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

Extracción y selección de características para el análisis de la profundidad del melanoma

Autor: María De la Cruz Arriero
Tutoras: Begoña Acha Piñero
Maria Del Carmen Serrano Gotarredona

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

Sevilla, 2024

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

Extracción y selección de características para el análisis de la profundidad del melanoma

Autora:
María De la Cruz Arriero

Tutoras:
Begoña Acha Piñero
María del Carmen Serrano Gotarredonda
Catedráticas de Universidad

Dpto. de Teoría de la Señal y Comunicaciones
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla
Sevilla, 2024

Trabajo Fin de Grado: Extracción y selección de características para el análisis de la profundidad del melanoma

Autora: María De la Cruz Arriero

Tutoras: Begoña Acha Piñero
María Del Carmen Serrano Gotarredonda

El tribunal nombrado para juzgar el Trabajo arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2024

El Secretario del Tribunal

A mi familia

A mis maestros

Agradecimientos

Quisiera expresar mi más sincero agradecimiento a todas las personas involucradas en la realización de este trabajo.

A mi familia, cuyo apoyo incondicional, paciencia y confianza han sido fundamentales para alcanzar esta meta.

Este logro es tan mío como suyo, y les dedico con gratitud cada esfuerzo reflejado en estas páginas.

Resumen

El melanoma es un tipo de cáncer de piel agresivo y de creciente incidencia, por lo que su detección temprana y precisa resulta fundamental. En este estudio, empleamos técnicas de aprendizaje automático (en inglés Machine Learning, ML) para clasificar el melanoma en superficial o profundo, enfocándonos en el análisis del color y explorando diversos espacios de color y sus propiedades.

Los resultados destacan la efectividad de las técnicas de selección de características, así como la importancia de aquellas que se asemejan a la percepción humana del color, lo cual mejora la precisión en la clasificación.

Sin embargo, los resultados presentan un número considerable de clasificaciones incorrectas, lo que indica margen de mejora en la precisión del modelo.

Abstract

Melanoma is an aggressive type of skin cancer with a rising incidence rate, making its early and accurate detection essential. In this study, we used Machine Learning (ML) techniques to classify melanoma as either superficial or deep, focusing on color analysis and exploring various color spaces and their properties. The results highlight the effectiveness of feature selection techniques and the importance of those features that resemble human color perception, enhancing classification accuracy. However, the results also show a considerable number of incorrect classifications, indicating room for improvement in model accuracy.

Índice

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

Notación

ML	*Machine Learning* (Aprendizaje Automático)
IA	*Inteligencia Artificial*
SVM	*Support Vector Machines* (Máquinas de Vectores de Soporte)
SBS	*Sequential Backward Selection* (Selección Secuencial hacia atrás)
SFS.	*Sequential Forward Selection* (Selección Secuencial hacia adelante)
SEOM	*Sociedad Española de Oncología Médica*
TDS	*Total Dermatoscopic Score* (Indice Dermatoscópico Total)
PI	*Permutation Importance* (Importancia por permutación)
CIE	*Comisión Internacional de Iluminación*
CAM	*Color Appearance Models*
UCS	*Uniform Color Space* (Espacio de color uniforme)
RBF	*Radial Basis Function* (Núcleo radial)
CNNs	*Redes neuronales convolucionales*
BT	*Breslow Thickness (*Profundidad de Breslow)
KD	*Knowledge Distillation* (Destilación de conocimiento)
DLS	*Deep Transfer Learning (Aprendizaje Profundo por Transferencia)*
TP	*True Positive* (Verdadero Positivo)
TN	*True Negative* (Verdadero Negativo)
FP	*False Positive* (Falso Positivo)
FN	*False Negative* (Falso Negativo)

1 INTRODUCCIÓN

"The goal of machine learning in dermatology is to assist clinicians in making faster, more accurate diagnoses".

- Ofer Mandelboim, dermatólogo y experto en IA -

La motivación de este estudio radica en la necesidad de mejorar el diagnóstico y la evaluación de los melanomas cutáneos. El melanoma es una forma agresiva de cáncer de piel que puede ser mortal si no se detecta y trata a tiempo. La precisión en la estimación de la profundidad del melanoma es crucial, ya que esta medida está estrechamente relacionada con el pronóstico y el plan de tratamiento del paciente. Sin embargo, los métodos actuales de diagnóstico presentan limitaciones significativas.

En este contexto, el uso de técnicas de *Machine Learning* (ML) ofrece una solución prometedora. ML puede analizar grandes volúmenes de datos de imágenes dermatoscópicas y detectar patrones complejos que podrían no ser evidentes para el ojo humano. Estas técnicas permiten una evaluación más precisa y automática de la profundidad del melanoma, mejorando así la precisión y eficiencia del diagnóstico médico en una variable de contextos.

1.1 Objetivos

El objetivo principal de este trabajo es obtener resultados favorables en la diferenciación entre melanomas superficiales y profundos utilizando características de color extraídas de imágenes dermatológicas, aplicando técnicas de *machine learning*.

Además, se plantean los siguientes objetivos específicos:

- Entender la importancia del factor profundidad en el melanoma.
- Explorar y conocer las características de los espacios de color CIELAB, CAM16 UCS y CAM16.
- Implementar un modelo de clasificación SVM (*Support Vector Machine*), optimizando su rendimiento mediante la validación cruzada *K-Fold*.
- Aplicar técnicas de selección de características como SBS (*Sequential Backward Selection*) y SFS (*Sequential Fordward Selection*).
- Procesar y analizar los resultados obtenidos para evaluar el rendimiento del modelo.

1.2 Inteligencia Artificial y *Machine Learning*

La inteligencia artificial (IA), un término acuñado por el profesor emérito de Stanford John McCarthy en 1955, fue definido por él como "la ciencia y la ingeniería de hacer máquinas inteligentes". [1] Es un campo de estudio dedicado a la creación de sistemas que pueden realizar tareas que requieren inteligencia humana. [2]

Dentro de la IA, el *machine learning* (ML) o aprendizaje automático es un subcampo que estudia cómo los agentes informáticos pueden mejorar su percepción, conocimiento, pensamiento o acciones a partir de la experiencia o los datos.

Dentro del aprendizaje automático tenemos subcategorías como el aprendizaje supervisado, una computadora aprende a predecir etiquetas dadas por humanos y el aprendizaje no supervisado no requiere etiquetas, a veces generando sus propias tareas de predicción. [1]

En resumen, la IA es un concepto más amplio que permite que una máquina o un sistema detecte, actúe o se adapte como una persona, mientras que el ML es una aplicación de la IA que hace posible que las máquinas extraigan conocimientos de los datos y aprendan de ellos de forma automática.

1.3 Análisis médico

1.3.1 Melanoma

El melanoma es un tipo de cáncer de piel que se desarrolla cuando los melanocitos (las células que dan a la piel su color bronceado o marrón) comienzan a crecer sin control. El melanoma es mucho menos común que otros tipos de cáncer de piel, sin embargo, es más peligroso porque tiene una mayor probabilidad de propagarse a otras partes del cuerpo si no se detecta y trata a tiempo. [3]

Un lunar, llaga, úlcera o tumor sobre la piel pueden ser un signo de melanoma o de otro tipo de cáncer de piel. Una úlcera o tumor que sangra o cambia de color también puede ser un signo de cáncer de piel. [5]

Figura 1-1. Melanoma cutáneo maligno.

La mayoría de los cánceres de piel comienzan en la capa superior de la piel, llamada epidermis. Hay tres tipos de células en esta capa:

- Células escamosas: Estas son células planas en la parte superior (externa) de la epidermis, que se desprenden constantemente a medida que se forman nuevas.
- Células basales: Estas células se encuentran en la parte inferior de la epidermis, llamada capa basal. Estas células se dividen constantemente para formar nuevas células que reemplazan a las células escamosas que se desprenden de la superficie de la piel. A medida que estas células se mueven hacia arriba en la epidermis, se aplanan, convirtiéndose eventualmente en células escamosas.
- Melanocitos: Estas son las células que pueden convertirse en melanoma. Normalmente producen un pigmento marrón llamado melanina, que da a la piel su color bronceado o marrón. La melanina protege las capas más profundas de la piel de algunos de los efectos dañinos del sol.

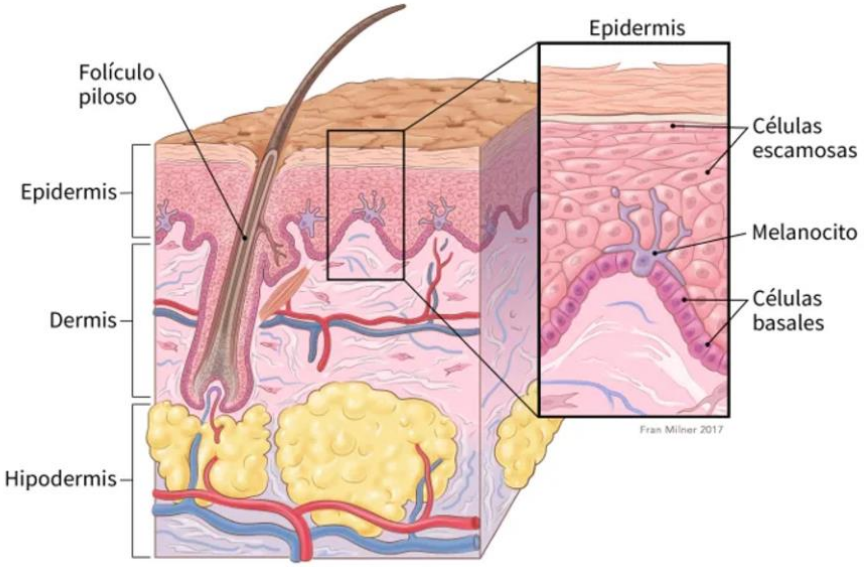

Figura 1-2. Capas de la piel y sus células [3]

La epidermis está separada de las capas más profundas de la piel (dermis y tejido subcutáneo) por la membrana basal. Cuando un cáncer de piel se vuelve más avanzado, generalmente crece a través de esta barrera y penetra en las capas más profundas. [3]

Según ha publicado la SEOM: Sociedad Española de Oncología Médica el 22 de mayo de 2024, la incidencia de melanoma cutáneo en España está al alza. La tasa anual por edad por cada 100.000 personas se ha incrementado de 2003 a 2024. En concreto, ha pasado de 12,0 a 15,1 casos en mujeres y de 12,0 a 15,4 en hombres, lo que supone una subida anual del 1,1% y del 1,2% respectivamente y lo que sitúa al melanoma cutáneo como el 8.º cáncer más frecuente en el primer caso y el 11.º, en el segundo.

Respecto a la mortalidad cabe destacar que en 2021 se contabilizaron 1.056 defunciones (2 por cada 100.000 personas/año), 586 en hombres y 471 en mujeres. Estas cifras sitúan al melanoma cutáneo en el 17.º lugar como el cáncer más mortal, en el primer caso, y en el 18.º en el segundo. Además, se observa que en los hombres la tasa ha crecido un 0,1% de forma anual entre 2003 y 2021 mientras en las mujeres ha bajado un 0,4%. [4]

1.3.2 Diagnóstico actual

Además de preguntar por los antecedentes médicos y familiares, y de hacer un examen físico, es posible que el médico realice las siguientes pruebas y procedimientos para detectar y diagnosticar el melanoma.

1.3.2.1 Dermatoscopia

La dermatoscopia es una técnica no invasiva que permite examinar detalladamente las lesiones cutáneas utilizando un instrumento llamado dermatoscopio. Esta técnica proporciona una visión más profunda de las estructuras de la piel que no son visibles a simple vista y se utiliza principalmente para la evaluación y diagnóstico de diversas afecciones cutáneas, incluyendo el melanoma. [6]

Existen diversos algoritmos diagnósticos que intentan facilitar y objetivar en la medida de lo posible el diagnóstico de melanoma. Hablaremos sobre el método ABCD, el método de Menzies y la lista de los 7 puntos, que son los que más repercusión han tenido. [7]

1. El sistema ABCD se basa en la valoración de 4 criterios:

- **A**simetría: Se divide la lesión pigmentada en dos ejes de 90º, de manera que consigamos la mayor simetría posible, y se valora la asimetría con respecto el color, la forma y estructuras en ambos lados del eje. Se otorga una puntuación de 0 si no existe asimetría en ningún eje; de 1, si existe asimetría en un eje y de 2, si la presenta en ambos ejes. De manera que una lesión puede tener una puntuación de 0 a 2.
- **B**ordes: La lesión es dividida en 8 segmentos y puntuaremos con 1 cada porción que presente una finalización abrupta del borde. De tal manera que la puntuación mínima que se puede obtener es 0 y la máxima, 8.
- **C**olor: Se valora la presencia de 6 colores: blanco, marrón claro, marrón oscuro, azul-gris, rojo y negro; por lo que la puntuación máxima sería de 6 y la mínima de 1. El blanco tan sólo puntuará si es más claro que la piel adyacente; es decir cuando pertenece a áreas blancas de regresión.
- **E**structuras dermatoscópicas: Se consideran 5 estructuras cada una de las cuales puntúa con un punto, por lo que la puntuación máxima será de 5 y la mínima de 1.

  Posteriormente, hemos de multiplicar cada puntuación individual por un factor de corrección que depende del valor diagnóstico individual en cada caso calculado por análisis multivariable. Finalmente, sumaremos todas las puntuaciones parciales y obtendremos un valor llamado índice dermatoscópico total (TDS).

Criterio dermatoscópico	Puntuación	Factor de corrección
Asimetría	0 a 2	x 1,3
Bordes	0 a 8	x 0,1
Color	1 a 6	x 0,5
Estructuras dermatoscópicas	1 a 5	x 0,5
TDS - Índice dermatoscópico total		
< 4,75	4,8 - 5,45	> 5,45
Benigna	Sospechosa	Maligna

Tabla 1-1. Regla del ABCD

2. El método de los 7 puntos de Argenziano tiene en cuenta 7 criterios dermatoscópicos, a los cuales se les da una puntuación en función de su importancia:

Criterio dermatoscópico	Puntuación máxima
CRITERIOS MAYORES	
Retículo pigmentado atípico	2
Velo azul-blanquecino	2
Patrón vascular atípico	2
CRITERIOS MENORES	
Proyecciones irregulares	1
Puntos/glóbulos irregulares	1
Manchas de pigmento irregulares	1
Estructuras asociadas a regresión	1
PUNTUACIÓN TOTAL	
< 3	≥ 3
BENIGNA	MALIGNA

Tabla 1-2. Lista de los 7 puntos

Finalmente se realiza la suma de las puntuaciones obtenidas y si el cómputo general es igual o superior a 3, se diagnosticará la lesión de melanoma con una sensibilidad del 95% y una especificidad del 75%.

