

Trabajo de Fin de Grado
Grado en Ingeniería de las Tecnologías de
Telecomunicación

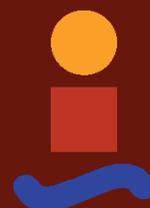
Evaluación de espacios y distancias de color en el
cálculo de los gradientes de color para detección
automática de glaucoma

Autor: Ismael Bugarin Kamour

Tutoras: Irene Fondón García y María Auxiliadora Sarmiento Vega

Dpto. Teoría de la Señal y Comunicaciones
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2018



Proyecto Fin de Carrera
Ingeniería de Telecomunicación

Evaluación de espacios y distancias de color en el cálculo de los gradientes de color para detección automática de glaucoma

Autor:

Ismael Bugarin Kamour

Tutoras:

Irene Fondón García

Profesora Titular.

María Auxiliadora Sarmiento Vega

Profesora Contratada Doctora Interina.

Dpto. de Teoría de la Señal y Comunicaciones

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2019

Proyecto Fin de Carrera: Evaluación de espacios y distancias de color en el cálculo de los gradientes de color para detección automática de glaucoma

Autor: Ismael Bugarin Kamour

Tutoras: Irene Fondón García y María Auxiliadora Sarmiento Vega

El tribunal nombrado para juzgar el trabajo arriba indicado, compuesto por los siguientes profesores:

Presidente:

Vocal/es:

Secretario:

Acuerdan otorgarle la calificación de:

El Secretario del Tribunal

Fecha: Sevilla, 2019

Agradecimientos

En primer lugar, me gustaría agradecer a mi madre porque sin ella no hubiera llegado a donde estoy ahora mismo, darle las gracias también por enseñarme todo lo que sé, por hacerme el hombre que soy. Si de alguien he aprendido a no dejar de luchar nunca es gracias ti.

Quiero agradecer también a mi padre por ayudarme y apoyarme siempre que lo he necesitado. Dándome ánimos en los momentos más difíciles de mi carrera.

Como no, agradecer a mis dos hermanos pequeños que, aunque ya son mayores para mi siguen siendo mis pequeñines, y en especial a mi hermano Anuar ayudándome mucho en este proyecto de fin de grado.

También agradecer a todos esos amigos que he conocido en la carrera, con los que he compartido muchos momentos y que sin ellos esta carrera habría sido mucho más dura. Quiero destacar a 3 personas con las que he pasado largos tiempo de estudios y de trabajos. Javi, Pablo y Ñaño gracias por todo.

Y no podía faltar, dar las gracias a mis tutoras, sin las cuales este proyecto no lo habría finalizado y haber aprendido tanto durante la carrera me hubiera sido imposible, y por último a todos los profesores que han hecho esto posible.

Gracias a todos.

Ismael Bugarin Kamour

Sevilla, 2019

Resumen

En el presente trabajo el objetivo final es evaluar un algoritmo automático para la segmentación de la excavación y el disco óptico en retinografías, basada en distintos gradientes de color y diferentes métricas.

Este método usa los gradientes de color como características principales para la segmentación de la excavación y el disco óptico, zonas de gran dificultad debido a la variedad de tamaños, tonalidades o bordes (éstos, a veces, difusos). La innovación aportada es la comparación entre diferentes espacios de color y métricas que permitirá concluir la necesidad o no del uso de un espacio de color uniforme junto con una distancia avanzada.

Para la clasificación se han usado 60 imágenes procedentes de diferentes bases de datos, con el fin de obtener un clasificador apto para cualquier imagen que se le presente.

Abstract

The main objective of this project is to evaluate an automatic segmentation algorithm for the excavation and the optical disk in retinographies, based on different colour gradients and different metrics.

This method uses the colour gradients as the main characteristic for segmentation of the excavation and the optical disk, areas of great difficulty due to the variety of sizes, shades or edges (those are sometimes diffuse). The innovation provided is the comparison between different color spaces and metrics that will allow to conclude the need or not of the use of a uniform color space along with an advanced distance.

For the classification, 60 images from different databases have been used, in order to obtain a classifier suitable for any image presented.

ÍNDICE

AGRADECIMIENTOS	VII
RESUMEN	X
ABSTRACT	XII
ÍNDICE	XIV
ÍNDICE DE TABLAS	XVI
ÍNDICE DE FIGURAS	XIX
1 OBJETIVO	1
2 INTRODUCCIÓN	3
2.1. SISTEMA VISUAL HUMANO.....	3
2.1.1 Nivel periférico.....	3
2.1.2 Nivel central.....	4
2.2. GLAUCOMA.....	5
2.3. RETINOGRAFÍA.....	5
2.4. ESPACIOS DE COLOR.....	6
2.4.1 Introducción.....	6
2.4.1.1 Luz.....	6
2.4.1.2 Color.....	7
2.4.2 Espacios de Color.....	8
2.4.2.1 CIE RGB.....	8
2.4.2.2 CIE L*a*b*.....	10
2.4.2.3 Espacio de color Gris.....	11
2.4.3 Distancias de Color.....	12
2.5. CLASIFICADORES.....	14
2.5.1 Árboles de Decisión.....	14
2.5.2 K-Nearest Neighbors (k-NN).....	15
2.5.3 Support Vector Machine (SVM).....	17
3 MÉTODO Y MATERIALES	21
3.1 IMÁGENES UTILIZADAS.....	22
3.2 ROI UTILIZADAS.....	23
3.3 EXTRACCIÓN DE CARACTERÍSTICAS POR PÍXEL DE LA IMAGEN.....	24
3.4 CLASIFICACIÓN	29
4 RESULTADOS	32
4.1 RESULTADOS GRIS EUCLÍDEA.....	34
4.2 RESULTADOS RBG EUCLÍDEA.....	39
4.3 RESULTADOS CIE L*a*b* EUCLÍDEA.....	44
4.4 RESULTADOS CIE94.....	49
4.5 RESULTADOS CIE2000.....	54
5 CONCLUSIONES.....	60
6 REFERENCIAS	61

ÍNDICE DE TABLAS

Tabla 1. Condiciones de Referencia	13
Tabla 2. Bases de datos utilizadas	22
Tabla 3. Matriz de Confusión	32
Tabla 4. Simple Tree Gris	34
Tabla 5. Medium Tree Gris	34
Tabla 6. Complex Tree Gris	35
Tabla 7. Fine KNN Gris	35
Tabla 8. Medium KNN Gris	36
Tabla 9. Coarse KNN Gris	36
Tabla 10. Cosine KNN Gris	37
Tabla 11. Weighted KNN Gris	37
Tabla 12. Linear SVM Gris	38
Tabla 13. Quadratic SVM Gris	38
Tabla 14. Simple Tree RGB	39
Tabla 15. Medium Tree RGB	39
Tabla 16. Complex Tree RGB	40
Tabla 17. Fine KNN RGB	40
Tabla 18. Medium KNN RGB	41
Tabla 19. Coarse KNN RGB	41
Tabla 20. Cosine KNN RGB	42
Tabla 21. Weighted KNN RGB	42
Tabla 22. Linear SVM RGB	43
Tabla 23. Quadratic SVM RGB	43
Tabla 24. Simple Tree CIE L*a*b*	44
Tabla 25. Medium Tree CIE L*a*b*	44
Tabla 26. Complex Tree CIE L*a*b*	45
Tabla 27. Fine KNN CIE L*a*b*	45
Tabla 28. Medium KNN CIE L*a*b*	46
Tabla 29. Coarse KNN CIE L*a*b*	46
Tabla 30. Cosine KNN CIE L*a*b*	47
Tabla 31. Weighted KNN CIE L*a*b*	47
Tabla 32. Linear SVM CIE L*a*b*	48
Tabla 33. Quadratic SVM CIE L*a*b*	48
Tabla 34. Simple Tree CIE94	49
Tabla 35. Medium Tree CIE94	49
Tabla 36. Complex Tree CIE94	50

Tabla 37. Fine KNN CIE94	50
Tabla 38. Medium KNN CIE94	51
Tabla 39. Coarse KNN CIE94	51
Tabla 40. Cosine KNN CIE94	52
Tabla 41. Weighted KNN CIE94	52
Tabla 42. Linear SVM CIE94	53
Tabla 43. Quadratic SVM CIE94	53
Tabla 44. Simple Tree CIE2000	54
Tabla 45. Medium Tree CIE2000	54
Tabla 46. Complex Tree CIE2000	55
Tabla 47. Fine KNN CIE2000	55
Tabla 48. Medium KNN CIE2000	56
Tabla 49. Coarse KNN CIE2000	56
Tabla 50. Cosine KNN CIE2000	57
Tabla 51. Weighted KNN CIE2000	57
Tabla 52. Linear SVM CIE2000	58
Tabla 53. Quadratic SVM CIE2000	58
Tabla 54. Resultados finales en tanto por ciento	59
Tabla 55. Tiempos de ejecución de los clasificadores.	59

ÍNDICE DE FIGURAS

Figura 1. Esquema General	1
Figura 2. Partes del Ojo Humano	3
Figura 3. Corte del Cerebro	4
Figura 4. Fondo del Ojo	6
Figura 5. Espectro	7
Figura 6. Diagrama de Chromacity	8
Figura 7. Representación del Espacio L*a*b*	11
Figura 8. Escala de Grises	11
Figura 9. Método Árbol	15
Figura 10. Método k-NN con distancia euclidea	16
Figura 11. Diagrama de Flujo	21
Figura 12. Imágenes de las Bases de Datos	22
Figura 13. Construcción de la verdad de referencia	23
Figura 14. Ejemplo de una ROI con diferentes gradientes	27
Figura 15. Cambios de color representados por gradiente	27
Figura 16. Matriz de características	29

1 OBJETIVO

El objetivo de este Trabajo Fin de Grado es evaluar un algoritmo automático de segmentación del nervio óptico y su excavación en retinografías, adaptado a la percepción humana del color. Para ello se han empleado diferentes gradientes de color y distintas métricas que son: RGB con distancia euclídea, CIE L*a*b* junto con la distancia euclídea, imagen en escala de gris y distancia euclídea y CIE L*a*b* con CIEDE 2000 y CIEDE 94 (estas dos últimas fundamentadas en CIE L*a*b*). Para la clasificación se han utilizado varios clasificadores tales como: “decision trees”, “K-nearest neighbors” (k-NN) y “support vector machine” (SVM) con el fin de ver cual obtiene una mejor precisión.

El algoritmo debe emular el diagnóstico realizado por un médico experto, en este caso el oftalmólogo. Por tanto, suponemos que la información de color debe ser análoga con la percepción humana y las diferencias entre colores. Gracias a este algoritmo se pretende que con una simple imagen se pueda determinar si el paciente padece de glaucoma, siendo una herramienta muy útil para la intervención del oftalmólogo. Es importante conseguir que este diagnóstico pueda hacerse automáticamente, ya que se pretende que la intervención humana sea mínima. Esto agiliza la detección y curación de este tipo de patologías.

El método utilizado es la segmentación de la excavación basada en imágenes de color de la retina, debido a que el posterior cálculo de la relación disco-copa (CDR), medida necesaria para el diagnóstico, se realiza por comparación entre el diámetro de la copa y el diámetro total del disco óptico.

Inicialmente estimaremos una región de interés de la imagen original, posteriormente para cada píxel de esa región de interés extraeremos las características espaciales y de color y su distancia al centro, para construir el vector de características. Este último paso tiene el mayor peso del trabajo, ya que para realizarlo utilizamos varios gradientes de color para ver cuál de ellos es el más conveniente usar. Finalmente emplearemos varios clasificadores que nos proporciona Matlab para determinar cuál es mejor. En la Figura 1 podemos observar un esquema general de los pasos seguidos en el proyecto.



Figura 1. Esquema General

2 INTRODUCCIÓN

El glaucoma es un trastorno ocular en el que el nervio óptico sufre daño, deteriora constantemente la visión en el ojo afectado y progresa hasta completar la ceguera si no se trata. A menudo, se asocia con una mayor presión del fluido intraocular [1].

2.1. Sistema Visual Humano

El Sistema Visual Humano (SVH) es el encargado de transformar las ondas electromagnéticas, que pertenecen al espectro visible, en señales nerviosas que son interpretadas por el cerebro [2].

Para definir el SVH vamos a distinguir entre dos niveles, el nivel periférico constituido por el ojo y el nivel central que está formado por el nervio óptico y el cerebro [3]. Aunque solemos asociar el ojo humano con el órgano de la visión, sería más correcto decir que es donde comienza la visión. Podríamos decir que “miramos con los ojos” pero “vemos con el cerebro” [4]. En la Figura 2 podemos ver una imagen del ojo humano en la que se indica donde se encuentran cada una de sus partes.

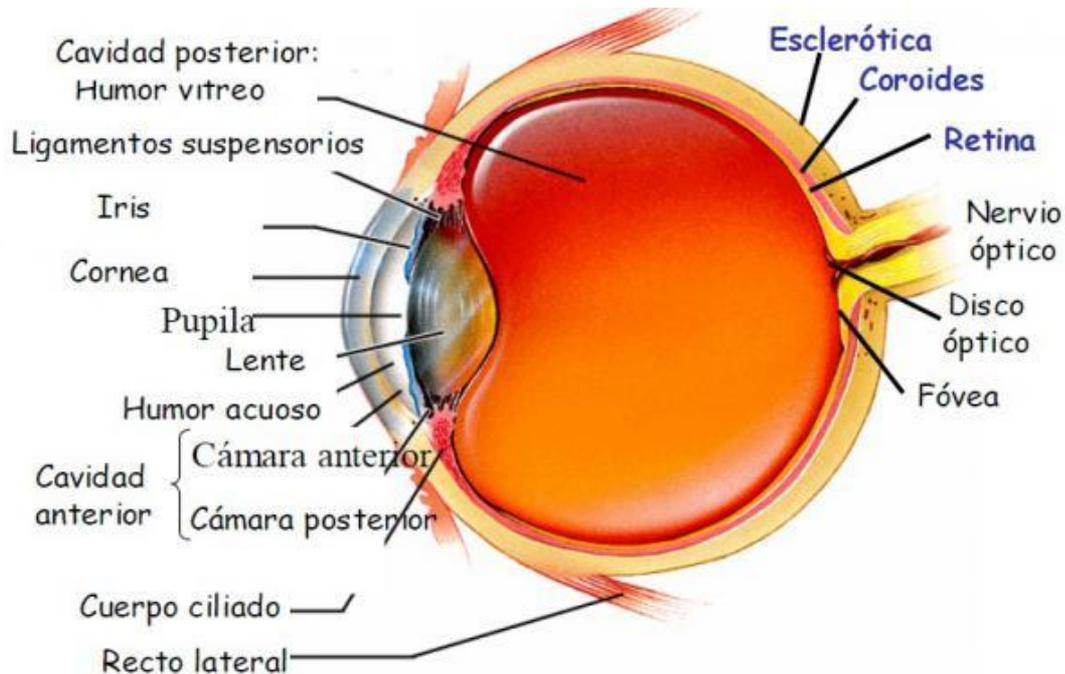


Figura 2. Partes del Ojo Humano

<http://lizyzulema.blogspot.es/1476312607/sistema-visual/>

2.1.1 Nivel periférico

El ojo humano es el encargado de convertir la luz en señales neuronales. Tiene una estructura prácticamente esférica en la que la luz entra por la pupila.

La cubierta externa del ojo es opaca y su interior es traslúcido, formada por dos membranas que dan forma al ojo. La esclerótica es una membrana blanca que en la parte anterior del ojo es transparente, dando lugar a la córnea, a través de la cual los rayos luminosos son orientados a la retina.

Detrás de la córnea existe una cavidad llena de un líquido nutritivo para ésta, el humor acuoso, esta cámara está limitada por una membrana circular de tejido muscular, el iris, que deja en su centro una apertura circular, la pupila. Gracias a su musculatura el iris puede regular el diámetro de la pupila, así el paso de luz

que llega a la retina funciona como un diafragma, además da el color a los ojos. Tras el iris tenemos el cristalino que permite el enfoque de la imagen en los fotorreceptores de la retina, funcionando como una lente. La luz después de atravesar el cristalino debe cruzar una segunda cavidad antes de alcanzar a la retina. Esa cámara está llena de un líquido llamado humor vítreo y sus paredes están limitada por una membrana denominada retina [5].

La retina es la encargada de convertir las imágenes en señales neurológicas. La retina tiene varias capas celulares donde podemos encontrar los fotorreceptores: los conos y los bastones.

- Conos: son los que proporcionan la información del color. Hay 3 tipos, los sensibles al rojo, al verde y al azul. Aportan la visión espacial, visión fotópica. Cada cono tiene una conexión directa al cerebro, sensibles con altos niveles de iluminación. Hay alrededor de unos seis millones en la retina.
- Bastones: son los que proporcionan información del brillo. Encargados de la visión a bajos niveles de iluminación, visión escotópica, no aportan información espacial. Al contrario que los conos, varios bastones comparten la misma fibra en el nervio óptico. Hay alrededor de 100 millones en la retina.

Otras partes importantes de la retina son el punto ciego y la fóvea. El punto ciego es el lugar de donde parte el nervio óptico, no tiene células sensibles. La fóvea es la zona central de la retina, donde se enfocan la imagen, contiene casi totalidad de los conos, también se conoce como mancha amarilla y es muy importante en el estudio del glaucoma.

2.1.2 Nivel central

Está formado por el nervio óptico y el cerebro, es el encargado de procesar las señales neuronales llegadas de la retina y las convierte en información visual, por este motivo dijimos anteriormente que “vemos con el cerebro”, aunque la información de la imagen entre por los ojos no es procesada ni interpretada hasta llegar al cerebro. Su funcionamiento es casi totalmente desconocido.

El nervio óptico es el nervio que conecta cada uno de los globos oculares con la parte del cerebro encargada de la visión, situada en el lóbulo occipital. Su función es la de transportar las imágenes recibidas por el sistema visual humano, transformadas en impulsos nerviosos, hasta el cerebro donde se procesan. El nervio óptico parte del disco óptico, que se encuentra en la retina, que es el tejido sensible a la luz que se localiza en la parte interior del ojo. Se producen una serie de fenómenos químicos y eléctricos por los que las imágenes se transforman en impulsos nerviosos [6].

En cuanto al cerebro nos vamos a centrar en el lóbulo dedicado a la visión, lóbulo occipital, está situado en la zona posterior del cerebro como vemos en la Figura 3, por detrás de los lóbulos parietal y temporal. Encargado del procesamiento visual, está formado por 3 áreas visuales, la primaria, la secundaria y la terciaria [7].

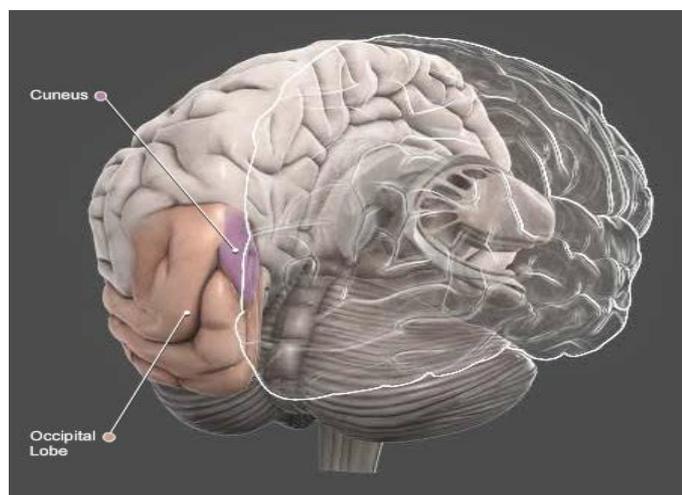


Figura 3. Corte del Cerebro

<http://www.g2conline.org/>

2.2. Glaucoma

Podríamos definir el glaucoma como una enfermedad del nervio óptico crónica, progresiva e irreversible que puede deberse a varias causas, cuyo factor de riesgo más importante es la presión intraocular (PIO) elevada. Es una de las enfermedades más importantes ya que supone la segunda causa de ceguera en los países desarrollados, el glaucoma no tiene síntomas, no causa dolor.

Solo podemos hablar de glaucoma cuando el nervio óptico está afectado. Por otro lado, la hipertensión ocular únicamente ocurre cuando la PIO está elevada, está determinada por la velocidad de producción del humor acuoso, y por la resistencia y dificultad en su drenaje.

Existen una gran variedad de tipos de glaucoma, aunque en términos generales lo podemos clasificar según su origen (primario o secundario) y según la amplitud angular (ángulo abierto o cerrado) [8].

- Glaucoma primario de ángulo abierto: es el más usual, suele aparecer a partir de la mediana edad, es de evolución lenta y no presenta síntomas hasta que la enfermedad está muy avanzada, produciendo una pérdida de la visión. Su diagnóstico se da en controles rutinarios del oftalmólogo.
- Glaucoma de ángulo cerrado: puede presentarse de forma aguda o crónica. Suele causar gran dolor ocular, enrojecimiento del ojo y visión borrosa. Tiene un elevado riesgo de pérdida de visión.
- Glaucomas secundarios: grupo heterogéneo de glaucomas. Hay una causa concreta, suele ser otra enfermedad ocular, que genera la elevación de la presión intraocular.
- Glaucoma congénito: se da durante los primeros días o meses de vida del paciente, producido por anomalías del desarrollo del ojo durante la gestación [9].

Para detectar el glaucoma existen diferentes pruebas que deben llevarse a cabo por el médico especialista:

- Tonometría: es la medida de la PIO, se mide con un tonómetro.
- Paquimetría: es la medida del grosor corneal, la tonometría puede estar influida por este grosor.
- Gonioscopia: permite hacer una evaluación del ángulo camerular, que es el lugar anatómico por donde debe drenar el humor acuoso. Según su aspecto se establece el tipo de glaucoma, de ángulo abierto o de ángulo cerrado.
- Oftalmoscopia: permite la evaluación directa del nervio óptico detectando los signos de glaucoma, para hacer un buen examen se requiere dilatar la pupila.
- Tomografía óptica de coherencia: cuantifica el número o la cantidad de fibras nerviosas que están lesionadas en el nervio óptico, es una técnica 3D de uso muy costoso.
- Campimetría: permite hacer una evaluación del campo visual de cada ojo. Se trata de una máquina que emite luces de distinta intensidad acompañados de un sonido, es una prueba subjetiva [10].

2.3. Retinografía

En el presente proyecto la retinografía es la técnica usada para diagnosticar el glaucoma, por ello vamos a describirla con mayor profundidad.

La retinografía es una técnica que, a modo de fotografía, permite adquirir una imagen del fondo del ojo o la retina. Mediante este procedimiento se pueden detectar diferentes enfermedades que afectan a la retina.

Es una prueba sencilla, indolora, útil y segura, la única molestia para el paciente es que hay que dilatar la pupila mediante gotas [11].

Esta técnica es muy parecida a la Tomografía, pero en 2D. Aunque no sea en 3D resulta mucho más eficiente y barato que la técnica anterior. Es muy útil para el estudio del nervio óptico y del daño producido por el glaucoma. Usamos el método llamado excavación papilar. Existe una regla muy básica: medimos el diámetro del nervio óptico y se compara con el diámetro de la copa. La relación entre el borde de excavación y el disco consiste en analizar si el tamaño del borde es relevante en función del disco óptico. Para ello, se realiza el cociente entre el diámetro (CDR) de ambos, dando lugar a un valor entre 0 y 1. Para poder diferenciar entre un paciente sano u otro con posibilidad de tener glaucoma, se considera glaucomatoso si su CDR supera 0,5, sin embargo, cuando este cociente es menor de 0,3 es normal; entre 0,3 y 0,6 es sospechoso; y si es mayor de 0,6, patológico [12]. En la Figura 4 tenemos dos imágenes: la (a) pertenece a un ojo sano y la (b) es un ojo con glaucoma. En rojo está marcado el disco óptico y en azul la copa óptica.

Las imágenes obtenidas en la prueba pueden almacenarse mediante sistemas informáticos y así compararse en el tiempo, esto nos permite evaluar la evolución de estas enfermedades a lo largo del tiempo.

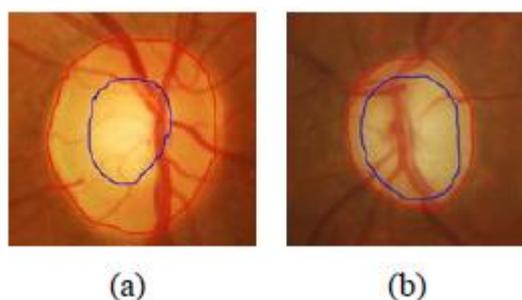


Figura 4. Fondo del Ojo; (a) ojo sano; (b) ojo con glaucoma [12]

2.4. Espacios De Color

2.4.1 Introducción

Podríamos definir el espacio de color como un modelo con el que se pretende describir la percepción humana del color. En un espacio de color se deben poder formar relaciones entre los diferentes colores, independientemente de sus intensidades, saturaciones, etc [13]. Empecemos desde el principio, vamos a separar los conceptos de luz y color por la relevancia de cada uno, aunque guardan relación entre ellos.

2.4.1.1 Luz

La luz es la forma de energía más común y conocida. Tiene dos aspectos fundamentales: el primero es un aspecto de onda electromagnética, y el segundo es corpuscular (teoría de los fotones). La luz es emitida por el sol como fuente principal, tiene una longitud de onda de 380×10^{-9} m y una frecuencia de 7.89×10^{14} Hz [14].

Al igual que los otros tipos de ondas, la luz experimenta los fenómenos de reflexión y refracción. La reflexión de luz es el cambio que percibe el rayo de luz cuando incide sobre el área que separa dos medios diferentes sin abandonar el medio por el cual se propaga. La refracción de la luz es el cambio de dirección de un rayo luminoso al pasar de un medio a otro de distinta densidad, a través del cual se propaga a distinta velocidad [15].

La frecuencia de la luz es muy alta y por ello los valores de medida los expresamos en longitudes de onda. La luz visible oscila entre 380 y 760 nm. Con la longitud de onda se determina el color, desde el rojo hasta el violeta formando el espectro de color. La mayoría de las fuentes de luz son policromáticas. El ojo percibe la amplitud de las ondas como luminancia [16].

2.4.1.2 Color

El color depende de la frecuencia de la luz, que a su vez depende de la longitud de caracterizado por la relación: $\lambda = CT$, donde λ representa la longitud de onda, C la longitud de los frentes de onda y T indica el período de la onda.

Denominamos espectro a la agrupación de todas las longitudes de onda que forman la radiación policromática, en la Figura 5 vemos una representación del espectro. El ojo humano no es capaz de distinguir los diversos componentes de esta radiación y percibe solamente las longitudes de onda entre 380 y 760 nanómetros, a este rango se le denomina espectro visible. La radiación con longitudes de onda menores de 380 nm se denomina radiación ultravioleta, mientras que las que superan los 760 nm se les llama radiación infrarroja [14].

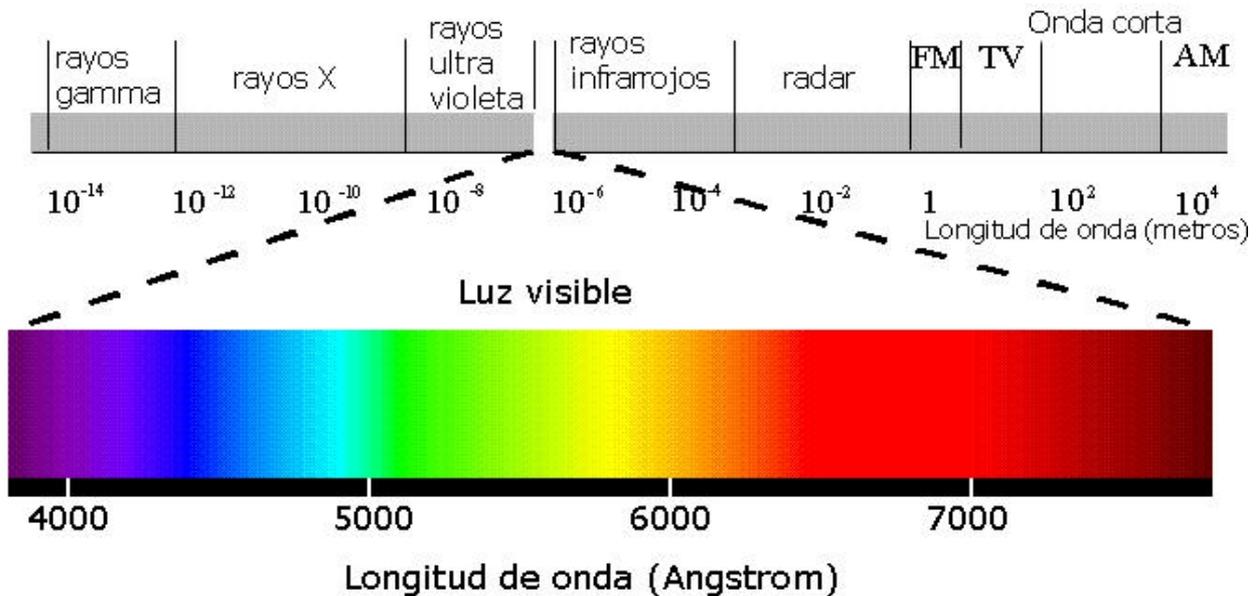


Figura 5. Espectro

<http://reflexionesfotograficas.blogspot.com/2011/01/el-espectro-visible-los-colores.html>

Se puede llegar a separar los colores del espectro con un prisma de cristal. Si quisiéramos el color blanco se consigue con la superposición de todas las frecuencias del espectro visible, en cambio, el color negro es la ausencia de componente espectral en la franja visible.

