos en cuenta.

Existen dos tipos de Retinopatía Diabética: **Retinopatía Diabética No Proliferativa (RDNP)** y **Retinopatía Diabética Proliferativa (RDP)**.

La RDNP es el estadio más temprano de la patología y está caracterizada por la aparición de los patrones anteriormente expuestos. Normalmente la RDNP progresa hasta convertirse en proliferativa (RDP) ya que muchos vasos sanguíneos en la retina se ocluyen, obstaculizando el correcto flujo sanguíneo. La respuesta de la retina a esto es el crecimiento de nuevos vasos sanguíneos (**neovascularización**). El problema es que estos nuevos vasos son anormales y no proporcionan el flujo sanguíneo correcto a la retina. Su fragilidad suele producir hemorragias bloqueando la visión y propiciando que aparezcan sombras en ella.

La RDP puede afectar tanto a la visión central como a la periférica, por lo que es mucho más agresiva que la RDNP.

1.8 Estado del arte en detección de hemorragias.

En la literatura referente a detección de lesiones oscuras y, en particular, de hemorragias, y tal y como se presenta en [7] se observa que los principales métodos presentados siguen siempre un proceso similar.

En primer lugar una fase de preprocesado de la imagen. A continuación se eliminan las características principales de la imagen que no son de interés en la detección y que pueden inducir a error (vasos sanguíneos, disco óptico,...). Por último se extraen los candidatos de la imagen o posibles hemorragias y se clasifican de manera que se obtengan los mejores resultados posibles.

De esta manera, podemos encontrar diferentes propuestas que se desvían ligeramente de este esquema o que no lo siguen de manera estricta.

Otro aspecto importante a tener en cuenta es la eliminación de características que no sean propiamente hemorragias como, por ejemplo, los vasos sanguíneos o el disco óptico, ya que pueden dar lugar a confusiones en el algoritmo de clasificación. La mayoría de los métodos utilizados con este objetivo se basan en técnicas de umbralización. [8]

Acharya et al.[9] utilizaron técnicas morfológicas. En primer lugar invirtieron el nivel de intensidad de la imagen y a continuación realizaron una ecualización adaptativa utilizando el histograma. Una vez realizado esto, mediante una serie de ‘aperturas’ se incrementa la intensidad de los vasos respecto del fondo de ojo y se binariza definiendo un umbral para la posterior eliminación de los vasos. En la misma línea, Marwan y Eswaran [10] eliminaron el disco óptico utilizando filtros de mediana. El problema de esta serie de operaciones, sobre la imagen original es que los vasos y los microaneurismas se parecen mucho y, eliminar los vasos puede llevar a eliminar también los microaneurismas. Es por esto, que en el algoritmo que proponemos, no se haya realizado este tipo de preprocesado.

No obstante, donde reside la verdadera importancia del proceso de detección de hemorragias es en la extracción de candidatos y en la clasificación.

Así pues, Hatanaka et al. [11] utilizaron un análisis de densidad para extraer las posibles hemorragias (candidatos) Una vez realizado esto se clasifican los candidatos utilizando tres clasificadores basados en distancias Mahalanobis.

Köse et al.[12] utilizaron otro enfoque. En vez de extraer las regiones patológicas, intentaron extraer las regiones sanas ya que la textura de las regiones sanas tiene mucha menos varianza.. Zhang et al. [13] propusieron un algoritmo basado en la estimación del fondo. Se consideró el primer plano las zonas de interés a detectar. Para ello aplicaron un esquema de múltiples umbrales basado en los métodos de umbral de histéresis [14].

Los métodos que solo utilizan como información el color son insuficientes. Es por ello que Zhang y Chutape [15] localizaron las hemorragias en la región de interés, después de una fase de preprocesado, mediante SVM (Support Vector Machines) utilizando como vector de características las extraídas con análisis bidimensional de componentes principales (2DPCA).

Kande et al.[16] propusieron un método para detectar lesiones oscuras donde primero aplican umbralización basada en entropía relativa para segmentar las lesiones después de preprocesar la imagen. Para extraer los vasos se usó filtrado ‘top-hat’ y SVM para clasificar las lesiones.

Narasimhan et al. [17] propusieron un método basado en operaciones de filtrado. La transformada ‘top-hat’ se aplica al canal verde. Una reconstrucción geodésica recupera las características lineales. Una técnica de umbralización mejora la imagen y la técnica de ‘region-growing’ se utiliza para extraer características y así detectar microaneurismas y hemorragias.

Esmaili et al. [18] utilizaron un algoritmo basado en aproximación multiescala llamado Digital Curvelet Transform (DCUT) donde se ajustaron una serie de coeficientes de la transformada para que especialmente las hemorragias fueran fácilmente distinguibles.

García et al. [19] utilizaron una red neuronal para segmentar automáticamente los candidatos a hemorragias. Las características del color y de forma se extraen con la ayuda de regresión logística y mediante cuatro clasificadores como son Multilayer Perceptron (MLP), Radial Basis Function (RBF), Support Vector Machine (SVM) y Majority Voting (MV) se segmenta la imagen.

Sinthanayothin et al.[20] también utilizar el concepto de *Region Growing* pero con otro enfoque. En primer lugar se extraen las características principales, tales como disco óptico, fovea o vasos ópticos. A continuación, mediante un operador introducido en el artículo referenciado y conocido como '*Moat Operator*' se extraen las características relacionadas con las hemorragias seguido por la aplicación de una umbralización de intensidad adaptativa. Es entonces cuando los candidatos a hemorragias pasan por un proceso de segmentación por *región growing* recursivo.

Capítulo 2. Objetivos.

El principal objetivo del proyecto en el cual se enmarca el presente trabajo de fin de grado, es el de conseguir desarrollar un sistema robusto, eficiente y fiable para el diagnóstico de las principales causas de ceguera en la actualidad, entre las que se encuentra la retinopatía diabética (objeto del presente trabajo).

Para ello, se puede sintetizar nuestro trabajo en los siguientes objetivos más específicos:

- Estudio de la retinopatía diabética y de todos los signos clínicos que se manifiestan en la evolución de la enfermedad y que permiten diagnosticarla.
- Análisis del estado del arte acerca de las técnicas de detección de lesiones oscuras en imagen de fondo de ojo.
- Estudio de descriptores de textura y forma que permitan caracterizar patrones anómalos en el fondo de ojo con la mayor precisión posible.
- Estudio de la técnica Support Vector Machine, para la detección local de tejido retiniano que manifieste signos de retinopatía diabética. Concretamente, en este trabajo se desea clasificar automáticamente entre tejido sano y tejido con lesión oscura.
- Diseño e implementación de un método que aprovechando la información obtenida a nivel local sea capaz de discriminar en sanas o patológicas las imágenes de fondo de ojo (casos clínicos) que se deseen analizar.

Capítulo 3. Metodología de trabajo

3.1 Base de datos ‘Diaretdb1’

En el sistema propuesto en el presente trabajo, es necesario contar con un conjunto de imágenes de una calidad lo suficientemente alta para que los resultados sean fiables. Es por ello que se ha empleado la base de datos *DiaretDb1* para la validación de los algoritmos presentados en este TFG.

La base de datos *Diaretdb1* es accesible de manera libre a través de la página web (<http://www.it.lut.fi/project/imageret/diaretdb1>) [21]. Esta base de datos consta de 89 imágenes de fondo de ojo de las cuáles 84 contienen como mínimo signos leves de retinopatía diabética no proliferativa.

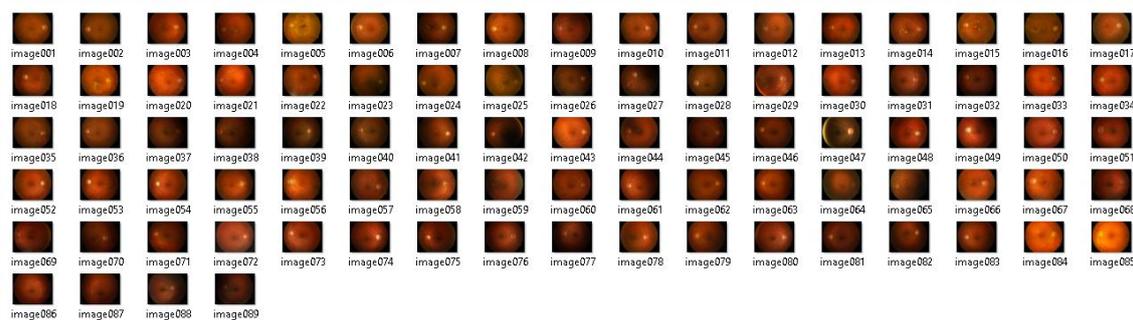


Figura 7. Imágenes de la base de datos diaretdb1

Todas las imágenes (RGB) se encuentran en el formato digital PNG y tienen unas dimensiones de 1500x1152 píxeles con una profundidad de 24 bits (8 para cada color). Las imágenes se tomaron sin tener ninguna información a priori (tal como la edad o el sexo) de los pacientes y todas fueron adquiridas con el mismo retinógrafo digital, empleando un ángulo de field of view (FOV) de 50 grados pero con una configuración de imagen (intensidad lumínica, apertura, etc.) distinta. No obstante las aberraciones ópticas (dispersión, distorsión, curvatura de campo, etc) son constantes para todas las imágenes. Así pues, la varianza introducida en las imágenes puede ser considerada muy pequeña.

3.1.1 Ground truth

Como comentamos, es preciso disponer de un material de calidad para poder testear cualquier sistema diseñado apropiadamente. Y, dentro de este material, es imprescindible tener lo que es conocido como ‘*ground truth*’ o **verdad absoluta**. En particular, en esta base de datos, se proporciona, para cada una de las imágenes de fondo de ojo, unas imágenes con cuatro niveles distintos de gris, en las cuales 4 especialistas en oftalmología, han marcado cualquier signo patológico observado.

Más concretamente, estos profesionales se han encargado de marcar en cada imagen los exudados duros o blandos, los microaneurismas y las hemorragias avistadas.

A continuación se adjunta un ejemplo de lo que comentamos. Concretamente se muestra el *ground truth* relativo a la imagen 008 de la base de datos Diaretdb1.

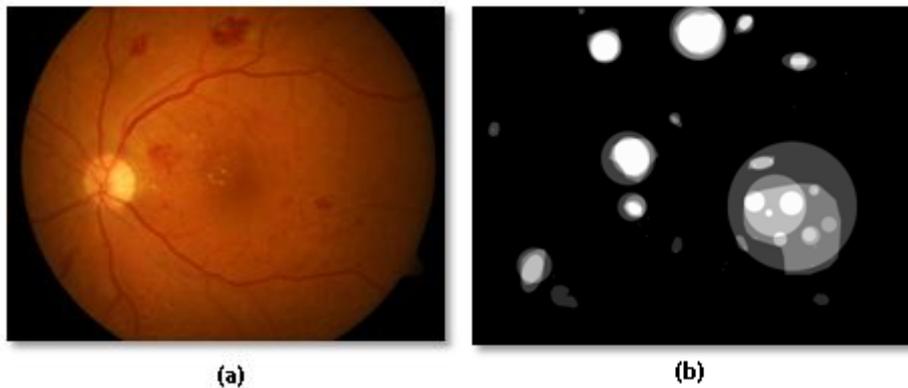


Figura 8. a) Imagen 008 (diaretdb1) Original . b) Ground truth (hemorragias) en escala de grises

Esta máscara indica, siempre según los expertos, en que parte de la imagen de fondo de ojo se ha detectado la lesión que la máscara identifica (en nuestro caso hemorragias)

Se pueden observar cuatro niveles de gris (uno por cada especialista), que se corresponden con las distintas anotaciones de cada profesional. Con el objetivo de generar una máscara binaria que indique si un píxel es sano o patológico, se define un nivel de confianza (porcentaje mínimo de detección por los distintos profesionales). Para este trabajo el nivel de confianza se ha fijado en el 75%. Así, todo píxel que supere este porcentaje será considerado como patológico, y todo aquel que no lo supere, se considerará como sano. En otras palabras, un píxel será marcado como patológico si res de los cuatro expertos coinciden en que dicho píxel es lesión. De lo contrario, el píxel será considerado sano.

De este proceso de binarización se obtiene un *groundtruth* de referencia tal y como muestra la figura 9:



Figura 9. Máscara binarizada de la figura 7b

3.2 Máscaras binarias

En este subapartado expondremos las distintas máscaras binarias que utilizaremos en nuestro sistema. Estas máscaras serán de gran utilidad a la hora de proporcionar robustez y eficiencia al algoritmo propuesto ya que, en ciertos casos, eliminarán regiones de la imagen del cálculo y acelerarán el proceso de clasificación y descartarán zonas de las imágenes de fondo de ojo que no sean propiamente zonas retinianas.

3.2.1 *Máscara del disco óptico*

Esta máscara (Figura 10) identifica el disco óptico en la región correspondiente de la imagen original. Esto nos será de gran ayuda ya que eliminará del análisis una región que, por su forma podría llegar a ser conflictiva.

El método utilizado para la segmentación de esta estructura, está basado en morfología matemática y Análisis de Componentes Principales (ACP) [22]. Dicho método utiliza transformaciones geodésicas y función de distancias generalizadas (GDF), entre otras.

Cabe destacar que el método está automatizado, es decir, únicamente introduciendo las imágenes originales será suficiente. Si bien esto es cierto, existe la posibilidad de utilizar un ‘modo semiautomático’ para mayor obtener una mayor precisión donde únicamente tendremos que indicar el centro aproximado del disco óptico.

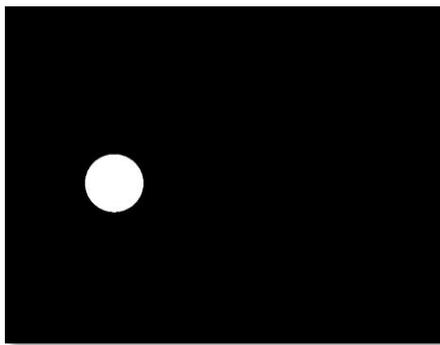


Figura 10. Máscara del disco óptico de la imagen 008 (diaretdb1)

3.2.2 *Máscara externa o del FOV (Field of View)*

Esta máscara (Figura 11) delimita la parte de la imagen referida propiamente a la retina, eliminando la parte de la imagen asociada a los equipos de adquisición de imágenes de fondo de ojo. Se puede extraer fácilmente realizando una umbralización del canal rojo.

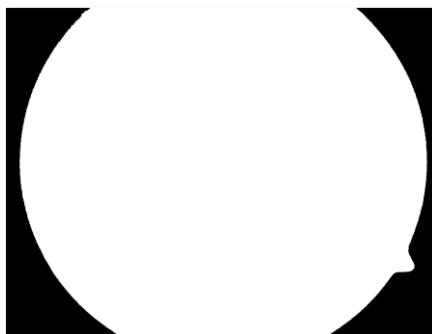


Figura 11. Máscara del FOV de la imagen 008 (diaretdb1)

3.2.3 Máscara de los vasos sanguíneos

Por último adjuntamos una máscara (Figura 12) que, si bien no es estrictamente necesaria, nos proporciona cierta flexibilidad para realizar un preprocesado de la imagen original y ‘eliminar’ de la imagen original los fragmentos de la imagen que corresponden a los vasos sanguíneos. Esta máscara se obtuvo aplicando un algoritmo basado en morfología matemática que también incluye una evaluación de la curvatura para la correcta detección del árbol vascular retiniano. [23]

En esta máscara se puede observar la distribución de los vasos sanguíneos que parten del disco óptico. En particular, esta máscara es utilizada para el cálculo de ciertos descriptores locales que se explicarán en el siguiente capítulo, provocando que las zonas asociadas a los vasos sanguíneos no entren en el cómputo de dichos descriptores.