3. El método de Menzies valora 11 criterios dermatoscópicos; éstos se dividen en criterios negativos, que no deben estar presentes para el diagnóstico de melanoma, y criterios positivos, es decir, que alguno de ellos debe cumplirse para el diagnóstico de malignidad.

   Con el presente método, para llegar al diagnóstico de melanoma, la lesión no debe presentar ninguno de los dos criterios negativos, es decir, la lesión no debe ser monocroma ni simétrica y, además, debe presentar, al menos, uno de los nueve criterios positivos.

Criterios Negativos: (Ninguno debe presentarse)
Simetría
Monocromía
Criterios Positivos: (al menos uno debe estar presente)
Velo Azul-blanquecino
Despigmentación pseudocicatricial
Múltiples colores (5-6)
Retículo pigmentado prominente
Pseudópodos
Proyecciones radiales
Múltiples puntos marrones
Puntos/glóbulos negros periféricos
Múltiples puntos azul-gris

Figura 1-3. Método de Menzies para el diagnóstico del melanoma

1.3.2.2 Biopsia

La biopsia es un procedimiento para extirpar el tejido anormal y una pequeña cantidad del tejido normal que lo rodea. Un patólogo observa el tejido al microscopio para detectar células cancerosas. Hay cuatro tipos de biopsias de piel, que dependen del lugar donde se formó el área anormal y el tamaño del área [6]:

1. Biopsia por rasurado: procedimiento para el que se usa una cuchilla de afeitar estéril para rasurar la masa que se ve anormal.
2. Biopsia con sacabocados: procedimiento que usa un instrumento principal llamado sacabocado o trépano para extraer una muestra en forma de círculo del tejido que se ve anormal.
3. Biopsia por incisión: procedimiento para el que se usa un bisturí a fin de extraer parte de una masa.
4. Biopsia por escisión: procedimiento para el que se usa un bisturí a fin de extraer toda la masa.

Sin embargo, estos métodos tienen limitaciones. La dermatoscopia depende en gran medida de la experiencia del dermatólogo y puede ser subjetiva, mientras que la biopsia es invasiva y puede causar cicatrices. Evaluar la profundidad del melanoma es particularmente desafiante, ya que implica la interpretación de características histológicas que pueden ser difíciles de discernir, por lo tanto, existe una necesidad urgente de métodos más precisos y automáticos para su estimación. Las técnicas de *Machine Learning* (ML) han mostrado un gran potencial en este ámbito. ML puede analizar grandes volúmenes de datos de imágenes dermatoscópicas y detectar patrones complejos que podrían no ser evidentes para el ojo humano, mejorando así la precisión del diagnóstico y reduciendo la subjetividad.

1.3.3 Clasificación del melanoma según su profundidad

La clasificación del melanoma en superficial o profundo describe el grado de invasión del tumor en la piel y es crucial para determinar tanto el tratamiento como el pronóstico del paciente. El índice de Breslow y los niveles de Clark son dos métodos usados para clasificar la profundidad y la extensión de los melanomas en la piel, lo cual ayuda a determinar el pronóstico y el tratamiento.

1.3.3.1 El índice de Breslow

En 1970, Alexander Breslow introdujo un método clave para evaluar el grado de invasión, conocido como Breslow Thickness (BT) o "grosor de Breslow". Este parámetro mide en milímetros la profundidad del melanoma desde la capa granulosa de la epidermis hasta la célula tumoral más profunda encontrada en la dermis o tejido subyacente. La medición del BT ha demostrado una correlación directa con el riesgo de metástasis. [8]

El grosor de Breslow se utiliza para dividir el melanoma en distintas etapas, las cuales tienen implicaciones pronósticas y terapéuticas claras:

- **Melanoma fino** (< 0.76 mm): En los estudios iniciales de Breslow, se demostró que los melanomas con un grosor inferior a este umbral tenían un riesgo prácticamente nulo de metástasis. Por ello, se consideran melanomas con buen pronóstico.
- **Melanoma intermedio** (0.76 - 4 mm): A medida que aumenta el grosor, el riesgo de invasión linfática y metástasis sistémica incrementa de manera significativa. En este rango, factores adicionales como la presencia de úlceras o el índice mitótico también son importantes para determinar el riesgo.
- **Melanoma grueso** (> 4 mm): Estas lesiones suelen asociarse con un pronóstico más desfavorable debido a su alta probabilidad de metástasis y recurrencia. Este subtipo requiere intervenciones más agresivas.

1.3.3.2 Los niveles de Clark

Los **Niveles de Clark** describen el grado de invasión del melanoma según las capas de la piel afectadas. Este sistema clasifica el melanoma en cinco niveles, cada uno relacionado con el pronóstico del paciente. A continuación, se detalla cada nivel según el artículo de Clark et al. (1969): [9]

- **Nivel I**: El melanoma está confinado a la epidermis y no ha invadido la dermis subyacente. Este estadio se conoce como *in situ* y tiene el mejor pronóstico.
- **Nivel II**: Las células cancerosas han penetrado la membrana basal y alcanzado la dermis papilar (la capa superior de la dermis), pero sin acumularse en la unión con la dermis reticular.
- **Nivel III**: El tumor ha invadido hasta la unión entre la dermis papilar y la dermis reticular, pero no ha penetrado significativamente en la dermis reticular.
- **Nivel IV:** La invasión alcanza la dermis reticular, donde las fibras de colágeno comienzan a organizarse en haces más gruesos. Este nivel indica una mayor agresividad del melanoma.
- **Nivel V:** El melanoma invade el tejido subcutáneo, alcanzando las capas más profundas de la piel. Este nivel está asociado con el peor pronóstico.

La clasificación de los niveles de Clark permite correlacionar el grado de invasión del melanoma con la supervivencia y el manejo clínico del paciente.

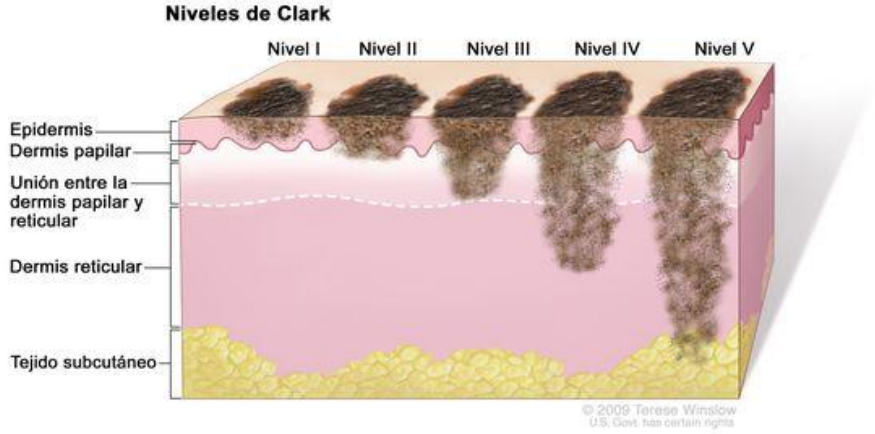

Figura 1-3 Niveles de Clark para el melanoma maligno. [10]

1.3.3.3 Clasificación del melanoma

En resumen, predecir con precisión el grosor del melanoma es esencial para planificar la extensión de la cirugía, evaluar el riesgo de propagación, decidir la necesidad de tratamientos adicionales y estimar el pronóstico. La detección temprana y la medición exacta del grosor del melanoma son, por tanto, fundamentales para mejorar los resultados en los pacientes.

El índice de Breslow se considera el indicador pronóstico más valioso, el cual clasifica el melanoma en cuatro niveles de gravedad, basados en distintos umbrales de profundidad.

Otro índice es el creado por G. Argenziano et al., que agrupa el grosor del melanoma en dos categorías, basado en el índice de Breslow. Básicamente, agrupa todas las etapas de II a IV en un solo grupo, diferenciando entre melanoma de baja profundidad (<0,76 mm) y de alta profundidad (>0,76 mm). [11]

Etapas	Profundidad (mm)
Etapa I	< 0,76
Etapa II	>0,76

Tabla 1-4. Etapas del melanoma

Las imágenes que conforman la base de datos utilizado para este trabajo están clasificadas en estas dos etapas.

1.4 Estado del arte

La clasificación del melanoma según su grosor de Breslow es un tema clave en dermatología, dado que influye en el pronóstico y las decisiones terapéuticas.

Un estudio internacional realizado por Polesie et al. (2022) [26] evaluó la capacidad de 438 lectores internacionales (incluyendo dermatólogos certificados, residentes y médicos generales) para clasificar melanomas *in situ* (MIS) e invasivos, así como para predecir el grosor de Breslow, utilizando imágenes dermatoscópicas. Este estudio, basado en una plataforma web abierta, reunió datos de lectores con diversos niveles de experiencia en dermatoscopia y comparó su desempeño con modelos de aprendizaje profundo (*deep learning*).

El estudio utilizó 1456 imágenes dermatoscópicas de melanomas categorizados como:

- Melanomas in situ (MIS): 788 imágenes.
- Melanomas invasivos con grosor ≤1 mm: 474 imágenes.
- Melanomas invasivos con grosor >1 mm: 194 imágenes.

Los resultados obtenidos fueron:

- Precisión general en la clasificación MIS vs invasivos: 63.4% (MIS) y 71.0% (invasivos).
- Precisión en melanomas finos (≤1 mm): 85.9%.
- Precisión en melanomas gruesos (>1 mm): 70.8%.

1.4.1 Domínguez-Morales et. al (2024)

Domínguez-Morales et al. presentan un estudio basado en redes neuronales convolucionales (CNNs) para la clasificación del melanoma según su profundidad (*Breslow thickness*, BT) utilizando técnicas avanzadas de aprendizaje profundo, específicamente la combinación de aprendizaje supervisado y semi-supervisado con destilación de conocimiento (*Knowledge Distillation, KD*). [27]

Este trabajo analiza dos tareas principales relacionadas con el diagnóstico del melanoma:

1. Clasificación binaria según BT:
    - Menor de 0.8 mm
    - Mayor o igual a 0.8 mm
2. Distinción entre melanomas:
    - In situ (Mis)
    - Invasivos (Miv)

El estudio utiliza cuatro conjuntos de datos heterogéneos, destacando uno del Hospital Universitario Virgen del Rocío, junto con otros públicos, como el archivo ISIC, alcanzando un total de 1449 imágenes para entrenamiento y evaluación.

En cuanto a la metodología, se emplearon dos enfoques de aprendizaje profundo:

1. Aprendizaje supervisado: Se entrenaron modelos CNN tradicionales con datos fuertemente etiquetados.
2. Aprendizaje semi-supervisado: Mediante un algoritmo Multi-Teacher *KD*, cinco modelos supervisados actuaron como "profesores" para pseudo-anotar datos no etiquetados. Los datos pseudo-etiquetados se usaron luego en el entrenamiento de un modelo "estudiante", junto con los datos etiquetados originales.

Resultados obtenidos

- Clasificación según BT: El enfoque semi-supervisado alcanzó un AUC[1] promedio de 0.8768, superando al mejor modelo supervisado, que obtuvo un AUC de 0.8329.
- Clasificación Mis vs Miv: El modelo semi-supervisado logró un AUC promedio de 0.8547, frente a 0.7743 del supervisado.

- Clasificación multicategoría (Mis, Miv<0.8 mm, Miv≥0.8 mm): Aunque los resultados fueron inferiores (AUC de 0.7787 en semi-supervisado), se destaca la dificultad añadida por la similitud entre Mis y melanomas invasivos finos.

1.4.2 Hernández Rodriguez et. al (2023)

Hernández-Rodríguez et al. presentan un estudio centrado en la predicción de la profundidad de invasión del melanoma (*Breslow thickness*, BT) utilizando técnicas de deep transfer learning (DTL)[2]. Se comparan tres redes neuronales convolucionales preentrenadas (*ResNetV2*, *EfficientNetB6* e *InceptionV3*) para clasificar melanomas en dos tareas principales: [28]

1. Clasificación entre melanomas in situ (Mis) e invasivos.
2. Clasificación por BT, distinguiendo melanomas con profundidad < 0,8 mm o ≥ 0,8 mm.

El estudio emplea un conjunto de datos retrospectivo compuesto por 1315 imágenes dermatoscópicas de melanomas confirmados histopatológicamente. Las imágenes provienen de el Hopital Universitario Virgen del Rocío, repositorios de la International Skin Imaging Collaboration y la base de datos de Polesie et al.

Los datos se etiquetaron según el diagnóstico histopatológico para las dos tareas, y se dividieron en conjuntos de entrenamiento (80%), validación (10%) y prueba (10%).

Resultados obtenidos

- Tarea Mis vs. Invasivo:
  - *EfficientNetB6* presentó la mayor precisión diagnóstica (61%) y sensibilidad (72%), aunque con una especificidad más baja (39%).
  - El AUC promedio fue de 0.59 para *ResNetV2*, 0.63 para *InceptionV3* y 0.54 para *EfficientNetB6*.
- Tarea BT < 0,8 mm o ≥ 0,8 mm:
  - *EfficientNetB6* obtuvo la mayor precisión diagnóstica (75%) y especificidad (84%), con un AUC promedio de 0.69.
  - *ResNetV2* y *InceptionV3* alcanzaron un AUC promedio de 0.76 y 0.75, respectivamente, superando a los dermatólogos (AUC promedio de 0.70).

[1] El AUC (*Area Under the Curve*, por sus siglas en inglés) es una métrica utilizada en modelos de clasificación para evaluar su desempeño. Representa el área bajo la curva ROC (*Receiver Operating Characteristic*), que es una gráfica que muestra la relación entre la sensibilidad o recall (verdaderos positivos) y la especificidad o specificity (falsos positivos) del modelo en diferentes umbrales de decisión.
AUC = 1.0: Desempeño perfecto del modelo; separa correctamente todas las clases.

[2] Deep Transfer Learning (DLS) es una técnica en *machine learning* que aprovecha los modelos de aprendizaje profundo (*deep learning*) ya entrenados en un dominio o tarea específica y los reutiliza para resolver otra tarea relacionada.

Por lo tanto, *EfficientNetB6* destacó como el modelo más eficaz para la clasificación de BT, superando a dermatólogos expertos en la distinción de melanomas de profundidad ≥ 0.8 mm. Sin embargo, resaltan la necesidad de estandarizar las imágenes dermatoscópicas y de incluir poblaciones más diversas para mejorar la generalización de los modelos.