Los seres humanos podemos llegar a distinguir miles de colores y sin embargo sólo dos docenas de niveles de gris. La percepción del color se basa en la teoría tricromática, es posible conseguir cualquier color mezclando los tres colores primarios.

Newton pudo comprobar que el color no es una propiedad de los objetos, sino que la superficie de un objeto refleja o absorbe las distintas longitudes de onda, recibiendo únicamente los colores reflejados (los objetos rojos absorben todas las longitudes de onda menos la del rojo). Rojo, verde y azul son los colores primarios aditivos del espectro de color. Al variar la cantidad de luz roja, verde y azul es posible producir todos los colores del espectro visible [17].

Las características más importantes del color son:

- Matiz: es el color reflejado o transmitido a través de un objeto. Está directamente relacionado con la longitud de onda, lo que le da nombre del color. Se mide como un ángulo en grados, entre 0° y 360°.
- Saturación: se refiere a la intensidad de un matiz y es una medida de la pureza del color. Se mide como porcentaje entre 0% (gris) y 100% (saturación completa).

- Brillo: El brillo es la luminosidad u oscuridad relativa del color y normalmente se expresa como un porcentaje comprendido entre 0% (negro) y 100% (blanco) [18].

2.4.2 Espacios de Color

A consecuencia de la teoría tricromática y de las características del ojo humano, todos los colores que reconocemos son una combinación de los llamados colores primarios. La finalidad de un modelo de color es hacer más fácil la especificación de los colores de una forma normalizada. Un modelo de color es la especificación de un sistema de coordenadas tridimensional y de un subespacio de este sistema en el que cada color queda representado por un único punto [19]. La colorimetría es una ciencia que estudia y desarrolla el color para obtener una escala de valores numéricos.

2.4.2.1 CIE RGB

En 1920 W. David Wright y John Guild realizaron una serie de experimentos sobre la vista humana, a partir de los cuales trataron de determinar los colores dándole una base matemática. Sabemos que el ojo reacciona a los tres primarios RGB por separado, y los colores que vemos son simplemente sensaciones que estas longitudes de onda causan en nosotros. En un gráfico no pueden ocupar el mismo espacio [19].

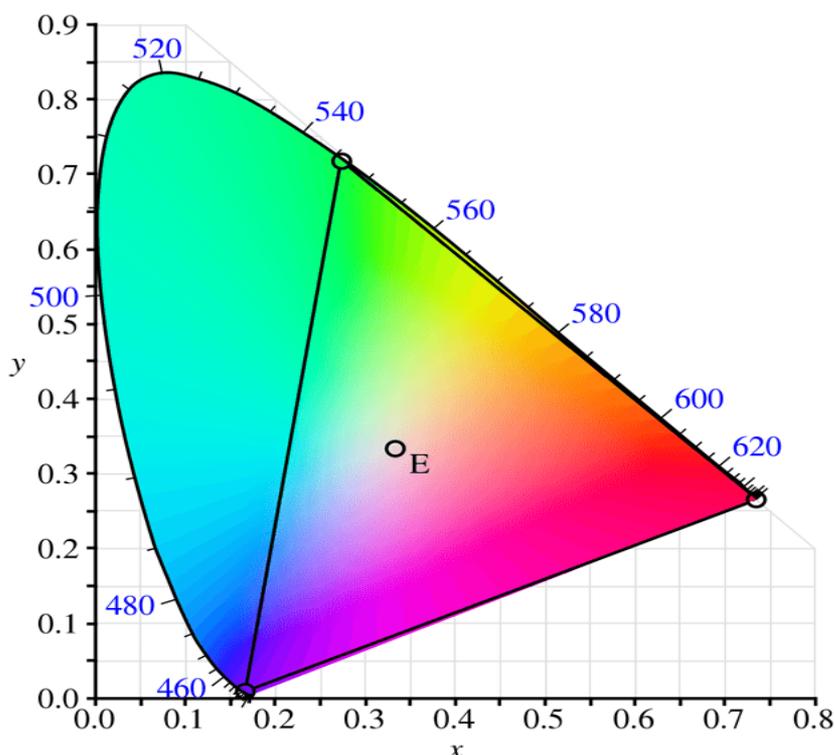


Figura 6. Diagrama de Chromacity

<https://wolfcrow.com/blog/what-is-the-difference-between-cie-lab-cie-rgb-cie-xyy-and-cie-xyz/>

La Figura 6 se corresponde con el diagrama de ‘Chromaticity’, en él podrás observar que el blanco está en el centro y los bordes tienen cada color a la máxima saturación. El triángulo dentro de la superficie se denomina triángulo CIE RGB.

Todos los resultados de los experimentos de Wright y Guild se resumieron mediante las funciones de coincidencia de color CIE RGB estandarizadas $\bar{r}(\lambda)$, $\bar{b}(\lambda)$ y $\bar{g}(\lambda)$.

Las funciones de concordancia de color y las primarias fueron resueltas por el CIE. El ojo humano es capaz de ver luz con longitudes de onda de hasta 810 nm, pero con una sensibilidad mucho menor que la luz verde. Estas funciones de coincidencia de color definen lo que se conoce como el “observador estándar CIE 1931”. Hay que tener en cuenta que, en lugar de especificar el brillo de cada elemento primario, las curvas se normalizan para tener un área constante debajo de ellas. Esta área se fija a un valor particular al definir que [20]

$$\int_0^{\infty} \bar{r}(\lambda) d\lambda = \int_0^{\infty} \bar{g}(\lambda) d\lambda = \int_0^{\infty} \bar{b}(\lambda) d\lambda.$$

(1)

La linealidad de la percepción del color en el ojo humano se expresa en la ley de Grassmann.

- El efecto sobre el ojo de un color cualquiera (C), es el mismo que el producido por la suma aditiva de tres colores cualesquiera (R), (G), y (B), siempre que se mezclen en las proporciones correctas [21]. En la **¡Error! No se encuentra el origen de la referencia.** observamos la gráfica RGB con su representación de cada canal.

$$C = \alpha \cdot (R) + \beta \cdot (G) + \gamma \cdot (B)$$

(2)

- Si se suman dos colores cualesquiera, obteniéndose otro color resultante, este podría haberse obtenido también sumando los primarios de cada uno de los colores origen [20].

$$C_3 = C_1 + C_2$$

(3)

$$C_3 = (\alpha_1 + \alpha_2) \cdot R + (\beta_1 + \beta_2) \cdot G + (\gamma_1 + \gamma_2) \cdot B$$

(4)

Si se expresa en función de la longitud de onda (λ):

$$C(\lambda) = \bar{r}(\lambda)R + \bar{g}(\lambda)G + \bar{b}(\lambda)B$$

(5)

Normalizando las coordenadas anteriores, podemos definir unas nuevas:

$$r(\lambda) = \bar{r}(\lambda) \bar{r}(\lambda) + \bar{g}(\lambda) + \bar{b}(\lambda)$$

(6)

$$g(\lambda) = \bar{g}(\lambda) \bar{r}(\lambda) + \bar{g}(\lambda) + \bar{b}(\lambda)$$

(7)

$$b(\lambda) = \bar{b}(\lambda) \bar{r}(\lambda) + \bar{g}(\lambda) + \bar{b}(\lambda)$$

(8)

Llegando a la conclusión que $r(\lambda) + g(\lambda) + b(\lambda) = 1$. A los coeficientes r, g y b se les denomina coordenadas tricromáticas de un color, en el espacio CIE RGB.

2.4.2.2 CIE L*a*b*

El espacio de color L*a*b*, también conocido como CIELAB, es en este momento uno de los espacios de color más conocidos y uniformes que se usan para evaluar el color de un objeto, además está adaptado como norma "Una Norma Española" (UNE). Este espacio correlaciona los valores numéricos de color en relación con la percepción visual humana. El espacio de color L*a*b* fue modelado en base a una teoría de color antagonista, que establece que dos colores no pueden ser rojo y verde al mismo tiempo o amarillo y azul. L* indica la luminosidad, a* y b* son las coordenadas cromáticas [21]:

- L* = luminosidad, puede tomar valores entre 0 y 100.
- a* = coordenadas rojo/verde (+a indica rojo, -a indica verde).
- b* = coordenadas amarillo/azul(+b indica amarillo, -b indica azul).

Los instrumentos de medición de color pueden cuantificar estos atributos de color fácilmente, determinan el color de un objeto dentro del espacio de color y muestran las coordenadas L*, a*, y b*.

Las proporciones de L*, a* y b* se obtienen de los triestímulos de acuerdo con las siguientes transformaciones [22]:

$$L^* = 116 \cdot \left(\frac{Y}{Y_n}\right)^{\frac{1}{3}} - 16$$

(9)

$$a^* = 500 \cdot \left[\left(\frac{X}{X_n}\right)^{\frac{1}{3}} - \left(\frac{Y}{Y_n}\right)^{\frac{1}{3}} \right]$$

(10)

$$b^* = 200 \cdot \left[\left(\frac{Y}{Y_n}\right)^{\frac{1}{3}} - \left(\frac{Z}{Z_n}\right)^{\frac{1}{3}} \right]$$

(11)

Donde $X_n, Y_n,$ y Z_n son los valores de X, Y y Z con el iluminante que se haya elegido para obtener los valores X, Y y Z de la muestra. Los cocientes son todos superiores a 0,008856, en caso de ser menores se usan las siguientes ecuaciones.

$$L^* = 903.293 \cdot \left(\frac{Y}{Y_n}\right)$$

(12)

$$a^* = 3893.5 \cdot \left(\frac{X}{X_n}\right) - \left(\frac{Y}{Y_n}\right)$$

(13)

$$b^* = 1557.4 \cdot \left(\frac{Y}{Y_n}\right) - \left(\frac{Z}{Z_n}\right) \quad (14)$$

Algunas de las ventajas de CIE L*a*b*:

- El cálculo de las coordenadas es mucho más sencillo.
- Las unidades de las escalas entre las magnitudes CIE L*a*b* son casi iguales, a diferencia de lo que ocurre en las escalas del sistema Munsell.
- Comparado con otros espacios es más rápido hacer correcciones eficientes de color. El hecho de que la luminosidad es completamente ignorada en los canales A y B, hace que sea mucho menos sensible a errores [23].

Una representación visual del espacio L*a*b* lo tenemos en la **¡Error! No se encuentra el origen de la referencia.**, en ella se puede distinguir el paso de '-a' a 'a' o de '-b' a 'b', así como la influencia de variar 'L'. Con una esfera es muy fácil de entender su funcionamiento.

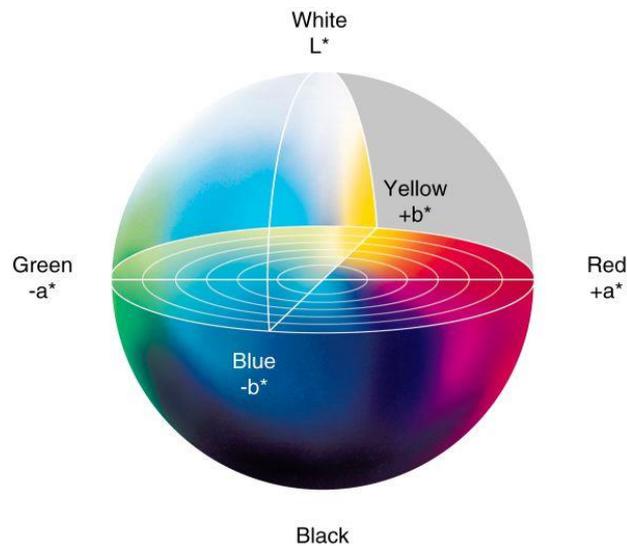


Figura 7. Representación del Espacio L*a*b*

<http://sensing.konicaminolta.com.mx/2014/09/entendiendo-el-espacio-de-color-cie-lab/>

2.4.2.3 Espacio de color Gris

El espacio de color gris utiliza diferentes tonos de gris en una imagen como vemos en la **¡Error! No se encuentra el origen de la referencia.** Por ejemplo, en imágenes de 8 bits puede haber hasta 256 tonos de gris. Cada píxel en escala de grises tiene un valor de brillo comprendido entre 0 que corresponde al negro y 255 que es blanco. En imágenes de 16 y 32 bits, el número de tonos de gris es mucho mayor que en las imágenes de 8 bits [24].

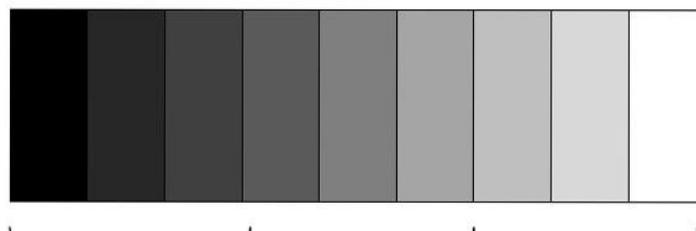


Figura 8. Escala de Grises

<http://cursodibujoypintura.com/abc-del-dibujo-escala-de-valores/>

Los valores de la escala de grises también se pueden medir como porcentajes, 0% es igual a blanco, 100% a negro.

A partir de la escala de gris es posible clasificar cualquier muestra de color de acuerdo con un tono u otro de gris. La claridad como atributo de color puede definirse como aquella cualidad de un color percibido que permite clasificarlo como equivalente a alguno de los grises de una escala de intervalos uniformes que vaya del negro al blanco.

Para hacer la conversión a escala de gris se hace un mezclado de los tres canales RGB para obtener uno solo en gris tomando los siguientes porcentajes: Rojo 30%, Verde 59%, Azul 11%, que es según los expertos lo más parecido a como nuestros ojos captan la intensidad de luz.

Con la ecuación de la luminancia llegamos a la expresión matemática de ese fenómeno. Los factores de ponderación de cada componente nos indican la sensibilidad del ojo humano a las frecuencias del espectro expuestas: $y = R*0.3 + G*0.59 + B*0.11$. Por tanto, para efectuar esta conversión basta con aplicar esta ecuación a cada píxel de una imagen RGB. Dando lugar a una nueva matriz de un byte por píxel que da la información de luminancia [25].

2.4.3 Distancias de Color

La distancia entre dos colores es una métrica de interés en la ciencia del color. Permite el examen cuantificado de una noción que solo se podía describir con adjetivos [26].

Un espacio de color que es uniforme y sus dimensiones son ortogonales, la diferencia entre dos colores se puede obtener como la distancia euclidiana entre sus coordenadas. Los espacios uniformes como CIE $L^*a^*b^*$ intentan proporcionar ese espacio de color uniforme mediante una transformación de los triestímulos XYZ, y al hacerlo, superar la no uniformidad significativa del espacio.

La diferencia de color CIE $L^*a^*b^*$ continúa siendo ampliamente utilizada, pero es inadecuada en ciertas aplicaciones, como aquellas en las que se requiere un solo valor para definir una tolerancia en el espacio de color. Los principales problemas que tiene son [27]:

- Uniformidad de las componentes de la distancia de color: croma (en la diferencia perceptual), matiz (falta de uniformidad) y brillo (muchas variaciones de luminosidad).
- Condiciones de visualización: no se tiene en cuenta las condiciones bajo las que se toman las muestras, ya que los detectores solo registran los estímulos que se les presentan. A medida que se añaden condiciones de visualización, la uniformidad del espacio se ve deteriorada, lo que conlleva dificultades para cuantificar las distancias.
- Superficie de características: el brillo y la textura entre otras características, inducen cambios en la forma en la que se percibe el color.

Todos estos problemas se pueden abordar de dos maneras:

- El espacio de color en sí mismo puede modificarse de una manera que haga que su uniformidad perceptual sea adecuada.
- Las limitaciones pueden superarse al ponderar las ecuaciones de distancia de color, de manera que corrijan las principales causas de no uniformidad.

La mayoría de las ecuaciones de distancia de color que están en uso en la actualidad se basan en una transformación de coordenadas en el espacio de color CIE $L^*a^*b^*$. Las diferentes distancias usadas en este proyecto son:

- **Euclídea:** el medio estándar para determinar distancias es la euclidiana. Si disponemos de una imagen RGB y se desea encontrar la diferencia de color, computacionalmente una de las más fáciles es llamar a las dimensiones lineales R, G, B que definen el espacio de color [26].

$$d = \sqrt{(R_2 - R_1)^2 + (G_2 - G_1)^2 + (B_2 - B_1)^2} \quad (15)$$

Cuando se necesita que el resultado sea computacionalmente simple, es aceptable eliminar la raíz cuadrada y simplemente usar:

$$d^2 = (R_2 - R_1)^2 + (G_2 - G_1)^2 + (B_2 - B_1)^2 \quad (16)$$

Esto es válido en los casos en que un único color se compara con otro color para saber si la distancia es mayor. Si se suman estas distancias de color al cuadrado, dicha métrica se convierte en la varianza de las distancias de color.

Ha habido muchos intentos de ponderar los valores RGB para que se ajusten mejor a la percepción humana, las aproximaciones más cercanas serían los coeficientes de 2, 3 y 4 [26].

$$d = \sqrt{2 \cdot \Delta R^2 + 4 \cdot \Delta G^2 + 3 \cdot \Delta B^2} \quad (17)$$

- **CIE94:** un comité del CIE fue creado para el estudio de la evaluación de las diferencias de color en la industria, publicó sus recomendaciones como CIE Technical Report, en 1995. La nueva fórmula fue conocida como CIE94. El comité propuso una nueva ecuación de distancia de color, cuyas funciones de ponderación se basan en los datos de la RIT/DuPont CMC(l:c), abordando las no uniformidades perceptuales [27].

$$\Delta E_{94}^* = \sqrt{\left(\frac{\Delta L^*}{k_L S_L}\right)^2 + \left(\frac{\Delta C_{ab}^*}{k_C S_C}\right)^2 + \left(\frac{\Delta H^*}{k_H S_H}\right)^2} \quad (18)$$

Donde $S_L = 1$, $S_C = 1 + 0.045C_{ab}^*$, $S_H = 1 + 0.015C_{ab}^*$ y $k_L = 1$, $k_C = 1$, $k_H = 1$ para un conjunto de muestras de referencias, condiciones de visualización e iluminación. Pero puede configurarse de forma diferente según el contexto, en la Tabla 1 podemos ver las condiciones de referencia.

El papel de los factores paramétricos (k_L , k_C y k_H) es permitir que la ecuación sea sensible al ajuste de los componentes de matiz, saturación y brillo de la distancia de color, permitiendo tareas de evaluación que difieran de las condiciones básicas por la cual se creó CIE94 [28].

Iluminación	D65
Illuminance	1000lux
Background	Uniform neutral gray, $L^*=50$
Sample size	Angular subtense greater $\tan 4^\circ$
Sample separation	Direct contact
Sample structure	Visually homogeneous
Magnitude of colour difference	$\Delta E_{ab} \leq 5.0$

Tabla 1. Condiciones de Referencia

P. G. a. L. MacDonald, Colour Engineering, John Wiley & Sons, LTD, 2003

- **CIEDE2000:** El CIE propuso la primera revisión importante de CIE94 en 2001, donde se utilizó el término hue-chroma, similar al de la ecuación de distancia de color de BFD, introduciendo ajustes a las funciones de ponderación S_L y S_H . La revisión se basó en datos experimentales que se acumularon a través de varios estudios diferentes. La ecuación de la distancia de color es la siguiente [27]:

$$\Delta E_{00} = \sqrt{\left(\frac{\Delta L'}{k_L S_L}\right)^2 + \left(\frac{\Delta C'_{ab}}{k_C S_C}\right)^2 + \left(\frac{\Delta H'_{ab}}{k_H S_H}\right)^2 + RT \left(\frac{\Delta C'_{ab}}{k_C S_C}\right) \cdot \left(\frac{\Delta H'_{ab}}{k_H S_H}\right)}$$

(19)

Las correcciones que añadió el CIE sobre CIE94 son:

- 1) Un término de rotación de matiz (R_T), para tratar la región azul problemática.
- 2) Compensación en los colores neutros.
- 3) Compensación por brillo (S_L).
- 4) Compensación por croma (S_C).
- 5) Compensación por tono (S_H).

2.5. Clasificadores

A partir de un conjunto de muestras que importamos en un clasificador, éste es capaz de separar dichas muestras como pertenecientes a una clase u otra. Esto es posible ya que el clasificador se entrena previamente con un conjunto de muestras conocidas, en el cual se definen las diferentes clases que serán clasificadas.

Se pueden implementar dos tipos de clasificadores: los supervisados y los no supervisados. En el entrenamiento de los supervisados es necesario indicar de qué clase es cada vector de características, mientras que los no supervisados lo hacen por sí mismos.

Ningún píxel del conjunto de entrenamiento puede estar presente en el conjunto de clasificación y viceversa. Así, una proporción de píxeles se utiliza para enseñar al clasificador, y el restante, para comprobar la fiabilidad de este. Es recomendable que todos los píxeles estén normalizados.

Si se consigue un porcentaje de acierto admisible, el clasificador será capaz de clasificar píxeles de los cuales no sabemos su clase con una fiabilidad significativa.

En este proyecto usaremos la aplicación de MATLAB 'Classification Learner' que entrena modelos para clasificar datos. Esta aplicación es capaz de explorar el aprendizaje automático supervisado utilizando varios clasificadores. Se puede explorar datos, seleccionar características, especificar esquemas de validación, entrenar modelos y evaluar resultados. Los tipos de clasificadores incluidos son: los árboles de decisión, el análisis discriminante, las máquinas de vectores de soporte, la regresión logística, los vecinos más cercanos y la clasificación de conjuntos [28].

A continuación, explicaremos los clasificadores usados en el trabajo.

2.5.1 Árboles de Decisión

La finalidad de los árboles de decisión es crear un modelo que prediga el valor de una variable de salida en función de diversas variables de entrada. Cada una de las hojas es un valor de la variable de salida y cada rama representa la conjunción de características o condiciones que conducen a ese valor, es decir, para predecir una respuesta hay que seguir las decisiones en el árbol desde el nodo raíz (inicio) hasta un nodo hoja., el nodo hoja contiene la respuesta. Si la variable de salida toma valores continuos, se le denomina árboles de regresión.

En la clasificación de píxeles de este proyecto, cada hoja del árbol representa un tipo de clase, y cada rama, un conjunto de características [29].

Los clasificadores con un solo árbol de decisión que se utilizan son los siguientes:

- **Simple Tree:** es un árbol de decisión simple con pocas hojas que da unos resultados bastante menos finos, debido a su número máximo de divisiones que es 4.
- **Medium Tree:** Es un árbol de decisión de mediana complejidad con algo más de hojas. El número máximo de divisiones es 20. Da mejores resultados que Simple.
- **Complex Tree:** Es un árbol de decisión con muchas hojas que hace una distinción muy buena entre clases. El número máximo de divisiones es 100, da muy buenos resultados.

También hay técnicas que construyen clasificadores con más de un árbol de decisión. Es el caso de los métodos: Boosted y Bagged. En la **¡Error! No se encuentra el origen de la referencia.** observamos el clasificador de Matlab utilizando un único árbol de decisión.

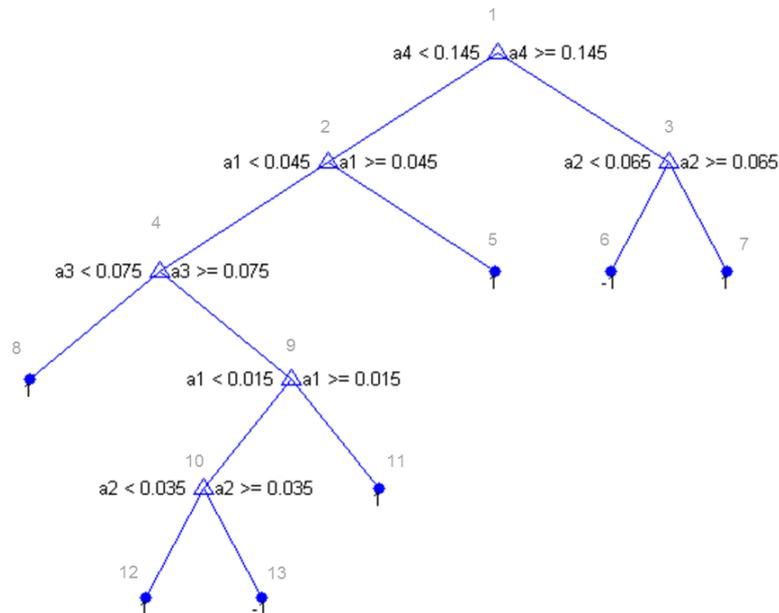


Figura 9. Método Árbol

<https://quantdare.com/arboles-clasificacion-matlab/>

2.5.2 K-Nearest Neighbors (k-NN)

k-NN es un método de clasificación supervisada basado en un conjunto de entrenamientos y prototipos. Sirve para estimar la probabilidad a posteriori de que un elemento x pertenezca a la clase C . Este es un método no paramétrico, que estima el valor de la función de densidad de probabilidad a partir de la información proporcionada por el conjunto de prototipos. Durante el aprendizaje no se hace ninguna suposición acerca de la distribución de las variables predictoras.

Se toma cada vector de características como las coordenadas de un espacio de dimensión N . Si se desea clasificar una nueva muestra entre los k vecinos más próximos, se posiciona y se cuantifican el número de elementos de las diferentes clases que le rodean, es decir, un punto en el espacio es asignado a la clase C si esta es la clase más frecuente entre los k ejemplos de entrenamiento más cercano [30]. Una representación gráfica de cómo funciona este método la tenemos en la **¡Error! No se encuentra el origen de la referencia.**

A la hora de tomar la decisión de a qué clase pertenece cada muestra, influye el tipo de distancia que usemos. Si tenemos una matriz X de datos m_x -por- n y una matriz Y de datos m_y -por- n de vectores fila, las distancias entre el X_s y Y_t con p atributos perteneciente a q clases se definen de las siguientes maneras [31]:

- Distancia euclídea: Caso especial de la distancia de Minkowski, donde $p=2$. Generalmente es la más usada.

$$d_{st}^2 = (X_s - Y_t) - (X_s - Y_t)' \quad (20)$$

- Distancia City Block: Caso especial de la distancia de Minkowski, donde $p=1$

$$d_{st} = \sum_{j=1}^n |X_{sj} - Y_{sj}| \quad (21)$$

- 3. Distancia de Minkowski:

$$d_{st} = \sqrt[p]{\left(\sum_{j=1}^n |X_{sj} - Y_{sj}|^p \right)} \quad (22)$$

donde P puede tomar los siguientes valores:

- $P=1$, d_{ij} es la distancia de City Block (21).
 - $P=2$, d_{ij} es la distancia euclídea (20).
 - $P=\infty$, d_{ij} es la distancia de Chebychev
- 4. Distancia de Mahalanobis:

$$d_{st}^2 = (X_s - Y_t)C^{-1}(X_s - Y_t)' \quad (23)$$

donde C es la matriz de covarianza.