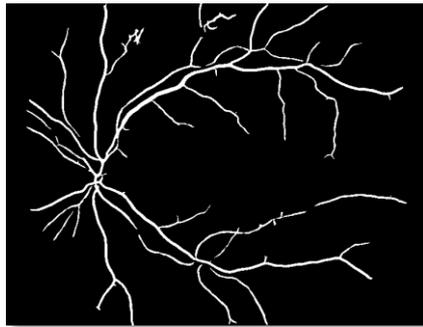


Figura 12. Máscara de los vasos sanguíneos de la imagen 008 (diaretdb1)

3.3 Espacio de color RGB.

El espacio de color RGB (por sus siglas en inglés Rojo, Verde y Azul) define la composición de una imagen a partir de una síntesis aditiva de los tres canales presentes, esto es, se representa el color de cada píxel a partir de la suma de los tres canales, teniendo en cuenta su intensidad, que suele variar en un rango entre [0,255]. [24]

Aunque suele ser el más común, existen otros espacios de color con los que se puede trabajar, no obstante en el trabajo aquí expuesto trabajaremos con este espacio RGB. En particular, se trabajará únicamente con el **canal G** ya que, de todos los canales, es el que mejor contraste presente entre las lesiones que se querrán detectar y el fondo de ojo, como se puede observar en la figura 12.

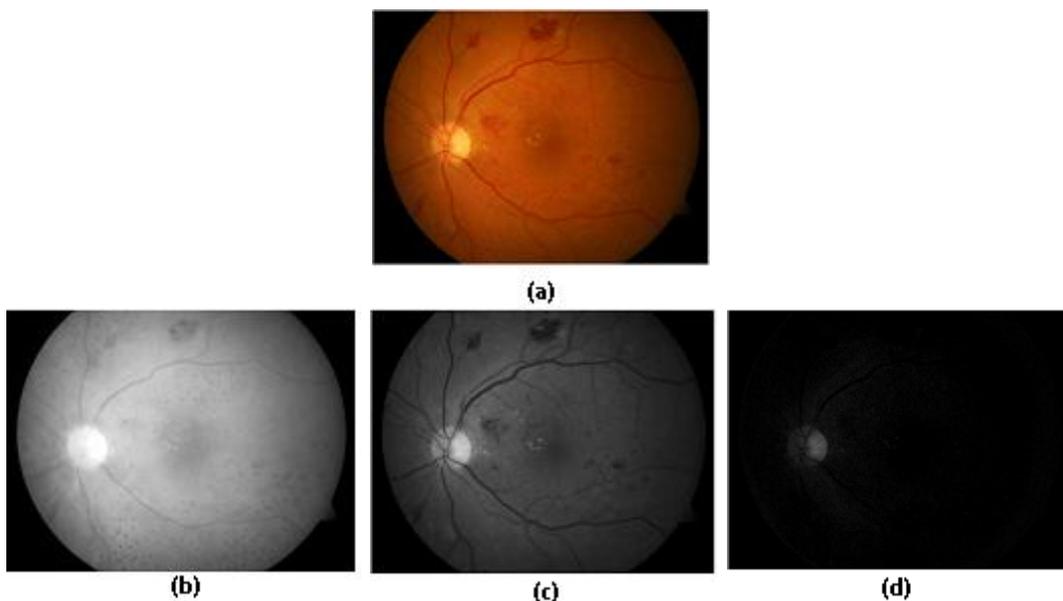


Figura 13. a) Imagen 008 RGB. b) Canal R. c) Canal G. d) Canal B.

3.4 Reparto de Tareas y Diagrama Temporal.

A continuación se detallan las distintas tareas realizadas para la consecución de los objetivos expuestos en el capítulo 2:

1. Estudio del estado del arte para familiarizarse con las técnicas utilizadas.
2. Estudio y análisis de los distintos operadores morfológicos utilizados así como de las distintas técnicas de Machine Learning aplicadas.
3. Obtención de las máscaras de los casos sanguíneos del campo de visión y del disco óptico para la base de datos diaretdb1
4. Adaptación del método propuesto en el apartado 4.5 para la detección de lesiones oscuras.
5. Realización de las distintas simulaciones para obtener los resultados utilizando los distintos operadores propuestos.
6. Implementación del algoritmo de clasificación a nivel de imagen.
7. Obtención de los resultados de clasificación a nivel de imagen.
8. Redacción de la memoria.

| | Enero | Febrero | Marzo | Abril | Mayo | Junio | Julio | Agosto |
|---|-------|---------|-------|-------|------|-------|-------|--------|
| 1 | | | | | | | | |
| 2 | | | | | | | | |
| 3 | | | | | | | | |
| 4 | | | | | | | | |
| 5 | | | | | | | | |
| 6 | | | | | | | | |
| 7 | | | | | | | | |
| 8 | | | | | | | | |

Tabla 1. Diagrama Temporal

Capítulo 4. Detección de lesiones oscuras

4.1 Introducción

Una vez se han expuesto las bases teóricas sobre las que se asienta este proyecto, así como la metodología que se ha seguido durante todo el proceso de creación del algoritmo estamos en condiciones de proceder a presentarlo. No obstante, es necesario explicar ciertos aspectos que se han utilizado y sin los cuáles no se entendería el diseño en su totalidad.

La peculiaridad de nuestro trabajo reside en que, para la extracción de las características de la imagen, aspecto imprescindible en cualquier tipo de labor de clasificación de imágenes, se han utilizado descriptores locales de textura y forma, tales como los *local binary patterns* o granulometría. Este concepto será explicado en profundidad en las siguientes subsecciones, no obstante, es importante tenerlo presente desde un inicio ya que, a diferencia de lo que mucha de la literatura existente propone, en nuestro trabajo no se realiza ningún tipo de segmentación de imagen.

Hemos de decir que la implementación del algoritmo, junto con la obtención de los resultados arrojados de las distintas simulaciones, se ha realizado en el entorno MATLAB.

4.2 Local Binary Pattern (LBP)

Como hemos comentado anteriormente, nuestro trabajo se particulariza por estar basado en métodos locales de análisis de texturas. En este apartado se describirán los fundamentos de uno de los dos descriptores de texturas empleados en la descripción de los patrones anómalos que presenta la retina patológica.

Los **Local Binary Patterns** (LBP) [25] [26] son descriptores de texturas basados en la cuantificación de las variaciones locales de la intensidad de cierto pixel con respecto a las de su vecindario.. A partir de imágenes en escala de grises es posible extraer determinados patrones binarios locales que se convierten en una herramienta fundamental en el reconocimiento de texturas. La estabilidad respecto a cambios monotónicos en el nivel de gris hace de éstos unos descriptores muy robustos. Además, se trata de un descriptor muy simple y computacionalmente muy eficiente.

En aplicaciones biomédicas (y en otros muchos campos), cuando se aborda el análisis de texturas de imagen, ha de tenerse en cuenta que existen posibles variaciones como la orientación, la escala u otros muchos aspectos visuales debidos a distintas causas como, por ejemplo, diferencias de iluminación en el instrumental de captación de imágenes. De aquí que este no sea un problema trivial y se trate de una difícil tarea de clasificación. Este hecho exige contar con métodos descriptivos invariantes a cambios en la escala de grises como son los LBP.

Los LBP extraen información para cada píxel de la imagen a partir de un conjunto de píxeles vecinos definidos a priori de forma arbitraria, mediante la regulación de una serie de parámetros que también se verán más tarde.

A continuación, procederemos a explicar cómo es el funcionamiento de los LBP y cómo se consigue la tan deseada invarianza respecto a la variación monótonica en el nivel de grises y respecto de la rotación.

El operador LBP básico se basa, tal y como hemos mencionado, en obtener una matriz de píxeles de tamaño igual a la imagen original en la cual se codificará información no solo del píxel implicado en cuestión si no de sus vecinos. Hay que aclarar que el concepto de vecindad es arbitrario y viene definido por los parámetros P y R . El parámetro P define el número de actores que participan en la vecindad, mientras que el parámetro R define el radio o distancia a la que se encuentran dichos vecinos.

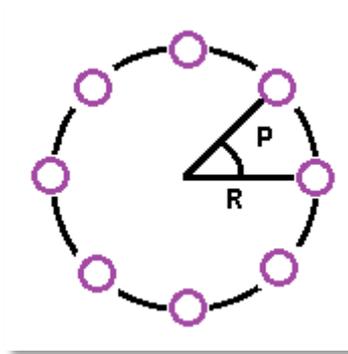


Figura 14. Parámetros LBP

Se puede apreciar que, debido a la definición de **vecindad**, puede haber ciertos valores vecinos que no caigan exactamente en el centro de un píxel por lo que su valor será obtenido mediante interpolación.

Una vez definidos estos parámetros pasaremos a codificar, píxel a píxel, la imagen. Para ello, habrá de definirse un orden de vecinos, para ir recorriendo toda la imagen. Un ejemplo de este orden se incluye en la Figura 15:

| | | |
|---|---|---|
| 1 | 8 | 7 |
| 2 | 0 | 6 |
| 3 | 4 | 5 |

Figura 15. Ejemplo de orden de comparación de vecinos

La anterior figura es un ejemplo de un posible orden que se seguirá para comparar cada vecino con el píxel central en cuestión. Siendo el píxel central el marcado como '0' en la imagen se obtendrá una cadena binaria por medio de la siguiente regla: si el valor de la intensidad del nivel gris del píxel vecino que se está comparando es mayor que el del central se marcará como '1', en caso contrario se le asignará un '0' al código. Esto se puede expresar en forma de ecuación de la siguiente manera:

$$LBP_{P,R}(i,j) = \sum_{p=0}^{P-1} s(g_p - g_c) \cdot 2^p, \quad s(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \quad (4.1)$$

Donde P es el número de vecinos de radio R , g_c es el valor de gris del píxel central considerado y g_p el valor de gris de cada vecino.

Cabe destacar que tal y como indica la ecuación 4.1, lo que se codifica es el signo de la diferencia entre el píxel central y el vecino en cuestión, ya que aunque es posible una pequeña pérdida de información, esta es la forma de conseguir estabilidad ante cambios monótonicos en

el nivel de gris de la imagen. Este signo, convertido a posteriori en '1' o '0' se multiplica por un factor de 2^p donde p indica el orden predefinido en el conjunto de vecinos.

También es importante remarcar que, para aquellos píxeles situados en el borde de la imagen también ha de definirse una táctica de codificación. Se puede asignar el valor del píxel más próximo o, simplemente, descartar esta zona en la transformación *LBP*.

Para una mayor comprensión, adjuntamos en la Figura 16 un esquema visual del proceso:

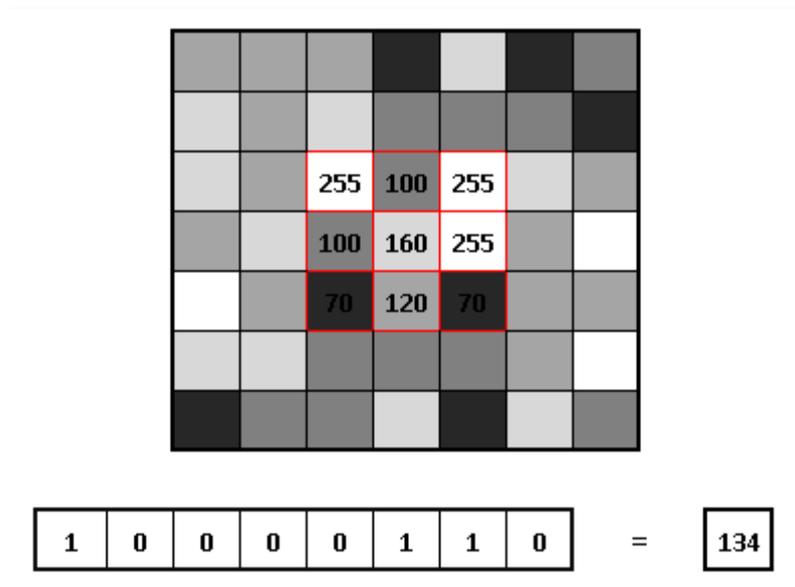


Figura 16. Ejemplo de codificación LBP para un píxel dado según el orden de la Figura 14

En la figura 15 se muestra el proceso de codificación anteriormente expuesto. La obtención de estos códigos binarios para cada píxel da nombre a este descriptor de textura (*local binary pattern*).

Además del LBP aquí expuesto, existen muchas variantes propuestas en la literatura. No obstante, únicamente expondremos las que son de interés para nuestro trabajo: LBP rotacionalmente invariante (*LBPri*) y el LBP uniforme o *LBP(riu2)*[*Rotation-Invariant Uniform*]. [25]

El LBP rotacionalmente invariante es una variante del LBP básico. Se basa en el hecho de que, al existir ciertos patrones que se pueden obtener mediante la rotación de otros, se puede reducir el número de patrones utilizados y utilizar un único patrón para todas las rotaciones de un esquema LBP. Un ejemplo más visual de lo aquí expuesto es, si se observan los patrones binarios 11000011 y 11100001 se aprecia que son distintas rotaciones de una misma cadena binaria normalizada como 00001111. Con el objetivo de evitar este efecto de rotación, se propone el operador $LBP_{P,R}^{ri}$ definido de la siguiente forma:

$$LBP_{P,R}^{ri} = \min\{ROR(LBP_{P,R}, t) \mid t = 0, 1, \dots, P - 1\} \quad (4.2)$$

El término uniforme se refiere a que en vez de utilizar 256 patrones binarios que se salen de las posibles combinaciones de una cadena binaria de 8 bits) se agrupan en 59. Estos patrones uniformes son aquellos que tienen como máximo dos transiciones '1/0' o viceversa en su cadena binaria. Asimismo, cada patrón binario tendrá su propio código representativo mientras que todos los 'no uniformes' se definirán con un único patrón. En el caso de descripción de texturas este tipo de operador es muy interesante ya que permite observar en su histograma de manera bastante clara los patrones básicos de las dichas texturas ya que representan alrededor del 90% de los patrones presentes en las texturas de los experimentos realizados en el artículo propuesto.

La textura de imagen se considera como un fenómeno bidimensional, que está caracterizado por dos propiedades ortogonales: la estructura espacial y el contraste. Mientras que la rotación afecta a la estructura espacial pero no lo hace al contraste, la variación de grises afecta al contraste pero no a la estructura. Por lo tanto, si queremos conseguir un operador invariante a cambios monotónicos en el nivel de gris, se tendrá en cuenta únicamente la estructura espacial, mientras que si se quiere un operador invariante a la rotación se hará lo propio con el contraste.

El operador $LBP_{P,R}^{riu2}$ obtiene una medida muy buena de la estructura espacial pero descarta el contraste. Para paliar este efecto se puede combinar el análisis de este operador con medidas de varianza invariantes a la rotación.