1.4.3 Saez et. al (2016)

Sáez et al. [29] presentan un sistema computacional basado en imágenes dermatoscópicas para estimar el grosor del melanoma (*Breslow thickness*, BT) utilizando técnicas de clasificación supervisada y aprendizaje automático.

El estudio utiliza un dataset compuesto por 250 imágenes dermatoscópicas de melanomas extraídas del *Interactive Atlas of Dermoscopy*, todas ellas con diagnóstico histopatológico confirmado. Estas imágenes se clasificaron en función del grosor (*Breslow thickness*, BT), dividido en dos esquemas:

1. Clasificación binaria:
    I. Finos (< 0.76 mm): Incluye 167 melanomas, entre los cuales hay 64 melanomas in situ**,** ya que comparten características clínicas similares con los melanomas finos invasivos.
    II. Gruesos (≥ 0.76 mm): Incluye 83 melanomas, distribuidos entre los intermedios y gruesos.
2. Clasificación en tres clases:
    I. Finos (< 0.76 mm): Mismos 167 melanomas que en el esquema binario.
    II. Intermedios (0.76-1.5 mm): 54 melanomas, considerados como un grupo de transición en términos de profundidad y pronóstico.
    III. Gruesos (> 1.5 mm): 29 melanomas, representando los casos más avanzados.

El sistema incluye los siguientes pasos:

1. Extracción de características: Inspirada en estudios clínicos, se extrajeron un total de ***81 descriptores*** basados en forma, color, textura y red de pigmnetos.
2. El modelo de clasificación empleado en este estudio es LIPU (*Logistic Regression using Initial variables and Product Units*), un enfoque híbrido que combina elementos de la regresión logística tradicional y las redes neuronales con unidades producto.

<u>Resultados obtenidos</u>

- Para la clasificación binaria se obtuvo una precisión del 77,6%.
- Para la clasificación en tres clases, ya que el problema radica en la distinción de la etapa II y III, se obtuvo una precisión de más del 55% en la peor clase clasificada.

2 MÉTODO

En esta sección se presentan los fundamentos teóricos de los métodos y algoritmos que se aplicarán en el desarrollo de este trabajo, incluyendo los espacios de color CIELAB, CAM16-UCS y CAM16, los métodos de cuantificación de color con *K-Means* y la dispersión. Se abordan también el clasificador SVM (*Support Vector Machines*), los enfoques de selección de características SBS (*Sequential Backward Selection*) y SFS (*Sequential Forward Selection*), y el Permutation Importance (p-*value*) como herramienta para evaluar la relevancia de las características. Con ello se establece la base técnica para el desarrollo de los métodos utilizados en el proyecto.

2.1 Espacios de color

La importancia del color en el análisis de características en imágenes es fundamental para mejorar la precisión y la fiabilidad de los resultados. El color esta intrínsecamente relacionado con la biología de las lesiones cutáneas, permitiendo a los profesionales de la salud identificar patrones y anomalías que pueden indicar la presencia de cáncer de piel.

Para percibir el color se necesita que la luz de un objeto llegue a nuestros ojos y estimule el proceso visual que, como tal tiene un aspecto fisiológico y otro psicológico. Por lo tanto, a la hora de especificar un color, es necesario la intervención de tres factores fundamentales:

- El tono o matiz, atributo por el que decimos si el color es un estímulo verde, rojo, azul...
- La cantidad de luz, luminosidad o brillo, atributo por el que decimos si el color es claro u oscuro.
- La saturación o croma, atributo por el que decimos si el color es más suave o fuerte.

Luego lo ideal sería especificar un color mediante tres números relacionados con esos tres atributos.

Figura 2-1. Atributos perceptuales para expresar un determinado color

A lo largo del tiempo se han ido desarrollando distintos sistemas para conseguir este objetivo, uno de ellos llamado los espacios de color, que son modelos matemáticos que permiten describir colores de manera estándar. [12]

2.1.1 El espacio de color CIE RGB

El espacio de representación de color CIE RGB es el que surge al representar en un sistema de ejes cartesianos las coordenadas tricromáticas de los estímulos de color en el sistema de primarios RGB de la CIE (Comisión Internacional de Iluminación). Representa el color como una mezcla de tres componentes: rojo, verde y azul.

Los primarios del CIE RGB están definidos en longitudes de onda específicas (700 nm para rojo, 546.1 nm para verde, y 435.8 nm para azul), cada uno contribuyendo a un estímulo percibido que es una combinación lineal de estos tres colores.

La representación matemática de cualquier color en este espacio es una combinación lineal de los tres componentes, que se expresa como:

$$C = R * r + G * g + B * b \quad (1\text{-}1)$$

donde R, G, y B son los valores de intensidad de cada color, y r, g y b son las respuestas espectrales de los primarios rojo, verde y azul. [13]

2.1.2 El espacio de color CIE XYZ

El espacio CIE XYZ fue diseñado a partir de la base científica de la percepción del color, especialmente la teoría tricromática, que establece que el ojo humano percibe el color a través de tres tipos de conos. Sin embargo, debido a que los primarios RGB no pueden cubrir todo el espectro visible, se construyeron funciones de igualación de color que permitieron definir tres primarios imaginarios X, Y, y Z. Estos primarios son matemáticamente derivados y no corresponden a colores físicos, pero garantizan una cobertura completa del espectro visible sin valores negativos.

Para representar cualquier color percibido por el ojo humano, se usaron funciones de igualación ($\bar{x}(\lambda)$, $\bar{y}(\lambda)$, $\bar{z}(\lambda)$, que describen cómo los colores percibidos responden a distintas longitudes de onda.

La representación de un color en CIE XYZ se calcula a partir de su espectro de luz reflejada (S(λ)) y una fuente de luz (E(λ)). Esto se integra con las funciones de igualación:

$$X = \int S(\lambda)E(\lambda)\bar{x}(\lambda)d\lambda \quad (1\text{-}2)$$

$$Y = \int S(\lambda)E(\lambda)\bar{y}(\lambda)d\lambda \quad (1\text{-}3)$$

$$Z = \int S(\lambda)E(\lambda)\bar{z}(\lambda)d\lambda \quad (1\text{-}4)$$

Estos valores tristímulo, X, Y, y Z, representan las cantidades de los primarios imaginarios necesarias para igualar un color dado. En este modelo, la componente Y representa la luminancia y se corresponde con la percepción de brillo del color, mientras que X y Z no tienen correspondencia directa con propiedades visuales específicas. [13]

Las coordenadas cromáticas se definen como:

$$x = \frac{X}{X+Y+Z} \quad (1\text{-}5)$$

$$y = \frac{Y}{X+Y+Z} \quad (1\text{-}6)$$

2.1.3 El espacio de color CIELAB

El espacio de color CIELAB (o CIE L*a*b*), desarrollado por la Comisión Internacional de Iluminación (CIE) en 1976, es un modelo perceptual diseñado para representar los colores de manera más alineada con la percepción humana. Esto hace que CIELAB sea un modelo perceptualmente uniforme, es decir, que los cambios iguales en sus valores representan aproximadamente cambios iguales en la percepción de color.

El espacio CIELAB está compuesto por tres componentes principales:

1. **L***: Representa la luminosidad o claridad del color y varía entre 0 (negro) y 100 (blanco).
2. **a***: Representa la dimensión de verde a rojo, con valores negativos para los tonos verdes y positivos para los tonos rojos.
3. **b***: Representa la dimensión de azul a amarillo, con valores negativos para los tonos azules y positivos para los tonos amarillos.

Los valores de L*, a* y b* se derivan a partir de los valores tristímulo X, Y, y Z del espacio CIE XYZ mediante una transformación no lineal, que se expresa matemáticamente como:

$$L^* = 116 \cdot f\left(\frac{Y}{Y_n}\right) - 16 \quad (1\text{-}7)$$

$$a^* = 500 \cdot \left(f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right)\right) \quad (1\text{-}8)$$

$$b^* = 200 \cdot \left(f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right)\right) \quad (1\text{-}9)$$

donde Xn, Yn , y Zn son los valores tristímulo de referencia del blanco en el entorno de visualización y f(t)f(t)f(t) es una función de transformación que suaviza la respuesta del color:

$$f(t) = \begin{cases} \sqrt[3]{t} & \text{si } t > \delta^3 \\ \frac{t}{3\delta^2} + \frac{4}{29} & \text{si } t \leq \delta^3 \end{cases} \quad (1\text{-}10)$$

con δ=6/29.

Al estar diseñado para ser independiente del dispositivo (a diferencia de RGB), CIELAB también asegura una representación consistente en diferentes sistemas de captura y visualización, lo que es crucial en contextos médicos donde la estandarización y la reproducibilidad de los resultados son claves. Además, facilita el procesamiento computacional para segmentación de tejidos y análisis de lesiones, siendo un espacio de color robusto para algoritmos de detección y clasificación en imágenes médicas, como en el caso del análisis de melanomas.[13]

2.1.4 Color Appearance Models

Los *Color Appearance Models* (CAM) son herramientas teóricas que predicen cómo los humanos percibimos los colores bajo diversas condiciones de iluminación y contextos visuales. Estos modelos fueron desarrollados para estandarizar la representación del color en varias aplicaciones, como la industria, la impresión y la imagen digital, permitiendo una representación precisa de los colores tal como los percibimos en el mundo real, considerando factores como luminancia, contexto y entorno de visualización [14, 15]

2.1.4.1 El espacio de color CAM16

El modelo **CAM16** es una versión avanzada del modelo CIECAM para la gestión del color, que proporciona una representación más precisa de cómo los humanos perciben los colores bajo diversas condiciones de iluminación y contexto visual. **CIECAM16** fue desarrollado para mejorar la predicción de la percepción del color y proporciona una evaluación detallada de la apariencia del color, incorporando correlatos perceptuales clave como luminancia, cromatismo y matiz.

Los correlatos perceptuales del **CAM16** son:

- Luminancia (Y)**:** Determina la cantidad de luz que refleja o emite una superficie, influyendo en la percepción del brillo del color.
- Chroma (C)**:** Representa la pureza o intensidad del color, indicando colores más saturados o apagados.
- Hue (h)**:** Este componente corresponde al matiz o tono del color, como el rojo, verde o azul.

Además de estos, el CAM16 incluye atributos adicionales que permiten una representación más rica del color, como:

- Brillo (B): Relacionado con la claridad del color, proporcionando una medida más precisa de su luminosidad en comparación con otros modelos.
- Saturación (S): La pureza del color en relación con el gris, donde los colores más saturados son percibidos como más intensos y vívidos.
- Riqueza (R): Se refiere a la cantidad de color presente en relación con la luminancia, proporcionando contexto sobre cómo se percibe el color bajo diferentes fuentes de luz

lo que facilita su aplicación en contextos donde el color necesita ser evaluado de manera precisa y detallada, como en el análisis de imágenes dermatológicas. [14]

2.1.4.2 El espacio de color CAM16UCS

El espacio CAM16UCS se deriva del modelo CAM16 y fue desarrollado para proporcionar una representación más precisa y perceptualmente uniforme del color, particularmente en condiciones cambiantes de iluminación. UCS significa *Uniform Color Space*, lo que implica que las diferencias entre colores en este espacio se perciben de manera uniforme, lo que facilita la comparación de colores en diversas aplicaciones prácticas.

Los tres principales correlatos perceptuales del CAM16UCS son la Luminancia (Y), el Chroma (C) y el matiz **(**Hue, h).

El modelo CAM16UCS está diseñado para mejorar la coherencia en la representación del color, lo que resulta particularmente útil en sistemas de gestión del color donde la precisión es fundamental. [15]

2.2 Cuantificación del color con K-means

La segmentación es un paso crucial en el procesamiento de imágenes que consiste en dividir una imagen en diferentes regiones o grupos homogéneos, facilitando su análisis posterior. Esta tarea puede abordarse mediante distintos métodos, cada uno centrado en características específicas de la imagen, dependiendo del atributo más relevante para el problema en cuestión. Un enfoque popular para la segmentación es el *clustering* o agrupamiento, una técnica de aprendizaje no supervisado que clasifica los datos en grupos basándose en su similitud, sin necesidad de información previa sobre las etiquetas o características específicas de los datos. [16]

Para llevar a cabo el *clustering* de nuestras imágenes utilizaremos el algoritmo *K-means*.

El algoritmo K-*means* [17] es un método de agrupamiento (*clustering*) ampliamente utilizado en la minería de datos. Su principal función es clasificar patrones en un conjunto de datos en un número predefinido de clústeres. La calidad de los resultados obtenidos por este algoritmo se mide generalmente mediante una función objetivo, como la suma de los cuadrados de las distancias Euclidianas entre cada patrón y su centroide:

$$J = \sum_{j=1}^{K} \sum_{i=1}^{N_j} ||x_i^{(j)} - \mu_j||^2 \qquad (1-11)$$

Donde:

- K es el número de clústeres.
- μj es el centroide del cluster j.
- N_j es el número de patrones pertenecientes al clúster j.
- $x_i^{(j)}$ es el i-ésimo patrón asignado al clúster j.

La minimización de esta función, llamada función de costo, garantiza que los puntos dentro de cada *cluster* estén lo más cerca posible de su centroide, lo que conduce a una agrupación eficiente. La función de costo o dispersión puede revelar diferencias en la uniformidad de color, que pueden correlacionarse con variaciones estructurales y de profundidad en las lesiones cutáneas (González et al., 2018; Raman et al., 2017). Por ejemplo, áreas con alta dispersión podrían corresponder a lesiones con heterogeneidad en el color, un factor importante en el diagnóstico de melanomas.

El procedimiento básico del algoritmo K-means es [17]:

1. Inicialización: Se seleccionan aleatoriamente K puntos como los centroides iniciales de los clusters. En nuestro caso estamos utilizando 8 centroides, cada uno correspondiente a un color.
2. Asignación de *Clusters*: Cada punto *xi* se asigna al *cluster* más cercano, basado en una medida de distancia, generalmente la distancia euclidiana. Esto se expresa matemáticamente como:

   $$ci = arg\min_j ||xi - \mu j||^2 \quad (1\text{-}12)$$

   donde *ci* es el índice del *cluster* al que se asigna *xi* y *μj* es el centroide del *cluster j*.
3. Actualización de Centroides: Una vez que todos los puntos han sido asignados a un *cluster*, se recalculan los centroides de cada grupo tomando la media de todos los puntos asignados:

   $$\mu j = \frac{1}{|Nj|} \sum_{xi \epsilon Cj} xi \quad (1\text{-}13)$$

   donde *Nj* es el conjunto de puntos asignados al *cluster j* y |*Nj*| es el número de puntos en ese *cluster*.