- 5. Distancia del coseno:

$$d_{st} = \left(1 - \sqrt{\left(\frac{X_s Y_t'}{(X_s X_s')(Y_t Y_t')} \right)} \right) \quad (24)$$

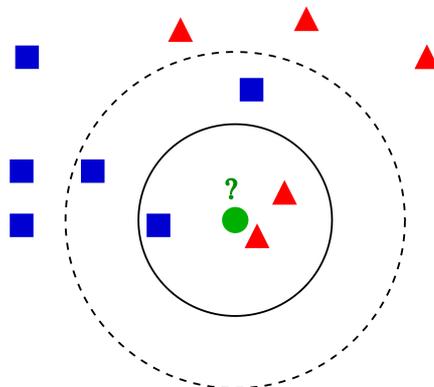


Figura 10. Método k-NN con distancia euclídea

De Antti Ajanki AnAj- Trabajo propio

<https://commons.wikimedia.org/w/index.php?curid=2170282>

2.5.3 Support Vector Machine (SVM)

SVM es un clasificador supervisado desarrollado por V. Vapnik y su equipo en los laboratorios AT&T. La finalidad de SVM es dividir un espacio de muestras, separando las clases a 2 espacios lo más amplios posibles mediante un hiperplano de separación definido como el vector entre los 2 puntos, de las 2 clases, más cercanos al que se llama vector soporte. Cuando las nuevas muestras están a un lado del hiperplano se etiquetarán como una clase, y cuando están al otro como otra clase diferente. Una buena separación entre las clases permitirá una clasificación correcta [32].

Sea el vector de características, con P atributos, perteneciente a la muestra i del conjunto N como en (20)

La información de entrada es clasificada encontrando el mejor hiperplano que separa los puntos de una clase con otra. El mejor clasificador es aquel con mayor distancia, perpendicular al hiperplano, entre dos vectores soporte de clases diferentes sin que ningún otro pertenezca a ella [32]. Observando la Figura podemos llegar a entender mucho mejor este método.

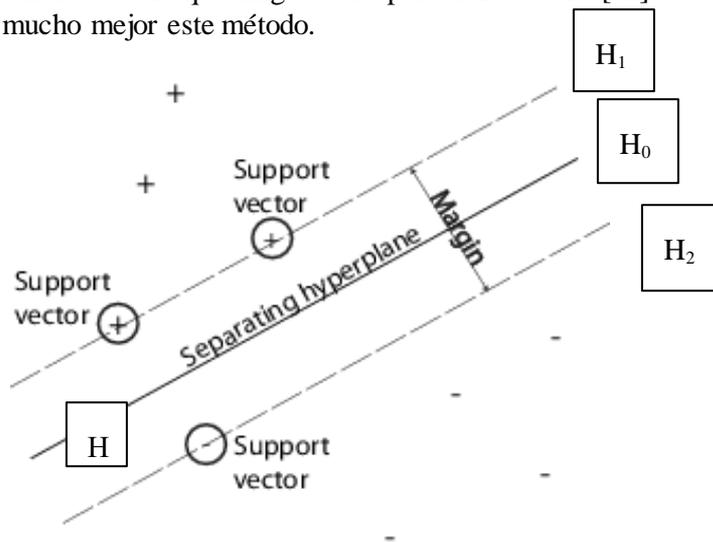


Figura 11. Método SVM

<https://es.mathworks.com/help/stats/support-vector-machines-for-binary-classification.html>

En muchas ocasiones no es una tarea sencilla por lo que se define un margen suave, permitiendo que cierto número de muestras pertenezcan al margen.

Para definir el hiperplano H visto en la Figura 11 tenemos que [33]:

$$\begin{aligned} w \cdot x_i + b &\geq 1, & \text{cuando } y_j = 1 \\ w \cdot x_i + b &\geq -1, & \text{cuando } y_j = -1 \end{aligned} \quad (25)$$

Donde w denota el vector normal al hiperplano, b el término independiente y 'x' e 'y' coordenadas en el espacio. Por lo tanto, los planos H₁ y H₂ se definen como [33]:

$$\begin{aligned} w \cdot x_i + b &= 1 \\ w \cdot x_i + b &= -1 \end{aligned} \quad (26)$$

Si el conjunto de entrenamiento es linealmente separable podemos elegir dos hiperplanos en el borde de los conjuntos de vectores de soporte maximizando su distancia.

La distancia entre H₀ y H₁ es [34]:

$$\left(\frac{|w \cdot x + b|}{\|w\|} \right) = \left(\frac{1}{\|w\|} \right)$$

(27)

Para maximizar el margen, debemos minimizar $|w|$. Desarrollando por el método Lagrange [34]:

$$\min \frac{1}{2} \cdot \|w\|^2 - \sum_i a_i y_i (x_i \cdot w + b) + \sum_i a_i$$

(28)

donde los términos a_i constituyen una representación del vector de pesos en el conjunto de entrenamiento. Analizando el problema primal y dual encontramos que la ecuación del plano es [33]:

$$w = \sum_i a_i y_i x_i$$

(29)

Cuando el problema de separación no es lineal se definen unas funciones o kernels que transforman el espacio de entrada en un espacio que sea separable linealmente.

Los diferentes kernels son: cuadrático, cubico, gaussiano, etc. Dependiendo de la forma del espacio de muestras habrá que seleccionar el kernel más adecuado que se adapte para obtener una separabilidad clara.

3 MÉTODO Y MATERIALES

Una vez acabada la introducción en la cual se han explicado y desarrollado los puntos mínimos necesarios para comprender el presente proyecto, procedemos a explicar los métodos y algoritmos ya implementados cambiando gradientes de color con distintos espacios y métricas también usados en el proyecto para la formación del clasificador. El objetivo principal del clasificador es conseguir un sistema con la capacidad de segmentación de la excavación del nervio óptico y la copa en imágenes de retinografías. El clasificador es a nivel de píxel. En la Figura 11 podemos observar un diagrama de flujo en el cual tendremos una visión general del algoritmo utilizado.

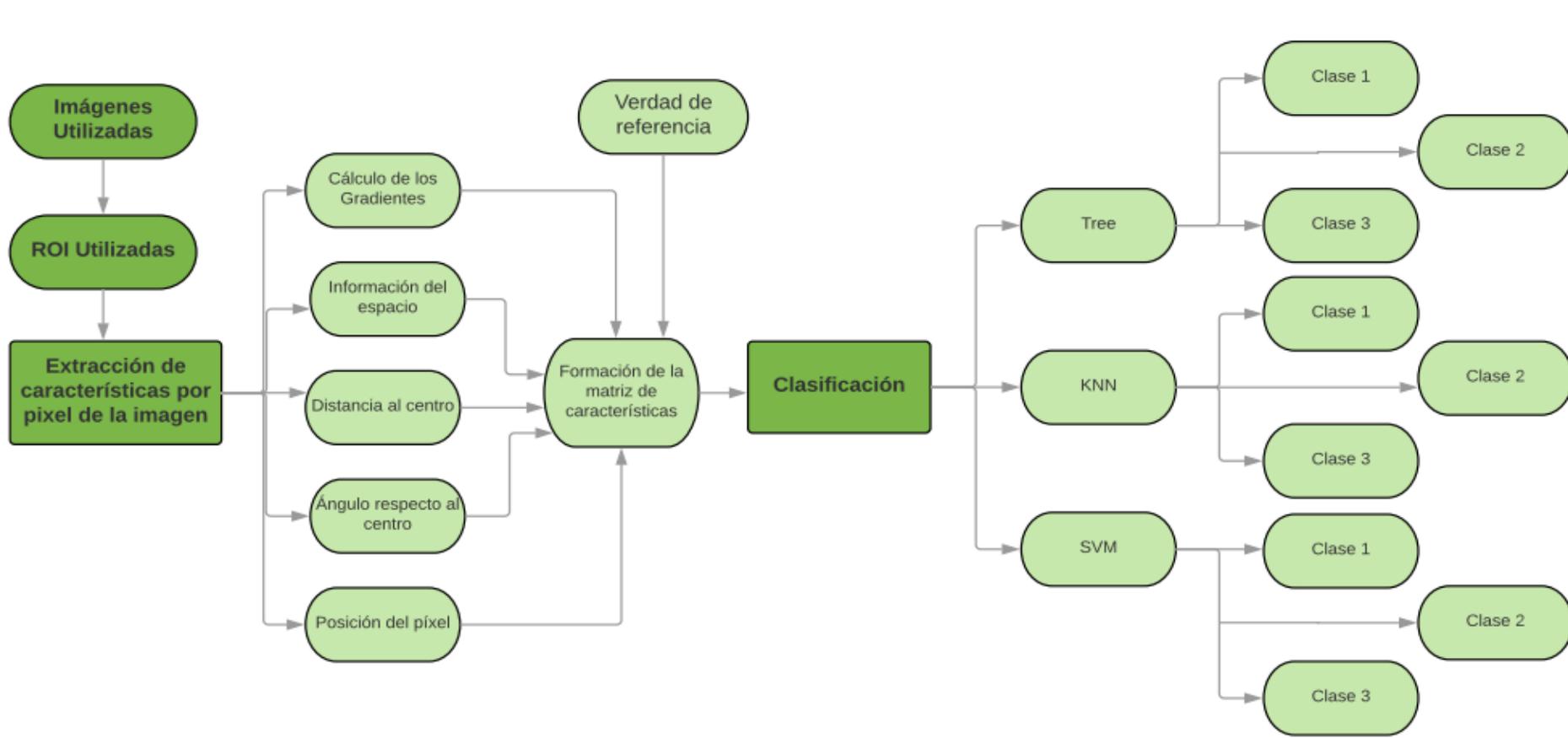


Figura 11. Diagrama de Flujo

3.1 Imágenes Utilizadas

Hemos usado una base de datos con 60 imágenes del fondo de la retina, denominada *dataset1*. Estas imágenes abarcan una gran diversidad de casos, ya que es necesario que sean representativas del contenido que el algoritmo pueda encontrar en la práctica [35].

En la selección de las 60 imágenes, varios expertos exploraron 6 bases de datos diferentes disponibles públicamente las cuales están recogidas en la Tabla 2. Todas ofrecen una amplia gama de iluminación, apariencias o de color. Las bases de datos tenían tanto imágenes sanas, como la de pacientes que sufren glaucoma o retinopatía diabética. Esto hace que el problema sea más realista y difícil. Un ejemplo de las diferentes imágenes seleccionadas lo vemos en la

Figura 12.



Figura 12. Imágenes de las Bases de Datos

	FORMATO	TAMAÑO	ADQUISICIÓN	Nº TOTAL
MESSIDOR	TIFF	900x600	Retinógrafo Topcon TRC NW6	1200
ERLANGEN	JPEG	900x600	Retinógrafo Canon CR-1	45
DRIVE	TIFF	565x584	Retinógrafo Canon CR-5	40
DMED	JPEG	781x600	No disponible	169
STARE	JPG	700x605	No disponible	400
ARIA	JPEG	768x576	No disponible	105

Tabla 2. Bases de datos utilizadas

Grado, T. F. De. (2015). Autora: Ana Isabel García Noguera Tutora: Irene Fondón García. Segmentación automática de la excavación en retinografías basada en gradientes de color y clasificador Complex Tree.

Por lo que contamos con retinografías que:

- A simple vista se ve la excavación.
- Retinografías con excavación difusa.
- Retinografías con vasos sanguíneos dentro y fuera de la excavación

Es necesario tener tanta variedad para obtener un algoritmo fiable, de manera que no sólo detecte la excavación que sea clara, sino también la que sea difusa.

Para evaluar de manera efectiva el rendimiento del algoritmo, los expertos introdujeron manualmente los centros de los diferentes discos ópticos de todas las imágenes, evitando así propagación de errores.

Al no disponer de la verdad de referencia de las retinografías obtenidas de las diferentes bases de datos, construyeron una para poder realizar el entrenamiento.

Partiendo de la ROI seleccionaron la zona que corresponde con el borde de la copa, así como la que corresponde con el borde del disco óptico (todo el procedimiento se hizo de forma manual por un experto).

Cuando tenían diferenciada las distintas zonas, crearon una imagen con 3 valores de gris: el 0 si no es borde de copa ni borde de disco óptico, 127 si es borde de disco óptico y 255 si es borde de copa. Se observa claramente en la Figura 13 (b) que el negro es el valor 0, el gris es 127 y blanco el 255 [33].

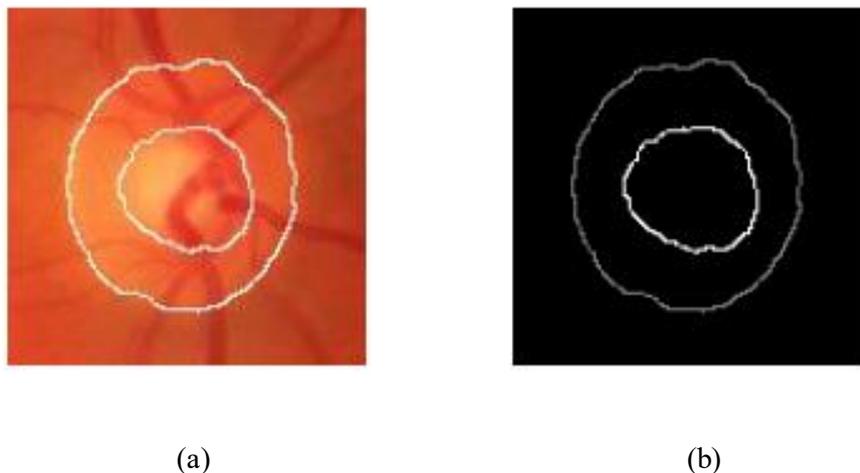


Figura 13. Construcción de la verdad de referencia

(a) ROI original; (b) Verdad de referencia

Es importante señalar que las retinografías usadas en el entrenamiento no se deben utilizar para la clasificación. Además, es fundamental que las imágenes estén bien ajustadas con mismo formato, iluminación y tamaño para conseguir una buena clasificación, por ello es necesario realizar una serie de cambios y extraer cierta información de las imágenes.

3.2 ROI Utilizadas

Para reducir el tiempo computacional se han seleccionado automáticamente de cada imagen una región de interés (ROI). Las ROI tienen las siguientes características [35]:

- Forma cuadrada.
- Centrado en el disco óptico.

- Área equivalente a 1/7 del tamaño de la retina.

Una vez tenemos todas las ROI el siguiente paso sería extraer de ella sus diferentes características que serán necesarias para crear la matriz. Dicha matriz es la que utilizamos para entrenar el clasificador, lo que buscamos es realizar una k-cross validation.

3.3 Extracción de Características por Píxel de la Imagen

La parte más importante en este proyecto es calcular el gradiente de color de cada imagen, es decir, su derivada. La derivada de una señal continua nos aporta las modificaciones locales, de forma que cuanto mayor es la derivada, mayor son estas variaciones. En nuestro caso como usamos imágenes tenemos funciones bidimensionales $I(x,y)$, la derivada es el vector que apunta a la dirección de máxima variación de $I(x,y)$. Este vector se denomina gradiente.

$$\nabla I(x, y) = \begin{bmatrix} \frac{\partial I(x, y)}{\partial x} \\ \frac{\partial I(x, y)}{\partial y} \end{bmatrix} \quad (30)$$

En el caso discreto, las diferentes aproximaciones del operador gradiente se basan en diferencias entre los niveles de la imagen. Su derivada parcial se puede aproximar por la diferencia de píxeles adyacentes de la misma fila.

$$\frac{\partial I(x, y)}{\partial x} \approx \nabla_x I(x, y) = I(x, y) - I(x - 1, y) \quad \begin{array}{|c|c|} \hline -1 & 1 \\ \hline \end{array} \quad (31)$$

$$\frac{\partial I(x, y)}{\partial y} \approx \nabla_y I(x, y) = I(x, y) - I(x, y - 1) \quad \begin{array}{|c|} \hline -1 \\ \hline 1 \\ \hline \end{array} \quad (32)$$

Para el computo de las derivadas direccionales, hemos utilizado una máscara de Sobel en cada espacio de color con su respectiva distancia. En general podríamos caracterizar la siguiente función entera discreta, donde consideramos las coordenadas de un píxel de la imagen (x,y) denotadas como:

$$I(x, y) = [CP_1(x, y), CP_2(x, y), CP_3(x, y)] \quad (33)$$

Donde $CP_1(x, y)$, $CP_2(x, y)$ y $CP_3(x, y)$ corresponden con los planos de color y (x, y) a la ubicación de los píxeles. Por ejemplo, en el espacio de color RGB [35]:

$$\begin{aligned} CP_1(x, y) &= R(x, y) \\ CP_2(x, y) &= G(x, y) \\ CP_3(x, y) &= B(x, y) \end{aligned} \quad (34)$$

Para obtener las derivadas direccionales tenemos que restar dos vectores, uno contiene los coeficientes positivos de la máscara de Sobel, lo denotamos como V_+ , y el otro los coeficientes negativos, denotado como V_- , la

estructura de la máscara es la siguiente:

$$\begin{bmatrix} I_1 & I_2 & I_3 \\ I_4 & I_5 & I_6 \\ I_7 & I_8 & I_9 \end{bmatrix} \quad (35)$$

Si despejamos para 0° obtenemos:

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (36)$$

Si despejamos para 45° obtenemos:

$$\begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix} \quad (37)$$

Concretamente si queremos obtener las derivadas debemos de restar los coeficientes positivos con los negativos, para comprenderlo mejor haremos un ejemplo con la máscara de Sobel para 0° y usando CIE94.

Según (33) tenemos:

$$V^+ = I_1 + 2I_2 + I_3 \quad (38)$$

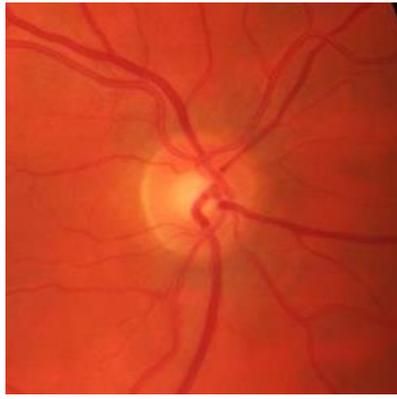
$$V^- = -I_7 - 2I_8 - I_9 \quad (39)$$

Esto da lugar a la distancia entre ambos vectores, $\Delta E_{94} = (V_+, V_-)$, de manera que si queremos la diferencia de color de la imagen:

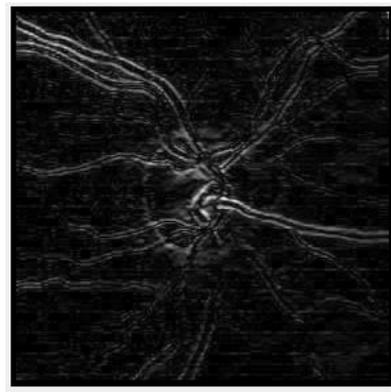
$$d = |\Delta E_{94} = (V_+, V_-)| \quad (40)$$

Donde de d es la norma de las diferencias de color.

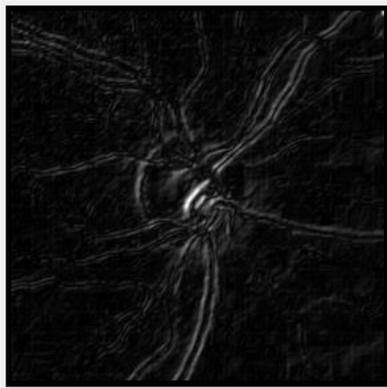
El cálculo del gradiente se realiza píxel a píxel en 360° , con pasos de 15° . En la Figura 14 tenemos un ejemplo en el cual a una ROI le aplicamos diferentes gradientes. La Figura 14 (a) corresponde con la ROI original y desde la (b) hasta la (j) son los resultados de emplear los gradientes. Estos gradientes van de 45° en 45° empezando por 0° y acabando en 360° . Hemos usado varios detectores de bordes de color para determinar las derivadas direccionales, los cuales los mencionamos posteriormente y están desarrollados en la introducción. En el caso de la Figura 14 se está usando la distancia de color CIEDE2000 [35].



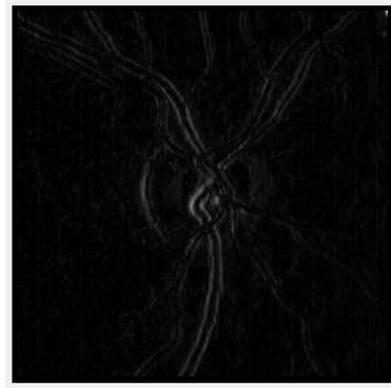
(a)



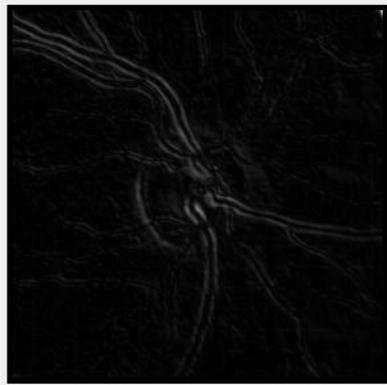
(b)



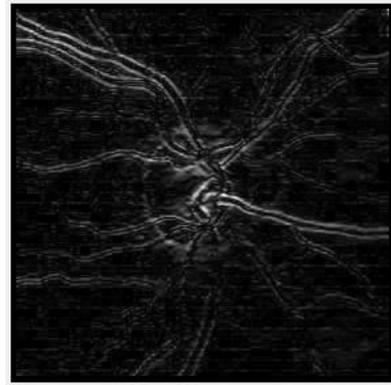
(c)



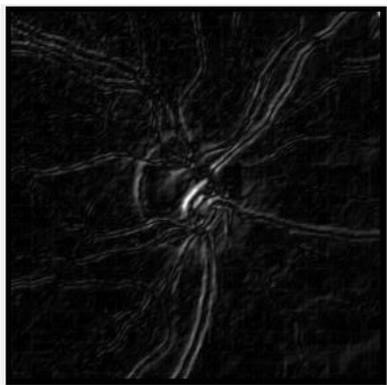
(d)



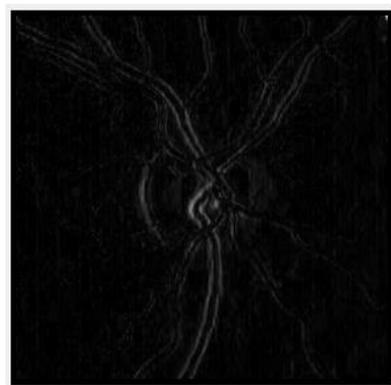
(e)



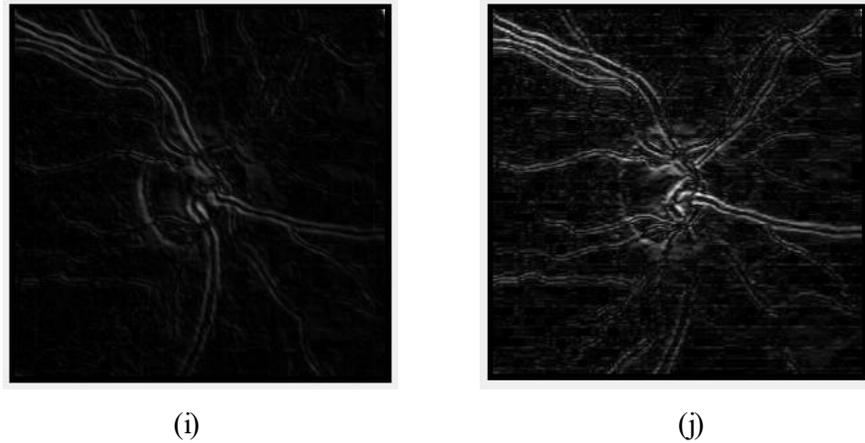
(f)



(g)



(h)



(i) (j)
Figura 14. Ejemplo de una ROI con diferentes gradientes
En (a) ROI original; (b) gradiente 0° ; (c) gradiente 45° ; (d) gradiente 90° ;
(e) gradiente 135° ; (f) gradiente 180° ; (g) gradiente 225° ;
(h) gradiente 270° ; (i) gradiente 315° ; (j) gradiente 360°

En estas imágenes podemos apreciar que los puntos con grandes cambios de color coinciden con las zonas más brillantes, donde se encuentran los vasos sanguíneos. Gracias a esto, el método es más eficiente ya que podemos apreciar la curvatura de los vasos y no es necesario añadir esa información.

Usamos las derivadas de imágenes para poder identificar los límites del disco óptico y la copa, gracias a la capacidad de determinar cambios entre los píxeles.

Los bordes se deben buscar con cambios de color ya que las retinografías son imágenes a color, donde se tratan como dos campos de vectores dimensionales (ubicación de píxeles), de tres canales (planos de color). La explicación de por qué se calcula el gradiente en 360 grados es debido a que los píxeles del borde de la copa tienen un color distinto a los píxeles del disco óptico colindantes. Esta variación del color es lo que se pretende medir con el cálculo de los gradientes como vemos en la Figura 15. En otras palabras, los valores absolutos de color no son relativos, pero sus cambios en la retina sí lo son [35].

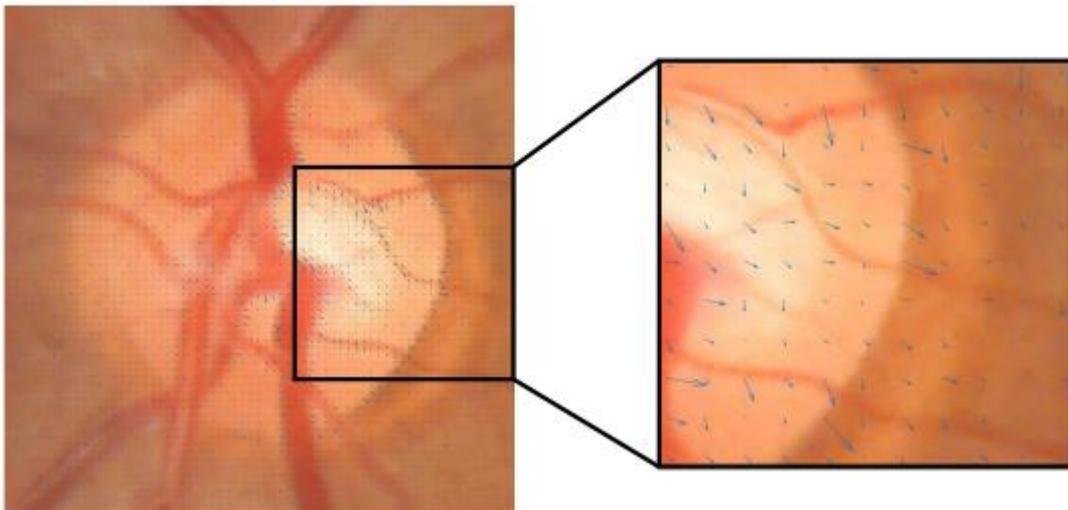


Figura 15. Cambios de color representados por gradiente [35]

En este proyecto hemos usado varios detectores de bordes de color que determinan las derivadas

direccionales, con el fin de hacer una buena comparación y determinar cuál es mejor.

Los gradientes de color que hemos calculado son:

- RGB Euclídea: hallamos la distancia euclídea usando el espacio RGB.

$$d_{rgb} = \sqrt{(CP_1(y) - CP_1(x))^2 + (CP_2(y) - CP_2(x))^2 + (CP_3(y) - CP_3(x))^2}$$

(41)

- Gris Euclídea: el espacio de color gris está compuesto por un único plano, por lo que se halla la distancia euclídea a ese plano. Para poder calcularlo también es necesario pasar la imagen a escala de grises.

$$d_{gris} = \sqrt{(CP(y) - CP(x))^2}$$

(42)

- CIE L*a*b* Euclídea: para obtenerlo calculamos la distancia euclídea a cada uno de los planos del espacio de color CIE L*a*b*. Es necesario recalcar que para poder operar con dicho espacio hay que hacer una conversión previa y pasar la imagen del espacio RGB al CIE L*a*b*.

$$d_{Lab} = \sqrt{(CP_L(y) - CP_L(x))^2 + (CP_a(y) - CP_a(x))^2 + (CP_b(y) - CP_b(x))^2}$$

(43)

Donde $CP_L, CP_a,$ y CP_b son las conversiones del espacio de color RGB al CIE L*a*b*, representando cada uno al plano L, a, y b respectivamente.

- CIE94: es una distancia del espacio de color CIE L*a*b*, la cual la hemos calculado con la ecuación (18), usando los parámetros por defecto.
- CIEDE2000: es una distancia del espacio de color CIE L*a*b*, la cual la hemos calculado con la ecuación (19) (18), usando los parámetros por defecto.