Por este motivo, es común añadir alguna medida de contraste como anteriormente se comentaba. En particular, se define el VAR (*Rotational Invariant Local Variance*) como:

$$VAR_{P,R}(i,j) = \frac{1}{P} \sum_{p=0}^{P-1} (g_p - \mu)^2, \quad \mu = \frac{1}{P} \sum_{p=0}^{P-1} g_p \quad (4.3)$$

Para combinar la información de contraste proporcionada por este operador y la información estructural proporcionada por LBP (LBP^{riu2}) se ha propuesto [25] la distribución conjunta $LBP_{P,R}/VAR_{P,R}$. El problema de esta distribución es que $VAR_{P,R}$, a pesar de ser poderoso, tiene algunas limitaciones como son que se necesita de un proceso de entrenamiento o que, al proporcionar valores continuos, se ha de cuantificar y para ello se ha de decidir el número de *bins* que tendrá el histograma a extraer (proceso que no es para nada trivial).

Es por ello, que en [27] se propone un descriptor LBPV que soluciona los problemas comentados anteriormente. Aquí la varianza $VAR_{P,R}$, debido a que el cálculo del histograma LBP no se ve influido para nada por la varianza, se usa como un peso adicional para calcular la contribución del código LBP en el dicho histograma, asignando a las regiones más frecuentes, mayores pesos. Por lo tanto, el histograma LBPV se calcula como:

$$LBPV_{P,R}(k) = \sum_{i=1}^N \sum_{j=1}^M w(LBP_{P,R}(i,j), k), k \in [0, K] \quad (4.4)$$

$$w(LBPV_{P,R}(i,j), k) = \begin{cases} VAR_{P,R}(i,j), & LBP_{P,R}(i,j) = k \\ 0, & otherwise \end{cases} \quad (4.5)$$

Gráficamente, se puede imaginar que LBPV es como una proyección integral a lo largo del eje VAR. Como LBPV es una simplificación de la distribución LBP/VAR, la longitud de sus vectores de características es mucho menor. Además, el descriptor LBPV libera la necesidad de un proceso de entrenamiento ya que no necesita la cuantificación de la que hablábamos.

De esto se extrae la importancia de la obtención del histograma LBPV ya que, una vez realizado esto, con el número de *bins* (valores en el eje x) elegido de manera arbitraria (p.e. 64 o 128) tendremos ya el vector de características para cada trozo de imagen analizado y que será entrada del clasificador que posteriormente se explicará.

Para asentar lo explicado anteriormente, adjuntamos, en las figuras 17 y 18 algunas imágenes que representan lo expuesto en esta sección

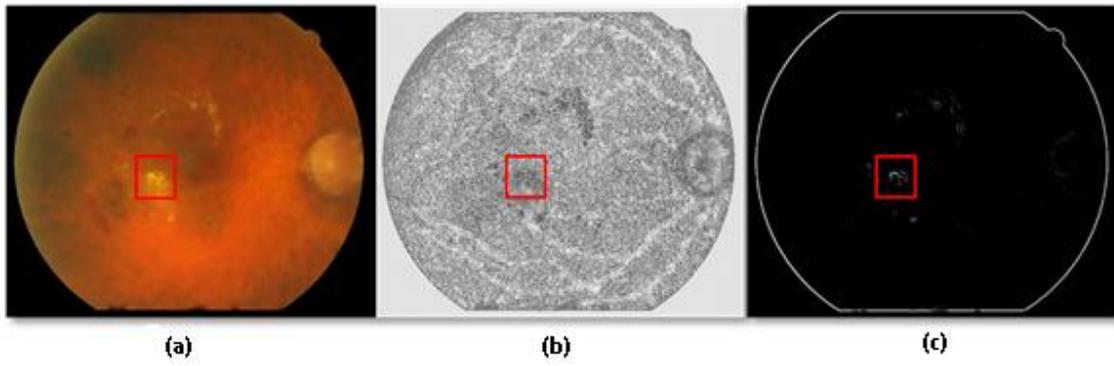


Figura 17 a) Imagen Original. b) Imagen LBP. c) Imagen VAR.

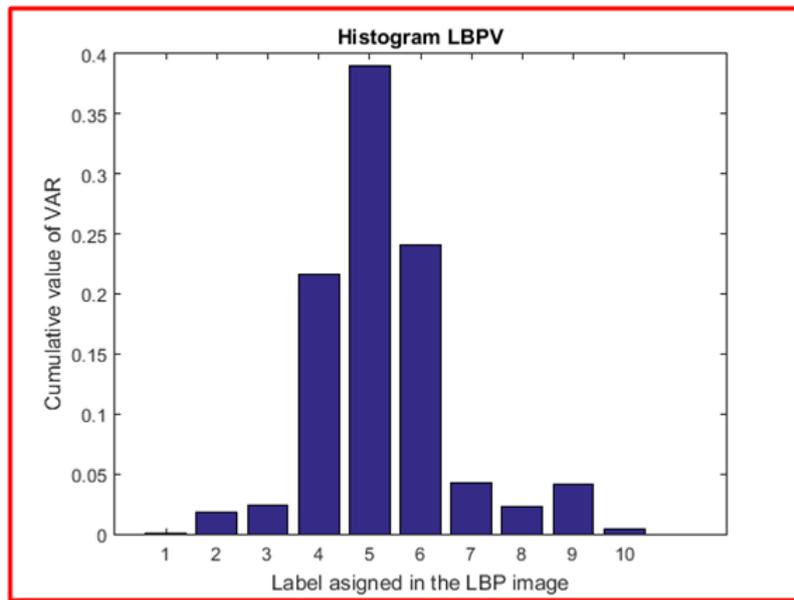


Figura 18. Histograma LBPV

4.3 Granulometría

Una de las técnicas más interesantes basadas en morfología matemática es la granulometría. La granulometría y algunas métricas que se extraen de ella se pueden aplicar a un gran rango de tareas como segmentación, extracción de características (*feature extraction*), caracterización de texturas y estimación de tamaños entre otros.

Aplicando una serie de operaciones morfológicas (cierres y aperturas), con un tamaño incremental del elemento estructurante (SE), se pueden obtener distribuciones de tamaños de diferentes objetos en una imagen.

Esta técnica se puede procesar en imágenes binarias o en escala de grises proporcionando firmas que pueden usarse para alimentar a un clasificador capaz de reconocer diferentes patrones o discriminar entre varias texturas.

4.3.1 Operadores morfológicos básicos

Las operaciones morfológicas simplifican las imágenes conservando sus características de forma más importantes o descriptivas. Los principales operadores morfológicos son: la dilatación y la erosión. A continuación se expone su formulación matemática.

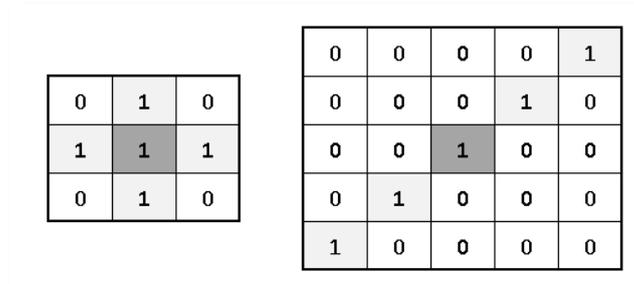


Figura 19. Elemento estructurante en forma de disco con radio 1 (derecha) y de línea de longitud 5 (izquierda)

Teniendo en cuenta lo anteriormente expuesto, las operaciones de dilatación y erosión para imágenes en escala de grises se definen como:

$$\text{Dilatación: } [\delta_B(f)](x) = \max_{b \in B(x)} f(x + b) \quad (4.6)$$

$$\text{Erosión: } [\varepsilon_B(f)](x) = \min_{b \in B(x)} f(x + b) \quad (4.7)$$

El objetivo de estas operaciones es, respectivamente, expandir objetos claros u oscuros. A modo de ejemplo visual, se adjuntan las siguientes imágenes:

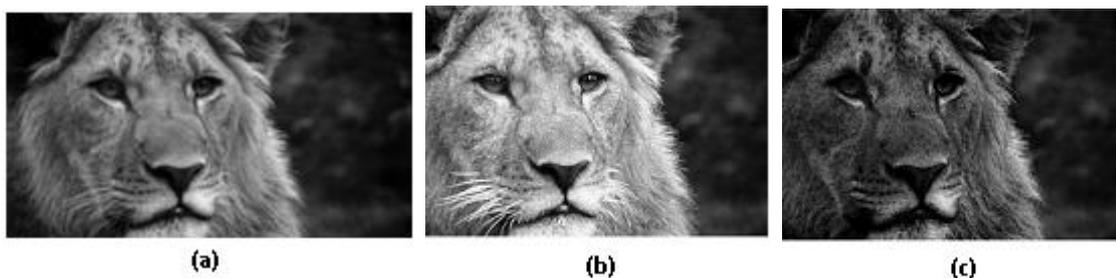


Figura 20. a) Imagen original. b) Imagen dilatada. c) Imagen erosionada. (Utilizando SE en forma de disco de radio 4)

Mediante la combinación de los operadores definidos anteriormente, se pueden obtener unos nuevos filtros morfológicos. A continuación exponemos las aperturas y los cierres:

$$\text{Apertura: } \gamma_B(f) = \delta_B(\varepsilon_B(f)) \quad (4.8)$$

$$\text{Cierre: } \phi_B(f) = \varepsilon_B(\delta_B(f)) \quad (4.9)$$

Las aperturas permiten filtrar elementos claros mientras que los cierres hacen lo propio con los elementos oscuros. En la figura 15 se observa el resultado de aplicar los operadores a la imagen propuesta en la figura 14:

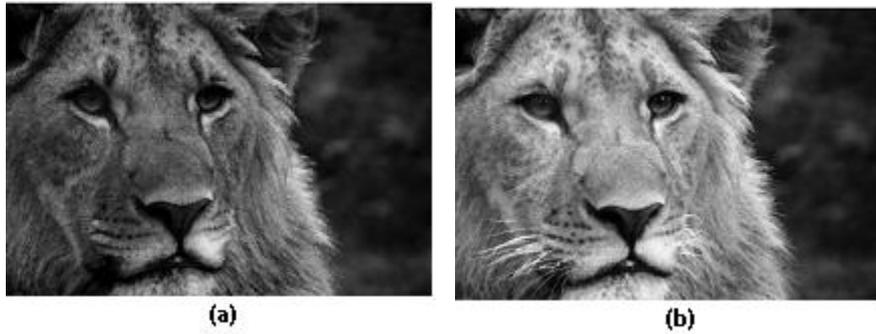


Figura 21. a) Imagen aplicando apertura. b) Imagen aplicando cierre (Utilizando SE en forma de disco de radio 4)

No obstante, en ocasiones es necesaria más de una operación morfológica para obtener el resultado deseado. De aquí nace la necesidad de obtener lo que conocemos como perfiles morfológicos.

4.3.2 Perfiles morfológicos

Un perfil morfológico es una herramienta para modelar cierta información estructural extrayendo una serie de características de los objetos de una imagen gracias a una serie de operadores morfológicos.

La imagen de entrada se procesa de manera iterativa con los operadores morfológicos elegidos en función de cuáles son las características a detectar. Evaluar como las estructuras de la imagen interactúan con los operadores proporcionará la información deseada.

Se define un *perfil morfológico de aperturas* [28] como:

$$\Pi_{\gamma}(f) = \left\{ \Pi_{\gamma\lambda} : \Pi_{\gamma\lambda} = \gamma^{\lambda}(f), \forall \lambda \in [0, \dots, n] \right\} \quad (4.10)$$

Esto se traduce en un procesado de la imagen mediante una serie de aperturas con un elemento estructurante de tamaño incremental (λ). El tamaño del SE se incrementa mediante cierto paso previamente definido hasta llegar a un valor máximo. Estas operaciones iterativas dan lugar a n imágenes adicionales a la original que pueden ser almacenadas en forma de volumen.

Este volumen, obtenido mediante una serie de aperturas sucesivas, proporciona información sobre como los elementos brillantes o claros de la imagen van desapareciendo, gracias a esto, será posible detectarlos y relacionarlos con su tamaño.

De la misma forma que para las aperturas, se define el *perfil morfológico de cierres* tal que:

$$\prod_{\emptyset}(f) = \left\{ \prod_{\emptyset\lambda} : \prod_{\emptyset\lambda} = \emptyset^{\lambda}(f), \forall \lambda \in [0, \dots, n] \right\} \quad (4.11)$$

En este caso, es válida la misma explicación que la anterior. La única diferencia es que en este perfil se utilizan como operadores morfológicos los cierres. Esto hace que lo que se observe en volumen generado es como se van filtrando los elementos oscuros.

Para formar un perfil morfológico completo, lo único que es necesario, una vez se han obtenido el *perfil morfológico de aperturas* y el *perfil morfológico de cierres*, es concatenar ambos perfiles.