4. Criterio de convergencia: Si se cumple un criterio de convergencia, como un número máximo de iteraciones, la estabilización de los centroides, o una variación aceptable en el valor de la función objetivo, el algoritmo se detiene. En caso contrario, se regresa al paso 2.

2.3 Support Vector Machines

Las Máquinas de Vectores de Soporte (por sus siglas en inglés SVM) son algoritmos de aprendizaje supervisado desarrollados para la clasificación y la regresión, introducidas en 1995 por Vapnik y fundamentadas en la teoría del aprendizaje estadístico. Este modelo se basa en encontrar un hiperplano que divida de manera óptima los datos en distintas clases de manera que maximice la distancia entre los puntos más cercanos de cada clase y dicho hiperplano y reducir así el error de clasificación en nuevas muestras. [18, 19]

Definiciones clave

El objetivo de un modelo SVM es identificar un límite de decisión (hiperplano) que separe los datos de dos clases de forma óptima, manteniendo el mayor "margen" posible entre estas clases.

Hiperplano: en un espacio *n*-dimensional, un hiperplano es un subespacio de dimensión *n*-1 que divide al espacio en dos mitades. Si es espacio es bidimensional, el hiperplano es una línea y si es tridimensional es un plano.

Margen**:** distancia entre el hiperplano y los puntos más cercanos de cada clase, conocidos como vectores de soporte.

Principios matemáticos de los SVM

Dado un conjunto de datos de entrenamiento (xi, yi), donde $xi \in R^n$ representa el vector de características de la i-ésima observación (imagen) e $yi \in \{-1, 1\}$ indica su clase, el problema de optimización en SVM puede formularse de la siguiente manera:

Para una separación óptima en SVM, el modelo busca un vector de pesos w y un sesgo b que maximicen el margen entre las clases. El hiperplano de decisión se define como:

$$w * x + b = 0 \quad (1\text{-}15)$$

Para un clasificador lineal SVM, el objetivo es satisfacer las siguientes restricciones para todos los puntos xi:

$$yi\ (w * x + b) \geq 1 \quad (1\text{-}16)$$

Este planteamiento asegura que cada punto esté correctamente clasificado y se encuentre a una distancia de al menos $1/||w||$ del hiperplano de separación.

Para maximizar el margen, se debe minimizar ||w||, lo que se traduce en el siguiente problema de optimización:

$$\min_{w,b} \frac{1}{2}\ ||w||^2 \quad (1\text{-}17)$$

sujeto a:

$$yi\ (w * xi + b) \geq 1,\ \forall i \quad (1\text{-}18)$$

Esta es una optimización cuadrática con restricciones lineales, que puede resolverse mediante métodos de optimización convexa, obteniendo un modelo con un margen óptimo entre las clases.

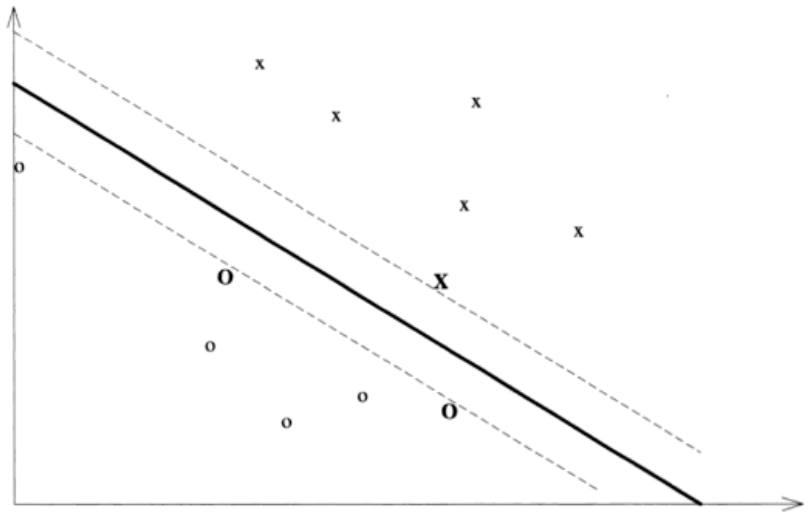

Figura 2-2. SVM: hiperplano de margen máximo con sus vectores de soporte [18]

Función de pérdida y Margen blando

En muchos casos, no es posible una separación lineal perfecta de los datos, especialmente cuando los datos tienen ruido o no son linealmente separables. Para abordar esto, se introduce el concepto de *margen blando* mediante variables de relajación ξi, que permite que algunos puntos se encuentren en el margen o incluso en el lado incorrecto del hiperplano de separación.

La función de pérdida con margen blando permite al modelo ser más flexible en situaciones en las que los datos no son perfectamente separables en su espacio original.

El problema de optimización con margen blando se formula como:

$$\min_{w,b} \frac{1}{2} ||w||^2 + C \sum_{i=1}^{n} \xi i \quad (1\text{-}19)$$

donde C (por defecto está fijado a 1) es un parámetro que controla el equilibrio entre maximizar el margen y minimizar los errores de clasificación. Está sujeto a:

$$yi\ (w * xi + b) \geq 1 - \xi i,\ \xi i \geq 0,\ \forall i \quad (1\text{-}20)$$

SVM no lineales y Función de núcleo (kernel)

Cuando los datos no son linealmente separables en el espacio original, SVM puede aplicar una transformación a un espacio de mayor dimensión, donde sí sean separables. Para evitar la carga computacional de transformar explícitamente los datos, se usa la función de núcleo o kernel, que calcula el producto interno en el espacio transformado sin necesidad de realizar la transformación directamente.

Las funciones de núcleo más comunes son: núcleo lineal, núcleo polinomial y núcleo radial (RBF). Esta se elige en función de la distribución de los datos y el tipo de separación requerida.

2.4 K-Fold Cross Validation

La validación cruzada K-*Fold* es un método estadístico en aprendizaje automático utilizado para evaluar el rendimiento de un modelo en datos nuevos y no vistos. [20]

En K-*Fold*, el conjunto de datos D se divide aleatoriamente en K subconjuntos de tamaño similar D1, D2,…,DK. En cada iteración, uno de estos subconjuntos se utiliza como conjunto de prueba y los restantes como conjunto de entrenamiento. Este proceso se repite K veces, de modo que cada subconjunto es usado exactamente una vez como conjunto de prueba y K−1 veces como parte del conjunto de entrenamiento.

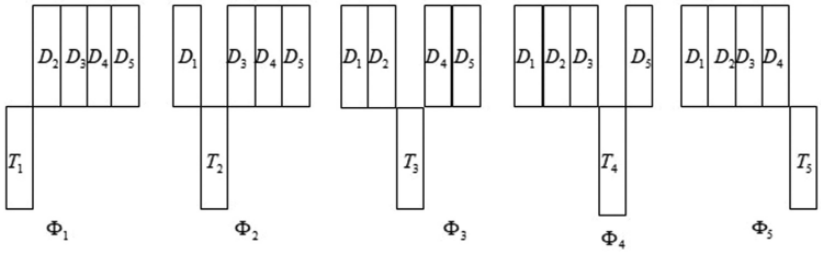

Figura 2-3. Esquema de una validación cruzada con 5 folds (K=5). [20]

Al final, se promedia el rendimiento de todas las iteraciones, proporcionando una estimación más estable de la calidad del modelo que una simple división de entrenamiento y prueba.

Elegir un valor K adecuado es clave. Valores comunes son K=5 o K=10, los cuales equilibran eficiencia computacional con precisión en la estimación de rendimiento. Con valores de K pequeños, el tiempo de cómputo es menor, pero puede reducirse la precisión de la estimación.

Ventajas de la Validación Cruzada K-Fold [21]

1. Reducción de la Varianza y Robustez: La validación cruzada K-*Fold* ayuda a obtener una estimación más estable y confiable del rendimiento general del modelo al reducir la varianza en las métricas de evaluación. Dado que el rendimiento del modelo se calcula como el promedio a lo largo de varias divisiones, la evaluación no se ve influenciada por una sola división del conjunto de datos. Esto ofrece una medida de rendimiento más robusta y confiable en comparación con una evaluación basada en una única división de datos, lo cual es especialmente útil en conjuntos de datos pequeños o limitados.
2. Mejora de la generalización del modelo: al evaluar el rendimiento en diferentes subconjuntos del conjunto de datos, K-*Fold* permite observar cómo el modelo responde a variaciones en los datos. Esto contribuye a una mejor capacidad de generalización, es decir, a la habilidad del modelo para realizar predicciones precisas sobre nuevos datos no vistos. Al probar en varios subconjuntos, se reduce el riesgo de sobreajuste a características específicas de un único conjunto de entrenamiento, lo cual fortalece su capacidad para adaptarse a datos reales y diversos.
3. Utilización Óptima de los Datos: Este método optimiza el uso del conjunto de datos completo al permitir que cada muestra sea utilizada tanto para el entrenamiento como para la prueba en al menos una de las iteraciones. En lugar de reservar una porción significativa del conjunto de datos solo para las pruebas, cada pliegue asegura que todos los datos

participen en el proceso, logrando así un aprendizaje más presentativo de los patrones subyacentes.

2.5 Selección de Características

La selección de características es una técnica utilizada para reducir la dimensionalidad de los datos y optimizar el rendimiento del modelo. Consiste en seleccionar un subconjunto de características relevantes para la tarea de clasificación, eliminando aquellas que son redundantes o no contribuyen significativamente al modelo. Con esto se consigue mejorar la eficiencia del modelo, reduciendo el tiempo de procesamiento y el riesgo de sobreajuste.

Existen diversos métodos de selección de características, pero en este estudio solo analizaremos dos: la Selección Secuencial Hacia Atrás (SBS) y la Selección Secuencial Hacia Adelante (SFS). [22, 23]

2.5.1 *Sequential Backward Selection*, SBS

La Selección secuencial hacia atrás (SBS) inicia con el conjunto completo de características y elimina una característica en cada paso, en función de cuál resulta menos relevante para el modelo en ese contexto. Permite evaluar la importancia de cada característica en relación con todas las demás, ya que parte de un conjunto completo y elimina progresivamente las menos significativas. Esto permite a SBS capturar interacciones complejas entre variables, siendo especialmente útil en casos donde algunas características son relevantes solo en combinación con otras.

Aunque SBS puede consumir más recursos computacionales debido a la necesidad de evaluar todas las características en cada paso de eliminación, este método puede obtener resultados más robustos en escenarios donde la relación entre las variables es compleja.

2.5.2 *Sequential Forward Selection*, SFS

La Selección secuencial hacia delante (SFS) comienza con un conjunto vacío y agrega una característica en cada paso. En cada iteración, se evalúan las características que aún no han sido seleccionadas y se elige aquella que mejora más el rendimiento del modelo al añadirse al conjunto actual de características. Este proceso continúa hasta alcanzar un número deseado de características o hasta que la adición de nuevas características no mejore significativamente el modelo.

Una de las principales ventajas de SFS es su eficiencia computacional. Dado que inicia desde cero, no necesita evaluar la interacción de cada característica con todas las demás en el conjunto completo, lo cual ahorra recursos. Sin embargo, esta eficiencia tiene un coste: SFS tiende a ignorar las interacciones complejas entre características porque evalúa cada característica individualmente antes de añadirla al modelo. Esto puede hacer que SFS no capture combinaciones de características que, en conjunto, podrían ofrecer un rendimiento mejor que las características añadidas en solitario.

2.6 Permutation Importance Value

El p-value (*Permutation Importance Value*) es una técnica utilizada para evaluar la importancia relativa de cada característica en un modelo de clasificación. Es fundamental en el contexto de análisis de datos, donde se necesita comprender qué características o variables influyen más en el rendimiento de un modelo predictivo. [24, 25]

El p-value se basa en la técnica de *Permutation Importance*, la cual mide la relevancia de una característica mediante la alteración aleatoria de sus valores y la observación del impacto en el rendimiento del modelo.

Para calcularlo, seguimos los siguientes pasos:

1. Entrenamiento del modelo en el conjunto de datos completos.
2. Permutación de los valores de una característica específica en el conjunto de datos, manteniendo las demás constantes.
3. Evaluación del cambio en el rendimiento del modelo después de la permutación. Si el rendimiento empeora significativamente, indica que la característica permutada es importante, mientras que si empeora mínimamente sugiere que la característica tiene apenas relevancia para el modelo.

El p-value es un método fácil de implementar, de interpretar y aplicable en una variedad de modelos de aprendizaje automático, como las máquinas de soporte vectorial (SVM). Sim embargo, presenta algunas limitaciones. La permutación aleatoria puede interrumpir dependencias entre variables, lo que es especialmente importante a la hora de capturar la importancia de una característica.

En nuestro caso, cada característica tiene una interpretación específica en términos de percepción de color, lo que permite que el modelo clasifique de forma más precisa y justificada. A través de p-valuealue, es posible determinar qué atributos colorimétricos son más informativos para distinguir entre diferentes niveles de profundidad de las lesiones cutáneas.

3 IMPLEMENTACIÓN

La implementación de este estudio se desarrolló utilizando como lenguaje Python en un entorno Jupyter Notebook de Anaconda, lo cual facilitó la escritura, ejecución y depuración del código en un entorno interactivo y visualmente accesible.

En esta sección se detalla la implementación del código desarrollado para llevar a cabo la clasificación de imágenes de melanomas en superficiales o profundos mediante técnicas de *Machine Learning*.

3.1 Preparación del entorno

El primer paso en la implementación es la configuración del entorno de trabajo. Para ello, se instalan las librerías necesarias que permiten realizar operaciones de procesamiento de imágenes, clasificación, y análisis de resultados.

Primero es necesario ejecutar el siguiente bloque de código para instalar las dependencias:

```
!pip install numpy==1.23.5
!pip install numpy pillow opencv-python colorspacious colour-science==0.4.4
```

A continuación, se importan las librerías necesarias para el desarrollo del código.

```
import os
import cv2
import numpy as np
import PIL.Image as Image
import matplotlib.pyplot as plt
from sklearn import svm
from sklearn.model_selection import KFold, cross_val_predict
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import make_scorer, precision_score, recall_score, confusion_matrix, ConfusionMatrixDisplay
from sklearn.feature_selection import SequentialFeatureSelector
from sklearn.inspection import permutation_importance
import colour as clr
from colour import convert
```

os: Esta librería proporciona funciones para interactuar con el sistema operativo, como la manipulación de rutas y archivos. Es esencial para la carga y gestión de los datos de imágenes y modelos.

cv2: OpenCV es una de las librerías más utilizadas para el procesamiento y análisis de imágenes. En este proyecto, se usa para realizar operaciones en las imágenes, como la conversión de espacios de color, redimensionamiento y manipulación de píxeles.

numpy: NumPy es una librería fundamental en Python para trabajar con matrices y operaciones matemáticas. Se emplea para manejar arrays multidimensionales y facilitar el procesamiento numérico de las características extraídas de las imágenes.