Es conveniente recalcar que el código está programado para usar el espacio de color RGB, por lo tanto, cuando se use uno distinto es necesario hacer una conversión previa.

Todo el trabajo realizado anteriormente con los gradientes sirve para formar la matriz de características con todas las imágenes seleccionadas, es decir, hay una matriz por cada espacio de color con su distancia. En nuestro caso disponemos de 5 matrices las cuales engloban las 60 imágenes de las bases de datos. Ahora vamos a pasar a explicar cómo conseguimos dicha matriz.

Cada fila corresponde a un vector de características de un píxel de la imagen y cada elemento del vector es una característica del píxel. En la matriz también incluimos las componentes de color, según el espacio que estemos usando tendremos componentes R, G y B o L, a y b o una única componente en el caso de gris. Conforme a la teoría que hemos visto el espacio L*a*b* debe ser el mejor ya que es el más parecido a la percepción humana. Otro dato importante que incluir son las distancias y el ángulo del píxel respecto al centro de la copa, usando las siguientes ecuaciones [33]:

$$d = \sqrt{(i - c_x)^2 + (j - c_y)^2}$$

(44)

$$\theta = \tan^{-1} \left(\frac{c_y - j}{i - c_x} \right)$$

(45)

Donde d es la distancia al centro, θ el ángulo y (c_x, c_y) son los valores de la posición (i, j) de la ROI en la que se encuentra el centro de la copa.

Las últimas características que se incluyen en la matriz son la información de la posición (i, j) del píxel en la imagen y la clase a la que pertenece. Si examinamos la Figura 16 veremos a modo de esquema como se construye la matriz de características.

A modo de resumen disponemos de un vector con 32 características o 30 para el caso de gris, más la verdad de referencia en ambos casos, que son:

- 25 valores de gradiente de color.
- 3 valores de color, en caso de gris un único valor de color
- 1 valor de distancia al centro.
- 1 valor del ángulo con respecto al centro.
- 2 valores de la posición (i, j) del píxel.

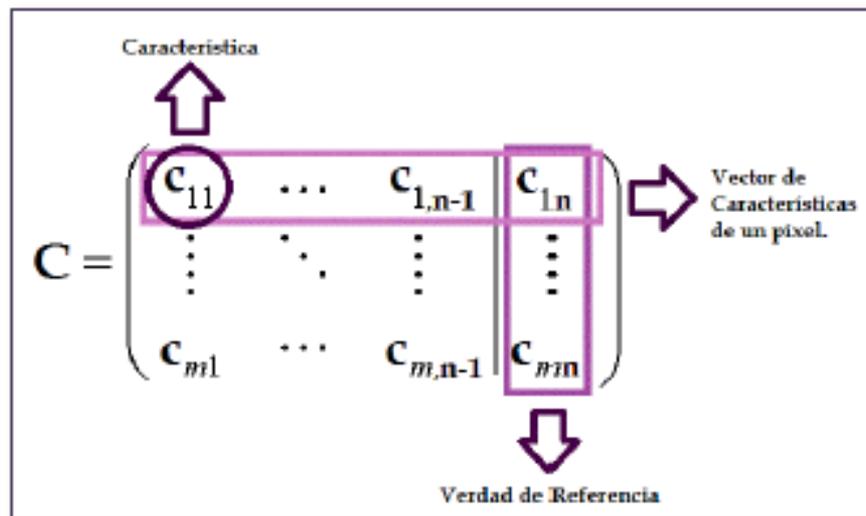


Figura 16. Matriz de características [33]

3.4 Clasificación

Cuando ya tenemos toda la información de todos los píxeles de cada imagen, y de todas las imágenes que se usarán para el entrenamiento, ya podría formarse la matriz C para comenzar el entrenamiento.

Vamos a realizar una clasificación a nivel de píxel definiendo 3 clases que son:

- Clase 1: El píxel pertenece al fondo.
- Clase 2: El píxel pertenece al disco óptico.
- Clase 3: El píxel pertenece a la copa.

Para que nuestro clasificador sea capaz de realizar una correcta clasificación entre las clases anteriores es necesario realizar un entrenamiento con imágenes conocidas, así conseguimos que aprenda a diferenciar las clases. Es conveniente recalcar que las imágenes que se quieren clasificar no se pueden utilizar en el entrenamiento.

Al principio la matriz resultante daba una clasificación nefasta, esto era debido a que no lograba evaluar bien las clases que tenemos. Este fallo se solucionó realizando un balance entre los píxeles que se toman, cogiendo igual número de píxeles de cada clase en cada imagen. El problema venía porque hay muchos más píxeles pertenecientes al fondo que a las otras clases, lo que hacía al fondo más representativo. Gracias a esto se consigue una mejor clasificación.

Para el entrenamiento hemos usado todos los clasificadores que nos proporciona la aplicación de Matlab que son:

- Simplex Tree
- Medium Tree
- Complex Tree
- K-Nearest Neighbor (KNN) Fine
- K-Nearest Neighbor (KNN) Medium
- K-Nearest Neighbor (KNN) Coarse
- K-Nearest Neighbor (KNN) Cosine
- K-Nearest Neighbor (KNN) Weighted
- Support Vector Machine (SVM) Linear
- Support Vector Machine (SVM) Quadratic

4 RESULTADOS

En este apartado vamos a mostrar los resultados obtenidos con el 'Classification Learner', una vez realizado el entrenamiento con los diferentes gradientes de color y sus respectivas distancias. Para hacer la comparación usaremos las tablas de confusión que es una herramienta que nos facilita el clasificador, la cual está basada en diferentes porcentajes que explicaremos a continuación.

Vamos a recordar las diferentes clases que teníamos y a que hacen referencia:

- Clase 1: El píxel pertenece al fondo.
- Clase 2: El píxel pertenece al disco óptico.
- Clase 3: El píxel pertenece a la copa.

True Class	Clase 1 (0)	Bien Clasificado (%)	Es clase 1 y se clasifica como 2 (%)	Es clase 1 y se clasifica como 3 (%)	TPR FNR
	Clase 2 (127)	Es clase 2 y se clasifica como 1 (%)	Bien Clasificado (%)	Es clase 2 y se clasifica como 3 (%)	TPR FNR
	Clase 3 (255)	Es clase 3 y se clasifica como 1 (%)	Es clase 3 y se clasifica como 2 (%)	Bien Clasificado (%)	TPR FNR
		Clase 1 (0)	Clase 2 (127)	Clase 3 (255)	
Predict Class					

Tabla 3. Matriz de Confusión [33]

Para comprender la Tabla 3 vamos a explicar ciertos términos que son necesarios, además de que se usarán a la hora de sacar las conclusiones, los cuales son:

- TP: Verdadero positivo, indica que es de clase X y el clasificador lo identifica como tal.
- FP: Falso positivo, indica que no es de clase X y el clasificador lo identifica como tal.
- TN: Verdadero negativo, indica que es de clase X y el clasificador no lo identifica como tal.
- FN: Falso negativo, indica que no es de clase X y el clasificador no lo identifica como tal.
- TPR: Tasa de verdaderos positivos, relación entre número de píxeles segmentados y la cantidad total de píxeles segmentados por el experto. También se le conoce como sensibilidad [35].

$$TPR = \frac{TP}{(TP + FN)}$$

(46)

- TNR: Tasa de verdaderos negativos, relación entre negativos correctamente identificados y los negativos de la imagen segmentada por el experto [35].

$$TNR = \frac{TN}{(TN + FP)}$$

(47)

- Accuaracy: Precisión, representa el número de píxeles correctamente identificados en % [33].

$$ACC = \frac{(TP + TN)}{(TP + FP + TN + FN)}$$

(48)

Una vez que hemos entendido estos conceptos vamos a pasar a mostrar los resultados que hemos obtenido de los diferentes gradientes y clasificadores. Lo vamos a organizar por los diferentes gradientes y dentro de ellos pondremos todos los clasificadores que se han usado de dicho gradiente. Elegiremos el clasificador que sea más eficaz y se adapte al algoritmo, para ello haremos las comparaciones con la precisión (accuaracy) de cada matriz que hemos obtenido.

4.1 Resultados Gris Euclídea

- Matriz de confusión Simple Tree: Ofrece una precisión del 68%. No hace una buena clasificación de clases y por ello da un resultado muy bajo, por lo que lo descartamos como clasificador. Su coste computacional es mínimo, tardando pocos segundos en ejecutarse.

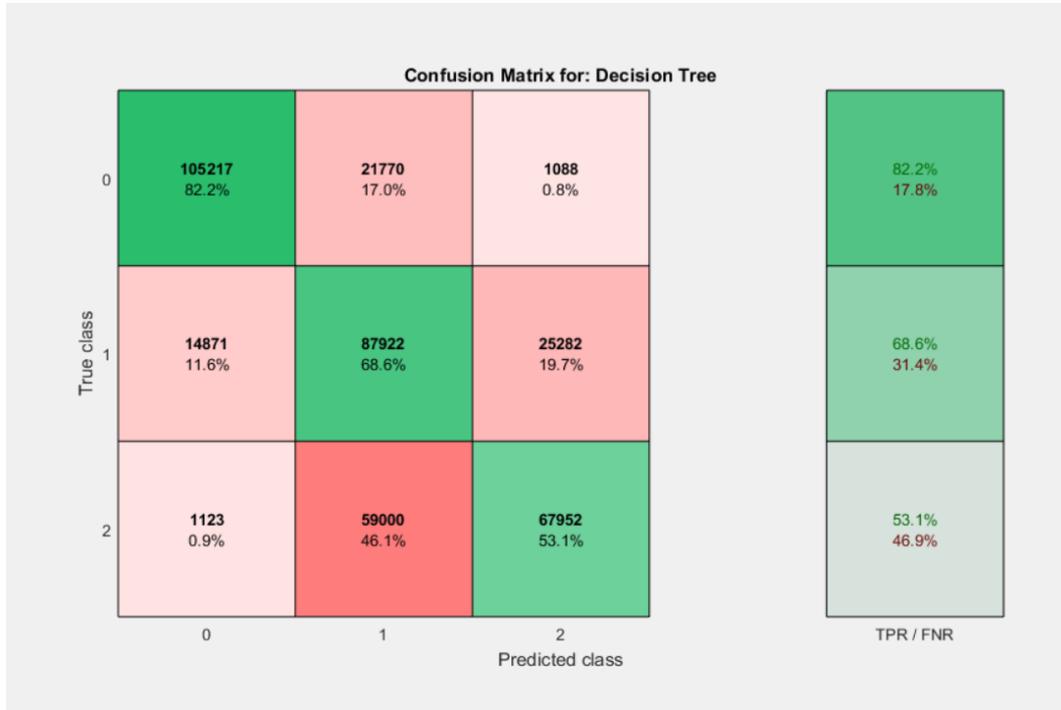


Tabla 4. Simple Tree Gris

- Matriz de confusión Medium Tree: Obtenemos una precisión del 76.9%. Hace una clasificación decente de clases y su coste computacional también es mínimo, aun así, descartamos su uso.

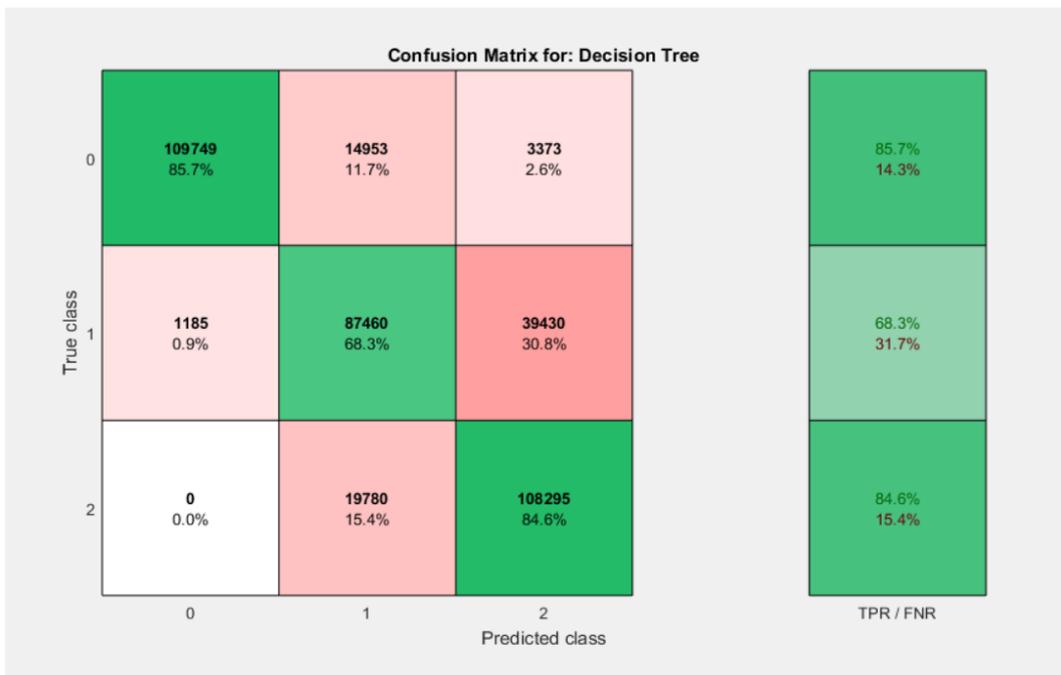


Tabla 5. Medium Tree Gris

- Matriz de confusión Complex Tree: Obtenemos una precisión del 79.5%. Hace una buena clasificación de las clases siendo su coste computacional ínfimo. Sería un posible candidato como clasificador.

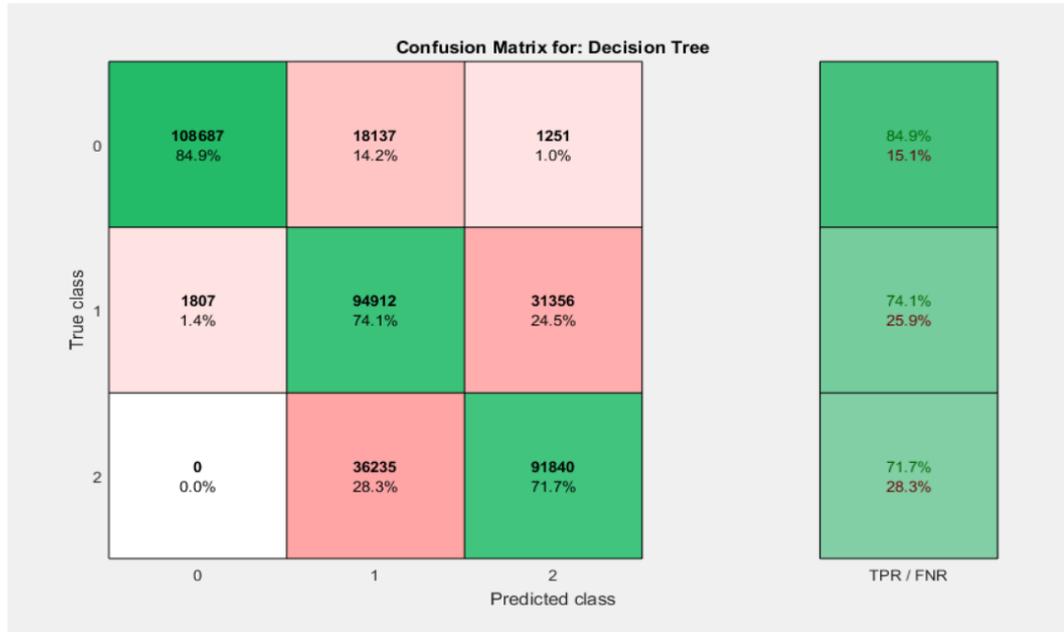


Tabla 6. Complex Tree Gris

Con estos resultados podemos concluir que el clasificador 'Decision Tree' es aceptable, aunque no es el más apropiado, pero tiene un coste computacional muy bajo haciendo que se ejecute en pocos segundos. El mejor es Complex Tree.

- Matriz de confusión Fine KNN: Conseguimos una precisión de 84.5% siendo buena la clasificación de clases. Tiene un coste computacional medio, tardando alrededor de 1 hora en ejecutarse. Sería un posible candidato como clasificador.



Tabla 7. Fine KNN Gris

- Matriz de confusión Medium KNN: Ofrece una precisión del 85%. Hace bastante bien la clasificación de clases. Tiene un coste computacional muy parecido a Fine KNN y es también posible candidato como clasificador.

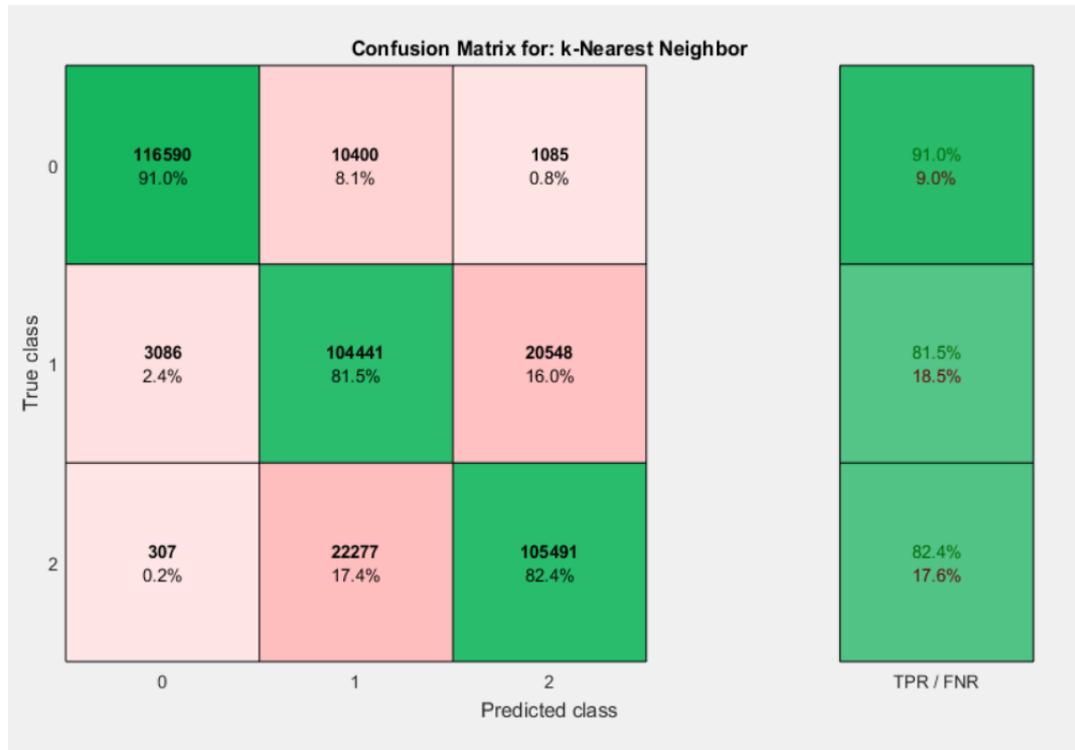


Tabla 8. Medium KNN Gris

- Matriz de confusión Coarse KNN: Conseguimos una precisión del 81.8%, siendo peor la clasificación de clases que sus anteriores. Su coste computacional es mayor, pudiendo tardar varias horas. Lo descartamos como clasificador.

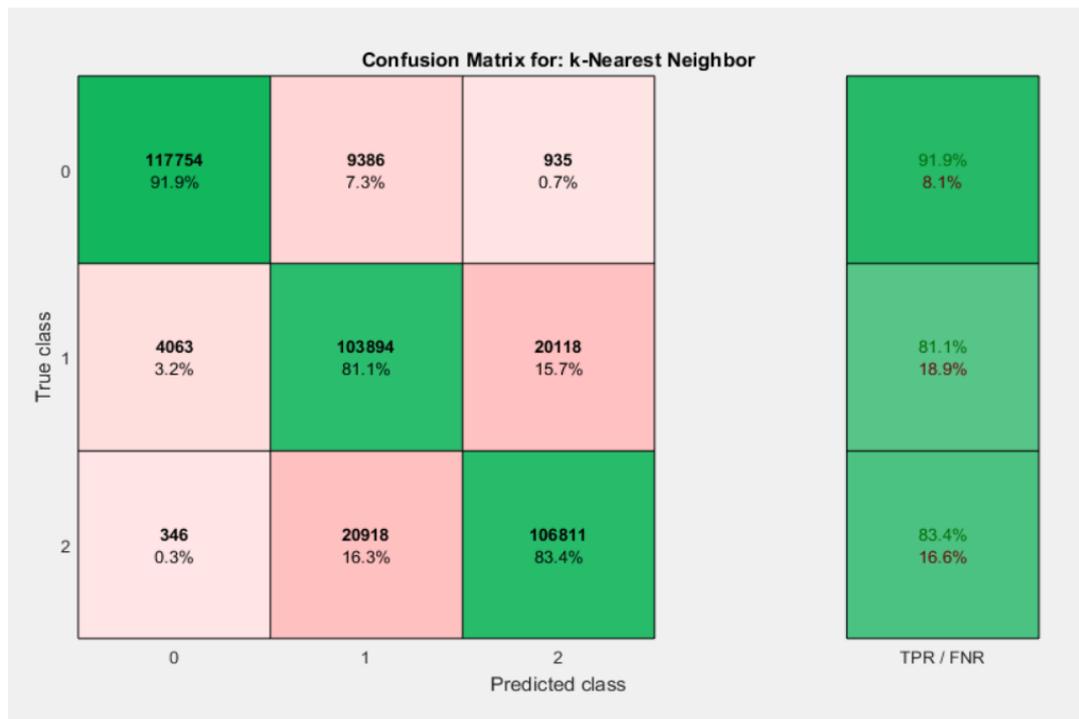


Tabla 9. Coarse KNN Gris

- Matriz de confusión Cosine KNN: Obtenemos una precisión del 85.5%, clasificando muy bien entre clases. Su coste computacional es muy parecido a Coarse KNN. Es un posible candidato a clasificador.

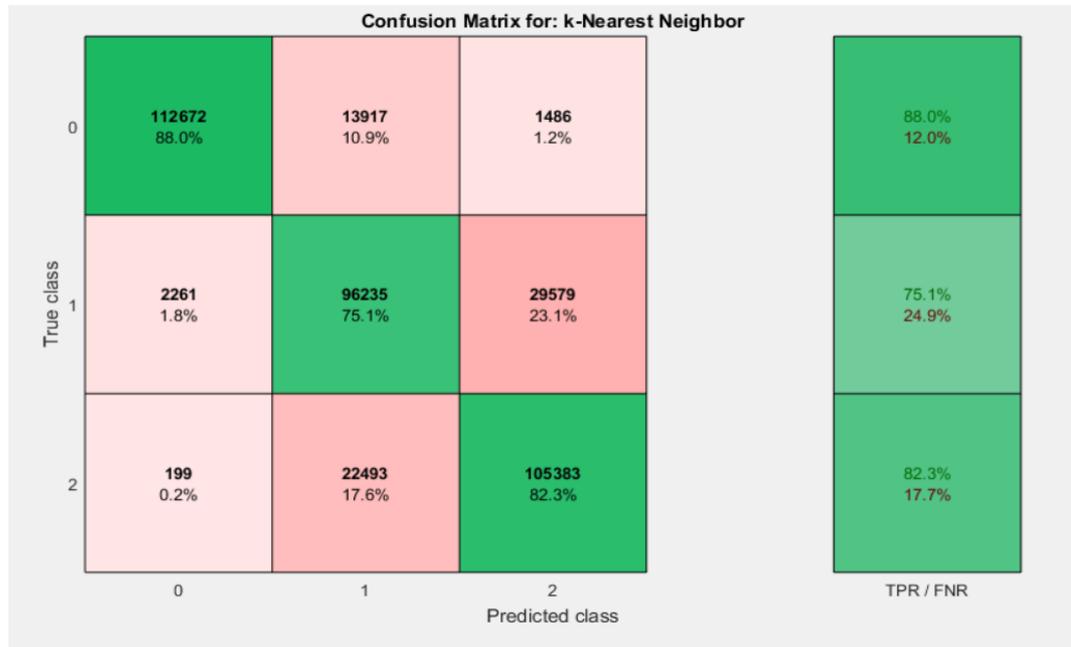


Tabla 10. Cosine KNN Gris

- Matriz de confusión Weighted KNN: Da una precisión del 86%, siendo la mayor de todas por lo que es el que mejor diferencia entre clases. Su coste computacional es medio, muy parecido al de Medium KNN, aunque tarda un poco más. Es el mejor candidato a clasificador.

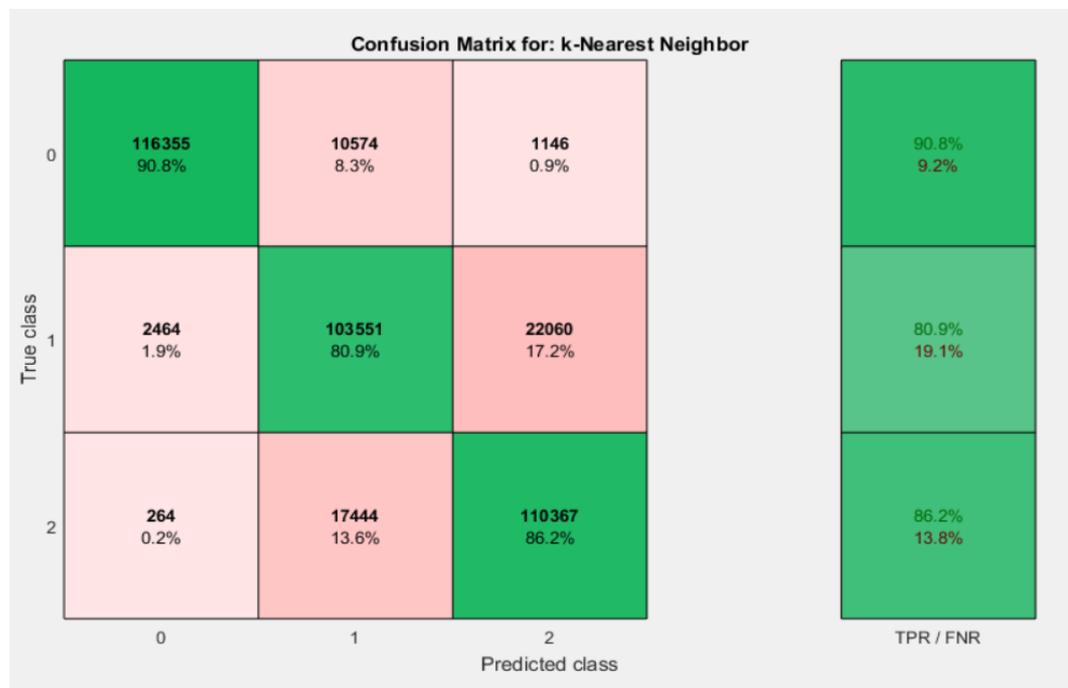


Tabla 11. Weighted KNN Gris

A la vista de estos resultados podemos concluir que KNN es buen clasificador siendo el más apropiado de usar. Hace una buena distinción de clases, además su tiempo de ejecución y computacional es medio. El mejor es Weighted KNN.

- Matriz de confusión Linear SVM: Ofrece una precisión del 80.6% siendo aceptable su distinción entre clases. Su coste computacional es muy alto, llegando a tardar un día en ejecutarse. Lo descartamos como clasificador.