Si se tiene en cuenta que de las $n+1$ imágenes que se generan en cada perfil, para un tamaño del elemento estructurante 0 corresponde a procesar un elemento estructurante de tamaño 0 sobre la imagen, lo cual es equivalente a hablar de la imagen original. Al procesarse el elemento estructurante de tamaño 0 en ambos perfiles, se tendrán dos imágenes correspondientes a la imagen original, y por tanto, con tal de no duplicar la información, solo se incluye una de las dos. De aquí resulta un perfil morfológico que consta de un conjunto de $2n+1$ imágenes (n imágenes del *perfil de cierres*, 1 imagen original y n imágenes del *perfil de aperturas*) y se puede expresar de la siguiente manera:

$$\prod(f) = \left\{ \prod_i : \left(\begin{array}{ll} \prod_i = \prod_{\emptyset\lambda}, & \text{con } \lambda = (n - i + 1), \quad \forall i \in [1, n] \\ \prod_i = \prod_{\gamma\lambda}, & \text{con } \lambda = (i - n - 1), \quad \forall i \in [n + 1, 2n + 1] \end{array} \right) \right\} \quad (4.12)$$

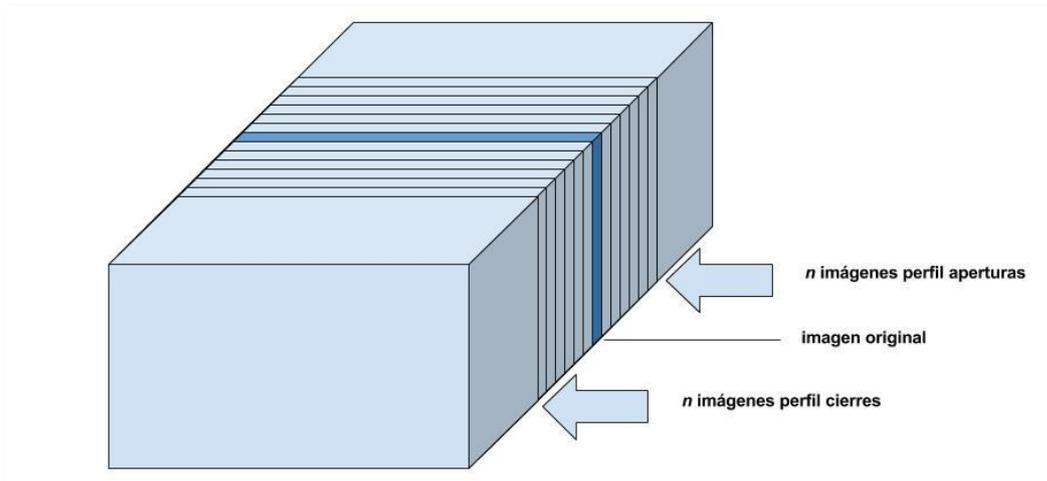


Figura 22. Esquema visual del perfil morfológico

Si se calcula la derivada de un perfil morfológico se obtiene un *perfil morfológico diferencial* que se define como:

$$\Delta(f) = \left\{ \Delta_i : \left(\begin{array}{ll} \Delta_i = \Delta_{\gamma\lambda}, & \text{con } \lambda = (n - i + 1), \quad \forall i \in [1, n] \\ \Delta_i = \Delta_{\emptyset\lambda}, & \text{con } \lambda = (i - n - 1), \quad \forall i \in [n + 1, 2n] \end{array} \right) \right\} \quad (4.13)$$

Dónde aparece un *perfil diferencial de aperturas* (Δ_γ) y un *perfil diferencial de cierres* (Δ_\emptyset) definido respectivamente como:

$$\Delta_\gamma(f) = \left\{ \Delta_\lambda: \Delta_{\gamma\lambda} = \prod_{\gamma(\lambda-1)} - \prod_{\gamma\lambda}, \forall \lambda \in [1, n] \right\} \quad (4.14)$$

$$\Delta_\emptyset(f) = \left\{ \Delta_{\emptyset\lambda}: \Delta_{\emptyset\lambda} = \prod_{\emptyset\lambda} - \prod_{\emptyset(\lambda-1)}, \forall \lambda \in [1, n] \right\} \quad (4.15)$$

Un *perfil morfológico diferencial* almacena el residuo de las diferencias de las sucesivas transformaciones aplicadas a la imagen original con lo que consta de $2n$ imágenes (al ser diferencias se elimina una capa del perfil). Este tipo de perfiles, que se suelen denominar como *pattern spectrum* y se pueden normalizar dividiendo entre el sumatorio del valor de los píxeles de la imagen original, es más práctico ya que los componentes principales son más evidentes.

Por último, cabe comentar que, como ya se ha comentado en esta sección, en la implementación del sistema propuesto se computa este tipo de información a nivel local y no a nivel de imagen.

4.4 Support Vector Machine (SVM)

4.4.1 Desarrollo visual del SVM

Una técnica muy útil en *machine learning* son los *support vector machines* (SVM). [29] Los SVM son unos predictores (o clasificadores) lineales que se suelen utilizar en espacios de características de dimensiones muy altas. Precisamente la alta dimensionalidad de estos espacios hace que la complejidad y el coste computacional sean unos aspectos a tener mucho en cuenta.

Este método de clasificación, en su formulación más básica, es un clasificador lineal binario (2 clases), no obstante se puede ampliar su utilidad a casos multiclase y a conjuntos no separables linealmente. Esta última extensión en particular, se expone al final de esta sección.

Antes de realizar una formulación matemática de los SVM, introduciremos brevemente cuál es su funcionamiento y como consigue el objetivo buscado, que no es otro que, clasificar las muestras deseadas de la mejor manera posible.

A diferencia de otros clasificadores, los SVM no utilizan una estimación de la función de densidad de probabilidad (*fdp*) del conjunto de muestras (Modelos Generativos) para la clasificación de las mismas, sino que utiliza un conjunto de entrenamiento sin necesidad de modelar la *fdp* (Modelos Discriminativos).

Cabe comentar que la tarea de clasificación normalmente consiste en, una vez se han separado los datos en un conjunto de entrenamiento y un conjunto de prueba, donde el conjunto de entrenamiento contiene un valor objetivo (el cuál corresponde a la etiqueta de clase) y diversos atributos (*features*) El objetivo de los SVM es producir un modelo basado en la información de entrenamiento que prediga el valor de la etiqueta de clase de las muestras de prueba.

Vamos a considerar el siguiente conjunto de muestras, donde las muestras de color amarillo corresponden a una clase, y las muestras de color azul corresponden a otra distinta:

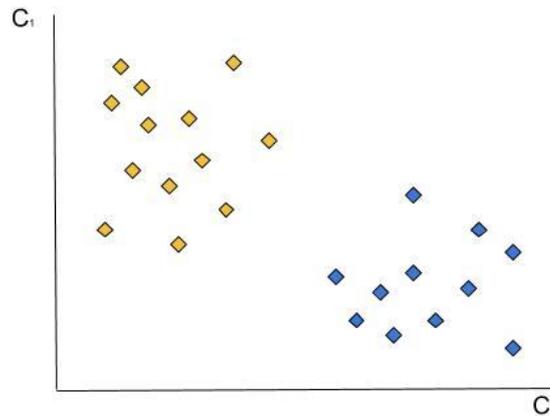


Figura 23. Muestras de dos clases en un espacio de características bidimensional

Para simplificar esta breve explicación, se han dibujado las muestras sobre un espacio de únicamente dos características, pero se ha de tener en cuenta que, normalmente, se trabajará con espacios de características mucho mayores.

En la figura 23 se pueden apreciar las dos clases bien diferenciadas, con lo que nos sería fácil trazar una línea que separara las dos clases. Esto se puede observar en la siguiente imagen:

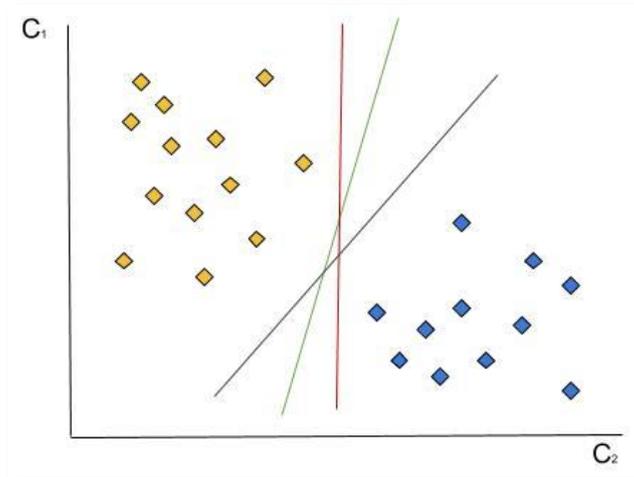


Figura 24. Distintos hiperplanos que separan las muestras en dos clases

En la figura 24 se han trazado distintas líneas que separan las muestras. Cuando hablamos en el argot de los SVM, estas líneas se conocen como **hiperplanos**. Esto es debido a que en el caso bidimensional, un separador binario puede ser interpretado como una línea (recta en este caso), pero como normalmente trabajamos con espacios multidimensionales, se necesitará de hiperplanos para poder separar adecuadamente las muestras.

Además de esto, se observa que, aunque todas las líneas marcadas separen correctamente el conjunto del ejemplo, existirá una de ellas que sea la que más se ajuste a la distribución de muestras y, por ende, sea la solución óptima.

Esto nos ayuda a introducir el concepto de **margen**. Para ello adjuntamos una nueva imagen:

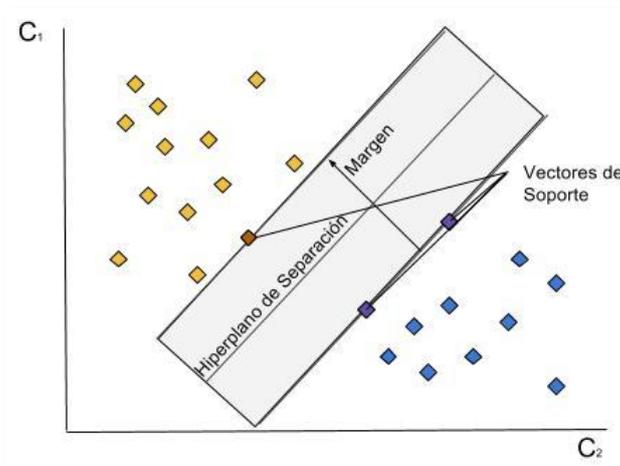


Figura 25. Concepto de Margen óptimo

Precisamente, la característica principal de este clasificador, es la introducción de este concepto: el margen. El margen es la distancia entre los hiperplanos que incluyen los **vectores de soporte** de cada clase. Un vector de soporte es la muestra (o las muestras) de cada clase más cercana al hiperplano de separación.

El proceso de búsqueda del margen máximo es equivalente a pensar en encontrar la región del espacio de características más amplia posible que separe las muestras de distintas clases y que está vacía de las mismas. Esta región, pues, quedará definida por 2 hiperplanos paralelos al hiperplano de separación. Más adelante se verá que esta solución se puede plantear como un problema de optimización en el cuál la variable de optimización es el propio margen.

De esta definición de margen se obtienen unas propiedades muy interesantes. La primera, y la más obvia, es que en la obtención del margen máximo, solo se tienen en cuenta unas pocas muestras (solo los vectores de soporte) y no la totalidad del conjunto. La segunda es que, si se desplazan las muestras asociadas a los vectores de soporte la solución cambia (Figura 26a). Por último, se aprecia fácilmente que si, por el contrario, se desplazan las otras muestras, la solución sigue siendo la misma (Figura 26b).

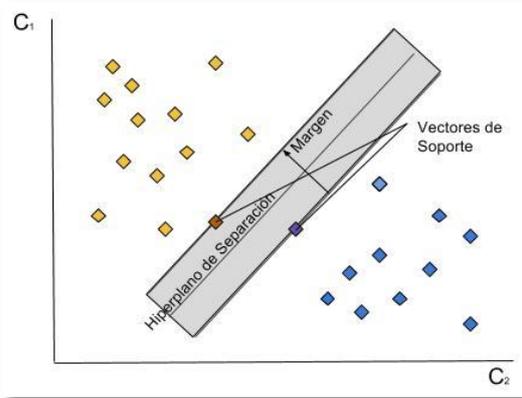


Figura 26a. Cambio vectores de soporte

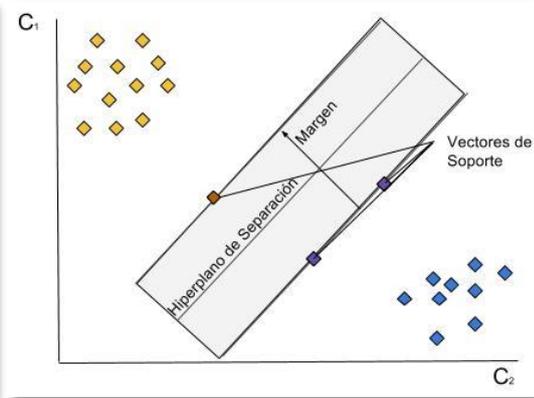


Figura 26b. Cambio muestras restantes

Esta última propiedad proporciona al SVM una cierta robustez estadística y, en particular, proporciona robustez sobre lo que se conoce como **overfitting** (o sobreajuste). Un cierto **overfitting** implicaría que la solución se ajustase demasiado a las muestras a riesgo de perder generalidad.

Para todo lo aquí expuesto, se tiene que asumir que se trata de un conjunto linealmente separable. No obstante, en la realidad no siempre vamos a estar ante este tipo de conjuntos. Es por ello que existen diferentes opciones para solucionar esto.

Por un lado, la condición de margen antes expuesta es conocida como condición de **Margen Duro**, ya que no permite que ninguna muestra se incluya dentro de la región del margen. No obstante, se puede relajar esta condición, lo que se conoce como **Margen Suave** y se posibilita que algunas muestras caigan dentro del margen proporcionando una cierta tolerancia a errores, que se controla con unos parámetros conocidos como **Slack Variables**.

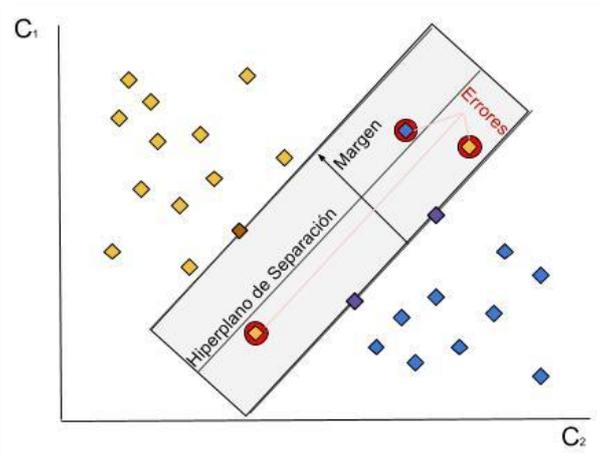


Figura 27. Concepto de Margen Suave

Por otro lado, cuando estamos ante conjuntos no separables linealmente, se puede utilizar una técnica conocida como **Kernel Trick** y que consiste en transformar los datos de un espacio dimensional en el que no son separables linealmente a otro de dimensión superior en el que sí lo son.

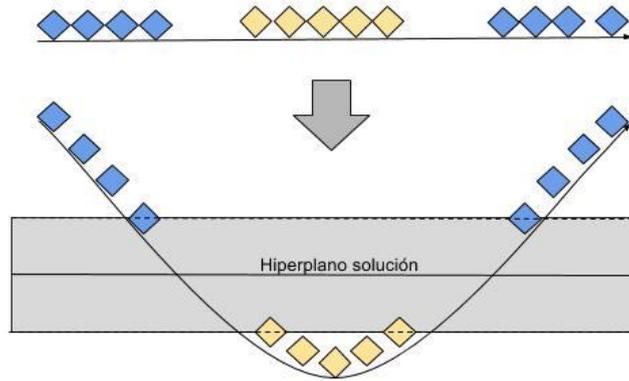


Figura 28. Concepto de Kernel Trick.

4.4.2 Desarrollo matemático del SVM

La solución al hiperplano se expresa de la siguiente manera:

$$w^T x_i + b = 0 \quad (4.16)$$

Donde el vector w es el ortogonal al hiperplano solución y el coeficiente b es el coeficiente de intersección.

A partir del hiperplano propuesto, se separa el espacio en dos regiones, una positiva y otra negativa, tal que el espacio que cumple $w^T x_i + b > 0$ corresponde a las muestras de la clase positiva (la decisión de que clase es positiva y que clase negativa es arbitraria) y la región que cumple que $w^T x_i + b < 0$ corresponde a las muestras negativas. En el caso de la imagen propuesta, las muestras azules se considerarán negativas, y las amarillas positivas.

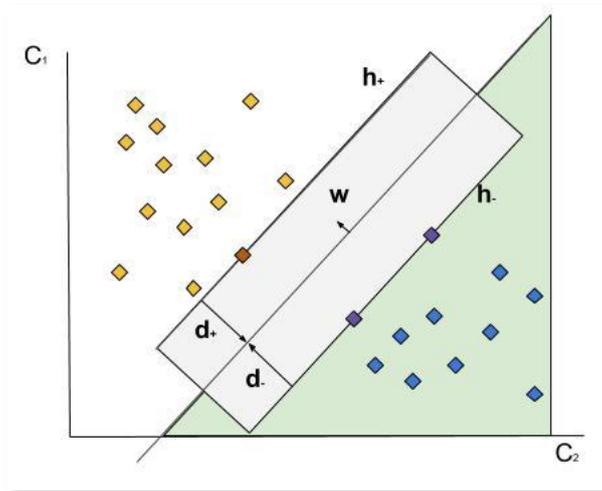


Figura 29. Definición Conceptos Básicos SVM

De esta manera la función de clasificación se podrá definir como:

$$f(x) = \text{sgn}(w^T x_i + b) \quad (4.17)$$

Ya que únicamente considerando el signo al aplicar el hiperplano a la muestra, sabremos a qué clase corresponde.