PIL (Pillow): Pillow es una librería para la manipulación de imágenes. Se utiliza para abrir, modificar y guardar imágenes en diversos formatos, y es útil para convertir entre representaciones de color en imágenes.

matplotlib.pyplot: Matplotlib es una librería de visualización que se utiliza para crear gráficos. La empleamos para visualizar imágenes, métricas de evaluación, y otros resultados gráficos que permiten analizar el rendimiento de los modelos de clasificación.

sklearn: Scikit-learn es una librería de machine learning que proporciona herramientas eficientes para la creación y evaluación de modelos predictivos. Se utiliza para:

- **SVM (Support Vector Machine)**: para la clasificación de imágenes.
- **KFold y cross_val_predict**: Herramientas para la validación cruzada, permitiendo una evaluación robusta del rendimiento del modelo.
- **StandardScaler**: Para la normalización de las características.
- **SequentialFeatureSelector**: Utilizada para la selección de características.
- **permutation_importance**: para evaluar la importancia de las características del modelo.
- **precision_score, recall_score, confusion_matrix**: Métricas de evaluación utilizadas para medir la exactitud, el recall y la matriz de confusión, que permiten evaluar el rendimiento del clasificador.

colour: Esta librería proporciona herramientas para trabajar con espacios de color.

convert (de colour): La función convert de la librería colour se utiliza para convertir imágenes entre diferentes espacios de color, permitiendo aplicar modelos de clasificación que dependen de la percepción del color.

3.2 Carga y procesamiento de los datos

3.2.1 Dataset

El conjunto de datos utilizado en este proyecto se ha construido a partir de la combinación de dos bases de datos reconocidas en el ámbito de la dermatología: el Interactive Atlas of Dermoscopy y la International Skin Imaging Collaboration (ISIC). Al fusionar estas fuentes, se conformó un *dataset* unificado que permite analizar y clasificar las muestras de manera consistente.

Siguiendo la clasificación propuesta por G. Argenziano et al. [11], basada en el índice de Breslow, que agrupa el grosor del melanoma en dos categorías, se organizó el conjunto de datos unificado de la siguiente manera:

Tabla 3-1. Dataset proyecto.

Etapas melanoma	Muestras
I	836
II	336

Como vemos nuestro *dataset* se encuentra desbalanceado, lo cual, como hemos comprobado, afecta negativamente a los resultados del clasificador. Decidimos balancearlo usando un número similar de muestras para ambas etapas, lo cual mejoró los resultados.

Al final utilizamos 343 muestras para la etapa I y 336 muestras para la etapa II

```
# Obtener listas de imágenes y sus etiquetas
imagenes_etapa1 = [(img, '<0.76mm') for img in os.listdir(path_etapa1) if es_imagen(img)]
imagenes_etapa2 = [(img, '>0.76mm') for img in os.listdir(path_etapa2) if es_imagen(img)]

# Unir las listas
imagenes = imagenes_etapa1 + imagenes_etapa2

# No filtrar duplicados, conservar todas las imágenes
imagenes_filtradas = []
etiquetas_filtradas = []

for img, etiqueta in imagenes:
    imagenes_filtradas.append(img)
    etiquetas_filtradas.append(etiqueta)
```

El código clasifica imágenes de dos directorios (path_etapa1: correspondiente a imágenes de la etapa I y path_etapa2: correspondiente a imágenes de la etapa II) según la profundidad de melanoma: <0.76mm para las imágenes de la primera carpeta y >0.76mm para las de la segunda. Primero, genera dos listas que contienen tuplas con el nombre de cada imagen y su etiqueta correspondiente. Luego, combina estas listas en una sola. Finalmente, separa los datos en dos listas independientes: una para los nombres de las imágenes y otra para sus etiquetas

3.2.2 Variables

```
X_lab = []
X_cam16ucs = []
X_cam16 = []
Y = []
```

X_lab, X_cam16ucs, X_cam16: Estas matrices se preparan para almacenar las características extraídas de cada imagen en los espacios de color Lab, CAM16-UCS y CAM16.

Y: Lista de etiquetas correspondientes a cada imagen, para que cada conjunto de características tenga su categoría asociada (<0.76mm o >0.76mm).

`ncolores = 8` , Este parámetro define el número de colores (centroides) que se consideran al aplicar el algoritmo K-means en cada imagen (en este caso, 8 colores).

3.3 Funciones

En este apartado se detallan las funciones implementadas.

```
def es_imagen(filename):
    return any(filename.endswith(extension) for extension in ['.png', '.jpg', '.jpeg', '.bmp'])
```

es_imagen(filename): Esta función verifica si un archivo es una imagen, comprobando su extensión.

```
def kmeans_cluster_centroides(imagen, ncolores):
    Z = np.float32(imagen)
    criterios = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 100, 1.0)
    _, label, centroides = cv2.kmeans(Z, ncolores, None, criterios, 10, cv2.KMEANS_RANDOM_CENTERS)
    res = centroides[label.flatten()]
    res2 = res.reshape((imagen.shape))
    return centroides, label, res2
```

kmeans_cluster_centroides (imagen, ncolores): Esta función aplica el algoritmo K-means sobre los píxeles de una imagen para obtener los centroides (colores dominantes) y la imagen cuantificada.

Parámetros de entrada:

- Imagen: imagen convertida a RGB
- Ncolores: la cantidad de centroides que queremos, que en este caso son 8.

La función nos devuelve:

- Centroides
- Label: las etiquetas de los pixeles (a que centroide pertenece cada pixel)
- Res2: la imagen cuantificada.

```
def calcular_dispersion(imagen, labels, centroides):
    dispersion = []
    for i, centroide in enumerate(centroides):
        mask = (labels == i).reshape(-1)
        dispersions = np.linalg.norm(imagen[mask] - centroide, axis=1)
        dispersion.append(np.mean(dispersions))
    return dispersion
```

calcular_dispersion (imagen, labels, centroides): Calcula la dispersión de cada color en la imagen con respecto a sus centroides.

Parámetros de entrada:

- Imagen: imagen convertida a RGB
- Label: las etiquetas de los pixeles
- Centroides

La función nos devuelve:

- Dispersión: una lista con la dispersión promedio para cada color dominante (centroide).

```
def convertir_a_cam16ucs(imagen_np):
    imagen_np_float = imagen_np.astype(np.float64) / 255.0
    imgCAM16_UCS = convert(imagen_np_float, 'sRGB', 'CAM16UCS')
    return imgCAM16_UCS
```

convertir_a_cam16ucs (imagen_np): dada una imagen en espacio de color RGB, la convierte al espacio de color CAM16UCS.

```python
def convertir_a_cam16(imagen_np):
    imagen_np_float = imagen_np.astype(np.float64) / 255.0
    imgXYZ = clr.sRGB_to_XYZ(imagen_np_float)
    L_A = 15
    Y_b = 20.0
    XYZ_w = np.array([95.05, 100.00, 108.88])    # Blanco de referencia D65
    surround = clr.VIEWING_CONDITIONS_CIECAM16["Average"]
    corrCAM16 = clr.XYZ_to_CIECAM16(100 * imgXYZ, XYZ_w, L_A, Y_b, surround)

    # Extraer componentes CAM16
    J = corrCAM16.J
    C = corrCAM16.C
    h = corrCAM16.h
    s = corrCAM16.s
    Q = corrCAM16.Q
    M = corrCAM16.M

    # Combinar en un solo array
    cam16_features = np.stack([J, C, h, s, Q, M], axis=-1)

    return cam16_features
```

convertir_a_cam16(imagen_np): dada una imagen en espacio de color RGB, la convierte al espacio de color CAM16. Usamos XYZ como paso intermedio porque es un espacio estándar y perceptual que representa fielmente los colores para el ojo humano. Esta conversión permite que CAM16 integre propiedades de percepción como claridad, cromaticidad y tono, ajustadas por las condiciones de visualización.

```python
def extraer_caracteristicas(imagen_np, ncolores):
    # Convertir a espacio de color Lab
    imagen_lab = cv2.cvtColor(imagen_np, cv2.COLOR_RGB2Lab)
    lab_reshaped = imagen_lab.reshape((-1, 3))

    # Aplicar k-means en espacio Lab
    centroides_lab, labels_lab, _ = kmeans_cluster_centroides(lab_reshaped, ncolores)

    # Calcular dispersión en Lab
    dispersion_lab = calcular_dispersion(lab_reshaped, labels_lab, centroides_lab)

    # Concatenar centroides y dispersión
    caracteristicas_lab = np.concatenate((centroides_lab.flatten(), dispersion_lab))

    # Convertir a espacio de color CAM16-UCS
    imgCAM16_UCS = convertir_a_cam16ucs(imagen_np)
    cam16ucs_reshaped = imgCAM16_UCS.reshape((-1, 3))

    # Aplicar k-means en espacio CAM16-UCS
    centroides_cam16ucs, labels_cam16ucs, _ = kmeans_cluster_centroides(cam16ucs_reshaped, ncolores)

    # Calcular dispersión en CAM16-UCS
    dispersion_cam16ucs = calcular_dispersion(cam16ucs_reshaped, labels_cam16ucs, centroides_cam16ucs)

    # Concatenar centroides y dispersión
    caracteristicas_cam16ucs = np.concatenate((centroides_cam16ucs.flatten(), dispersion_cam16ucs))

    # Convertir a espacio CAM16
    imagen_cam16 = convertir_a_cam16(imagen_np)
    cam16_reshaped = imagen_cam16.reshape((-1, 6))  # CAM16 tiene 6 componentes

    # Aplicar K-means en el espacio CAM16
    centroides, labels, _ = kmeans_cluster_centroides(cam16_reshaped, ncolores)

    # Calcular las dispersiones
    dispersiones = calcular_dispersion(cam16_reshaped, labels, centroides)

    # Concatenar características: 6 valores por 8 centroides + 8 dispersiones
    caracteristicas_cam16 = np.concatenate([centroides.flatten(), dispersiones])

    return caracteristicas_lab, caracteristicas_cam16, caracteristicas_cam16ucs
```

extraer_caracteristicas (imagen_np, ncolores): Esta función, recibiendo como entrada la imagen original y el numero de centroides (colores) que queremos, en este caso 8, genera las características necesarias para la clasificación de imágenes, transformando la imagen a los espacios de color CIELAB, CAM16 y CAM16-UCS, y aplicando K-means en cada espacio para obtener colores dominantes y dispersiones.

Como resultado obtendremos los vectores de características de la imagen en cada espacio de color.

3.3.1 Vectores de características

Para cada imagen obtenemos tres vectores de características, uno en el espacio CIELAB, otro en CAM16UCS y en el espacio CAM16.

Vamos a ver en profundidad como serán dichos vectores y las características que estamos generando en función del espacio de color usado.

Vector características_Lab: El espacio CIELAB, como hemos visto en el apartado 2.1, tenemos 3 componentes (L, a, b). Para cada uno de los 8 centroides obtenidos tenemos esas tres componentes y su dispersión. La estructura del vector de características tiene el siguiente orden:

- Primeras 24 posiciones (0 - 23): Cada una de las 3 componentes del espacio repetidas para los 8 centroides.
    i. 0-2: Las 3 características (L, a, b) para el primer centroide.
    ii. 3-5: Las 3 características para el segundo centroide.

    ... (Esto continúa para los 8 centroides).

    Esto da un total de 3×8= 24 características para las componentes LAB.
- Últimas 8 posiciones (24-31): Las dispersiones correspondientes a cada uno de los 8 centroides.

Total: 32 características para el espacio CIELAB.

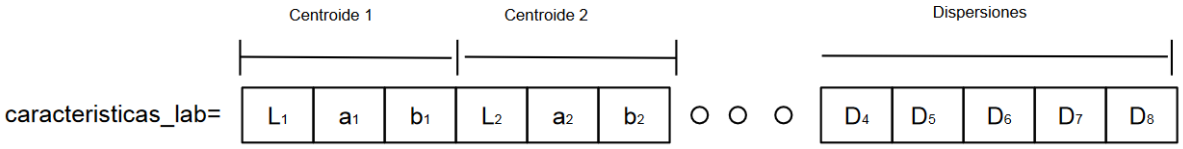

Figura 3-1. Vector de características LAB

Vector características_CAM16UCS: El espacio CAM16UCS, como hemos visto en el apartado 2.1, tenemos 3 componentes (Y, C, h). Para cada uno de los 8 centroides obtenidos tenemos esas tres componentes y su dispersión. La estructura del vector de características tiene el siguiente orden:

- Primeras 24 posiciones (0 - 23): Cada una de las 3 componentes del espacio repetidas para los 8 centroides.
    i. 0-2: Las 3 características (Y, C, h) para el primer centroide.
    ii. 3-5: Las 3 características para el segundo centroide.

    ... (Esto continúa para los 8 centroides).

    Esto da un total de 3×8= 24 características para las componentes CAM16UCS.
- Últimas 8 posiciones (24-31): Las dispersiones correspondientes a cada uno de los 8 centroides.

Total: 32 características para el espacio CAM16UCS.

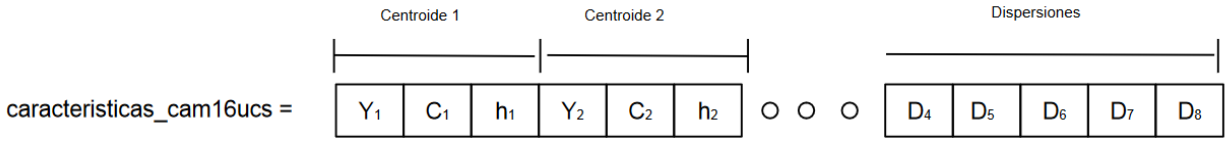

Figura 3-2. Vector de características CAM16UCS

Vector características_CAM16: El espacio CAM16, como hemos visto en el apartado 2.1, tenemos 6 componentes (Y, C, h, s, Q, M). Para cada uno de los 8 centroides obtenidos tenemos esas seis componentes y su dispersión. La estructura del vector de características tiene el siguiente orden:

- Primeras 48 posiciones (0 - 47): Cada una de las 6 componentes del espacio repetidas para los 8 centroides.
    i. 0-5: Las 6 características (Y, C, h, s, Q, M) para el primer centroide.
    ii. 6-11: Las 6 características para el segundo centroide.