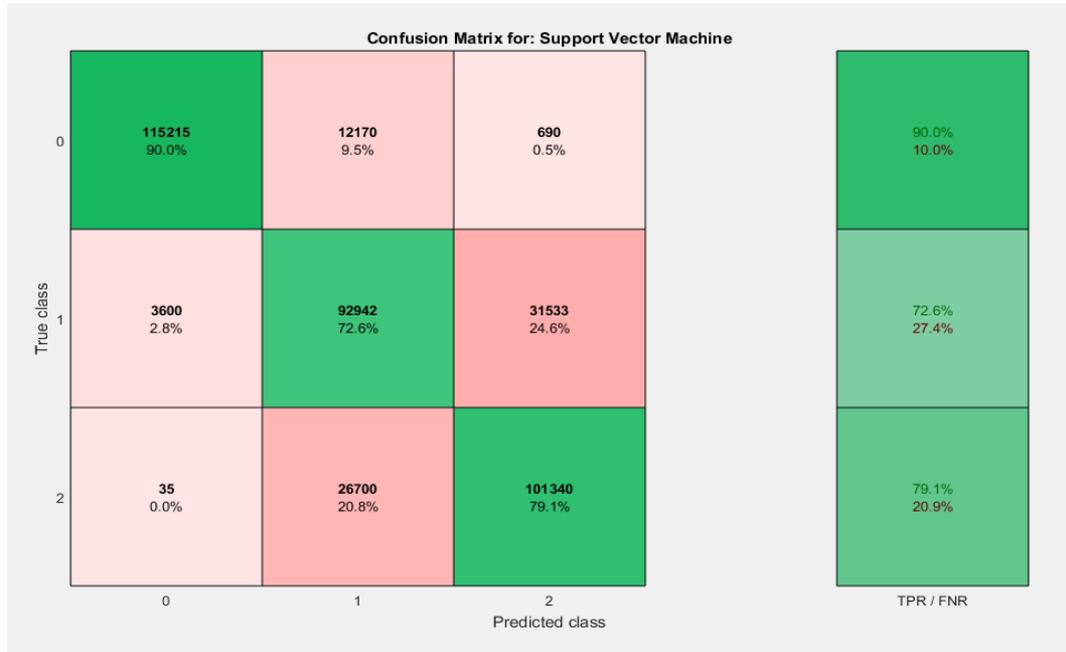


Tabla 12. Linear SVM Gris

- Matriz de confusión Quadratic SVM: Obtenemos una precisión del 83.8%. La clasificación de clases es mejor que en el caso de Linear SVM, pero su coste computacional es mayor, tarda casi dos días en ejecutarse completo. Lo descartamos como clasificador.

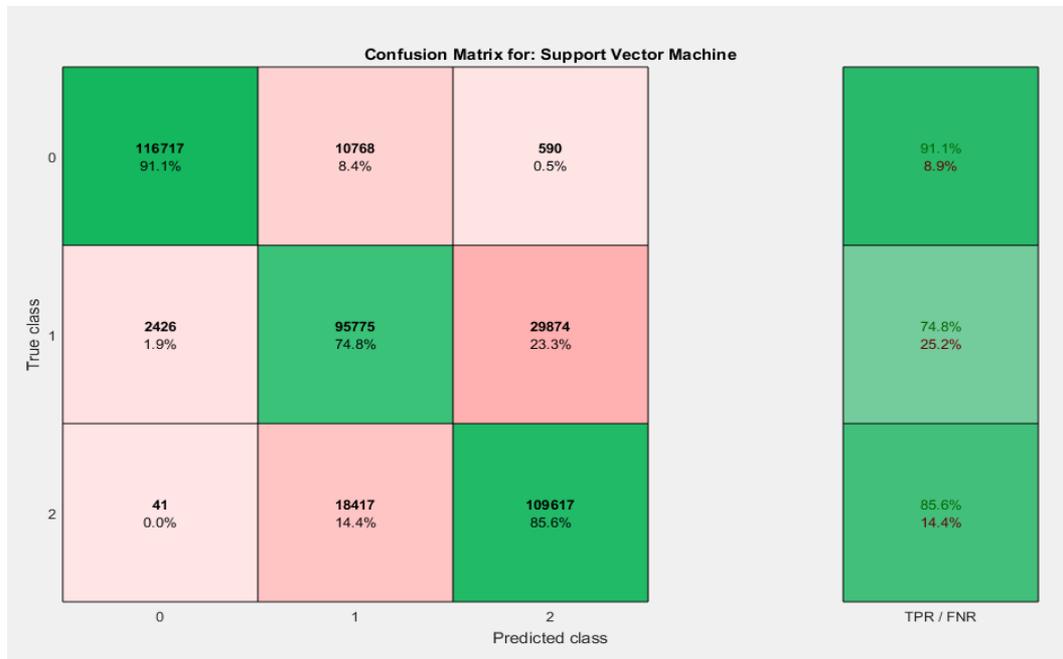


Tabla 13. Quadratic SVM Gris

Una vez visto los resultados anteriores podemos decir que SVM no es un buen clasificador para este caso, la principal causa es su elevado coste computacional. El mejor clasificador es Quadratic SVM que, aunque da una precisión buena no compensa con el tiempo que tarda en realizar la ejecución.

4.2 Resultados RGB Euclídea

- Matriz de confusión Simple Tree: Ofrece una precisión del 67.9%. No hace una buena clasificación de clases y por ello da un resultado muy bajo, por lo que lo descartamos como clasificador. Su coste computacional es mínimo, tardando pocos segundos en ejecutarse.

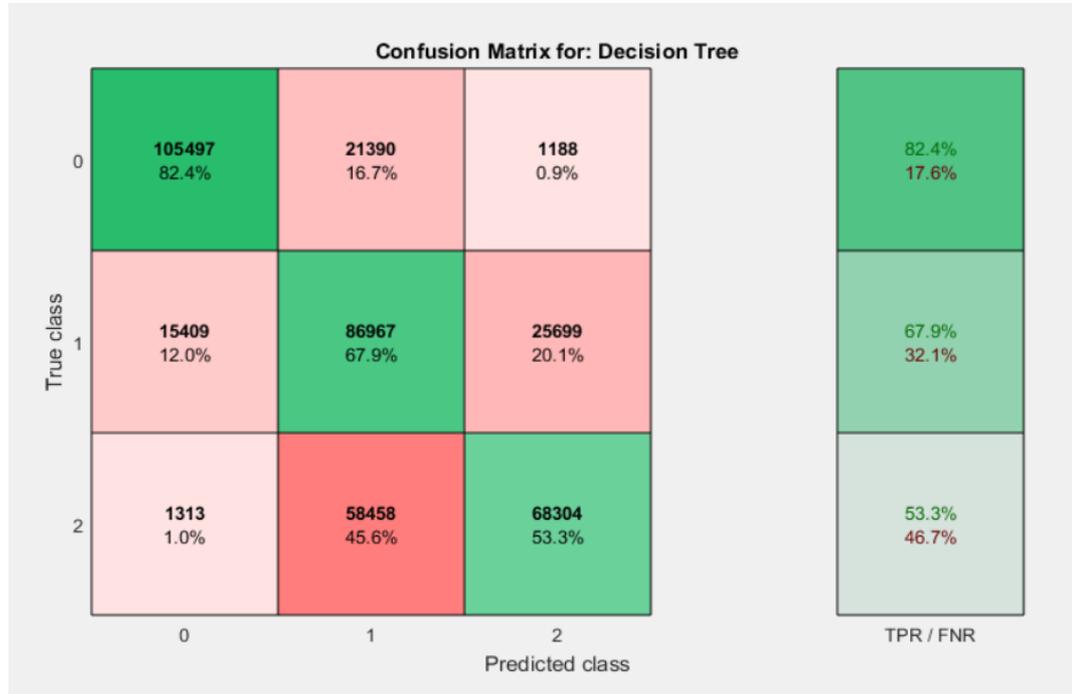


Tabla 14. Simple Tree RGB

- Matriz de confusión Medium Tree: Obtenemos una precisión del 76.7%. Hace una clasificación decente de clases y su coste computacional también es mínimo, aun así descartamos su uso.

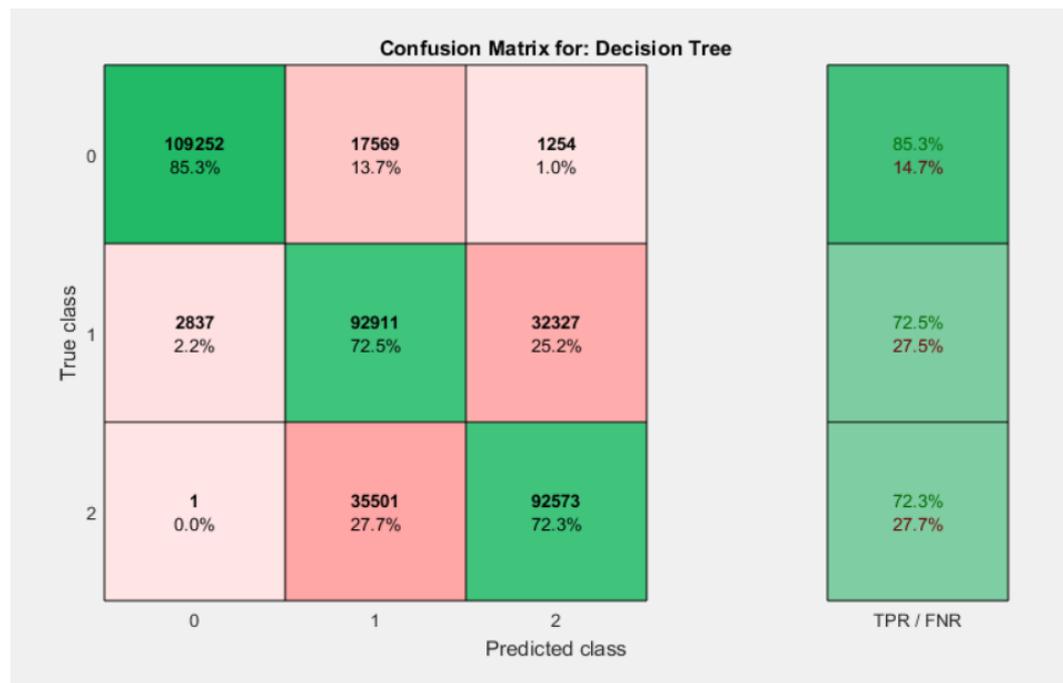


Tabla 15. Medium Tree RGB

- Matriz de confusión Complex Tree: Obtenemos una precisión del 79.7%. Hace una buena clasificación de las clases siendo su coste computacional ínfimo. Sería un posible candidato como clasificador.

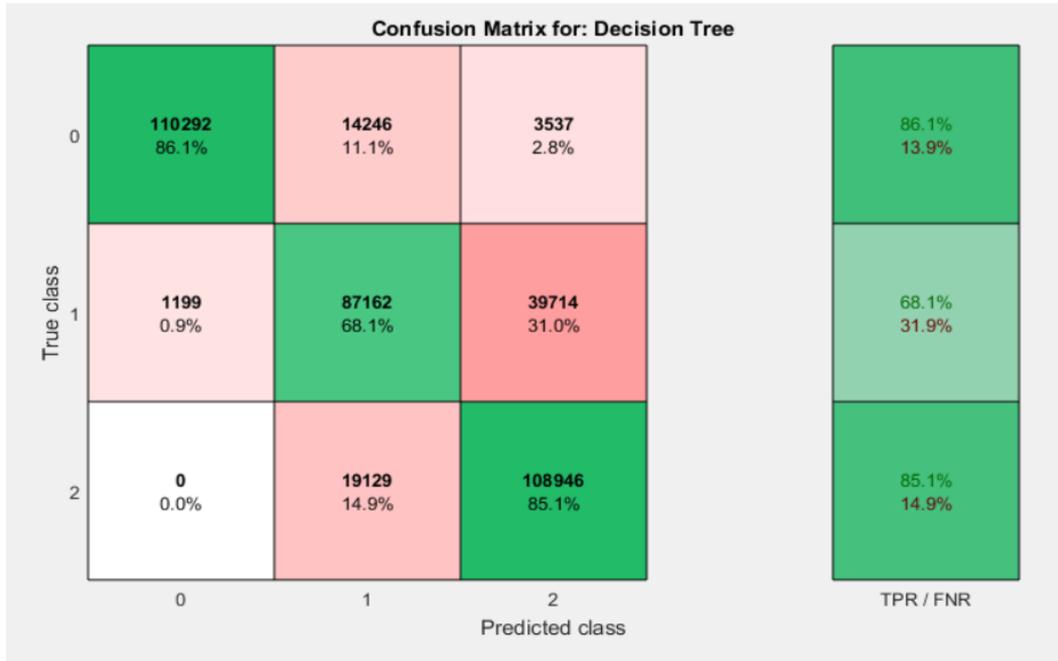


Tabla 16. Complex Tree RGB

Con estos resultados podemos concluir que el clasificador ‘Decision Tree’ es aceptable, aunque no es el más apropiado, pero tiene un coste computacional muy bajo haciendo que se ejecute en pocos segundos. El mejor es Complex Tree.

- Matriz de confusión Fine KNN: Conseguimos una precisión de 86.8% siendo buena la clasificación de clases. Tiene un coste computacional medio, tardando alrededor de 1 hora en ejecutarse. Sería un posible candidato como clasificador.

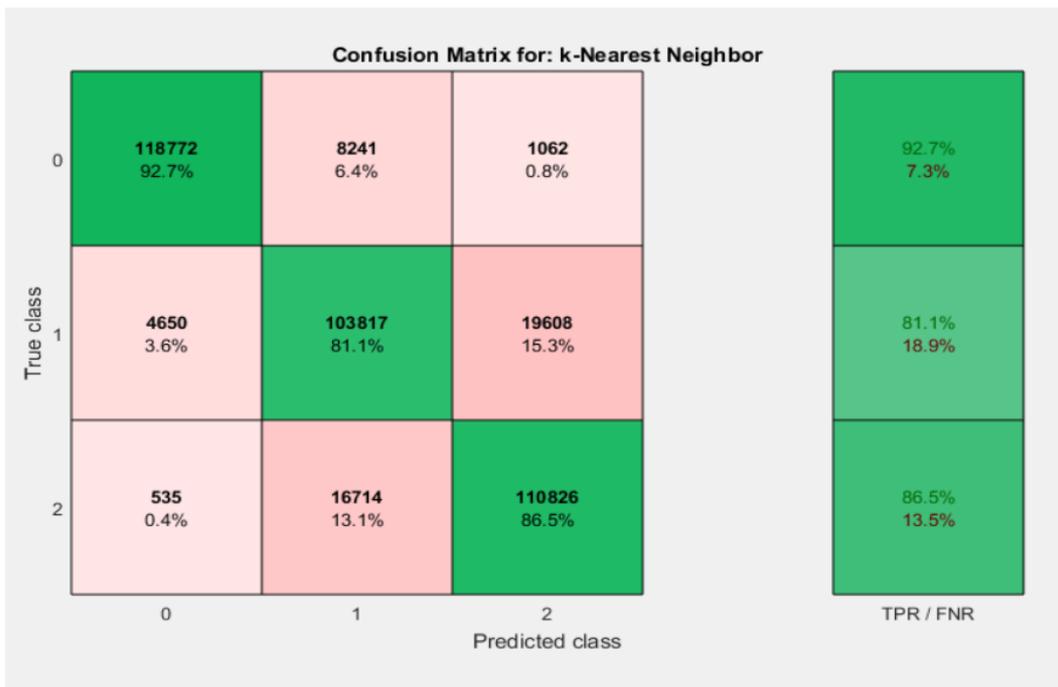


Tabla 17. Fine KNN RGB

- Matriz de confusión Medium KNN: Ofrece una precisión del 86.8%. Hace bastante bien la clasificación de clases. Tiene un coste computacional muy parecido a Fine KNN y es también posible candidato como clasificador.

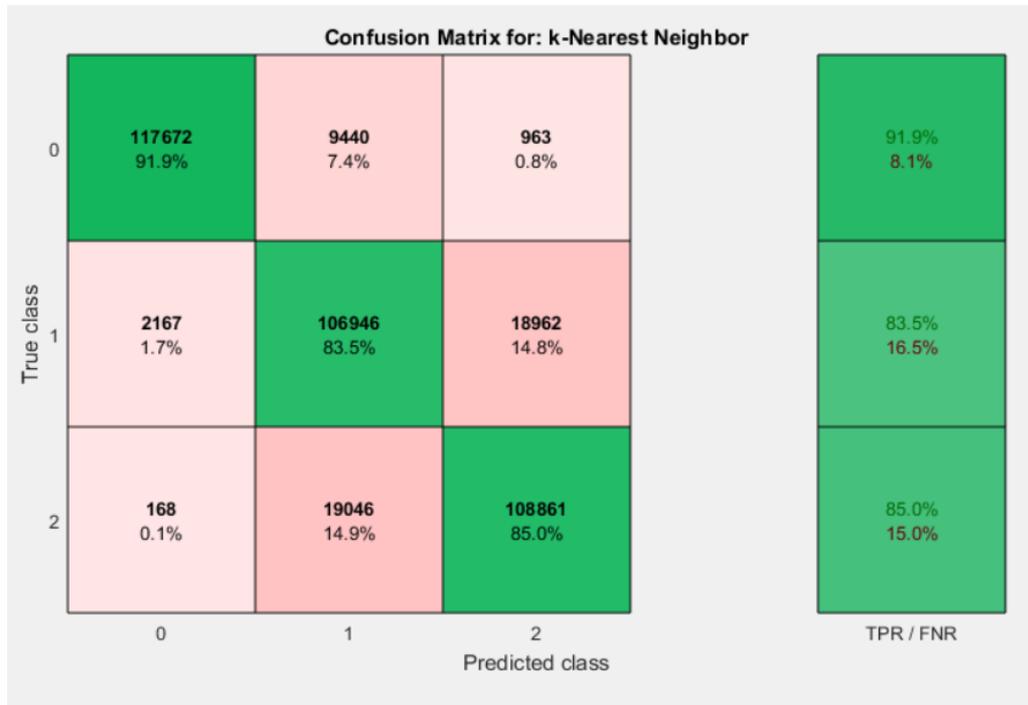


Tabla 18. Medium KNN RGB

- Matriz de confusión Coarse KNN: Conseguimos una precisión del 83.1%, siendo peor la clasificación de clases que sus anteriores. Su coste computacional es mayor, pudiendo tardar varias horas. Lo descartamos como clasificador.



Tabla 19. Coarse KNN RGB

- Matriz de confusión Cosine KNN: Obtenemos una precisión del 86.9%, clasificando muy bien entre clases. Su coste computacional es muy parecido a Coarse KNN. Es un posible candidato a clasificador.

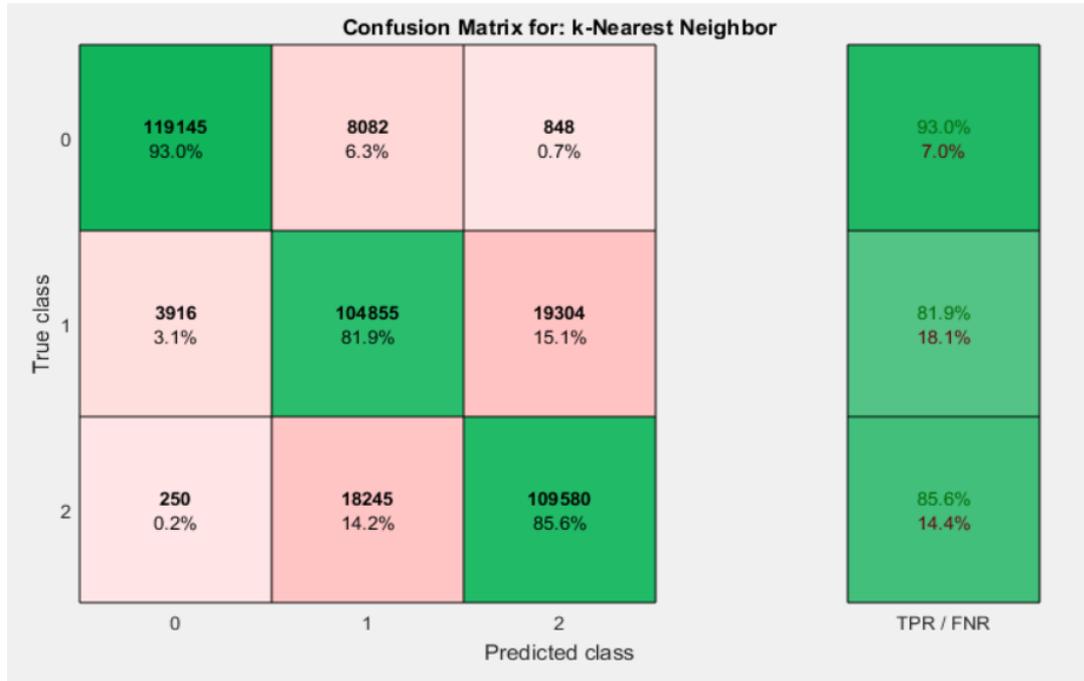


Tabla 20. Cosine KNN RGB

- Matriz de confusión Weighted KNN: Da una precisión del 87.8%, siendo la mayor de todas por lo que es el que mejor diferencia entre clases. Su coste computacional es medio, muy parecido al de Medium KNN, aunque tarda un poco más. Es el mejor candidato a clasificador.

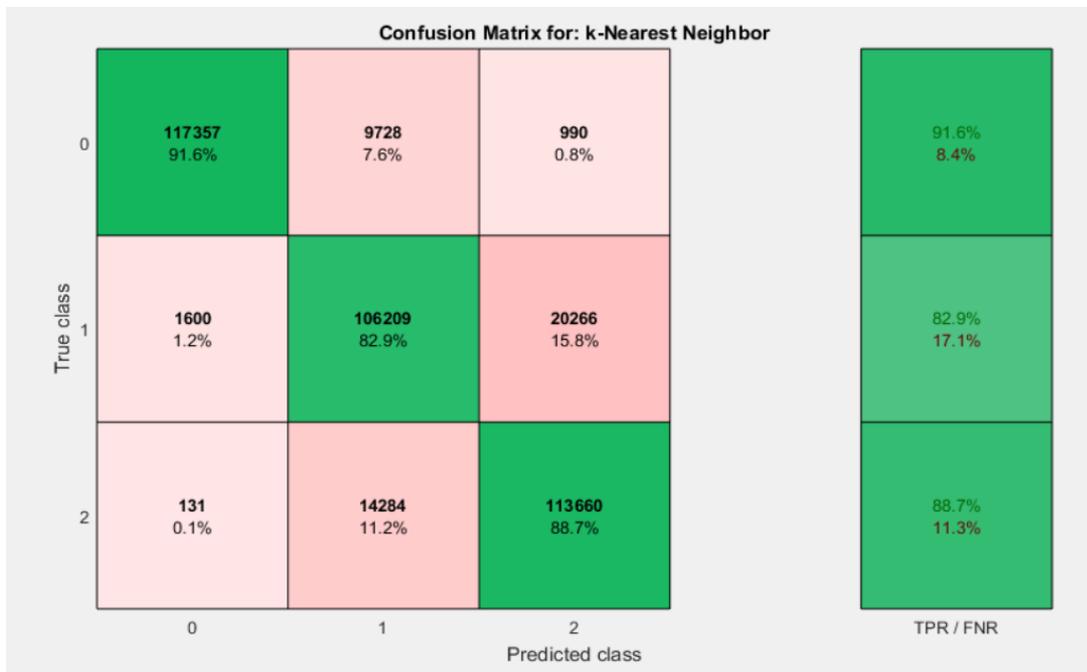


Tabla 21. Weighted KNN RGB

A la vista de estos resultados podemos concluir que KNN es buen clasificador siendo el más apropiado de usar. Hace una buena distinción de clases, además su tiempo de ejecución y computacional es medio. El mejor es Weighted KNN.

Evaluación de espacios y distancias de color en el cálculo de los gradientes de color para detección automática de glaucoma

- Matriz de confusión Linear SVM: Ofrece una precisión del 81% siendo aceptable su distinción entre clases. Su coste computacional es muy alto, llegando a tardar un día en ejecutarse. Lo descartamos como clasificador.

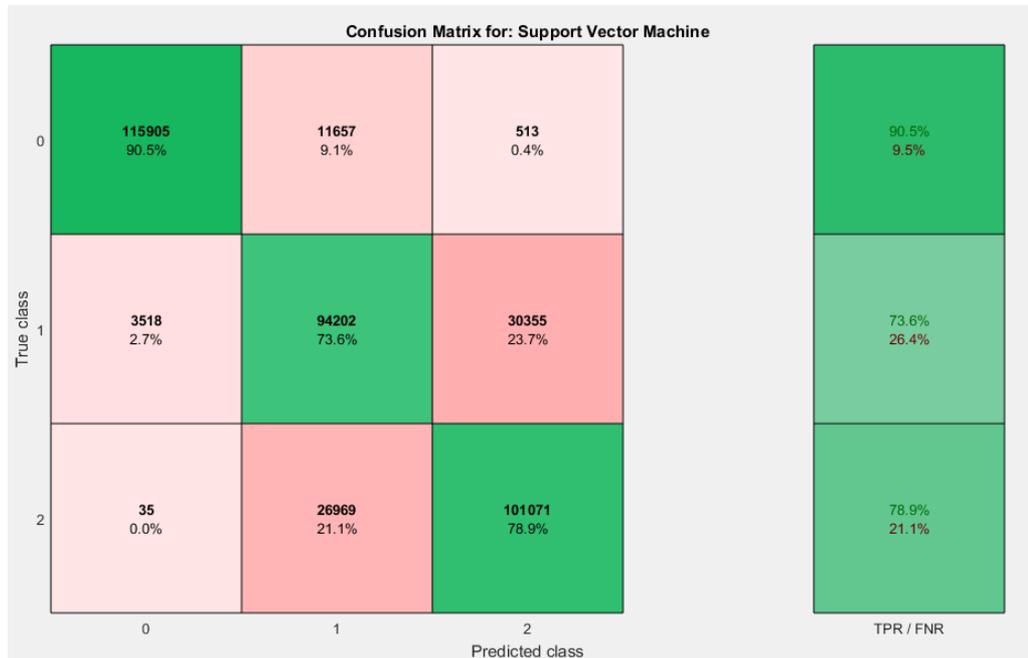


Tabla 22. Linear SVM RGB

- Matriz de confusión Quadratic SVM: Obtenemos una precisión del 85%. La clasificación de clases es mejor que en el caso de Linear SVM, pero su coste computacional es mayor, tarda casi dos días en ejecutarse completo. Lo descartamos como clasificador.

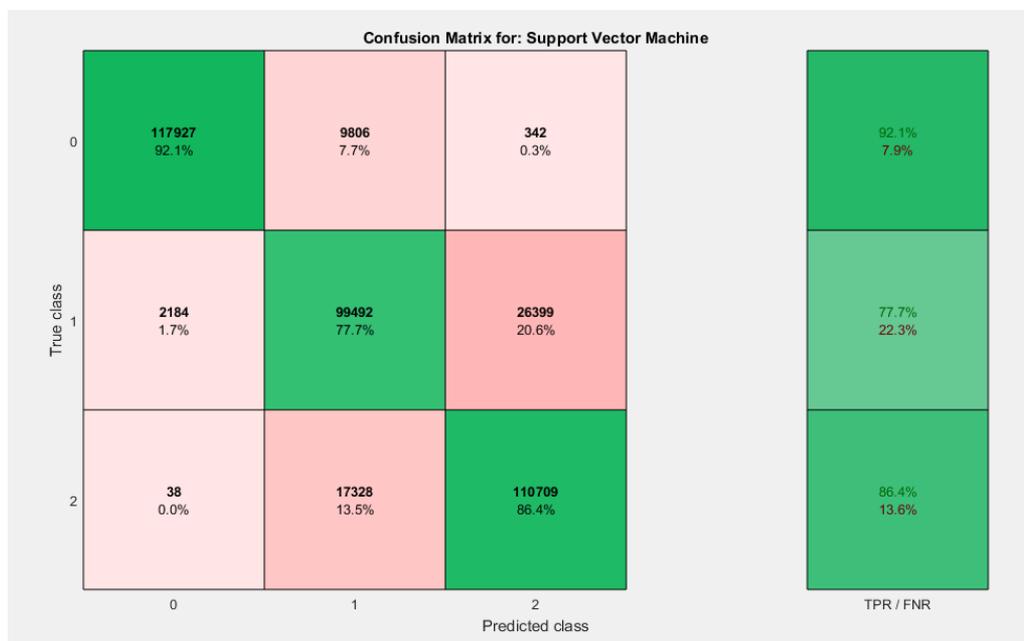


Tabla 23. Quadratic SVM RGB

Una vez visto los resultados anteriores podemos decir que SVM no es un buen clasificador para este caso, la principal causa es su elevado coste computacional. El mejor clasificador es Quadratic SVM que, aunque da una precisión buena no compensa con el tiempo que tarda en realizar la ejecución.

4.3 Resultados CIE L*a*b* Euclídea

- Matriz de confusión Simple Tree: Ofrece una precisión del 63.5%. No hace una buena clasificación de clases y por ello da un resultado muy bajo, por lo que lo descartamos como clasificador. Su coste computacional es mínimo, tardando pocos segundos en ejecutarse.