Los hiperplanos de los vectores de soporte, que son paralelos al hiperplano de solución se definen como $w^T x_i + b = 1$ para el hiperplano de la clase positiva (h_+ en el dibujo) y $w^T x_i + b = -1$ para el hiperplano de la clase negativa (h_-).

Por tanto para que una muestra sea etiquetada como positiva deberán cumplir que $w^T x_i + b \geq 1$ y para que sea etiquetada como negativa deberá cumplir $w^T x_i + b \leq -1$.

Así pues, considerando que las muestras positivas son etiquetadas como $\{+1\}$ y las muestras negativas como $\{-1\}$ las dos condiciones anteriores se pueden juntar en una única **condición de clasificación** que es la siguiente:

$$y_i(w^T x_i + b) \geq 1 \quad (4.18)$$

El margen podrá calcularse como la distancia entre los dos hiperplanos que incluyen a los vectores de soporte de la siguiente manera:

$$\text{margen} = d^+ + d^- = 2 \left(\frac{|w^T x + b|}{\|w\|} \right) = 2 \frac{1}{\|w\|} \quad (4.19)$$

El objetivo pues, será de maximizar el valor del margen. Si se tiene en cuenta que maximizar esta ecuación es equivalente a minimizar la función inversa que se puede representar como la siguiente función:

$$\Phi(w) = \frac{1}{2} \|w\|^2 \quad (4.20)$$

Siendo $\|w\|^2 = w^T w$ y estando sujeta a la condición de clasificación expuesta en la ecuación 4.18.

Estamos ahora pues ante un problema de optimización cuadrática donde el objetivo será obtener los valores w y b (ecuación 4.16) pertenecientes al hiperplano solución.

Resolviendo esto, tendríamos la solución óptima al problema planteado, no obstante, ha de tenerse en cuenta que, en un inicio, se supuso que el conjunto de muestras utilizado era linealmente separable. Debido a que muchos de los conjuntos utilizados en la realidad no serán de esta manera, se propuso el *kernel trick*.

Con esta técnica, se utiliza una función no-lineal (Φ) para mapear los datos a un espacio dimensional superior en el que sí que sean linealmente separables.

La función *kernel* se define como:

$$K(x_i, x_j) = \Phi(x_i)^T \Phi(x_j) \quad (4.21)$$

Aunque existen distintos *kernels* propuestos en el presente trabajo se ha utilizado un *kernel* lineal [30], debido a su velocidad, y simplicidad en el cálculo. Se define en la ecuación XXX:

$$K(x_i, x_j) = x_i^T x_j \quad (4.22)$$

4.4.3 Técnica de Bagging (Bootstrap Aggregating)

La técnica de Bootstrap Aggregating también conocida como Bagging se utiliza en el algoritmo propuesto para mejorar la estabilidad y la precisión del algoritmo de *machine learning* utilizado (SVM) [31]. Esta técnica reduce la varianza y ayuda a evitar lo que ya conocemos como *overfitting*. En particular, en este trabajo, es utilizada para solucionar el problema de desbalanceo de clases, esto es, que existen muchas más muestras (*patches*) sanas que patológicas. Para poder hacerse una idea, se comprobó en una de las simulaciones realizadas que el número de muestras patológicas es de 12.000 mientras que el número de sanas es de 130.000. Aquí se ve presente el gran desbalanceo entre clases.

Esta técnica pues, generará M clasificadores distintos según la siguiente ecuación

$$M = \left\lceil \frac{\text{muestras sanas}}{\text{muestras patológicas}} \right\rceil \quad (4.23)$$

Asimismo, se resume el proceso en la siguiente figura:

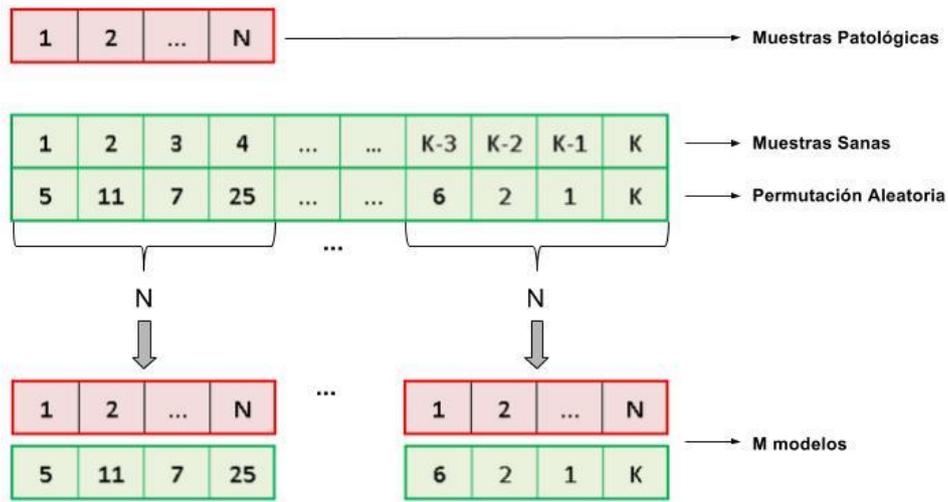


Figura 30. Técnica de Bagging

Como se puede apreciar en la figura 30 se crean M modelos de clasificadores SVM. Cada clasificador predice una etiqueta para cada patch y la decisión final se tomará en función del algoritmo que se decida: mayoría simple, unanimidad o mediante la definición de un cierto umbral.

4.4.4 LIBSVM

Debido a su gran aceptación en el mundo científico, existen diferentes librerías desarrolladas que facilitan el trabajo, proporcionando una serie de funciones que agilicen el proceso de diseño de algoritmos de clasificación. En nuestro caso se ha utilizado la librería LIBSVM disponible de forma gratuita (<http://www.csie.edu.tw/~cjlin/libsvm>) [32].

Para utilizar correctamente esta librería, se proporcionan dos funciones: *svmtrain* (entrenamiento) y *svmpredict* (testeo).

En ambos casos, se recomienda normalizar los datos de entrada, es decir, el vector de características (*features*) que se utilice para que tengan un valor de [-1,+1] o [0,1], además de, claro está, utilizar el mismo escalado para el el *train* que para el *test*.

También se recomienda representar *m* números para una atributo con *m* categorías utilizando únicamente un 1 para la categoría que sea cierta y un 0 para las restantes. Por ejemplo, en un atributo donde las categorías sean {negro, blanco, color} se podría representar como (1,0,0),(0,1,0) y (0,0,1) ya que, siempre según los desarrolladores de la librería, esta codificación podría ser más estable.

Por último, comentar que en el caso de LIBSVM, la función a optimizar se define como:

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^l \xi_i \quad (4.24)$$

sujeta a $y_i(w^T \Phi(x_i) + b) \geq 1 - \xi_i, \xi_i > 0$

El término introducido está relacionado con el error cometido ya que *C* es conocido como el parámetro de error.

4.5 Implementación del algoritmo

Para comprender de una manera más simple y visual el funcionamiento de nuestro sistema, en la Figura XXX se pueden observar los diferentes módulos o unidades en los que se puede dividir el método propuesto. Además se puede visualizar el diagrama de flujo que sigue el algoritmo.

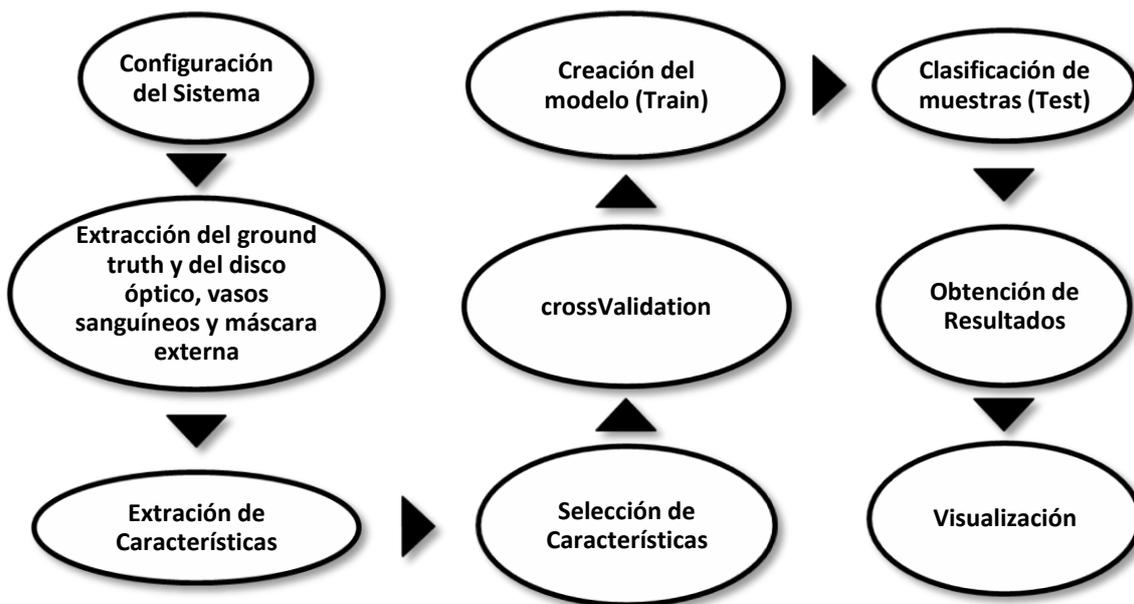


Figura 31. Diagrama de flujo del algoritmo propuesto en esta sección.

En esta figura, se pueden observar nueve unidades distintas que se explicarán, una a una, en las siguientes subsecciones. Cabe comentar que el apartado de Configuración del Sistema se expondrá el último, ya que de esta manera, se habrán expuesto previamente todos los parámetros implicados en la configuración.

4.5.1 Extracción del ground truth.

En este bloque del algoritmo se recorre la imagen con una **ventana deslizable** de tamaño (x,y) y con un cierto **solape** entre *patches* (cada una de las ventanas de la imagen) definido arbitrariamente con los parámetros (Δx , Δy). Se adjunta un ejemplo gráfico en la figura 32.

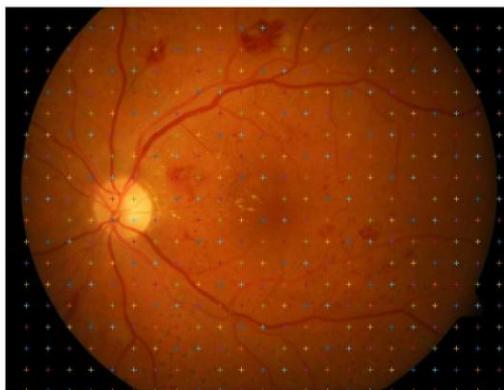


Figura 32. Rejilla de patches sobre una imagen original

El objetivo de este recorrido es determinar, para cada uno de los *patches* de la imagen, si se trata de una zona patológica, sana o no está incluida en la propia zona de la retina (bordes exteriores). Para ello, se define un **porcentaje mínimo** de píxeles patológicos en un *patch* (p_{pat}) definidos en la imagen de *ground truth* a partir del cual será considerado como patológico.

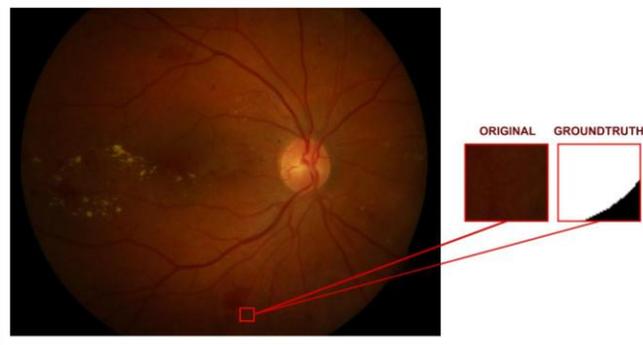


Figura 33. Patch imagen original y del ground truth

También se extraen y se almacenan las máscaras binarias expuestas en la sección 3.2 (disco óptico, vasos sanguíneos y del FOV).

Toda la información extraída en este bloque, se almacena en un fichero *.mat* para que, si se utiliza la misma configuración a posteriori, no sea necesario volver a realizar todo el proceso de extracción, sino que con cargar el archivo *.mat* sea suficiente. Además, cabe comentar que la información del *ground truth* se extrae para distintos niveles de la **pirámide espacial**.

Este concepto se basa en que, al reducir el tamaño de las imágenes originales, será posible detectar lesiones de distintos tamaños en cada capa. Asimismo, en la cima de la pirámide se detectaran lesiones grandes mientras que en la base lesiones de menor tamaño. Para una correcta transición entre los distintos niveles de la pirámide se realiza, previamente al reescalado un filtrado con el objetivo de que no aparezcan problemas en la imagen. En el caso de querer detectar lesiones oscuras, como es nuestro caso, se recomienda un filtrado de mínimos. Serán parámetros regulables el número de capas de la pirámide (L) así como el factor de reducción (N) y el tipo de filtrado aplicado.

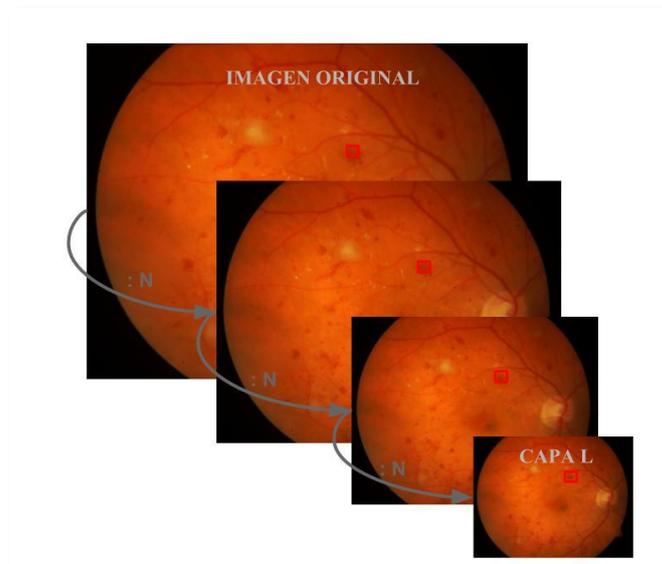


Figura 34. Pirámide Espacial

4.5.2 Extracción y selección de características.

Si bien en el esquema presentado se ha dividido en dos bloques, ya que es así como se indica en nuestro algoritmo, pensamos que es de mayor utilidad incluir ambos en único subapartado debido a que son enteramente dependientes.

Una vez se ha determinado el canal a utilizar (R, G, B o cualquier combinación de ellos) se realiza un análisis local de la textura de las imágenes en cuestión según las técnicas explicadas en las secciones 4.2 y 4.3. Este análisis depende de los parámetros elegidos tanto como para el análisis LBP como el análisis de granulometría (se detallan en la tabla 2).