    ... (Esto continúa para los 8 centroides).

    Esto da un total de 6×8= 48 características para las componentes CAM16.
- Últimas 8 posiciones (47-55): Las dispersiones correspondientes a cada uno de los 8 centroides.

Total: 56 características para el espacio CAM16.

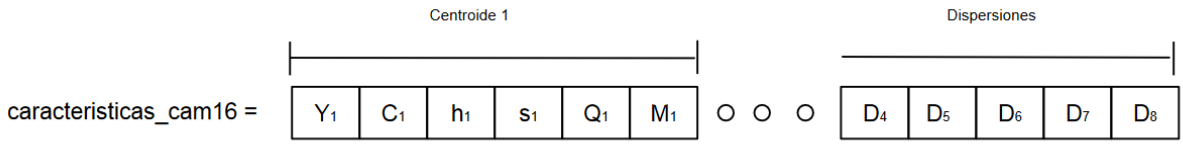

Figura 3-3. Vector de características CAM16

3.4 Procesamiento de imágenes

Esta sección detalla el proceso de creación de las matrices de características para cada espacio de color, las cuales usaremos posteriormente como datos de entrada para el clasificador.

```
for img, etiqueta in zip(imagenes_filtradas, etiquetas_filtradas):
    try:
        if etiqueta == '<0.76mm':
            imagen = Image.open(os.path.join(path_etapa1, img)).convert('RGB')
        else:
            imagen = Image.open(os.path.join(path_etapa2, img)).convert('RGB')

        imagen_np = np.asarray(imagen)

        # Extraer características
        caracteristicas_lab, caracteristicas_cam16, caracteristicas_cam16ucs = extraer_caracteristicas(imagen_np,
                                                                                                       ncolores)
        X_lab.append(caracteristicas_lab)
        X_cam16.append(caracteristicas_cam16)
        X_cam16ucs.append(caracteristicas_cam16ucs)
        Y.append(0 if etiqueta == '<0.76mm' else 1)
    except Exception as e:
        print(f"Error procesando la imagen {img}: {str(e)}")

# Convertir listas a matrices numpy
X_lab = np.array(X_lab)
X_cam16 = np.array(X_cam16)
X_cam16ucs = np.array(X_cam16ucs)
Y = np.array(Y)
```

1. Se recorre la lista de imágenes y sus etiquetas (imagenes_filtradas, etiquetas_filtradas), cargando cada imagen de acuerdo con su ruta y convirtiéndola al formato RGB. La ruta de cada imagen depende de su etiqueta: <0.76mm o >0.76mm, correspondiente a etapa1 o etapa2.

2. Para cada imagen cargada, la función extraer_caracteristicas () extrae sus representaciones en tres espacios de color: Lab, CAM16, y CAM16-UCS. Esta función devuelve tres conjuntos de características que se añaden a las listas X_lab, X_cam16, y X_cam16ucs, respectivamente, mientras que Y se actualiza con la etiqueta correspondiente (0 para <0.76mm, 1 para >0.76mm).

   Estas listas contienen los datos extraídos de las imágenes, pero en su forma original, no están optimizadas ni estructuradas para operaciones matemáticas rápidas o procesamiento avanzado.

3. Convertimos las listas de características y etiquetas (X_lab, X_cam16, X_cam16ucs y Y) en matrices de tipo numpy (arrays 2D). Esto se hace con el fin de facilitar su manipulación y

procesamiento, ya que las matrices numpy son más eficientes y compatibles con las funciones de aprendizaje automático.

```
scaler_lab = StandardScaler().fit(X_lab)
X_lab = scaler_lab.transform(X_lab)

scaler_cam16 = StandardScaler().fit(X_cam16)
X_cam16 = scaler_cam16.transform(X_cam16)

scaler_cam16ucs = StandardScaler().fit(X_cam16ucs)
X_cam16ucs = scaler_cam16ucs.transform(X_cam16ucs)
```

4. Finalmente, estandarizamos las matrices obtenidas. La estandarización es un proceso que transforma las características de los datos para que tengan una media de 0 y una desviación estándar de 1. La clase StandardScaler de la librería scikit-learn es la que se utiliza para estandarizar los datos. Razones por las que estandarizamos:
    a) Evitas que ciertas características dominen: Si una característica tiene valores muy grandes y otra tiene valores pequeños, la primera podría dominar las decisiones del modelo.
    b) Mejor rendimiento del modelo: Algunos algoritmos de machine learning (como SVM) funcionan mejor cuando las características están estandarizadas, ya que estas técnicas están basadas en distancias o gradientes.

En la Figura 3-4, vemos un ejemplo de cómo quedaría la matriz X_lab.

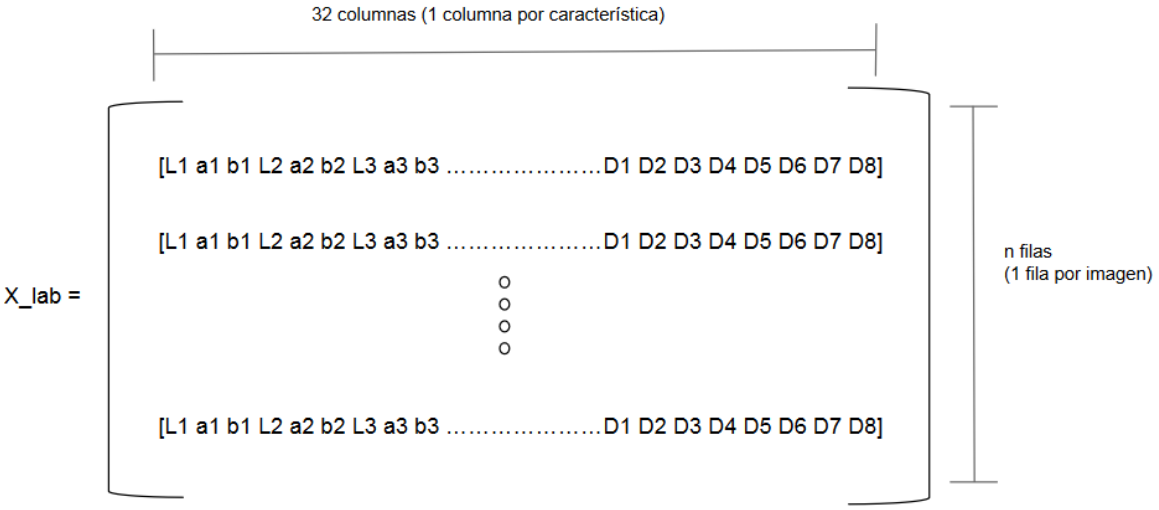

Figura 3-4. Matriz de características en el espacio de color LAB

3.5 Entrenamiento, validación y evaluación del modelo SVM

En este apartado, se presenta el proceso de entrenamiento, validación y evaluación del modelo de clasificación basado en Support Vector Machine (SVM) para la clasificación de imágenes.

El código proporcionado a continuación se repite exactamente igual para los distintos espacios de color (Lab, CAM16UCS y CAM16).

```
kf = KFold(n_splits=5, shuffle=True, random_state=42)

# Definir la métrica de evaluación usando average='macro' y zero_division=1
scoring = {
    'precision': make_scorer(precision_score, average='macro', zero_division=1),
    'recall': make_scorer(recall_score, average='macro')
}

# Clasificación SVM en espacio Lab con K-Fold
clf_lab = svm.SVC()
Y_pred_kfold = cross_val_predict(clf_lab, X_lab, Y, cv=kf)

# Crear la matriz de confusión
cm_lab = confusion_matrix(Y, Y_pred_kfold)
disp_lab = ConfusionMatrixDisplay(confusion_matrix=cm_lab, display_labels=np.unique(Y))
disp_lab.plot(cmap='Blues')
plt.title('Matriz de Confusión del Clasificador SVM LAB (Validación Cruzada)')
plt.show()

# Métricas de evaluación de la validación cruzada
precision_kfold = precision_score(Y, Y_pred_kfold, average='macro', zero_division=1)
recall_kfold = recall_score(Y, Y_pred_kfold, average='macro')

print("Resultados del clasificador SVM en espacio Lab con K-Fold:")
print("Recall promedio (K-Fold):", recall_kfold)
print("Precisión promedio (K-Fold):", precision_kfold)

# Importancia de las características (PI Value) en espacio Lab
clf_lab.fit(X_lab, Y)  # Ajustar el modelo con todos los datos
pi_lab = permutation_importance(clf_lab, X_lab, Y, n_repeats=10, random_state=42)

print("\nImportancia de las características (PI Value) en espacio LAB:")
for i in pi_lab.importances_mean.argsort()[::-1]:
    print(f"Característica {i}: {pi_lab.importances_mean[i]:.3f} ± {pi_lab.importances_std[i]:.3f}")
```

1. Para evaluar el rendimiento del modelo, se utiliza la validación cruzada K-*Fold* con 5 particiones, lo que indica que se harán 5 iteraciones siendo en cada una de ellas el conjunto de datos de prueba una partición diferente. La opción shuffle=True se emplea para mezclar aleatoriamente los datos antes de dividirlos en los diferentes pliegues, asegurando que cada partición sea representativa de todo el conjunto de datos.

2. Las métricas utilizadas para evaluar el rendimiento del modelo son la precisión (precision) y el recall (sensibilidad). Ambas métricas se calculan utilizando el promedio macro, lo que significa que se toman en cuenta los resultados de cada clase por igual, sin importar su frecuencia en los datos. La opción zero_division=1 asegura que no haya errores al calcular la precisión o recall en el caso de que no se detecten ciertos valores en las predicciones.

3. Creamos un clasificador SVM y realizamos la validación cruzada con la matriz de características X_lab generada anteriormente y el vector de etiquetas asociadas a las imágenes. Obtenemos el vector de predicciones Y_pred_kfold, resultado de las predicciones generadas por la validación cruzada. Cada valor en Y_pred_kfold representa la predicción del modelo (el valor predicho por el clasificador SVM) para la clase de la imagen correspondiente.

4. Una vez obtenidas las predicciones de validación cruzada, se genera la matriz de confusión para evaluar cómo el modelo ha clasificado correctamente las imágenes. Esto se explica en apartado 4.

5. Cálculo de las métricas de precisión y recall, que se realiza para obtener una visión cuantitativa del rendimiento del modelo. Esto se explica en apartado 4.

6. El análisis de la importancia de las características se realiza utilizando la técnica de importancia por permutación. Esta técnica evalúa el impacto de cada característica en el rendimiento del modelo al permutar sus valores y medir la caída en el rendimiento. Las características con mayor caída en el rendimiento indican que son más relevantes para el modelo. Este análisis se realiza:
    i. Usamos permutation_importance, función de scikit-learn, la cual permuta 10 veces cada característica. Obtenemos pi_lab, un objeto que contiene los resultados de la importancia de cada característica.
    ii. Se presenta la importancia de las características de forma ordenada, destacando las más influyentes en el modelo, en un formato en que se observa el promedio y la desviación estándar de cada característica.

3.5.1 Selección de características

En este estudio, se emplean métodos de selección de características con el objetivo de identificar y conservar las características más relevantes para la clasificación en los espacios de color LAB, CAM16-UCS y CAM16. Estos métodos, Sequential Backward Selection (SBS) y Sequential Forward Selection (SFS), permiten reducir la dimensionalidad del conjunto de datos, lo que disminuye la complejidad del modelo y mejora el rendimiento de clasificación del SVM.

Para el espacio Lab, aplicamos SBS donde se parte del conjunto completo y se eliminan, iterativamente, aquellas características que menos aportan al rendimiento del modelo. Este proceso continúa hasta obtener el conjunto óptimo de características que maximiza la precisión del clasificador.

El argumento n_features_to_select=None permite que el selector determine automáticamente el número óptimo de características necesarias. La transformación resultante (X_lab_sbs) es un conjunto reducido que contiene solo aquellas características esenciales para el espacio LAB.

```
# Selección de características utilizando SBS (Backward) con K-Fold
sbs = SequentialFeatureSelector(clf_lab, n_features_to_select=None, direction='backward')
X_lab_sbs = sbs.fit_transform(X_lab, Y)

# Clasificación SVM en espacio Lab con SBS y K-Fold
Y_pred_lab_sbs_kfold = cross_val_predict(clf_lab, X_lab_sbs, Y, cv=kf)
```

Después de obtener el vector de predicciones mediante validación cruzada, hacemos lo mismo que hicimos en el apartado anterior para el espacio sin selección de características.

Para el espacio CAMM16UCS y CAM16, aplicamos SFS el cual comienza con un conjunto vacío de características y, en cada paso, añade aquella característica que maximiza el rendimiento del modelo. Este proceso se repite hasta obtener un conjunto de características que optimiza la capacidad del clasificador SVM.

La transformación resultante (X_cam16ucs_sfs y X_cam16_sfs) es un conjunto reducido que contiene solo aquellas características esenciales para el espacio CAM16UCS o CAM16.

```
# Selección de características utilizando SFS (Forward) con K-Fold en espacio CAM16-UCS
sfs_cam16ucs = SequentialFeatureSelector(clf_cam16ucs, n_features_to_select=None, direction='forward')
X_cam16ucs_sfs = sfs_cam16ucs.fit_transform(X_cam16ucs, Y)

# Clasificación SVM en espacio CAM16-UCS con SFS y K-Fold
Y_pred_cam16ucs_sfs_kfold = cross_val_predict(clf_cam16ucs, X_cam16ucs_sfs, Y, cv=kf)
```

```
# Selección de características utilizando SFS (Forward) con K-Fold en espacio CAM16
sfs_cam16 = SequentialFeatureSelector(clf_cam16, n_features_to_select=None, direction='forward')
X_cam16_sfs = sfs_cam16.fit_transform(X_cam16, Y)

# Clasificación SVM en espacio CAM16 con SFS y K-Fold
Y_pred_cam16_sfs_kfold = cross_val_predict(clf_cam16, X_cam16_sfs, Y, cv=kf)
```

Después de obtener los vectores de predicciones mediante validación cruzada, hacemos lo mismo que hicimos en el apartado anterior para el espacio sin selección de características.

4 RESULTADOS

La implementación de este estudio se desarrolló utilizando como lenguaje Python en un entorno Jupyter Notebook de Anaconda, lo cual facilitó la escritura, ejecución y depuración del código en un entorno interactivo y visualmente accesible.

4.1 Métricas de clasificación

Al evaluar el rendimiento de un modelo de clasificación, es fundamental comprender cómo el modelo clasifica correcta e incorrectamente las muestras, ya que estas métricas reflejan la eficacia y precisión del sistema para identificar correctamente las imágenes en cada categoría de profundidad de la lesión (superficial o profunda).