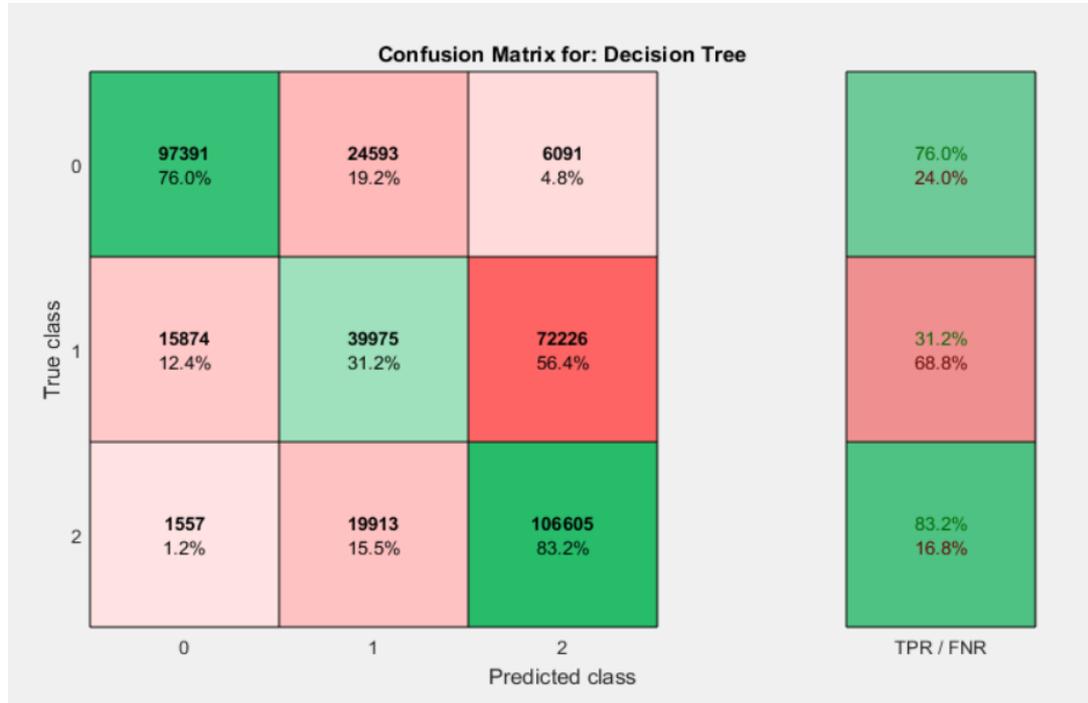


Tabla 24. Simple Tree CIE L*a*b*

- Matriz de confusión Medium Tree: Obtenemos una precisión del 76.9%. Hace una clasificación decente de clases y su coste computacional también es mínimo, aun así, descartamos su uso.

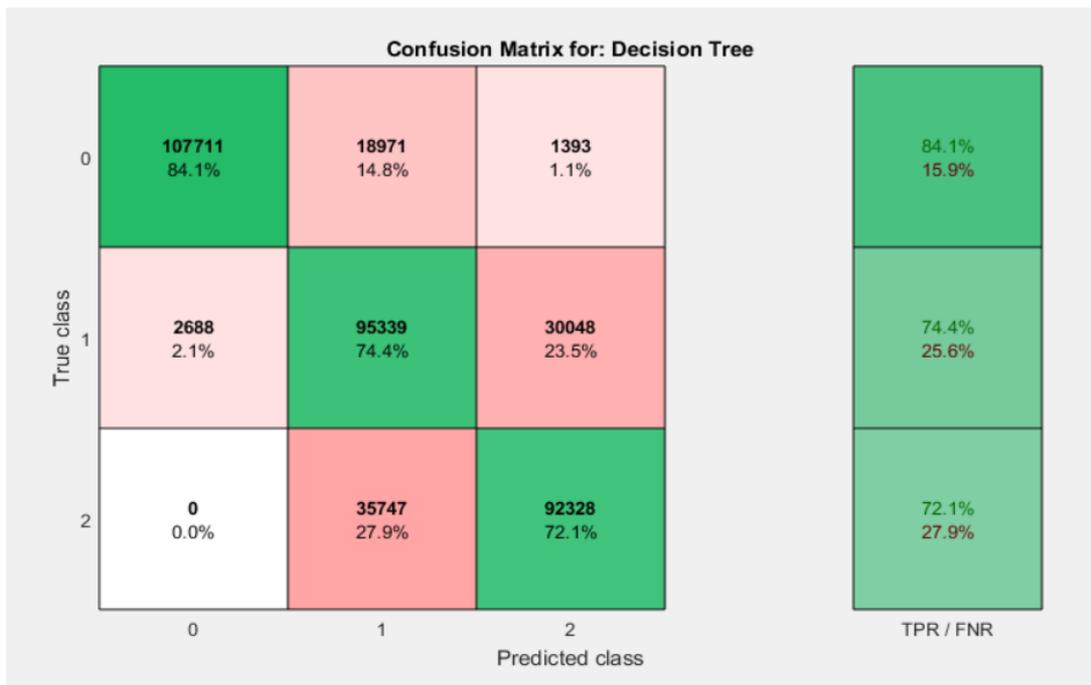


Tabla 25. Medium Tree CIE L*a*b*

- Matriz de confusión Complex Tree: Obtenemos una precisión del 80.9%. Hace una buena clasificación de las clases siendo su coste computacional ínfimo. Sería un posible candidato como clasificador.

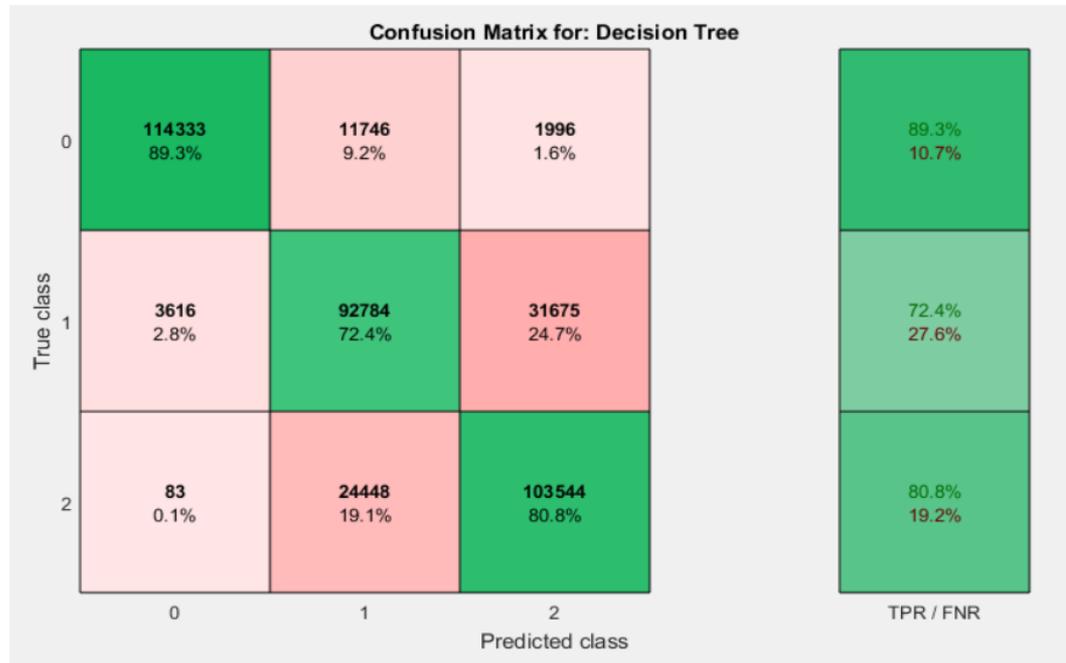


Tabla 26. Complex Tree CIE L*a*b*

Con estos resultados podemos concluir que el clasificador ‘Decision Tree’ es aceptable, aunque no es el más apropiado, pero tiene un coste computacional muy bajo haciendo que se ejecute en pocos segundos. El mejor es Complex Tree.

- Matriz de confusión Fine KNN: Conseguimos una precisión de 89.6% siendo buena la clasificación de clases. Tiene un coste computacional medio, tardando alrededor de 1 hora en ejecutarse. Sería un posible candidato como clasificador.

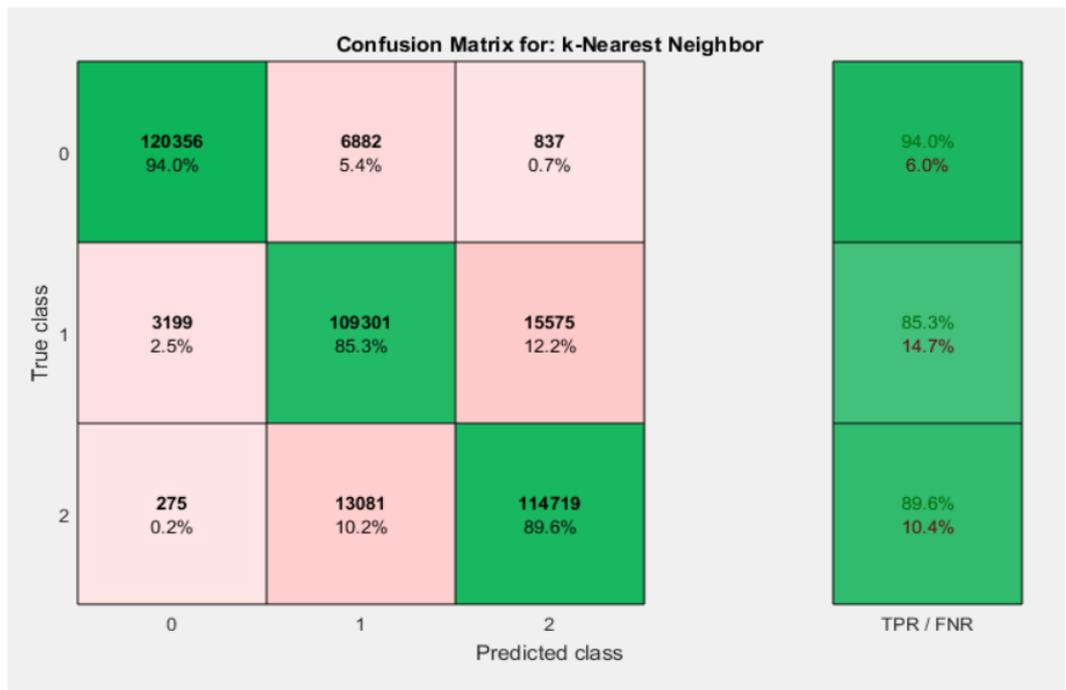


Tabla 27. Fine KNN CIE L*a*b*

- Matriz de confusión Medium KNN: Ofrece una precisión del 89%. Hace bastante bien la clasificación de clases. Tiene un coste computacional muy parecido a Fine KNN y es también posible candidato como clasificador.

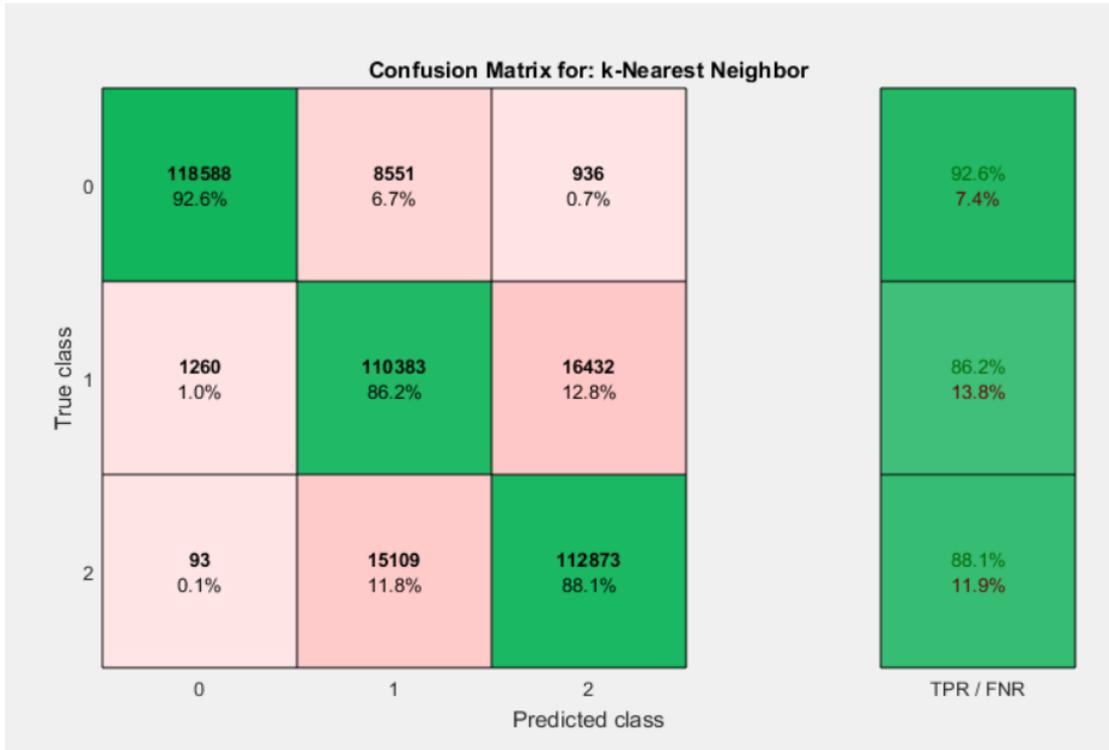


Tabla 28. Medium KNN CIE L*a*b*

- Matriz de confusión Coarse KNN: Conseguimos una precisión del 85%, siendo peor la clasificación de clases que sus anteriores. Su coste computacional es mayor, pudiendo tardar varias horas. Lo descartamos como clasificador.

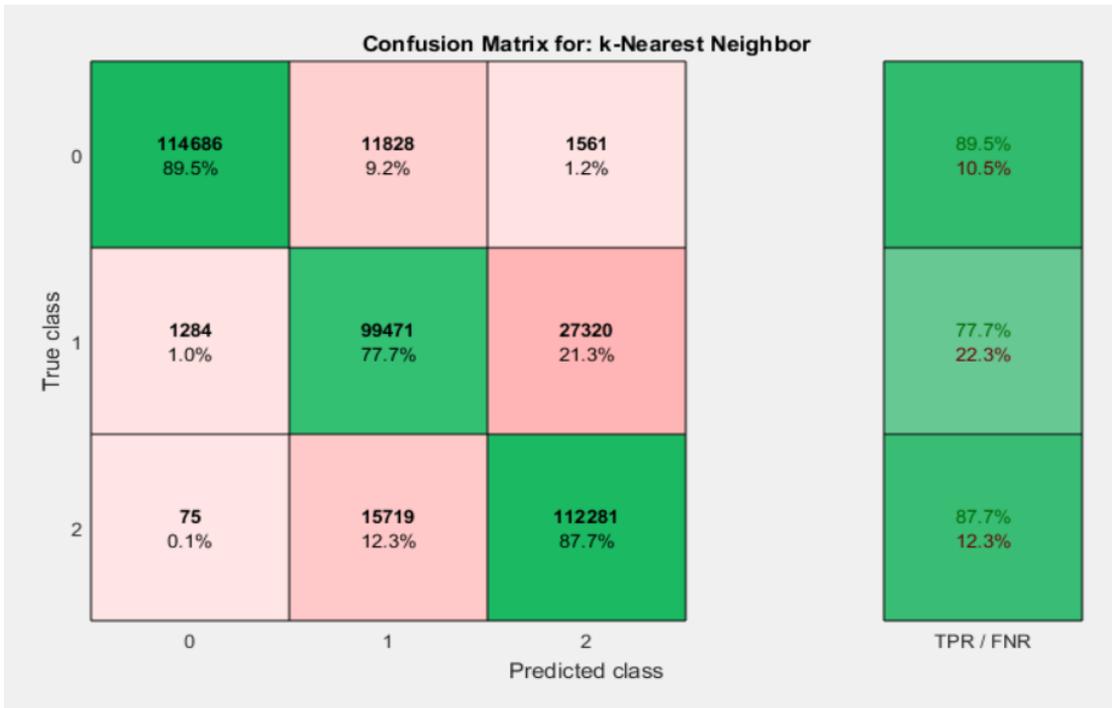


Tabla 29. Coarse KNN CIE L*a*b*

- Matriz de confusión Cosine KNN: Obtenemos una precisión del 89%, clasificando muy bien entre clases. Su coste computacional es muy parecido a Coarse KNN. Es un posible candidato a clasificador.



Tabla 30. Cosine KNN CIE L*a*b*

- Matriz de confusión Weighted KNN: Da una precisión del 90.1%, siendo la mayor de todas por lo que es el que mejor diferencia entre clases. Su coste computacional es medio, muy parecido al de Medium KNN, aunque tarda un poco más. Es el mejor candidato a clasificador.



Tabla 31. Weighted KNN CIE L*a*b*

A la vista de estos resultados podemos concluir que KNN es buen clasificador siendo el más apropiado de usar. Hace una buena distinción de clases, además su tiempo de ejecución y computacional es medio. El mejor es Weighted KNN.

- Matriz de confusión Linear SVM: Ofrece una precisión del 80% siendo aceptable su distinción entre clases. Su coste computacional es muy alto, llegando a tardar un día en ejecutarse. Lo descartamos como clasificador.

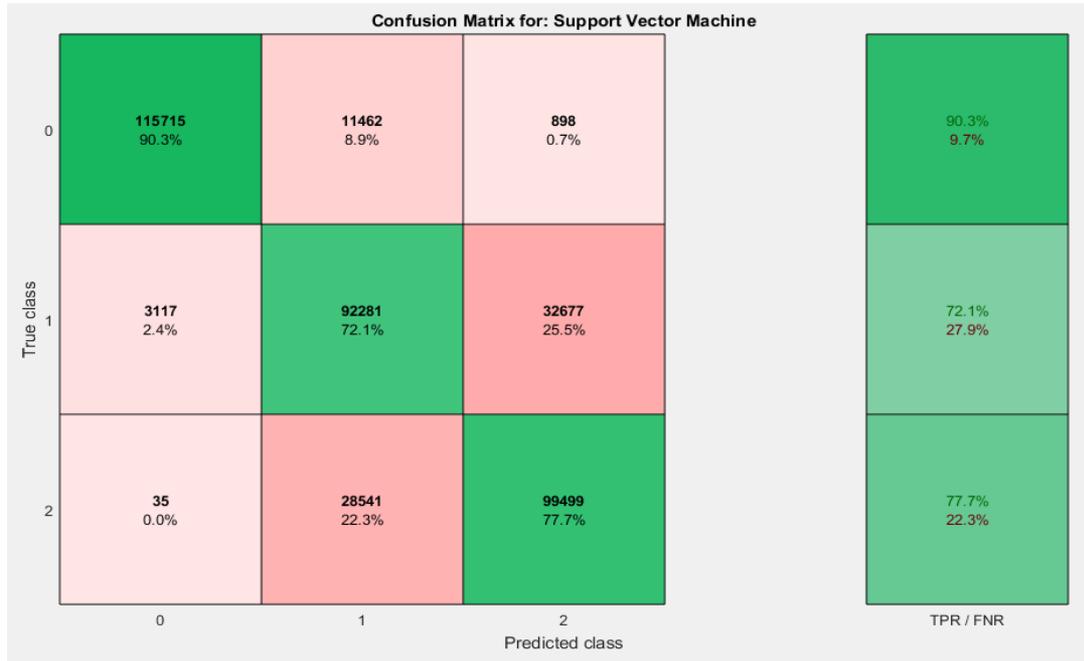


Tabla 32. Linear SVM CIE L*a*b*

- Matriz de confusión Quadratic SVM: Obtenemos una precisión del 84.3%. La clasificación de clases es mejor que en el caso de Linear SVM, pero su coste computacional es mayor, tarda casi dos días en ejecutarse completo. Lo descartamos como clasificador.

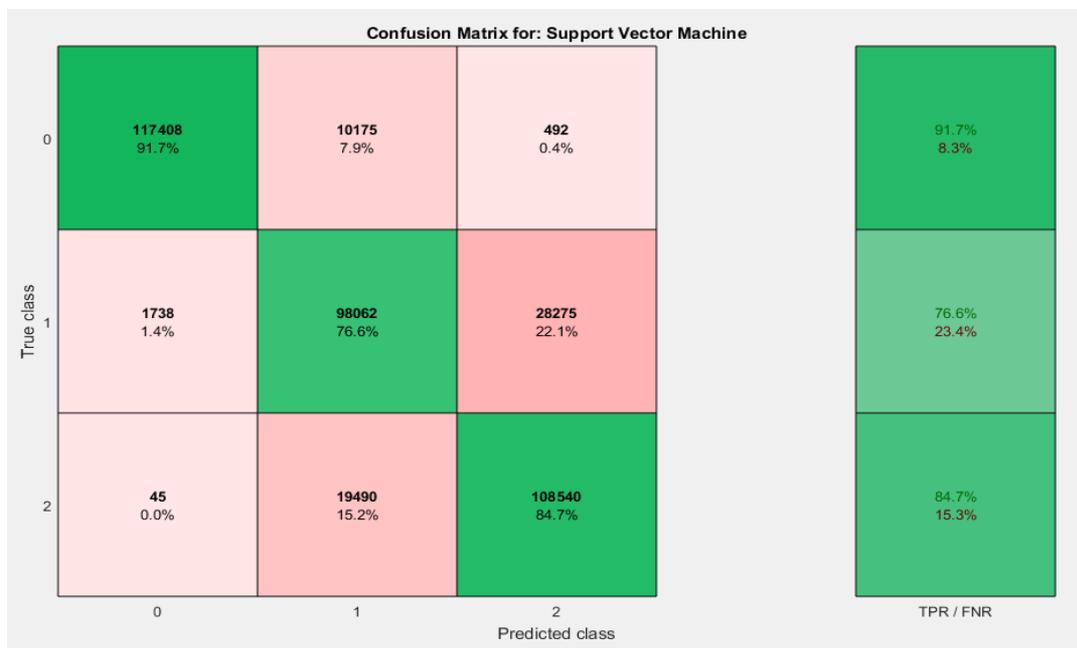


Tabla 33. Quadratic SVM CIE L*a*b*

Una vez visto los resultados anteriores podemos decir que SVM no es un buen clasificador para este caso, la principal causa es su elevado coste computacional. El mejor clasificador es Quadratic SVM que, aunque da una precisión buena no compensa con el tiempo que tarda en realizar la ejecución.

4.4 Resultados CIE94

- Matriz de confusión Simple Tree: Ofrece una precisión del 67.8%. No hace una buena clasificación de clases y por ello da un resultado muy bajo, por lo que lo descartamos como clasificador. Su coste computacional es mínimo, tardando pocos segundos en ejecutarse.

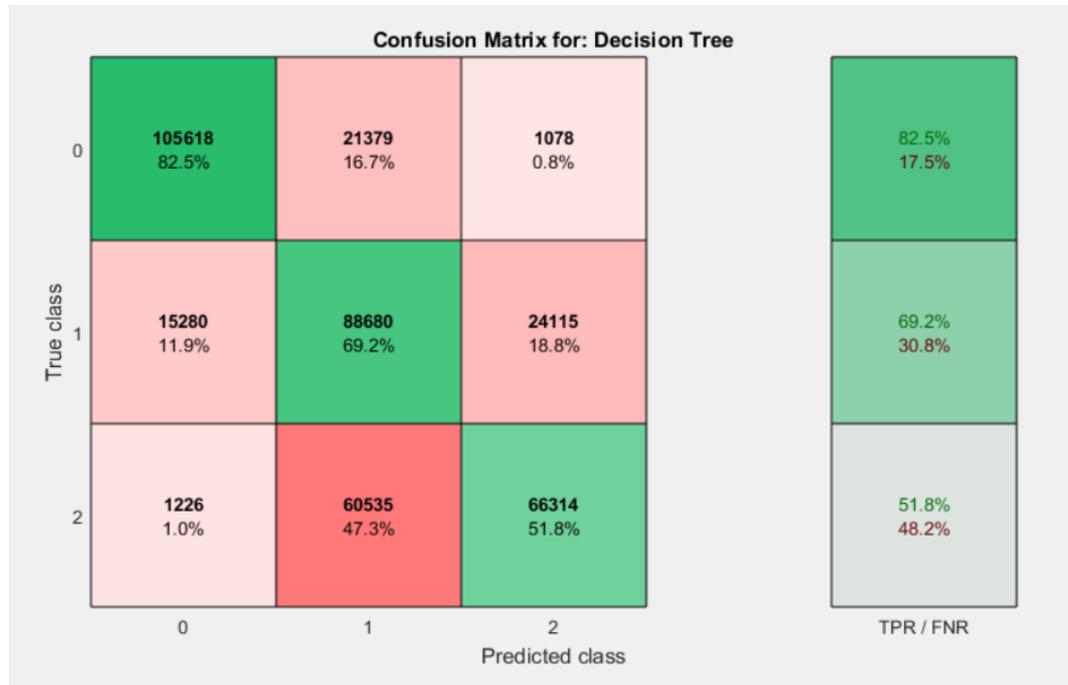


Tabla 34. Simple Tree CIE94

- Matriz de confusión Medium Tree: Obtenemos una precisión del 76.7%. Hace una clasificación decente de clases y su coste computacional también es mínimo, aun así, descartamos su uso.

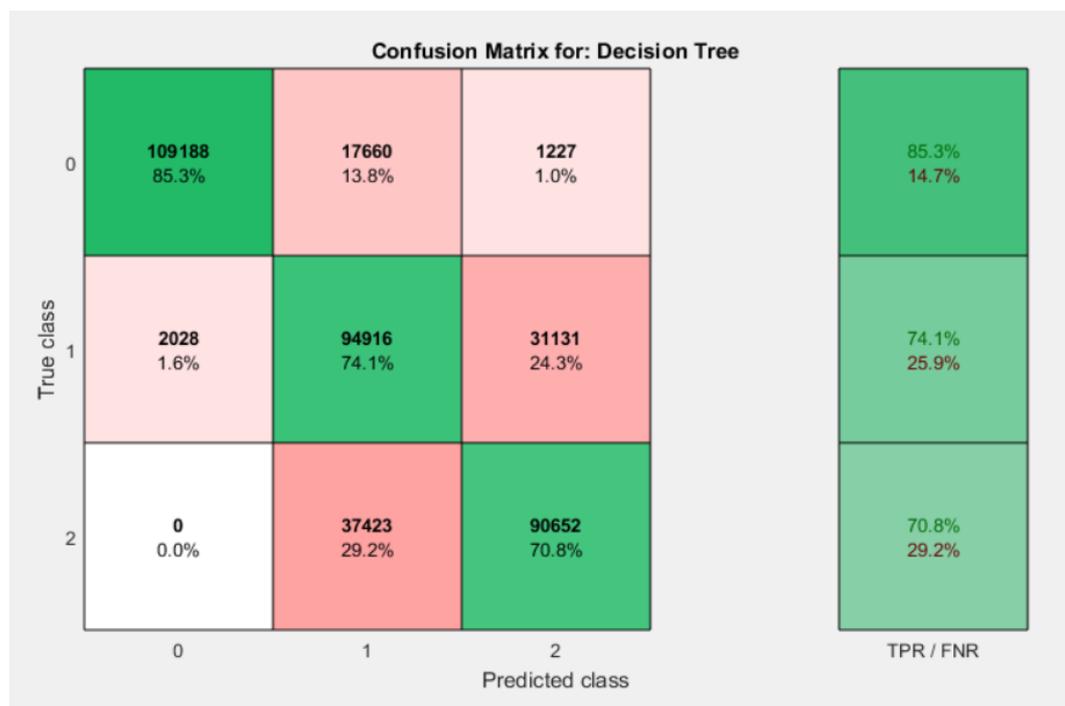


Tabla 35. Medium Tree CIE94

- Matriz de confusión Complex Tree: Obtenemos una precisión del 79.9%. Hace una buena clasificación de las clases siendo su coste computacional ínfimo. Sería un posible candidato como clasificador.