Aunque el análisis local de textura se computa para todos los niveles de la pirámide, en nuestro sistema se puede elegir entre utilizar la información de toda la pirámide o utilizar únicamente la de la base.

El vector de *features* para cada muestra se construye a partir de la concatenación de las características extraídas por todos los descriptores, es decir, tanto del histograma LBPV como de la granulometría.

No obstante, en el bloque de selección de características, podremos decidir utilizar únicamente aquellas que nos interese.

4.5.3 Cross Validation.

Este bloque es el encargado de dividir las imágenes en tantos grupos (*folders*) como se haya indicado en la configuración. Para ello, se utiliza un archivo definido a priori (*'kfold.mat'*) y que, de forma aleatoria, asigna un número entre 1 y el n° de *folders*, repartiendo equitativamente todas las imágenes. En nuestro caso (base de datos *diaretdb1*) al tener 89 imágenes, si ponemos que se eligen 5 se distribuyen de tal forma que 4 *folders* cuentan con 18 imágenes y el restante con 17 imágenes.

Esta distribución de las imágenes será utilizada para reservar uno de los *folders* (que se define en la configuración) para probar el modelo entrenado, mientras que los otros cuatro serán utilizados para entrenar el clasificador. Este proceso se repetirá utilizando cada uno de los *folders* para testear el sistema., obteniendo distintos resultados que se combinarán para extraer unas conclusiones más robustas ya que la varianza se reducirá.

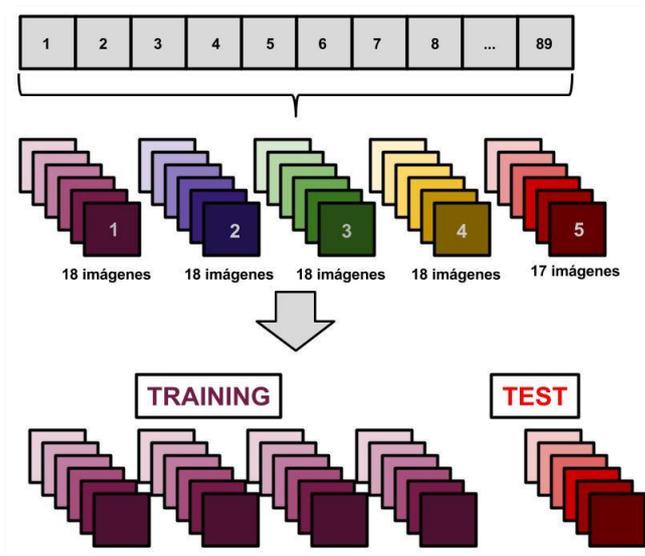


Figura 35. Esquema crossValidation

4.5.4 Creación del modelo (Train).

Mediante las muestras asociadas al grupo de entrenamiento se procede a entrenar el modelo según lo expuesto en la sección 4.4.

Como se expuso, se utiliza la técnica SVM y se crean tantos modelos como indique el parámetro M (sección 4.4.3) relacionado con la técnica de *Bagging*.

Una vez obtenido este valor, mediante la función de *libsvm* (sección 4.4.4) *svmtrain*, se entrenan los distintos modelos, los cuáles serán el parámetro necesario para el siguiente bloque.

Los distintos parámetros regulables relacionados con este bloque son el tipo de kernel utilizado (sección 4.4.2), parámetros específicos relacionados con el propio kernel, el algoritmo de votación. Estos se sintetizan al final de este capítulo.

4.5.5 Clasificación de muestras (Test).

Tras la etapa de entrenamiento, y ya con el modelo de clasificación generado se procederá a la validación del mismo empleando para ello el conjunto de test.

En este caso, se utilizarán los features extraídos en el bloque reservado a este fin y, mediante la función *svmpredict* que también pertenece a la librería *libsvm* se obtendrán los resultados.

4.5.6 Obtención de Resultados.

En la literatura perteneciente a la teoría de la clasificación, existen diversas medidas para evaluar el rendimiento del clasificador. Para entender algunas de estas medidas queremos mencionar primero el concepto de **matriz de confusión**, la cual constituye por si misma una de ellas:

| | | VALOR PREDICCIÓN | |
|------------|---|------------------|----|
| | | P | N |
| VALOR REAL | P | VP | FN |
| | N | FP | VN |

Figura 36. Matriz de confusión

Donde VP son los **verdaderos positivos**, es decir, muestras predichas como positivas que realmente lo son y VN son las muestras negativas estimadas correctamente según su etiqueta. De la misma forma FP son los **falsos positivos** (muestras negativas clasificadas como positivas) y FN son los **falsos negativos** (muestras positivas clasificadas como negativas). También es fácil deducir que P es el total de positivos en cada caso y N es el total de negativos, y que la suma de todos los positivos y negativos, tanto en el valor real como en la predicción da como resultado el mismo valor, es decir, el número total de muestras ($P+N$ =Número de muestras)

Esta herramienta permite visualizar la confusión entre un par dado de clases, en nuestro caso, patológico o sano. Como el lector se puede imaginar fácilmente, interesará tener el número más alto de verdaderos positivos y negativos, ya que será indicativo de una correcta clasificación.

A partir de la matriz de confusión se pueden extraer diversos parámetros [6]. Algunos que nos resultarán de interés son:

$$\text{Sensibilidad} = \frac{VP}{VP + FN} \quad (4.25)$$

$$\text{Especificidad} = \frac{VN}{VN + FP} \quad (4.26)$$

$$\text{Exactitud (Accuracy)} = \frac{VP + VN}{P + N} \quad (4.27)$$

La **sensibilidad** nos proporciona un indicador porcentual de cuántos positivos o enfermos (si asumimos esta clase como positiva) se detectan sobre el total, mientras que la **especificidad** caracteriza la capacidad del clasificador para dar como sanos a los pacientes verdaderamente sanos. La exactitud (o **accuracy**) indica, cuanto se desvía el clasificador respecto del caso utópico (ningún fallo de clasificación).

Por último, hemos de comentar lo que se conoce como **curva ROC** y, de la misma forma, el indicador **AUC** (o **área bajo la curva ROC**). En la figura 37 se adjunta un ejemplo de esta curva:

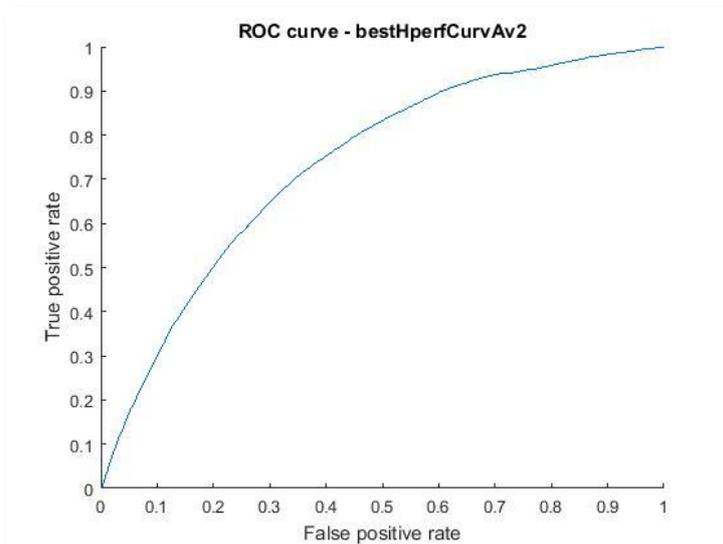


Figura 37. Curva ROC

Esta gráfica es muy utilizada para comprobar la eficacia en la detección. Es una representación del ratio de verdaderos positivos respecto del ratio de falsos positivos o lo que es lo mismo, de la sensibilidad respecto de la especificidad. El área bajo esta curva (AUC) es el indicador de cuán bien funciona el sistema clasificando. Cuánto mayor (más cercana a 1), mejor se comportará el clasificador. Por el contrario, una línea diagonal, desde el origen de coordenadas hasta el punto (1,1) sería el peor caso posible.

Todos los indicadores aquí expuestos, junto con otros, son calculados en el bloque de obtención de resultados y se almacenan en un archivo independiente para cada uno de los *folds* que, una vez realizado el cálculo para los 5 *folds* (*crossValidation*) se combinarán para obtener un resultado definitivo.

4.5.7 Configuración del sistema.

Para poder proceder a realizar nuestras simulaciones, hemos de saber a priori cuál es la configuración particular que queremos que el sistema diseñado adopte. Con este fin, el primer paso del algoritmo propuesto será una llamada a un script donde se cargarán a la configuración todos los parámetros deseados.

Para observar con mayor claridad todos los parámetros de los que hablamos, adjuntamos más abajo una tabla donde se mencionan aquellas variables modificables y los valores que se han asumido para los resultados que se expondrán en el capítulo 6.

| Bloque del algoritmo | Parámetros ajustables |
|-------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Imagen original | Canal a utilizar → Green (G) |
| Extracción de características | Análisis local: <ul style="list-style-type: none"> – Tamaño del patch (64 x 64) – Solape ($\Delta x=32$, $\Delta y=32$) LBP: <ul style="list-style-type: none"> – Radio (Parámetro $R = 1$) – Número de puntos (Parámetro $P = 8$) Granulometría: <ul style="list-style-type: none"> – Tipo de SE → Disco y Línea – Paso (<i>Step</i>) → 2 – $N_{\max} = 22$ |
| Pirámide espacial | Factor de reducción ($N = 2$) Número de capas ($L = 3$) Tipo de filtrado → <i>Máximos</i> |
| Clasificación | Tipo de kernel → Kernel Lineal Algoritmo de votación → Mayoría simple Número de folds =5 |
| Extracción del ground truth | $P_{pat} = 40\%$ |

Tabla 2. Parámetros variables del sistema propuesto y valores adoptados en las simulaciones realizadas

Capítulo 5. Clasificación a nivel de imagen

5.1 Introducción

Una vez explicado el algoritmo de clasificación a nivel de *patch* vamos a proceder a hacer lo propio con el algoritmo propuesto para realizar una clasificación global o de imagen.

Aunque la clasificación a nivel local como la presentada, nos permite obtener unos resultados muy precisos de las zonas de la imagen tratada que sufren algún tipo de patología, el objetivo de un sistema finalista de diagnóstico es poder predecir con suficiente rigor si un caso clínico en particular padece la enfermedad a diagnosticar o no. De aquí nace la necesidad de contar con un algoritmo de este tipo.

En el algoritmo propuesto, se utilizarán los resultados obtenidos de las simulaciones realizadas con el algoritmo propuesto en la sección 4.5 para predecir, para cada imagen de la base de datos *diaretdb1* si se trata de una imagen patológica (que presenta signos clínicos asociada a la enfermedad) o sana. Cabe decir que se han considerado patológicas, todas aquellas imágenes que cuentan con algún píxel de la imagen (definido en el *ground truth* por los expertos) marcado como lesión.

Hay que tener en cuenta también que el correcto funcionamiento de este algoritmo está estrechamente ligado a que ocurra lo propio con el algoritmo de clasificación a nivel local, ya que un etiquetado erróneo de los *patches* producirá que el algoritmo tenga datos de entrada, etiquetados en una clase y pertenecientes a otra distinta.

En este capítulo, debido a que toda la teoría asociada al algoritmo propuesto se encuentra incluida en el capítulo 4, únicamente se expondrá la implementación del mismo.

5.2 Implementación del algoritmo

Al igual que en la sección anterior, adjuntamos a continuación un diagrama de flujo para comprender más fácilmente la implementación del algoritmo.

Se han englobado los distintos bloques del algoritmo en 6 bloques distintos, no obstante, alguno de ellos, se subdividirá en otros, como veremos más tarde para una mayor comprensión del flujo de datos en el sistema.

A diferencia del capítulo anterior, aunque existe un bloque de configuración del sistema, no se adjuntará ninguna tabla con parámetros regulables ya que, al ser mínimos, será suficiente con mencionarlos en el apartado al que afecten.

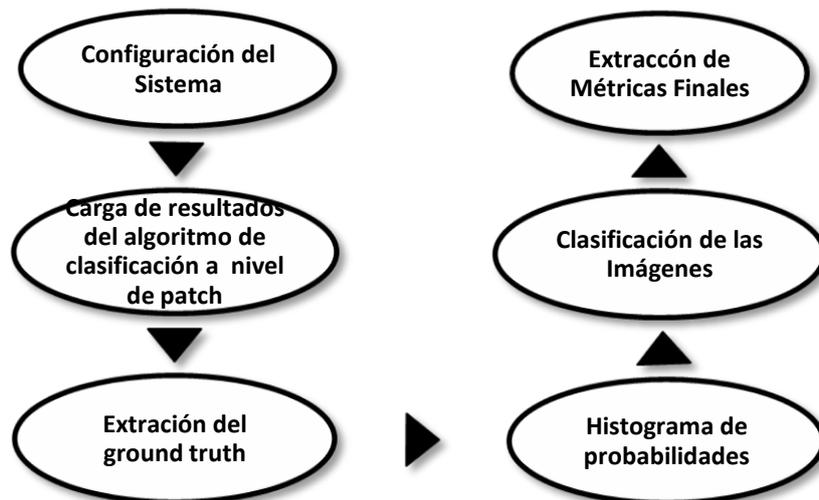


Figura 38. Diagrama de flujo del algoritmo de Clasificación a nivel de Imagen.

5.2.1 Carga de resultados del algoritmo de clasificación a nivel de patch

En primer lugar, una vez que se ha configurado el sistema (aspecto que en este caso sí que es bastante trivial) hay que añadir al sistema la información extraída del análisis y clasificación local. Esta información, debido al *crossValidation* realizado, se encuentra incluida en 5 ficheros distintos nombrados como *OutputSet-i* (siendo $i \in \{1, \dots, \text{número de folds}=5\}$). Es por ello que se repetirá este bloque íntegro 5 veces, un para extraer la información de cada *OutputSet*.

Esta información se encuentra organizada de la siguiente forma:

| Field ▲ | Value |
|-----------------|----------------------|
| predict_label_L | 1x35352 double |
| accuracy_L | 1x10 cell |
| dec_values_L | 1x10 cell |
| CP | 1x1 classperformance |
| GT | 1x35352 double |
| coordR | 1x35352 double |
| coordC | 1x35352 double |
| numIm | 1x35352 double |
| pyrLevel | 1x35352 double |
| ROC | 1x1 struct |
| AUC | 0.7794 |
| Fscore | 0.1659 |

Figura 39. Atributos del *OutputSet-i*

De todos los parámetros que se pueden apreciar en la figura 39, los que realmente serán de interés son: *predict_label_L* que indica la clase predicha para cada patch analizado en el *kFold*; *dec_values_L* que indica para todos los modelos entrenados y que equivalen al parámetro M de la técnica de *bagging* la probabilidad de cada patch de ser patológico; *numIm*, que indica a qué imagen corresponde cada patch y *pyrLevel*, que indica a que nivel de la pirámide espacial corresponde cada patch.