4.1.1 Verdadero Positivo (TP)

Definimos Verdadero Positivo (TP, del inglés *True Positive*) cuando el modelo clasifica correctamente las imágenes que pertenecen a la etapa II de profundidad de la lesión, es decir, aquellas lesiones con una profundidad mayor a 0,76 mm. En este contexto, un caso se considera como verdadero positivo cuando una imagen que efectivamente representa una lesión profunda es clasificada correctamente por el modelo como profunda. Esta métrica es particularmente importante para asegurar que los casos de mayor riesgo (lesiones profundas) sean identificados adecuadamente por el modelo, permitiendo un diagnóstico más certero en situaciones donde la intervención temprana puede ser crítica.

4.1.2 Verdadero Negativo (TN)

Definimos Verdadero Negativo (TN, del inglés *True Negative*) cuando el modelo clasifica correctamente las imágenes de la etapa I de profundidad de la lesión, es decir, aquellas lesiones que presentan una profundidad menor o igual a 0,76 mm. En este caso, un verdadero negativo ocurre cuando una lesión superficial (etapa I) es clasificada como superficial. La precisión en los verdaderos negativos ayuda a reducir falsos diagnósticos de gravedad, lo que es esencial en contextos médicos, ya que evita tratamientos innecesarios y permite a los pacientes recibir la atención adecuada.

4.1.3 Falso Positivo (FP)

Definimos Falso Positivo (FP, del inglés *False Positive*) cuando el modelo clasifica incorrectamente una imagen de la etapa I (superficial) como si fuera de etapa II (profunda). En otras palabras, un falso positivo ocurre cuando una lesión que debería haberse clasificado como superficial es clasificada erróneamente por el modelo como profunda. Este tipo de error puede tener implicaciones clínicas importantes, ya que podría llevar a decisiones de tratamiento innecesarias o más invasivas para una lesión que no lo requiere. Minimizar los falsos positivos es crucial para aumentar la precisión del modelo y reducir la carga de procedimientos adicionales.

4.1.4 Falso Negativo (FN)

Definimos Falso Negativo (FN, del inglés *False Negative*) cuando el modelo clasifica incorrectamente una imagen de la etapa II (profunda) como si fuera de etapa I (superficial). En este caso, un falso negativo ocurre cuando una lesión profunda es clasificada erróneamente por el modelo como superficial. Los falsos negativos son particularmente preocupantes en el diagnóstico de melanoma, ya que implican un riesgo significativo de no detectar una lesión de mayor gravedad. Un alto índice de falsos negativos puede comprometer la confiabilidad del modelo en situaciones críticas, por lo que la reducción de estos errores es esencial para mejorar la efectividad y seguridad del sistema en el contexto clínico.

4.1.5 Precisión

La Precisión (*Precision* en inglés) es una métrica de clasificación que mide la exactitud del modelo al identificar correctamente las predicciones positivas. En términos más concretos, la precisión indica la proporción de casos correctamente clasificados como positivos (TP) en relación con todos los casos que el modelo ha clasificado como positivos (es decir, la suma de los verdaderos positivos y los falsos positivos, FP).

$$Precisión = \frac{TP}{TP+FP} \quad (1\text{-}21)$$

En el contexto de la clasificación de lesiones cutáneas, una alta precisión implica que, de todas las lesiones que el modelo clasificó como profundas, la mayoría efectivamente corresponde a lesiones profundas. Esto es fundamental en el diagnóstico médico, ya que una alta precisión minimiza el número de falsos positivos, evitando intervenciones innecesarias o diagnósticos incorrectos para casos que en realidad no presentan una alta profundidad.

4.1.6 Sensibilidad

La Sensibilidad (*Recall* en inglés) es una métrica de clasificación que mide la capacidad del modelo para identificar correctamente todos los casos positivos presentes en el conjunto de datos. Es decir, la sensibilidad indica la proporción de verdaderos positivos (TP) en relación con todos los casos que realmente son positivos (la suma de verdaderos positivos y falsos negativos, FN).

$$Sensibilidad = \frac{TP}{TP+FN} \quad (1\text{-}22)$$

En la clasificación de lesiones cutáneas, una alta sensibilidad significa que el modelo tiene una capacidad elevada para detectar las lesiones profundas sin omitir casos importantes. Esto es particularmente importante en el diagnóstico del melanoma, donde un falso negativo (es decir, un caso profundo que se clasifica incorrectamente como superficial) podría implicar la falta de un tratamiento adecuado y a tiempo. Por tanto, la sensibilidad es crucial para evaluar la seguridad y confiabilidad del modelo en aplicaciones clínicas.

4.1.7 Matriz de confusión

La matriz de confusión es una herramienta fundamental en el análisis de la calidad de un modelo de clasificación. Se trata de una tabla de contingencia que permite visualizar el desempeño de un modelo al comparar las predicciones realizadas por el clasificador con las etiquetas reales de los datos. Cada celda de la matriz de confusión refleja el número de instancias que fueron clasificadas de una determinada manera, permitiendo identificar tanto las predicciones correctas como los errores cometidos.

La matriz de confusión para un problema binario se representa de la siguiente manera:

	Predicción Negativa	Predicción Positiva
Negativo Real	TN	FP
Positivo Real	FN	TP

Tabla 4-1. Matriz de confusión

4.1.8 Valor promedio

El valor promedio, también conocido como media aritmética, es una medida de tendencia central que proporciona un valor representativo de un conjunto de datos. Se calcula sumando todos los valores de las observaciones y dividiendo el resultado entre el número total de observaciones. En el contexto de la importancia de las características, el valor promedio de la importancia de cada característica refleja el impacto medio que esa característica tiene en el rendimiento del modelo a lo largo de múltiples iteraciones de permutación.

$$Media = \frac{1}{n}\sum_{i=1}^{n} xi \quad (1\text{-}23)$$

donde:

- xi son los valores individuales de las características
- n es el número total de observaciones.

En el caso de la importancia de las características (PI Value), el valor promedio indica cuán relevante es, en promedio, una característica para el modelo. Un valor promedio más alto significa que esa característica tiene una mayor influencia en la capacidad del modelo para hacer predicciones precisas. El valor promedio es útil para comparar las características entre sí y determinar cuáles son las más influyentes.

4.1.9 Desviación estándar

La desviación estándar es una medida estadística que indica cuánto se dispersan los valores de un conjunto de datos respecto al valor promedio. En otras palabras, la desviación estándar mide la variabilidad o dispersión de los datos. En el contexto de la importancia de las características, la desviación estándar proporciona información sobre la consistencia de la importancia de cada característica a lo largo de múltiples repeticiones de permutación.

$$Desviación\ estándar = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(xi - \mu)^2} \quad (1\text{-}24)$$

donde:

- xi son los valores individuales de las características
- n es el número total de observaciones.
- μ es el valor promedio

Una desviación estándar pequeña indica que los valores están muy concentrados alrededor de la media, lo que sugiere que la importancia de la característica es consistente en todas las permutaciones. Por el contrario, una desviación estándar grande indica que la importancia de la característica varía significativamente entre las repeticiones, lo que podría señalar que su influencia en el modelo no es estable o confiable. La desviación estándar, por lo tanto, es crucial para evaluar la estabilidad de las características seleccionadas y garantizar que las características más importantes sean consistentes a lo largo de las iteraciones.

4.2 Resultados obtenidos

4.2.1 Espacio de color CIELAB

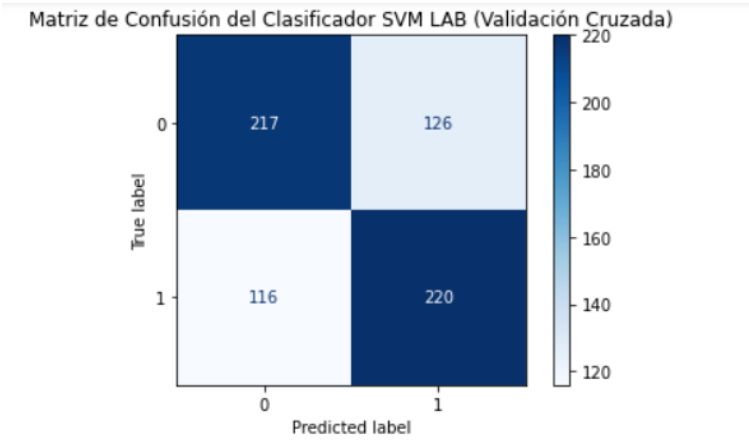

Figura 4-1. Matriz de confusión del Clasificador SVM LAB

Tabla 4-2. Resultados del clasificador SVM en espacio Lab con K-*Fold*

Métrica	Resultado
Precision	0.643707482993
Recall	0.64374490097

Tabla 4-3. Importancia de las características (p-value) en espacio LAB

N.º característica	Significado característica	Resultados (Valor promedio + Desviación)
21	L8	0.038 ± 0.008
3	L2	0.025 ± 0.008
12	L5	0.024 ± 0.006
28	D5	0.023 ± 0.004
27	D4	0.023 ± 0.008
24	D1	0.021 ± 0.006
6	L3	0.020 ± 0.005
0	L1	0.019 ± 0.007
31	D8	0.018 ± 0.005
15	L6	0.017 ± 0.005
23	b8	0.015 ± 0.003
2	b1	0.014 ± 0.006
5	b2	0.013 ± 0.004
1	a1	0.013 ± 0.004
19	a7	0.013 ± 0.007

22	a8	0.012 ± 0.004
25	D2	0.011 ± 0.006
18	L7	0.011 ± 0.004
8	b3	0.010 ± 0.007
14	b5	0.009 ± 0.007
13	a5	0.009 ± 0.006
29	D6	0.009 ± 0.003
4	a2	0.009 ± 0.004
30	D7	0.008 ± 0.004
7	a3	0.008 ± 0.003
9	L4	0.007 ± 0.003
10	a4	0.007 ± 0.003
26	D3	0.006 ± 0.003
17	b6	0.006 ± 0.006
16	a6	0.006 ± 0.005
20	b7	0.004 ± 0.007
11	b4	-0.006 ± 0.007

4.2.1.1 CIELAB con selección de características SBS

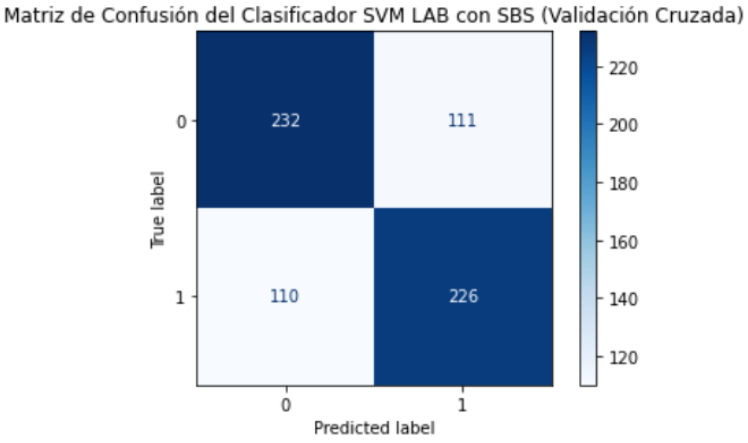

Figura 4-2. Matriz de confusión del Clasificador SVM LAB con SBS

Tabla 4-4. Resultados del clasificador SVM en espacio Lab con SBS y K-Fold

Métrica	Resultado
Precision	0.674501943635
Recall	0.67449285925

Tabla 4-5. Importancia de las características (p-valuealue) en espacio LAB con SBS

N.º característica	Significado característica	Resultados (Valor promedio + Desviación)
7	a3	0.056 ± 0.011
12	L5	0.056 ± 0.008
14	b5	0.049 ± 0.008
3	L2	0.043 ± 0.004
11	b4	0.042 ± 0.007
2	b1	0.041 ± 0.010
6	L3	0.037 ± 0.007
9	L4	0.034 ± 0.008
15	L6	0.033 ± 0.010
5	b2	0.028 ± 0.006
1	a1	0.027 ± 0.005
13	a5	0.025 ± 0.007
10	a4	0.024 ± 0.005
8	b3	0.024 ± 0.004
0	L1	0.022 ± 0.005
4	a2	0.016 ± 0.007

4.2.2 Espacio de color CAM16UCS

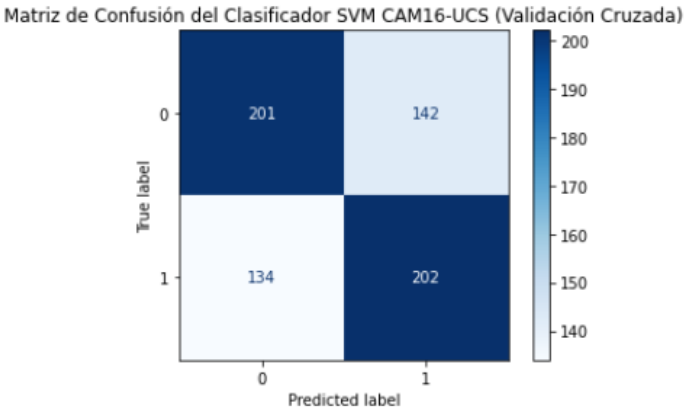

Figura 4-3. Matriz de confusión del Clasificador SVM CAM16UCS

Tabla 4-6. Resultados del clasificador SVM en espacio CAM16UCS con K-Fold

Métrica	Resultado
Precision	0.593598153547
Recall	0.593604651163

Tabla 4-7. Importancia de las características (p-value) en espacio CAM16UCS

N.º característica	Significado característica	Resultados (Valor promedio + Desviación)
28	D5	0.027 ± 0.004
21	Y8	0.022 ± 0.006
18	Y7	0.019 ± 0.003
12	Y5	0.018 ± 0.005
11	h4	0.018 ± 0.006
4	C2	0.015 ± 0.004
24	D1	0.015 ± 0.005
19	C7	0.014 ± 0.004
20	h7	0.013 ± 0.005
30	D7	0.013 ± 0.006
6	Y3	0.013 ± 0.007
15	Y6	0.012 ± 0.002
5	h2	0.012 ± 0.005
9	Y4	0.011 ± 0.008
0	Y1	0.011 ± 0.004
31	D8	0.011 ± 0.003
13	C5	0.010 ± 0.003
17	h6	0.010 ± 0.008
8	h3	0.010 ± 0.003
29	D6	0.010 ± 0.004
2	h1	0.009 ± 0.006
7	C3	0.009 ± 0.006
27	D4	0.009 ± 0.005
22	C8	0.008 ± 0.003
1	C1	0.008 ± 0.003
10	C4	0.008 ± 0.004
3	Y2	0.007 ± 0.004
16	C6	0.006 ± 0.005
23	h8	0.006 ± 0.006
25	D2	0.005 ± 0.005
26	D3	0.005 ± 0.003
14	h5	0.005 ± 0.003