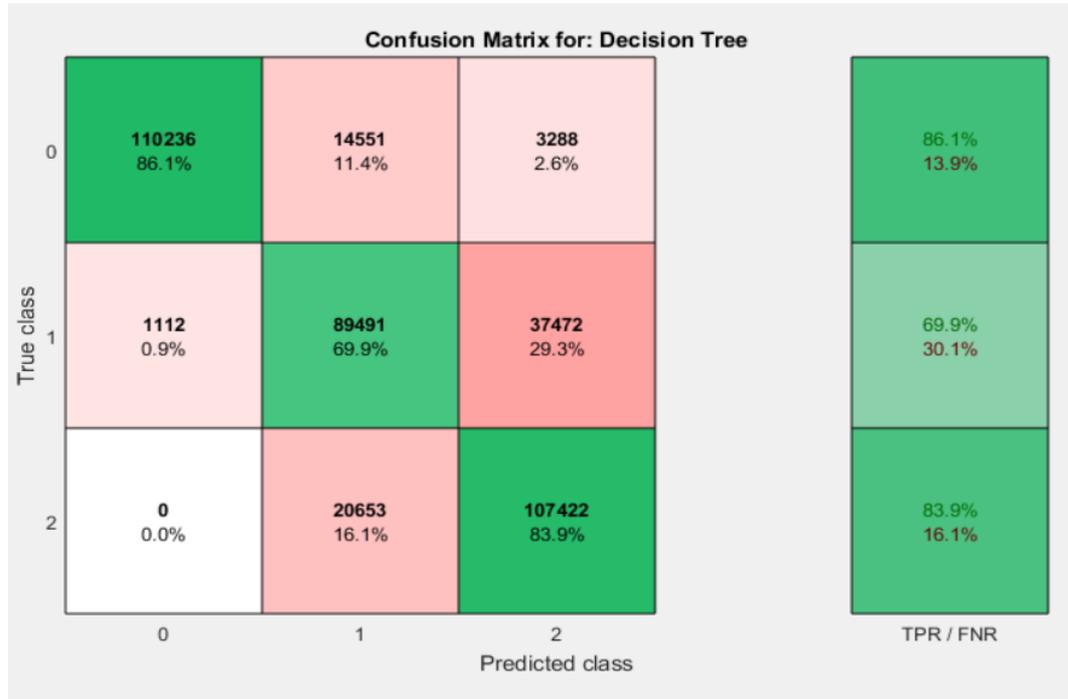


Tabla 36. Complex Tree CIE94

Con estos resultados podemos concluir que el clasificador ‘Decision Tree’ es aceptable, aunque no es el más apropiado, pero tiene un coste computacional muy bajo haciendo que se ejecute en pocos segundos. El mejor es Complex Tree.

- Matriz de confusión Fine KNN: Conseguimos una precisión de 87.6% siendo buena la clasificación de clases. Tiene un coste computacional medio, tardando alrededor de 1 hora en ejecutarse. Sería un posible candidato como clasificador.

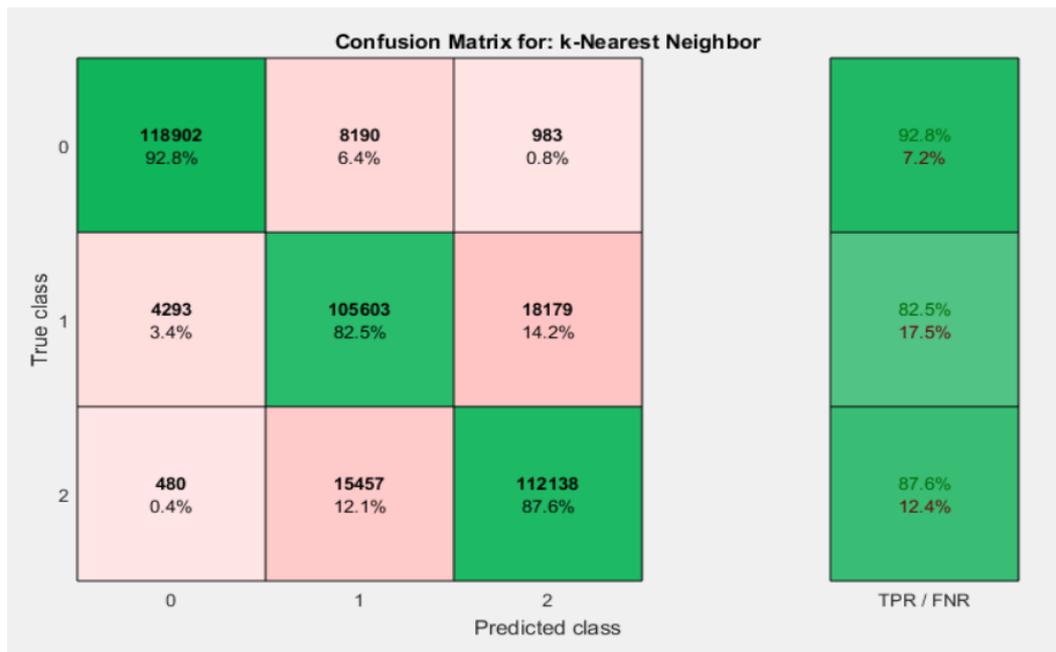


Tabla 37. Fine KNN CIE94

- Matriz de confusión Medium KNN: Ofrece una precisión del 87.4%. Hace bastante bien la clasificación de clases. Tiene un coste computacional muy parecido a Fine KNN y es también posible candidato como clasificador.

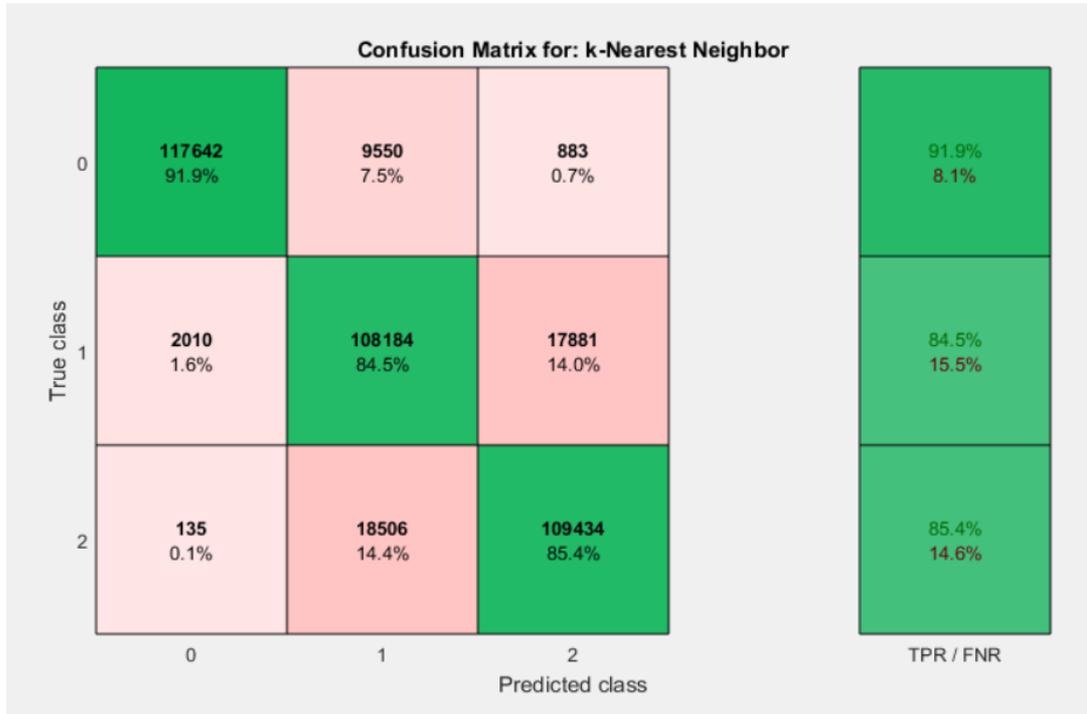


Tabla 38. Medium KNN CIE94

- Matriz de confusión Coarse KNN: Conseguimos una precisión del 83.5%, siendo peor la clasificación de clases que sus anteriores. Su coste computacional es mayor, pudiendo tardar varias horas. Lo descartamos como clasificador.



Tabla 39. Coarse KNN CIE94

- Matriz de confusión Cosine KNN: Obtenemos una precisión del 87.3%, clasificando muy bien entre clases. Su coste computacional es muy parecido a Coarse KNN. Es un posible candidato a clasificador.

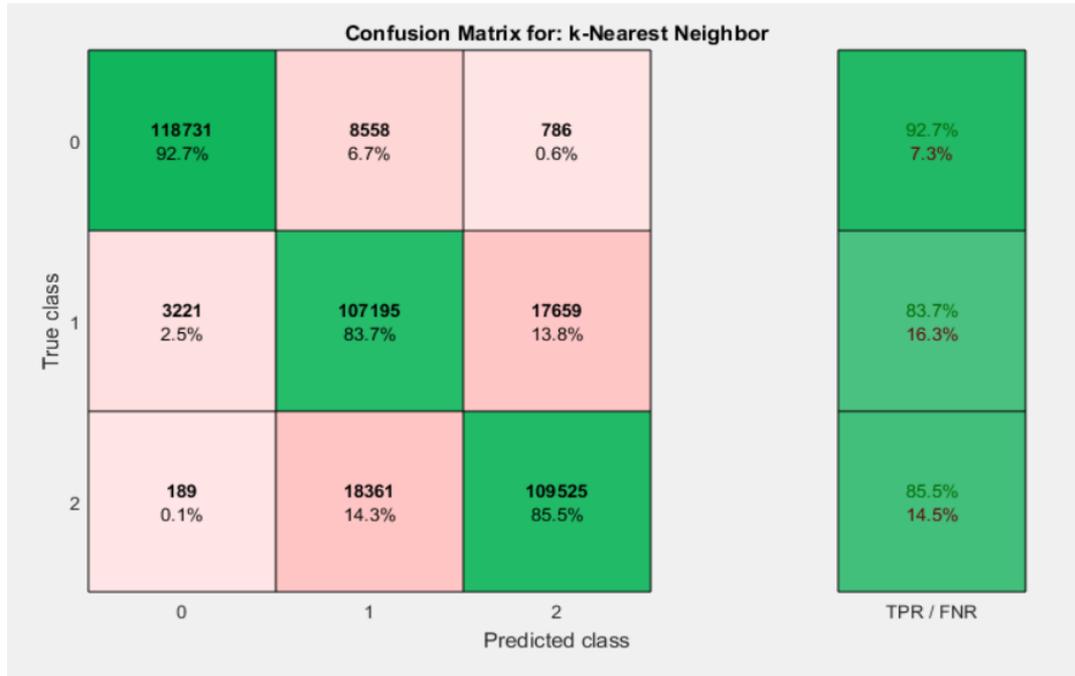


Tabla 40. Cosine KNN CIE94

- Matriz de confusión Weighted KNN: Da una precisión del 88.4%, siendo la mayor de todas por lo que es el que mejor diferencia entre clases. Su coste computacional es medio, muy parecido al de Medium KNN, aunque tarda un poco más. Es el mejor candidato a clasificador.

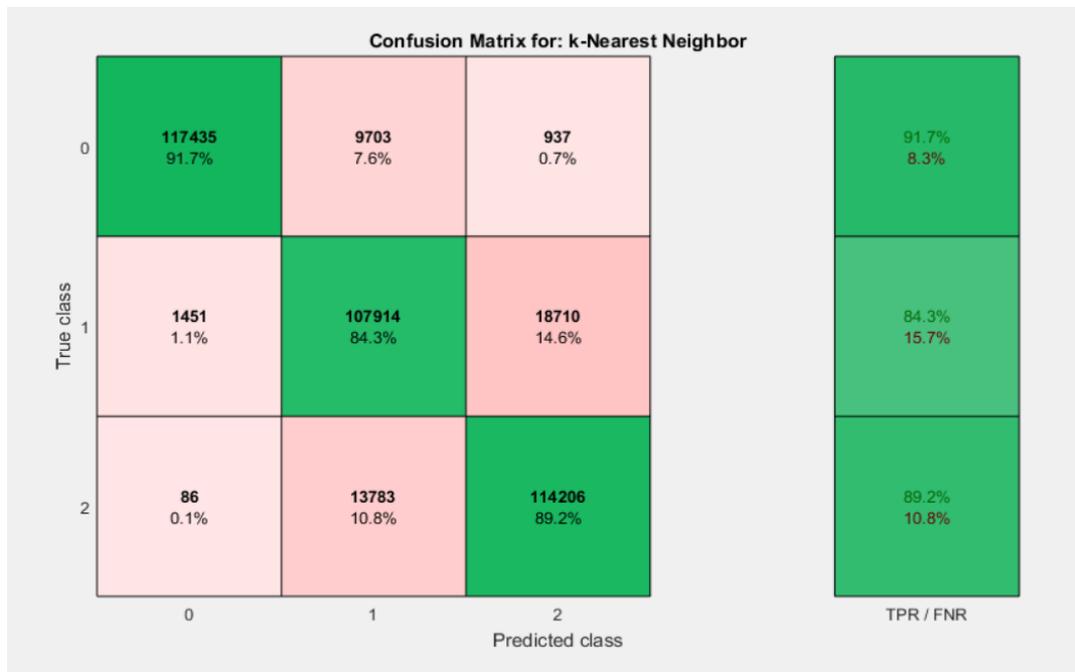


Tabla 41. Weighted KNN CIE94

A la vista de estos resultados podemos concluir que KNN es buen clasificador siendo el más apropiado de usar. Hace una buena distinción de clases, además su tiempo de ejecución y computacional es medio. El mejor es Weighted KNN.

Evaluación de espacios y distancias de color en el cálculo de los gradientes de color para detección automática de glaucoma

- Matriz de confusión Linear SVM: Ofrece una precisión del 80.7% siendo aceptable su distinción entre clases. Su coste computacional es muy alto, llegando a tardar un día en ejecutarse. Lo descartamos como clasificador.

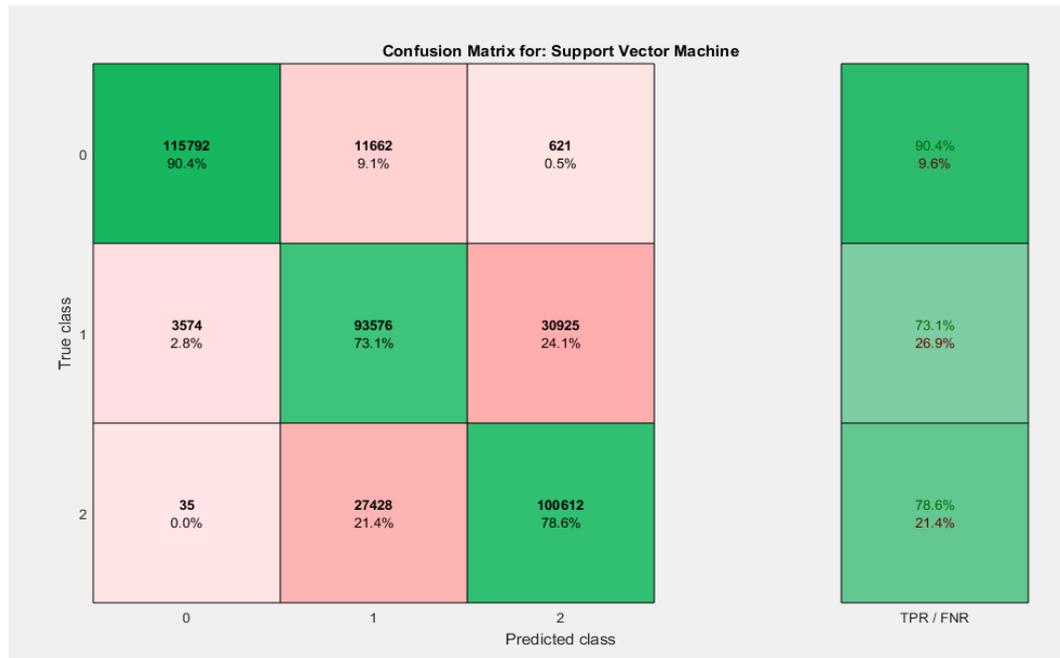


Tabla 42. Linear SVM CIE94

- Matriz de confusión Quadratic SVM: Obtenemos una precisión del 85.2%. La clasificación de clases es mejor que en el caso de Linear SVM, pero su coste computacional es mayor, tarda casi dos días en ejecutarse completo. Lo descartamos como clasificador.

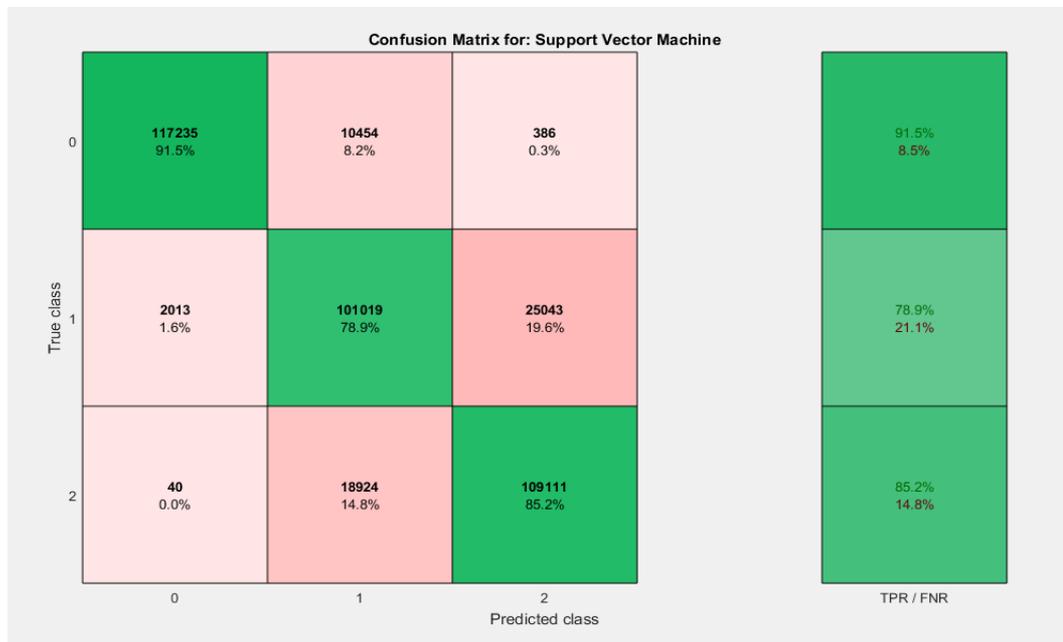


Tabla 43. Quadratic SVM CIE94

Una vez visto los resultados anteriores podemos decir que SVM no es un buen clasificador para este caso, la principal causa es su elevado coste computacional. El mejor clasificador es Quadratic SVM que, aunque da una precisión buena no compensa con el tiempo que tarda en realizar la ejecución.

4.5 Resultados CIE2000

- Matriz de confusión Simple Tree: Ofrece una precisión del 64%. No hace una buena clasificación de clases y por ello da un resultado muy bajo, por lo que lo descartamos como clasificador. Su coste computacional es mínimo, tardando pocos segundos en ejecutarse.

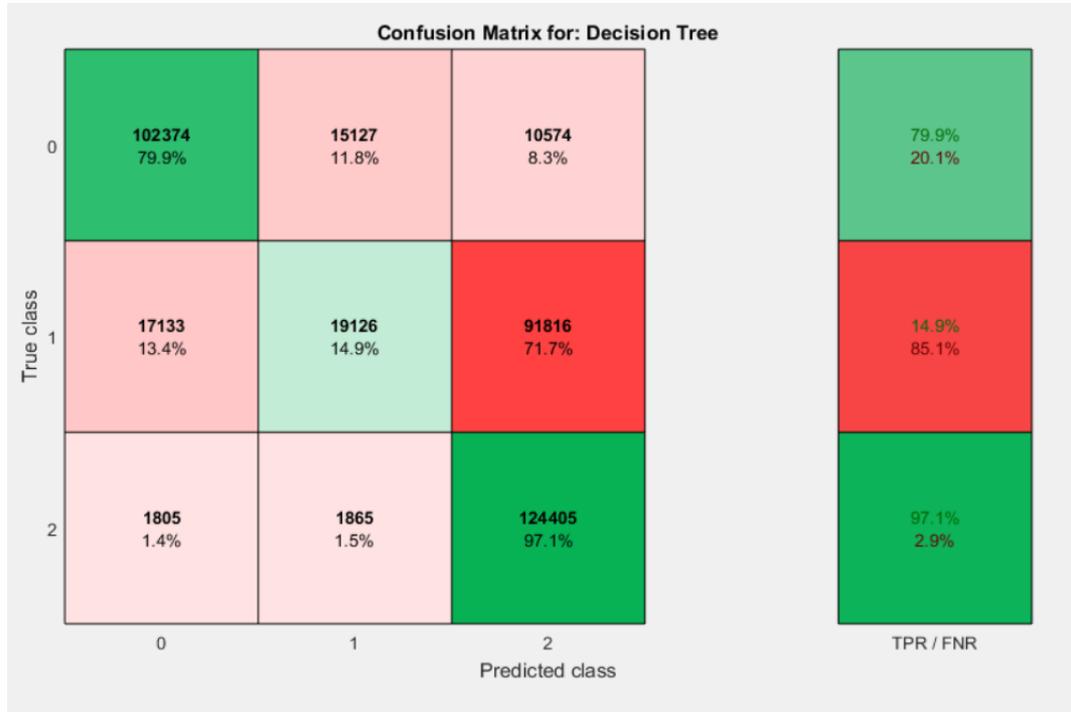


Tabla 44. Simple Tree CIE2000

- Matriz de confusión Medium Tree: Obtenemos una precisión del 74.8%. Hace una clasificación decente de clases y su coste computacional también es mínimo, aun así descartamos su uso.

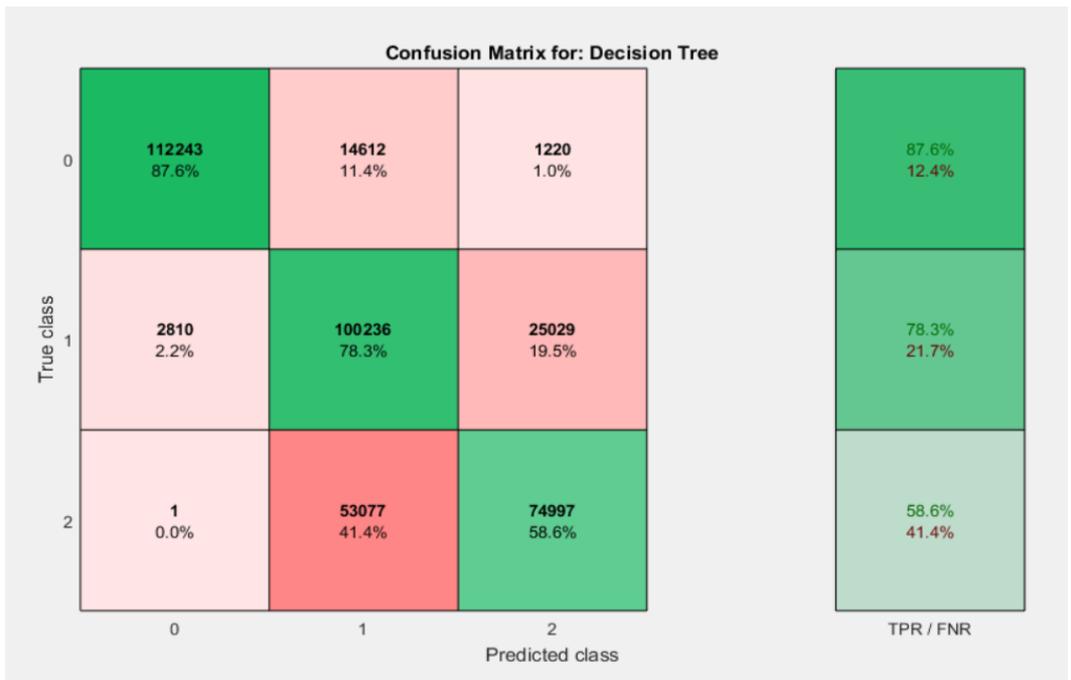


Tabla 45. Medium Tree CIE2000

- Matriz de confusión Complex Tree: Obtenemos una precisión del 80.1%. Hace una buena clasificación de las clases siendo su coste computacional ínfimo. Sería un posible candidato como clasificador.

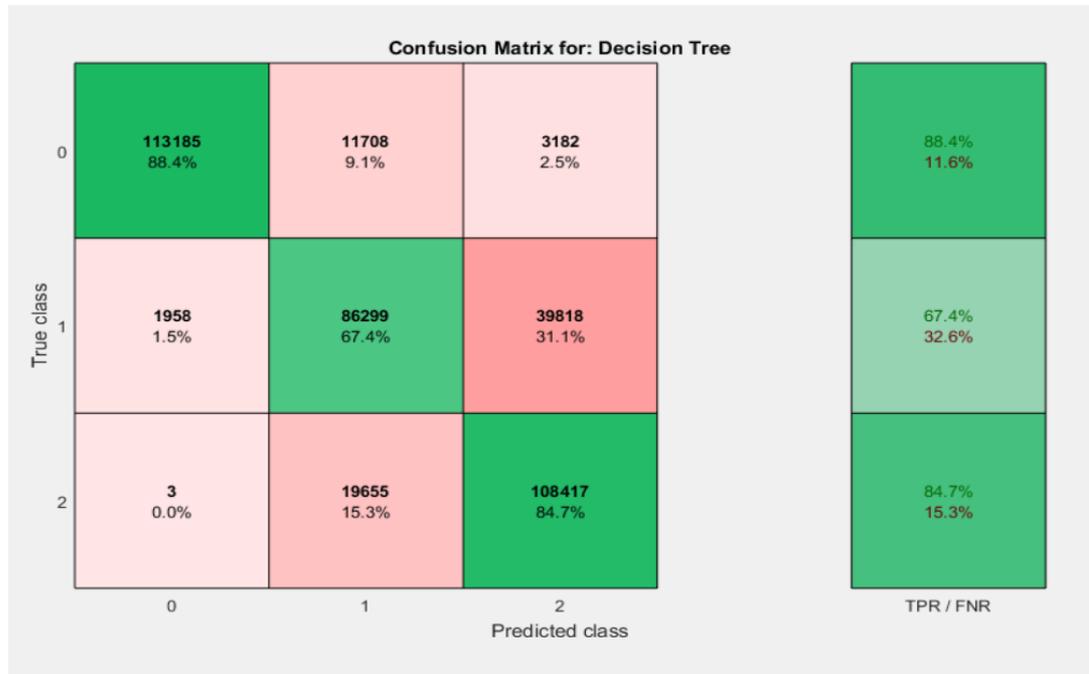


Tabla 46. Complex Tree CIE2000

Con estos resultados podemos concluir que el clasificador 'Decision Tree' es aceptable, aunque no es el más apropiado, pero tiene un coste computacional muy bajo haciendo que se ejecute en pocos segundos. El mejor es Complex Tree.

- Matriz de confusión Fine KNN: Conseguimos una precisión de 86.9% siendo buena la clasificación de clases. Tiene un coste computacional medio, tardando alrededor de 1 hora en ejecutarse. Sería un posible candidato como clasificador.

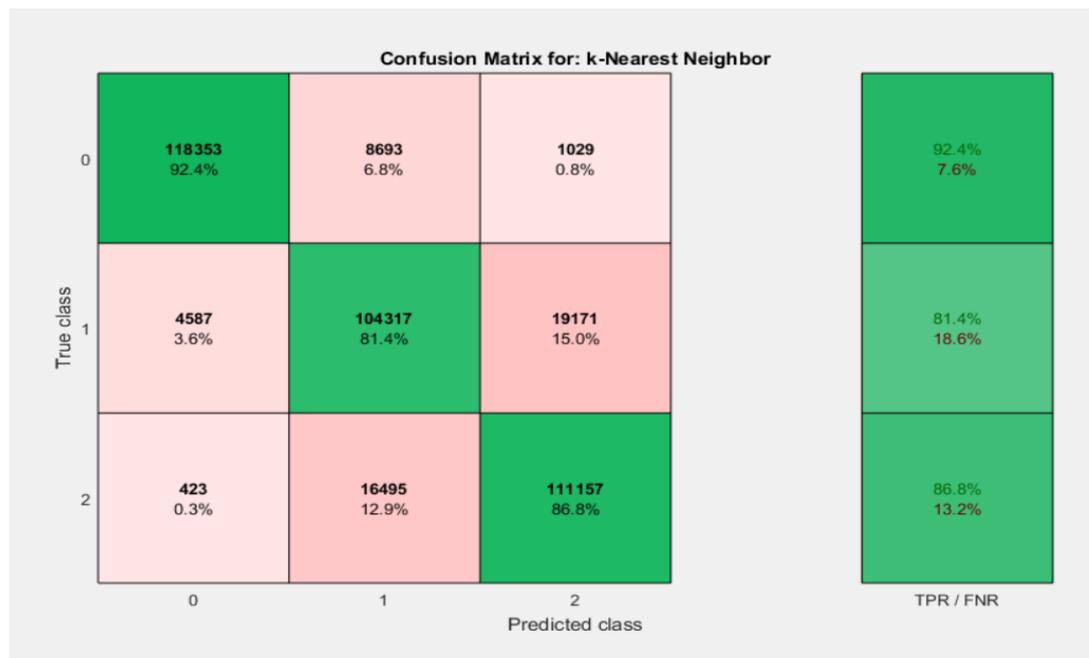


Tabla 47. Fine KNN CIE2000

- Matriz de confusión Medium KNN: Ofrece una precisión del 86.8%. Hace bastante bien la clasificación de clases. Tiene un coste computacional muy parecido a Fine KNN y es también posible candidato como clasificador.