Una vez extraída esta información, se ha de asignar cada patch a la imagen original para poder realizar un correcto análisis. Para ello, se necesita utilizar el archivo *kFold.mat* que, como se comentó anteriormente incluye a qué fold corresponde cada imagen. Esto es necesario debido a que la información que incluye *numIm* representa la imagen a la que corresponde pero dentro del Fold de trabajo.

Una vez realizado esto, es necesario calcular la probabilidad media de cada patch de ser patológico o no. Para ello se calcula la media de las probabilidades indicadas por cada uno de los M modelos entrenados anteriormente.

Una vez se ha realizado esto, solo queda introducir todos los datos de cada imagen asociados a la misma en una matriz que llamaremos *GT_matrix*. En la figura 40 se observa el aspecto de esta matriz:

| nº Im | Probabilidades patches | | | | |
|-------|------------------------|-------|-------|-------|-----|
| 3 | 0,873 | 0,798 | 0,384 | 0,799 | ... |
| 7 | 0,789 | 0,658 | 0,645 | 0,911 | ... |
| 13 | 0,388 | 0,389 | 0,236 | 0,444 | ... |
| 17 | 0,235 | 0,102 | 0,954 | 0,333 | ... |
| ... | ... | ... | ... | ... | ... |

Figura 40. Ejemplo de *GT_Matrix*

Este proceso, como comentamos se repetirá para los 5 folds para tener toda la información disponible. Para finalizar adjuntamos un esquema en la Figura 41 que resume todo lo aquí expuesto:

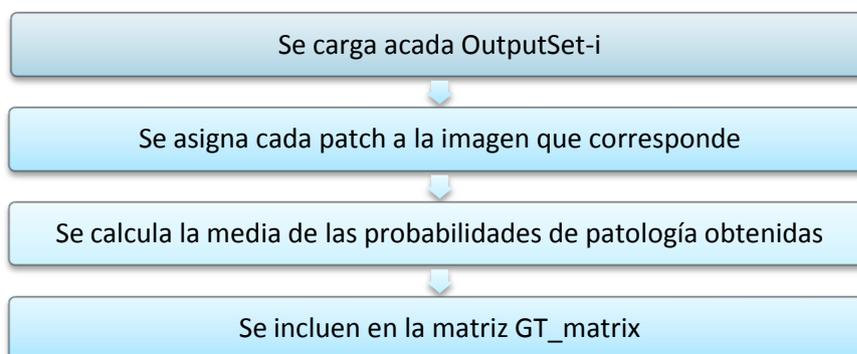


Figura 41. Esquema de flujo del bloque actual

5.2.2 Extracción del ground truth

Antes de proceder a extraer las etiquetas de clase de cada imagen, hemos de ordenar la anteriormente mencionada *GT_Matrix*. Como se observa en la Figura 44 en la columna asociada al número de Imagen, las imágenes están desordenadas por el proceso de extracción de cada fold. Para mayor facilidad, se ordenan en orden ascendente.

Una vez realizado esto se procede a extraer la etiqueta de clase a partir del *ground truth* de cada imagen. Para ello, como se mencionó anteriormente, se define que únicamente con que una imagen contenga un píxel patológico, se definirá como imagen patológica y se le asignará como etiqueta de clase {1}. Para las imágenes sanas (o no patológicas) se asigna como etiqueta de clase {0}.

Estas etiquetas se incluyen en la *GT_matrix* quedando con un aspecto similar al de la figura 42:

| nº Im | Probabilidades patches | | | | | Clase |
|-------|------------------------|-------|-------|-------|-----|-------|
| 3 | 0,873 | 0,798 | 0,384 | 0,799 | ... | 1 |
| 7 | 0,789 | 0,658 | 0,645 | 0,911 | ... | 0 |
| 13 | 0,388 | 0,389 | 0,236 | 0,444 | ... | 0 |
| 17 | 0,235 | 0,102 | 0,954 | 0,333 | ... | 1 |
| ... | ... | ... | ... | ... | ... | ... |

Figura 42. *GT_Matrix* con etiquetas de clase

5.2.3 Histograma de probabilidades

Una vez creada la *GT_Matrix* ya se está en condiciones de conseguir un vector de características que se utilizará en la etapa de entrenamiento para crear el modelo. En este caso, se ha decidido calcular el histograma de probabilidades de patología de los patches de cada imagen. Aquí aparece uno de los parámetros configurables del sistema.

El número de bins (valores en el eje x) del histograma. En nuestro caso hemos optado por utilizar 64, pero podría utilizarse cualquier otro valor que se pensara (p.e. 32,128,...).

Así pues, se procede a calcular el histograma normalizado. En la figura 43 mostramos un posible histograma:

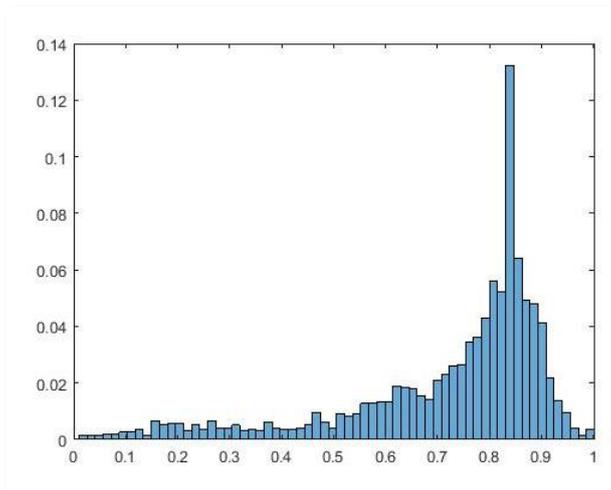


Figura 43. Histograma Normalizado de Probabilidades

Con este histograma se construye una matriz que tendrá en cada fila los valores de los *bins* del histograma y en la última columna llevará asociada una etiqueta de clase, quedando así una matriz de 89x65(89 imágenes y 64 *features* +1 etiqueta de clase).

5.2.4 Clasificación de las imágenes y Extracción de métricas Finales

Una vez tenemos construida la matriz que se utilizará para entrenar y testear los modelos solo nos queda separar las imágenes en 5 *folds* para, como ya es común, realizar un proceso de *crossValidation*. Así pues, se utilizará cada vez uno de los *folds* como *fold* de *test* mientras que los otros cuatros servirán para entrenar el modelo. Cabe comentar que a las imágenes asociadas, en cada caso, al *fold* de *test* se les eliminará la etiqueta de clase, ya que, este es el parámetro que se quiere predecir.



Figura 44. Diagrama de flujo del bloque

Cabe comentar que en el caso de este algoritmo se ha utilizado la aplicación de MATLAB *classificationLearner* y así exponer una nueva herramienta de clasificación.

Capítulo 6. Resultados

En esta sección se presentarán los resultados de las métricas extraídas mediante los métodos expuestos anteriormente. Es importante remarcar que todos estos, han sido obtenidos según la configuración expuesta en la tabla 2.

De esta manera, se han realizado diversas simulaciones con distintas combinaciones del vector de características con el objetivo de obtener la combinación que mejores resultados en cuanto a especificidad, sensibilidad y *accuracy* nos proporcionen.

Una vez obtenida la mejor combinación posible, se utilizarán los resultados elegidos como entrada al siguiente algoritmo (capítulo 5).

Por último, debido a que nuestro sistema proporciona una representación visual de los resultados, se adjuntarán algunos ejemplos de la misma, con tal de presentar una herramienta más gráfica para comprobar el estado del sistema de diagnóstico propuesto.

Antes de adjuntar una tabla con los resultados propiamente, se introduce la siguiente leyenda que representará cada uno de los componentes de las combinaciones utilizadas en el vector de características con el objetivo de sintetizar la tabla 3 de la mayor manera posible:

dirh_LBP_VAR representa la información de textura extraída con LBPV. La información extraída con granulometría se define como dirPSop_d, siendo esta la extraída con aperturas mediante un SE en forma de disco; dirPSop_l, información extraída con aperturas mediante un SE en forma de línea; dirPScI_d, información extraída con cierres con un SE en forma de disco; dirPScI_l, información extraída con cierres con un SE en forma de línea.

| | Accuracy | Sensibilidad | Especificidad | AUC |
|------------------------------------------------------|-----------------|-----------------|-----------------|-----------------|
| dirh_LBP_VAR | 0,6784 ± 0,0504 | 0,6751 ± 0,0504 | 0,6748 ± 0,0504 | 0,7340 ± 0,0635 |
| dirh_LBP_VAR-dirPSop_d | 0,6852 ± 0,0452 | 0,6855 ± 0,0446 | 0,6852 ± 0,0453 | 0,746 ± 0,0541 |
| dirh_LBP_VAR-dirPSop_d-dirPSop_l | 0,7085 ± 0,0319 | 0,7087 ± 0,0325 | 0,7084 ± 0,0319 | 0,7772 ± 0,0402 |
| dirh_LBP_VAR-dirPSop_d-dirPSop_l-dirPScI_d-dirPScI_l | 0,7236 ± 0,0302 | 0,7235 ± 0,0302 | 0,7236 ± 0,0302 | 0,7951 ± 0,0389 |
| dirPScI_d-dirPScI_l | 0,6805 ± 0,0334 | 0,6810 ± 0,0337 | 0,6804 ± 0,0334 | 0,7366 ± 0,0415 |
| dirPSop_d-dirPSop_l | 0,7048 ± 0,0297 | 0,7044 ± 0,0299 | 0,7048 ± 0,0297 | 0,7711 ± 0,0354 |
| dirPSop_d-dirPSop_l-dirPScI_d-dirPScI_l | 0,7172 ± 0,0284 | 0,7168 ± 0,0281 | 0,7172 ± 0,0284 | 0,7859 ± 0,0347 |

Tabla 3. Resultados de las Simulaciones.

Se puede apreciar en la tabla adjunta, que la combinación que mejores resultados arroja es la que engloba todos los descriptores mencionados anteriormente, es por esto que se tomará como referencia y será parámetro de entrada para el segundo algoritmo propuesto.

Adjuntamos también, una gráfica en la que se incluyen todas las curvas ROC de las simulaciones realizadas:

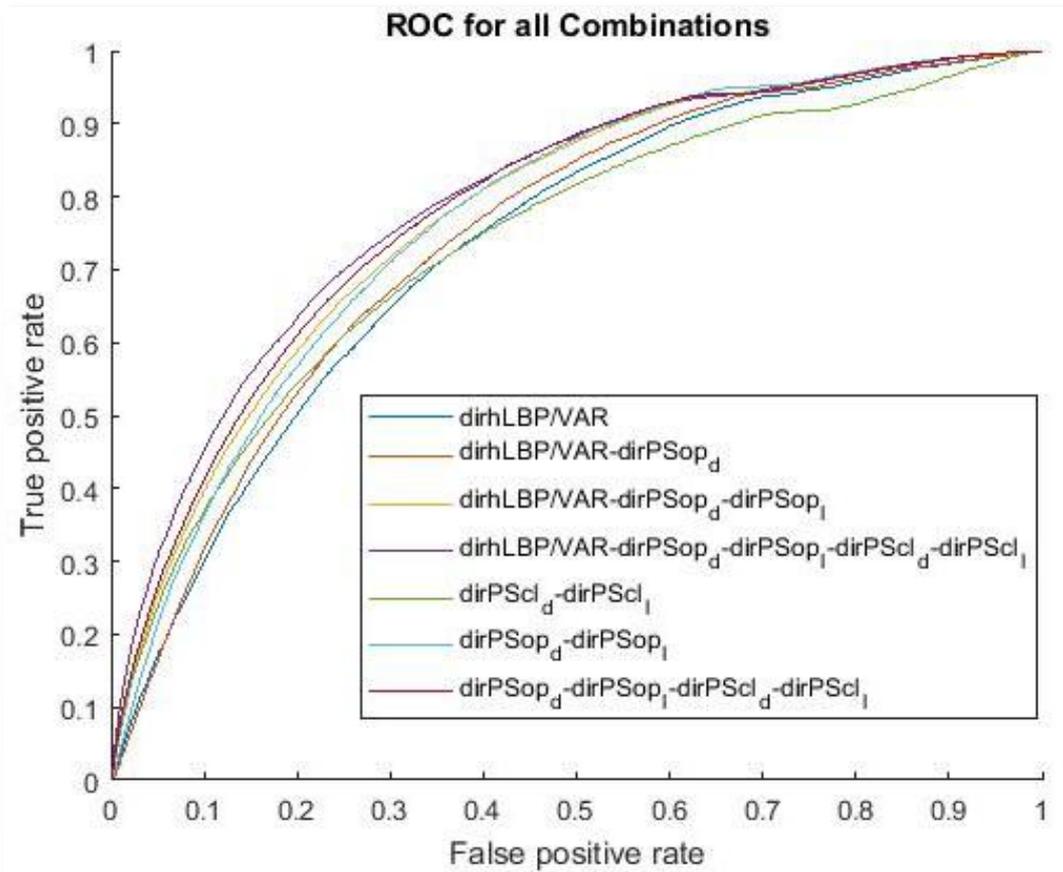


Figura 45. ROC de todas las combinaciones de descriptores utilizadas

En esta gráfica se comprueba lo expuesto en la tabla 3. La combinación que utiliza todos los descriptores mencionados es la que mejores resultados arroja.

Una vez expuesto esto, como comentamos, vamos a utilizar estos resultados para comprobar el funcionamiento del segundo algoritmo propuesto.

Mediante el algoritmo propuesto y utilizando la aplicación de MATLAB classificationLearner se obtienen los resultados que se pueden interpretar a partir de la matriz de confusión expuesta en la figura 46:

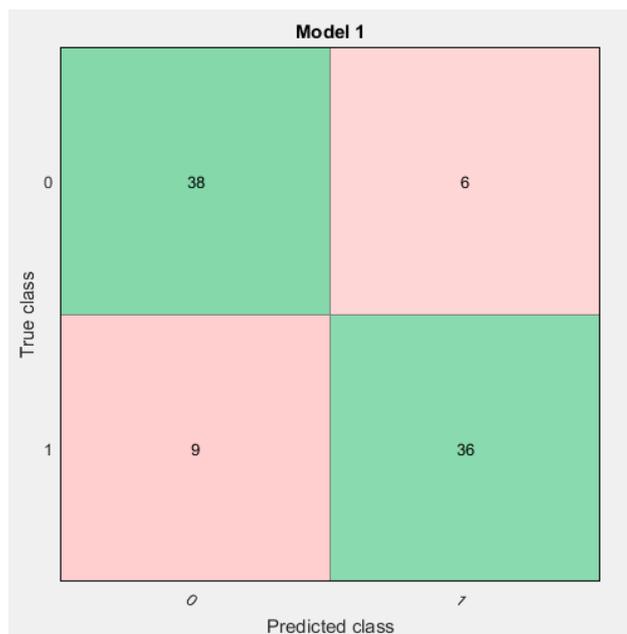


Figura 46. Matriz de confusión del modelo entrenado

A partir de la matriz de confusión, mediante las ecuaciones vistas en el apartado 4.5.6 se pueden obtener la *accuracy*, especificidad y la sensibilidad que toman en este caso los siguientes valores:

| | Accuracy | Sensibilidad | Especificidad | AUC |
|-----------------------------------|-------------|--------------|---------------|------|
| Modelo de Clasificación de Imgaen | 0,831460674 | 0,863636364 | 0,8 | 0,87 |

Tabla 4. Resultados del modelo de clasificación de Imagen

De la misma forma, enfrentando el ratio de verdaderos positivos, contra el de falsos positivos, como ya se vio, se puede obtener la curva ROC y el valor AUC adjuntado en la tabla 4 y presente en la Figura 47. Cabe comentar que, al contar con muchas menos instancias que la curva presentada en la figura 45, la curva ROC de la figura 47 es mucho menos suave.

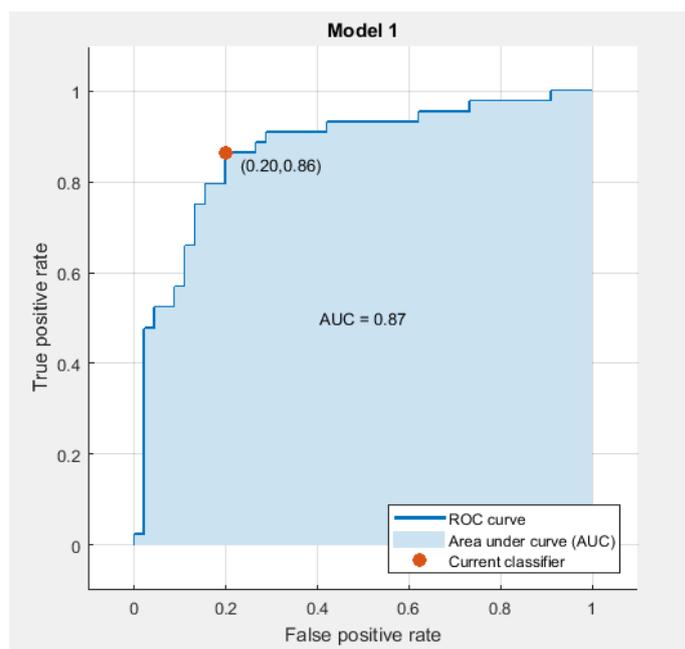


Figura 47. Curva ROC del modelo entrenado.

6.1.1 Visualización.

Por último, existe la posibilidad en nuestro sistema de visualizar en la imagen original el resultado arrojado de nuestra clasificación. Tanto los verdaderos positivos (marcados en rojo) como los falsos positivos (en verde) y los falsos negativos (en azul) son indicados en la imagen original a modo de diagnóstico final.

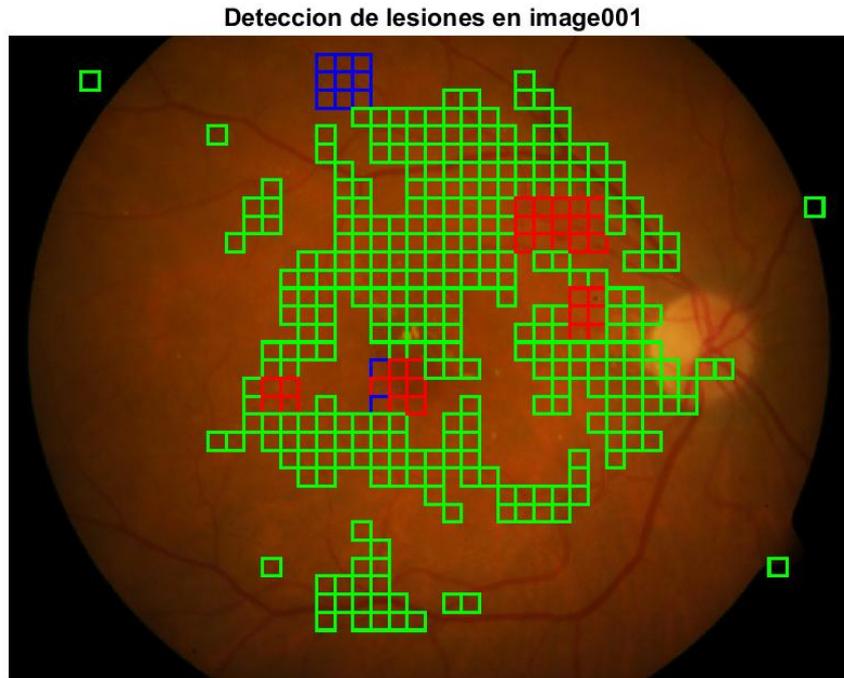


Figura 48. Ejemplo de diagnóstico final en la imagen 001 de la base de datos diartedb1.

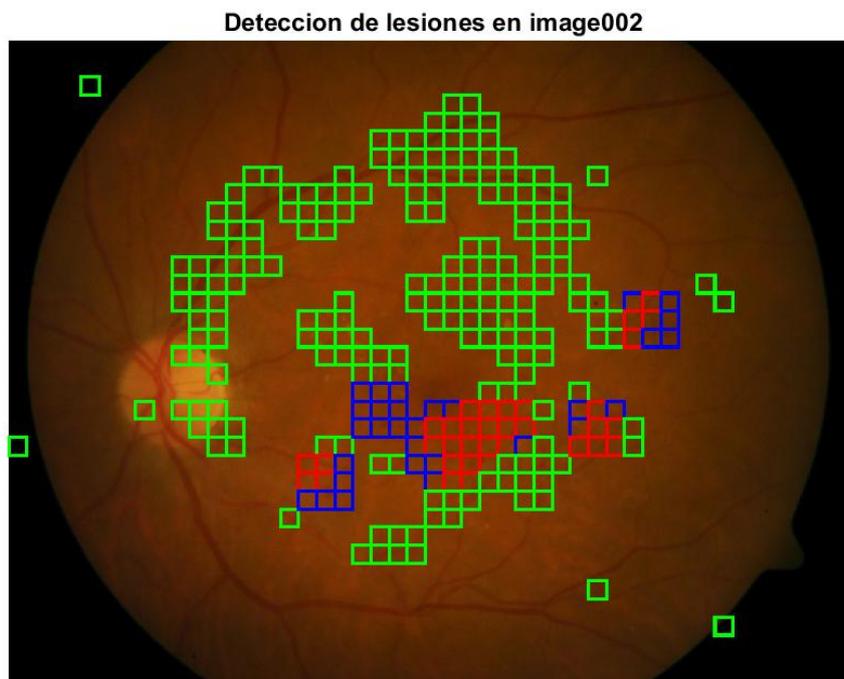


Figura 49. Ejemplo de diagnóstico final en la imagen 002 de la base de datos diartedb1.

Capítulo 7. Conclusiones y líneas futuras.

En el presente trabajo se ha presentado un sistema basado en análisis local de texturas combinado con técnicas de machine learning como son los support vector machine para detectar hemorragias en imágenes de fondo de ojo y ayudar a diagnosticar la retinopatía diabética.

Los resultados arrojados a pesar de no ser los ideales, nos sirven para evidenciar que es posible utilizar un sistema como el que planteamos para la detección de lesiones oscuras.

Se han conseguido resultados que pueden ser considerados aceptables como son un 72.4% de accuracy un 72.4% de sensibilidad junto con un 72.35% de especificidad, además de un AUC de 0.7995.

Aquí cabe destacar que al haberse utilizado un kernel lineal debido a su rapidez y sencillez, aún cabe un amplio margen de mejora, lo cual puede incluirse dentro de las líneas de trabajo futuras.

De esta manera, utilizar un SVM con un kernel *radial basis function* (RBF) podría mejorar notablemente los resultados. Además, en nuestro trabajo, aunque se ha extraído la información espacial correspondiente a varios niveles de la pirámide propuesta, solo se ha utilizado la información referente a la base de la misma. Otra línea futura sería utilizar la información de los distintos niveles de la pirámide.

Otra línea prometedora es la de implementar nuestro algoritmo mediante lenguajes como *CUDA* que permitan el procesamiento gráfico (GPU) y el consecuente incremento en la velocidad de las simulaciones, aspecto muy a tener en cuenta ya que en bases de datos con imágenes de gran calidad como es *diaretdb1* el proceso de entrenamiento y clasificación puede tomar una gran cantidad de tiempo y recursos.

Utilizar técnicas de postprocesado para intentar reducir el ratio de falsos positivos sería otra línea futura de mejora.

Por último, como se ha comentado ya, durante este trabajo se ha empleado el software *MATLAB* debido a su enorme cantidad de librerías relacionadas con el tratamiento de Imagen que incorpora, pero la implementación del algoritmo en lenguajes de programación de mayor nivel como C++ o Java sería de gran interés ya que permitiría la realización de un sistema de diagnóstico finalista.

Bibliografía

- [1] A. D. Fleming, K. A. Goatman, S. Philip, J. A. Olson and P. F. Sharp, “Automatic detection of retinal anatomy to assist diabetic retinopathy screening”, in *Biomedical Physics*, vol. 52, pp. 331-345, 2007.
- [2] Diabetes Teaching Center at the University of California, San Francisco <https://www.dtc.ucsf.edu/es>
- [3] “Informe mundial sobre la diabetes” *Organización Mundial de la Salud*, Ginebra, 2016.
- [4] Organización Mundial de la Salud (Julio 2017). <http://www.who.int/mediacentre/factsheets/fs312/es/>
- [5] American Academy of Ophthalmology (AAO) Retina/Vitreous Panel, *Preferred Practice Pattern® Guidelines. Diabetic Retinopathy*, American Academy of Ophthalmology, San Francisco, CA, 2016, Available at: www.aao.org/ppp
- [6] J. A. Swets, “Measuring the accuracy of diagnostic systems,” *Science*, vol. 240, no. 4857, pp. 1285–1293, 1988.
- [7] Smith Gulati, Nutnaree Kleawsirikul, and Bunyarit Uyyanonvara “A review on Hemorrhage Detection Methods for Diabetic Retinopathy Using Fundus Images”, *International Journal of Biological, Ecological and Environmental Sciences (IBJEES) Vol.1 No.6, 2012 ISSN 2277 -4394*
- [8] Matei D, Matei R, “Detection of diabetic symptoms in retinal images using analog algorithms,” in *Proc. 7th WSEAS Int. Conf. Signal Processing, Robotics and Automation*, pp. 198-203, 2008.
- [9] Acharya UR, Lim CM, Ng EYK et al., “Computer-based detection of diabetes retinopathy stages using digital fundus images,” in *Proc. Inst. Mech. Eng.*, pp. 545-553, July 2009
- [10] Marwan DS, Eswaran C, “An automated decision-support system for non-proliferative diabetic retinopathy disease based on MAs and Has detection,” *Comput. Methods Programs Biomed.*, electronically published, April 2012.

- [11] Hatanaka Y, Nakagawa T, Hayashi Y, Hara T, Fujita H, "Improvement of automated detection method of hemorrhages in fundus images," *In: 30th Annu. Int. IEEE EMBS Conf.*, pp. 5429-5432, August 2008.
- [12] Köse C, Şevik U, İkibaş C, Erdöl H, "Simple methods for segmentation and measurement of diabetic retinopathy lesions in retinal fundus images," *Computer Methods Programs Biomed.*, vol. 107(2), pp. 274-293, August 2012.
- [13] Zhang D, Li X, Shang X, Yi Y, Wang Y, "Robust hemorrhage detection in diabetic retinopathy image," *1st ACPR2011*, pp. 209-213, November 2011.
- [14] Grigorescu C, Petkov N, Westenberg MA, "Contour and boundary detection improved by surround suppression of texture edges," *J. Image Vision Computing*, vol.22(8), pp. 609–622, 2004.
- [15] Zhang X, Chutatape O, "A SVM approach for detection of hemorrhages in background diabetic retinopathy," *In: Proc. Int. Joint Conf. Neural Networks*, pp. 1435-1440, July-August 2005.
- [16] Kande GB, Savithri TS, Subbaiah PV, Tagore MRN, "Detection of red lesions in digital fundus images," *In: 2009 IEEE Int. Symp. Biomed. Imaging: From Nano to Macro*, pp. 558-561, June-July 2009.
- [17] Narasimhan K, Neha VC, Vijayarekha K, "An efficient automated system for detection of diabetic retinopathy from fundus images using support vector machine and bayesian classifiers," *In: 2012 Int. Conf. Computing, Electron. Elect. Technologies*, pp. 964-969, March 2012.
- [18] Esmaeili M, Rabbani H, Dehnavi AM, Dehghani A, "A new curvelet transform based method for extraction of red lesions in digital color retinal images," *in Proc. IEEE 17th Int. Conf. Image Processing*, Hong Kong, pp. 4093-4096, September 2010.
- [19] García M, López MI, Álvarez D, Hornero R, "Assessment of four neural network based classifiers to automatically detect red lesions in retinal images," *Med. Eng. Phys.*, vol. 32(10), pp. 1085-1093, December 2010.
- [20] Sinthanayothin C, Boyce JF, Williamson TH, Cook HL, Mensah E, Lal S et al., "Automated detection of diabetic retinopathy on digital fundus images," *Diabet. Med.*, vol. 19(2), pp. 105-112, February 2002.
- [21] Kauppi, T., Kalesnykiene, V., Kamarainen, J.-K., Lensu, L., Sorri, I., Raninen A., Voutilainen R., Uusitalo, H., Kälviäinen, H., Pietilä, J., "DIARETDB1 diabetic retinopathy database and evaluation protocol", Technical report
- [22] S. Morales, V. Naranjo, J. Angulo, M. Alcaniz, "Automatic detection of optic disc based on PCA and mathematical morphology", *Medical Imaging IEEE Transactions*, vol. 32, pp.786-796, 2013.

[23] S. Morales, V. Naranjo, J. Angulo, J. J. Fuentes, M. Alcaniz, "Segmentation and analysis of retinal vascular tree from fundus images processing", *BIOSIGNALS, SciTePress*, pp.321-324, 2012.

[24] I. Pitas, *Digital Image Processing Algorithms and Applications*. New York (USA): John Wiley & Sons, Inc., 1st ed. 2000.

[25] T.Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.24, no.7, pp. 971-987, 2002.

[26] Adrián Colomer, Pablo Ruiz Valery Naranjo, Rafael Molina and Aggelos K. Katsaggelos, "Hard exudates detection using local texture analysis and Gaussian processes", Instituto de Investigación e Innovación en Bioingeniería (I3B)

[27] Zhenhua Guo, Lei Zhang, David Zhang, "Rotation invariant texture classification using LBP variance (LBPV) with global matching," *Pattern Recognition*, Volume 43, Issue 3, March 2010, Pages 706-719.

[28] Mauro Dalla Mura, *Student Member, IEEE*, Jón Atli Benediktsson, *Fellow, IEEE*, Björn Waske, *Member, IEEE*, and Lorenzo Bruzzone, *Fellow, IEEE*, "Morphological Attribute Profiles for the Analysis of Very High Resolution Images", *IEEE Transactions on Geoscience and remote sensing*, vol.48, no.10, October 2010

[29] Shai Shalev-Shwartz y Shai Ben-David "Understanding Machine Learning", 2014, 32 Avenue of the Americas, New York, NY 10013-2473, USA, Cambridge University Press

[30] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, Cambridge University Press, 1 edition, 2000.

[31] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, Aug. 1996.

[32] Chih-Chung Chang; Chih-Jen Lin, "LIBSVM: A library for support vector machines", *ACM Transactions on Intelligent Systems and Technology* 2 (3) (2011).