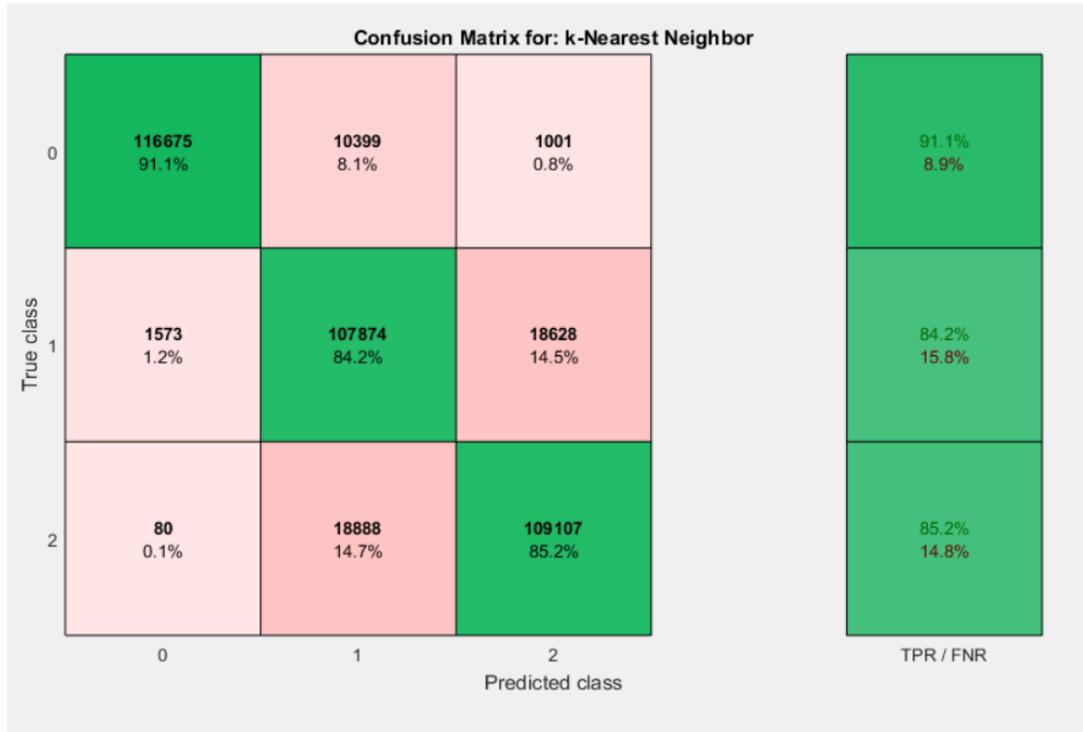


Tabla 48. Medium KNN CIE2000

- Matriz de confusión Coarse KNN: Conseguimos una precisión del 83.4%, siendo peor la clasificación de clases que sus anteriores. Su coste computacional es mayor, pudiendo tardar varias horas. Lo descartamos como clasificador.

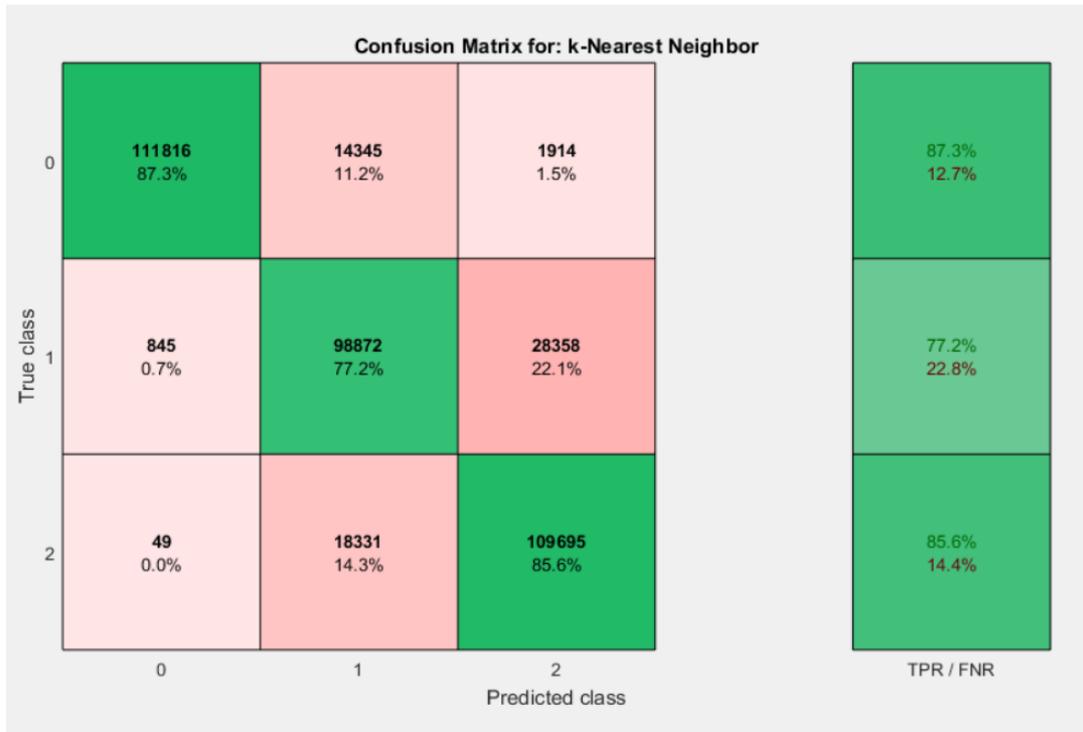


Tabla 49. Coarse KNN CIE2000

- Matriz de confusión Cosine KNN: Obtenemos una precisión del 87%, clasificando muy bien entre clases. Su coste computacional es muy parecido a Coarse KNN. Es un posible candidato a clasificador.

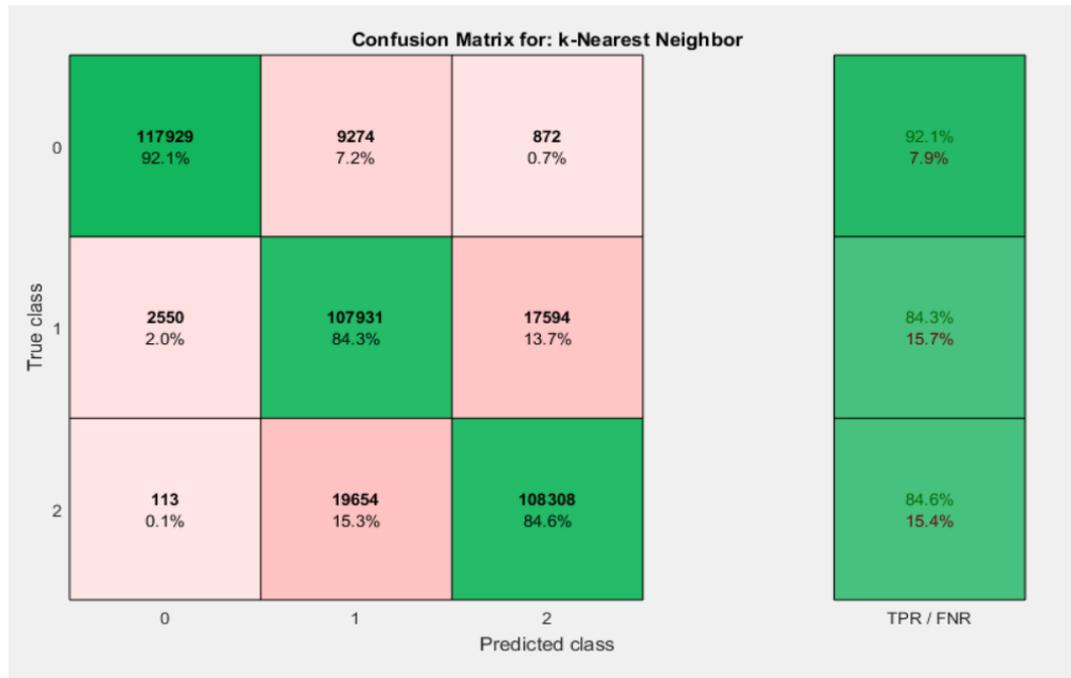


Tabla 50. Cosine KNN CIE2000

- Matriz de confusión Weighted KNN: Da una precisión del 87.8%, siendo la mayor de todas por lo que es el que mejor diferencia entre clases. Su coste computacional es medio, muy parecido al de Medium KNN, aunque tarda un poco más. Es el mejor candidato a clasificador.

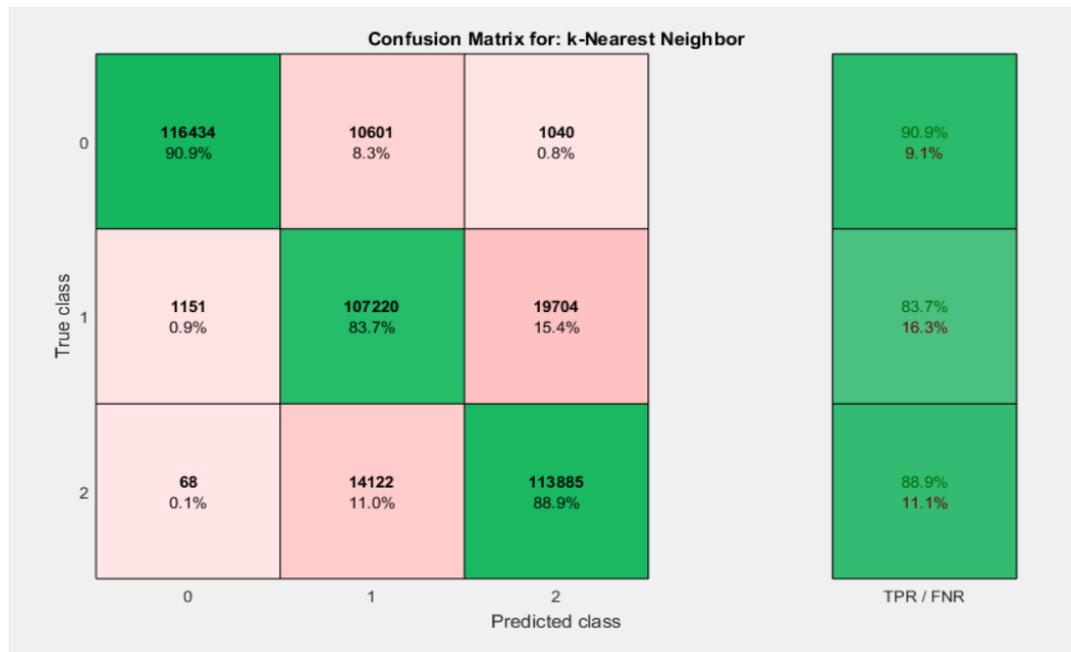


Tabla 51. Weighted KNN CIE2000

A la vista de estos resultados podemos concluir que KNN es buen clasificador siendo el más apropiado de usar. Hace una buena distinción de clases, además su tiempo de ejecución y computacional es medio. El mejor es Weighted KNN.

- Matriz de confusión Linear SVM: Ofrece una precisión del 79.6% siendo aceptable su distinción entre clases. Su coste computacional es muy alto, llegando a tardar un día en ejecutarse. Lo descartamos como clasificador.

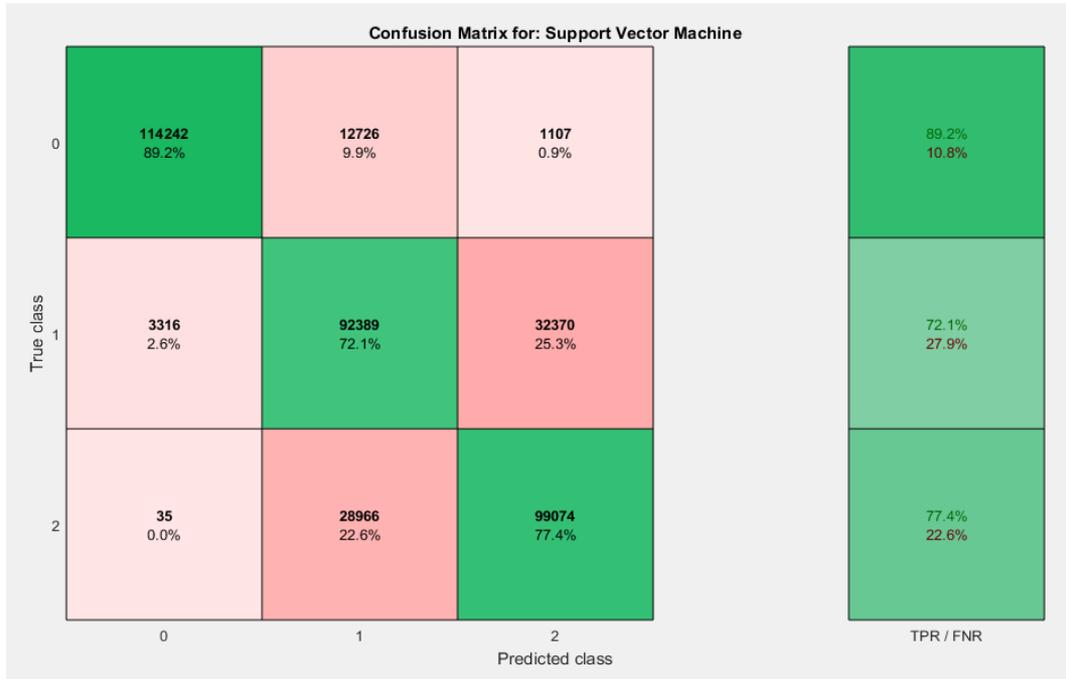


Tabla 52. Linear SVM CIE2000

- Matriz de confusión Quadratic SVM: Obtenemos una precisión del 83.9%. La clasificación de clases es mejor que en el caso de Linear SVM, pero su coste computacional es mayor, tarda casi dos días en ejecutarse completo. Lo descartamos como clasificador.

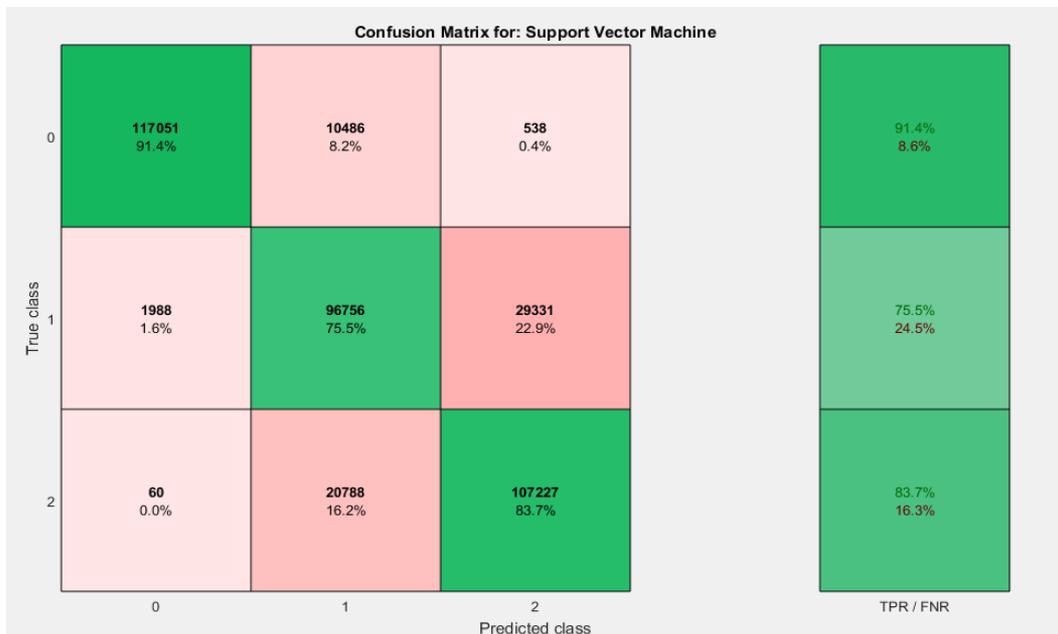


Tabla 53. Quadratic SVM CIE2000

Una vez visto los resultados anteriores podemos decir que SVM no es un buen clasificador para este caso, la principal causa es su elevado coste computacional. El mejor clasificador es Quadratic SVM que, aunque da una precisión buena no compensa con el tiempo que tarda en realizar la ejecución.

Evaluación de espacios y distancias de color en el cálculo de los gradientes de color para detección automática de glaucoma

A modo visual en la Tabla 54 tenemos un resumen de todos los resultados de precisión obtenidos en tanto por ciento.

Gradientes/ Clasificador	Simple Tree	Medium Tree	Complex Tree	Fine KNN	Medium KNN	Coarse KNN	Cosine KNN	Weighted KNN	Linear SVM	Quadratic SVM
Gris Euclídea	68	76.9	79.5	84.5	85	81.8	85.5	86	80.6	83.8
RGB Euclídea	67.9	76.7	79.7	86.8	86.8	83.1	86.9	87.8	81	85
CIE L*a*b* Euclídea	63.5	76.9	80.9	89.6	89	85	89	90.1	80	84.3
CIE94	67.8	76.7	79.9	87.6	87.4	83.5	87.3	88.4	80.7	85.2
CIE2000	64	74.8	80.1	86.9	86.8	83.4	87	87.8	79.6	83.9

Tabla 54. Resultados finales en tanto por ciento. En rojo clasificadores descartados; en verde candidatos a clasificadores; en negrita mejor clasificado.

	Simple Tree	Medium Tree	Complex Tree	Fine KNN	Medium KNN	Coarse KNN	Cosine KNN	Weighted KNN	Linear SVM	Quadratic SVM
Gradientes	<1min	< 1min	< 2min	<90min	<90 min	<150min	<150min	<90min	<1440min	<2880min

Tabla 55. Tiempos de ejecución de los clasificadores.

5 CONCLUSIONES

Una vez obtenido todos los resultados de los diferentes clasificadores podemos concluir que en general son bastante buenos, ya que la zona a segmentar destaca por su dificultad a la hora de distinguir correctamente entre las distintas clases que hemos propuesto.

Según la teoría los que mejor resultados deben dar son los referentes al espacio CIE $L^*a^*b^*$, concretamente CIE94 y su actualización CIE2000 ya que son los que mejor se adaptan al sistema visual humano, debido a que el usuario tiene el control sobre la relación de claridad contra croma. Está demostrado que ambos tienen una mejora del triple respecto de otras distancias de CIE $L^*a^*b^*$. Seguidos de los resultados del espacio RGB y finalmente el de Gris porque usa un único plano. Si observamos los resultados que tenemos vemos que no coinciden del todo con la teoría. Los mejores resultados han sido obtenidos por CIE $L^*a^*b^*$ euclídea, superando a CIE94 y CIE2000. La principal causa es que tanto para CIE94 como en CIE2000 se han utilizado los valores por defecto, esto provoca que su precisión se vea reducida al no utilizar los parámetros que mejor se adaptan a la visión humana. También se puede observar que no hay mucha variación entre CIE94 y CIE2000, la mayoría de las veces CIE2000 es superior que es lo normal ya que es una mejora de CIE94.

Si hacemos una comparación por espacios de color si coincide con la teoría, puesto que CIE $L^*a^*b^*$ da mejores resultados que RGB y este a su vez que Gris.

También comentar que los clasificadores que hemos descartado directamente son debido a que tiene una precisión muy baja, o que la precisión que se obtiene no compensa con el tiempo de ejecución. Los clasificadores que hemos considerado como candidatos son porque la precisión obtenida si compensaba con el tiempo de ejecución o porque da una muy buena precisión. Para nuestro caso el mejor clasificador es weighted KNN con un 90.1% de precisión, es decir, la mejor forma de clasificar nuestro algoritmo es por el método de vecindad usando pesos.

Como líneas futuras se podría proponer usar los valores en CIE 94 y CIE2000 que mejor se adaptan a la visión humana y no los de por defecto, viendo así si realmente influyen dichos factores o utilizar nuevos gradientes de color como JCH. Otra posibilidad sería optimizar este algoritmo e implementarlo como aplicación móvil, pudiendo así usarse en las consultas, ahorrando tiempo al médico experto pudiendo detectar la enfermedad de una forma más sencilla y precoz.

6 REFERENCIAS

- [1] Z.G. Fei y Seiji Zeng, Glaucoma: Etiology, Pathogenesis and Treatments, Nova Science Publishers, Incorporated, 2012-04-01.
- [2] R.C. González y R.E. Woods, Digital Image Porcessing, Addison Wesley, 1992.
- [3] X. F. Hermida, «Sistema Visual Humano,» Escola de Enxeñería de Telecomunicación, Pontevedra, 2008.
- [4] «El sistema visual humano | Luces CEI,» [En línea]. Disponible: <https://www.lucescei.com/estudios-y-eficiencia/extractos-libro-blanco-de-iluminacion/el-sistema-visual-humano/>. [Último acceso: 2 Agosto 2018].
- [5] «El Sistema Visual Humano,» [En línea]. Disponible: http://www7.uc.cl/sw_educ/neurociencias/html/115.html. [Último acceso: 2 Agosto 2018].
- [6] C. Baviera, «Nervio óptico dañado,» [En línea]. Disponible: <https://www.clinicabaviera.com/nervio-optico-danado/>. [Último acceso: 3 Agosto 2018].
- [7] L. España, «Lóbulo occipital: áreas y funciones,» 9 Agosto 2012. [En línea]. Disponible: <http://www.hela03.es/2012/08/lobulo-occipital-areas-y-funciones/>. [Último acceso: 4 Agosto 2018].
- [8] «¿Qué es el Glaucoma ocular? Síntomas y tratamientos,» [En línea]. Disponible: <https://www.barraquer.com/que-tratamos/glaucoma/>. [Último acceso: 4 Agosto 2018].
- [9] «Glaucoma | Causas, tipos, riesgos y tratamiento del glaucoma | ICR,» [En línea]. Disponible: <https://icrcat.com/enfermedades-oculares/glaucoma/>. [Último acceso: 5 Agosto 2018].
- [10] «Oftalmoseo: Diagnóstico de glaucoma,» [En línea]. Disponible: <https://www.oftalmoseo.com/patologias-frecuentes-2/diagnostico-de-glaucoma/>. [Último acceso: 5 Agosto 2018].
- [11] D. J. Monés, «Retinografía | Institut de la Màcula,» 25 Marzo 2013. [En línea]. Disponible: <http://www.institutmacula.com/procedimiento/retinografia/>. [Último acceso: 6 Agosto 2018].
- [12] B. Rodrigo, S. Morales, V. Naranjo, A. Colomer y M. Alcañiz, «Detección de glaucoma mediante la combinación de la relación copa/disco y la regla ISNT,» España, 2014.
- [13] «Espacio de color - Diccionario de fotografía y diseño,» [En línea]. Disponible: <https://www.fotonostra.com/glosario/espaciocolor.htm>. [Último acceso: 6 Agosto 2018].
- [14] C. Vialfa, «Luz y color,» 30 Noviembre 2017. [En línea]. Disponible: <https://es.ccm.net/contents/730-luz-y-color>. [Último acceso: 8 Agosto 2018].
- [15] «¿Qué es Luz? - Su Definición, Concepto y Significado,» Conceptodefinicion, 23 Julio 2014. [En línea]. Disponible: <https://conceptodefinicion.de/luz/>. [Último acceso: 8 Agosto 2018].

- [16] Julia Díaz y Kostadin Koruchev, «Luz y Color,» Escuela Politécnica Superior Universidad Autónoma de Madrid, Madrid, 2005.
- [17] «La percepción del color,» [En línea]. Disponible: <https://store.pantone.com/es/es/la-percepcion-del-color>. [Último acceso: 10 Agosto 2018].
- [18] GREGORIO MOLINERO DIEZ y TUTORA: BEGOÑA ACHA PIÑERO, «Segmentación de imágenes en color basada en el crecimiento de regiones,» Sevilla, 2010.
- [19] «Espacio de color CIE RGB – HiSoUR Arte Cultura Historia.,» [En línea]. Disponible: <https://hisour.com/es/cie-rgb-color-space-24862/>. [Último acceso: 13 Agosto 2018].
- [20] D. J. F. E. Granado, «La medida psicofísica del color,» Universidad de Rioja, Rioja, 2010.
- [21] «Entendiendo El Espacio de Color CIE L*A*B*,» [En línea]. Disponible: <http://sensing.konicaminolta.com.mx/2014/09/entendiendo-el-espacio-de-color-cie-lab/>. [Último acceso: 15 Agosto 2018].
- [22] S. Westland, «Qué es el espacio de color CIE L*a*b* | Imagen Digital,» 2001. [En línea]. Disponible: http://www.gusgsm.com/espacio_color_cie_lab. [Último acceso: 15 Agosto 2018].
- [23] D. J. F. E. Granado, «La medida práctica del color,» Universidad de Rioja, Rioja, 2010.
- [24] «Conceptos de modos de color en Photoshop,» [En línea]. Disponible: <https://helpx.adobe.com/es/photoshop/using/color-modes.html>. [Último acceso: 16 Agosto 2018].
- [25] J. G. F. ERAÑA, «SINTESIS DIGITAL DE COLOR UTILIZANDO TONOS DE GRIS,» UNIVERSIDAD AUTÓNOMA DE SAN LUIS POTOSI, San Luis Potosi, 2009.
- [26] Wikipedia, «Color Difference,» [En línea]. Disponible: https://en.wikipedia.org/wiki/Color_difference. [Último acceso: 18 Agosto 2018].
- [27] P. G. a. L. MacDonald, Colour Engineering, John Wiley & Sons, LTD, 2003.
- [28] «Train models to classify data using supervised machine learning - MATLAB,» [En línea]. Disponible: <https://es.mathworks.com/help/stats/classificationlearner-app.html>. [Último acceso: 19 Agosto 2018].
- [29] «Decision Trees - MATLAB,» [En línea]. Disponible: <https://es.mathworks.com/help/stats/decision-trees.html>. [Último acceso: 19 Agosto 2018].
- [30] Wikipedia, «KNN,» [En línea]. Disponible: https://es.wikipedia.org/wiki/K_vecinos_m%C3%A1s_pr%C3%B3ximos. [Último acceso: 20 Agosto 2018].
- [31] «Classification Using Nearest Neighbors - MATLAB,» [En línea]. Disponible: <https://es.mathworks.com/help/stats/classification-using-nearest-neighbors.html>. [Último acceso: 20 Agosto 2018].
- [32] Wikipedia, «SVM,» [En línea]. Disponible: https://es.wikipedia.org/wiki/M%C3%A1quinas_de_vectores_de_soporte. [Último acceso: 21 Agosto 2018].

- [33] Ana Isabel García Noguera y Tutora: Irene Fondón García, «Segmentación automática de la excavación en retinografías basada en gradientes de color y clasificador Complex Tree,» Sevilla, 2015.
- [34] «Support Vector Machines for Binary Classification - MATLAB,» [En línea]. Disponible: <https://es.mathworks.com/help/stats/support-vector-machines-for-binary-classification.html>. [Último acceso: 22 Agosto 2018].
- [35] Fernandez-Granero, M. A, Sarmiento, A, Sanchez-Morillo, D, Jiménez, S, Alemany, P y Fondón, I, «Automatic CDR Estimation for Early Glaucoma Diagnosis,» Journal of Healthcare Engineering, Sevilla, 2017.