4.2.2.1 CAM16UCS con selección de características SFS

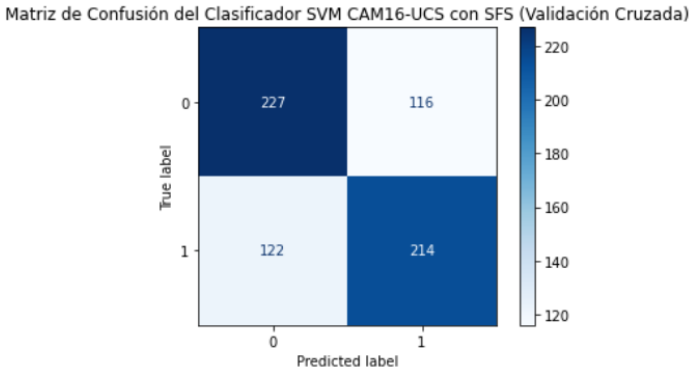

Figura 4-4. Matriz de confusión del Clasificador SVM CAM16UCS con SFS

Tabla 4-8. Resultados del clasificador SVM en espacio CAM16UCS con SFS y K-Fold

Métrica	Resultado
Precision	0.64935617104
Recall	0.649457323956

Tabla 4-9. Importancia de las características (PI Value) en espacio CAM16UCS con SFS

N.º característica	Significado característica	Resultados (Valor promedio + Desviación)
14	h5	0.045 ± 0.010
9	Y4	0.037 ± 0.005
12	Y5	0.036 ± 0.004
5	h2	0.033 ± 0.009
10	C4	0.033 ± 0.004
7	C3	0.032 ± 0.007
6	Y3	0.032 ± 0.005
15	Y6	0.030 ± 0.007
0	Y1	0.029 ± 0.007
13	C5	0.029 ± 0.004
8	h3	0.028 ± 0.006
11	h4	0.028 ± 0.007
1	C1	0.023 ± 0.007
4	C2	0.021 ± 0.006
2	h1	0.019 ± 0.005
3	Y2	0.018 ± 0.006

4.2.3 Espacio de color CAM16

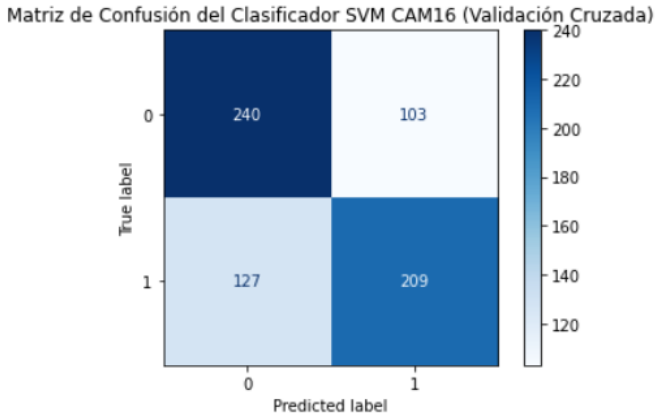

Figura 4-5. Matriz de confusión del Clasificador SVM CAM16

Tabla 4-10. Resultados del clasificador SVM en espacio CAM16 con K-Fold

Métrica	Resultado
Precision	0.660866132167
Recall	0.661911374275

Tabla 4-11. Importancia de las características (p-value) en espacio CAM16

N.º característica	Significado característica	Resultados (Valor promedio + Desviación)
2	h1	0.021 ± 0.006
54	D7	0.013 ± 0.003
38	h7	0.012 ± 0.004
0	Y1	0.011 ± 0.005
15	s3	0.010 ± 0.002
5	M1	0.009 ± 0.003
1	C1	0.009 ± 0.003
4	Q1	0.009 ± 0.004
8	h2	0.008 ± 0.001
32	h6	0.013 ± 0.006
3	s1	0.008 ± 0.003
45	s8	0.008 ± 0.004
18	Y4	0.007 ± 0.004
47	M8	0.007 ± 0.002
43	C8	0.007 ± 0.002
48	D1	0.007 ± 0.005
30	Y6	0.007 ± 0.006
39	s7	0.006± 0.004

46	Q8	0.006 ± 0.001
52	D5	0.006 ± 0.004
14	h3	0.006 ± 0.002
25	C5	0.006 ± 0.002
29	M5	0.006 ± 0.002
17	M3	0.006 ± 0.004
13	C3	0.006 ± 0.003
9	s2	0.006 ± 0.003
49	D2	0.006 ± 0.005
27	s5	0.005 ± 0.003
53	D6	0.005 ± 0.002
26	h5	0.005 ± 0.007
44	h8	0.005 ± 0.003
40	Q7	0.005 ± 0.004
50	D3	0.005 ± 0.004
22	Q4	0.005 ± 0.002
55	D8	0.005 ± 0.005
28	Q5	0.004 ± 0.003
36	Y7	0.004 ± 0.005
42	Y8	0.004 ± 0.003
34	Q6	0.004 ± 0.005
35	M6	0.004 ± 0.002
31	C6	0.004 ± 0.002
6	Y2	0.004 ± 0.004
10	Q2	0.004 ± 0.004
20	h4	0.003 ± 0.002
33	s6	0.003 ± 0.002
21	s4	0.003 ± 0.004
12	Y3	0.002 ± 0.003
16	Q3	0.002 ± 0.002
24	Y4	0.001 ± 0.002
11	M2	0.001 ± 0.003
7	C2	0.001 ± 0.003
37	C7	0.000 ± 0.003
41	M7	0.000 ± 0.003
23	M4	0.000 ± 0.003
19	C4	0.000 ± 0.003
51	D4	0.000 ± 0.003

4.2.3.1 CAM16 con selección de características SFS

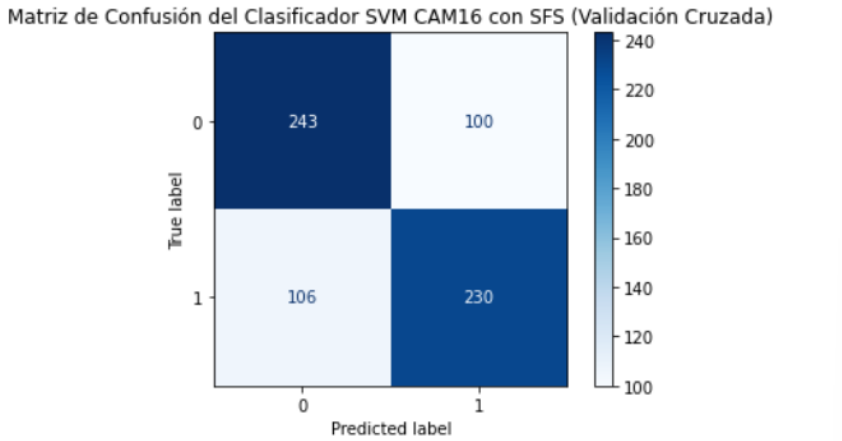

Figura 4-6. Matriz de confusión del Clasificador SVM CAM16 con SFS

Tabla 4-12. Resultados del clasificador SVM en espacio CAM16 con SFS y K-Fold

Métrica	Resultado
Precision	0.69648931001
Recall	0.696622384301

Tabla 4-13. Importancia de las características (p-value) en espacio CAM16 con SFS

N.º característica	Significado característica	Resultados (Valor promedio + Desviación)
13	C3	0.034 ± 0.007
27	s5	0.031± 0.009
2	h1	0.020 ± 0.007
16	Q3	0.020 ± 0.007
3	s1	0.017 ± 0.007
21	s4	0.015 ± 0.004
22	Q4	0.014 ± 0.003
25	C5	0.014 ± 0.003
24	Y5	0.014 ± 0.006
19	C4	0.014 ± 0.004
14	h3	0.013 ± 0.006
4	Q1	0.013 ± 0.006
0	Y1	0.013 ± 0.005
26	h5	0.011 ± 0.004
7	C2	0.010 ± 0.004
8	h2	0.010 ± 0.005

18	Y4	0.010 ± 0.002
20	h4	0.009 ± 0.002
15	s3	0.009 ± 0.004
17	M3	0.009 ± 0.004
23	M4	0.008 ± 0.006
6	Y2	0.008 ± 0.004
5	M1	0.007 ± 0.002
1	C1	0.007 ± 0.002
10	Q2	0.007 ± 0.005
11	M2	0.005 ± 0.006
9	Y2	0.003 ± 0.005
12	Y3	0.003 ± 0.005

5 CONCLUSIONES

La motivación de este trabajo radica en la importancia que tiene actualmente una detección rápida del melanoma para poder actuar de forma temprana y eficaz. La carga de trabajo de los médicos y la complejidad del diagnóstico pueden llevar más tiempo del deseado y, en algunos casos, la precisión del diagnóstico puede verse limitada. En este contexto, la Inteligencia Artificial ofrece un apoyo valioso para agilizar y mejorar la precisión de estos diagnósticos.

Este estudio ha explorado el uso de técnicas de *Machine Learning* y el análisis de distintos espacios de color, dado que el color es una característica crucial en el análisis de imágenes dermatológicas. Tras analizar los resultados obtenidos, se han alcanzado las siguientes conclusiones:

- La clasificación de melanomas superficiales muestra un mejor rendimiento en comparación con la de melanomas profundos, indicando que la profundidad puede influir en la precisión del modelo.
- El espacio de color CAM16 proporciona resultados superiores en comparación con los otros espacios de color analizados. Esto se debe a la inclusión de un mayor número de componentes que reflejan con mayor precisión la percepción del color en la visión humana.
- La aplicación de técnicas de selección de características mejora la precisión y sensibilidad de los resultados en los tres espacios de color.

Tras analizar todos los resultados obtenidos, concluimos que el espacio de color CAM16, combinado con la selección de características mediante SFS, es el que mejor se ajusta a los objetivos de este estudio. Este enfoque permite alcanzar una precisión y sensibilidad en el modelo de aproximadamente un 70%.

Además, al analizar la lista ordenada de características según su importancia, se observa que la croma, la saturación y el matiz son factores clave para diferenciar entre clases en la clasificación de melanomas.

Aun así, estos resultados no son del todo favorables, ya que el modelo tiene un elevado número de FP y FN, que podrían ser un problema a la hora de la detección correcta del melanoma. Como trabajo futuro se plantea el estudio en detalle de las lesiones que caigan cerca del umbral entre los dos tipos (0,76mm), planteando clasificadores que dividan en más de dos clases. Y, si la base de datos pasa a tener más imágenes, se pueden plantear técnicas de aprendizaje profundo (*deep learning*).

REFERENCIAS

[1] Manning, C. (2020). *Artificial Intelligence Definitions*. Stanford University. Disponible en: https://hai.stanford.edu/sites/default/files/2020-09/AI-Definitions-HAI.pdf [Accedido el 25 de noviembre de 2024].

[2] McCarthy, J., Minsky, M. L., Rochester, N., & Shannon, C. E. (2006). A proposal for the Dartmouth Summer Research Project on Artificial Intelligence. *AI Magazine*, 27(4), pp. 12-12.

[3] American Cancer Society (n.d.). ¿Qué es el cáncer de piel tipo melanoma? Disponible en: https://www.cancer.org/ [Acceso: 25 noviembre 2024].

[4] SEOM (2024). Se incrementa la incidencia interanual de melanoma en España con 7.881 casos nuevos. *Sociedad Española de Oncología Médica*. Disponible en: https://www.seom.org/ [Acceso: 25 noviembre 2024].

[5] MedlinePlus (n.d.). Melanoma. Disponible en: https://medlineplus.gov/spanish/ency/article/000850.htm [Acceso: 25 noviembre 2024].

[6] Skin Cancer Foundation (2023). Melanoma. Disponible en: http://www.skincancer.org/skin-cancer-information/melanoma/ [Acceso: 25 noviembre 2024].

[7] Zaballos, P., Carrera, C., Puig, S., & Malvehy, J. (n.d.). Dermoscopic Criteria in the Diagnosis of Melanoma. Disponible en: https://www.medigraphic.com/pdfs/cutanea/mc-2004/mc041b.pdf [Acceso: 25 noviembre 2024].

[8] Rashed, H., Flatman, K., Bamford, M., Teo, K.W. & Saldanha, G., 2017. Breslow density is a novel prognostic feature in cutaneous malignant melanoma. *Histopathology*, 70(2), pp.264–272. Available at: https://doi.org/10.1111/his.13060.

[9] Clark, W. H. Jr., From, L., Bernardino, E. A., & Mihm, M. C. (1969). The histogenesis and biologic behavior of primary human malignant melanomas of the skin. *Cancer Research, 29*(3), pp. 705–727.

[10] National Cancer Institute, 2024. Niveles de Clark. [en línea] Disponible en: https://www.cancer.gov/espanol/publicaciones/diccionarios/diccionario-cancer/def/niveles-de-clark [Accedido el 26 de noviembre de 2024].

[11] arXiv (2024). Robust Melanoma Thickness Prediction via Deep Transfer Learning enhanced by XAI Techniques. Disponible en: https://ar5iv.labs.arxiv.org/html/2406.13441v1 [Acceso: 25 noviembre 2024].

[12] Campos Acosta, J. (n.d.). La especificación del color: espacios de representación del color. *Instituto de Física Aplicada (CSIC)*. Disponible en: http://grupoorion.unex.es/divulgacion/especificacio [Acceso: 25 noviembre 2024].

[13] Fairchild, M.D. (2013). *Color Appearance Models*. John Wiley & Sons.

[14] Li, C.J., Luo, M.R., Brill, M.H., Melgosa, M., Pointer, M.R., Teunissen, C., Wei, M.T. (2022). *The CIE*

2016 Colour Appearance Model for Colour Management Systems: CIECAM16. CIE 248:2022. International Commission on Illumination (CIE).

[15] Li, C., Li, Z., Wang, Z., Xu, Y., Luo, M.R., Cui, G., Melgosa, M., Brill, M., and Pointer, M. (2017). Comprehensive color solutions: CAM16, CAT16, and CAM16UCS. *Color Research and Applications*, 42(6), pp. 703-718.

[16] Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data Clustering: A Review. *ACM Computing Surveys*, 31(3), 264-323.

[17] Xu, T.-S., Chiang, H.-D., Liu, G.-Y., & Tan, C.-W. (2017). Hierarchical K-means Method for Clustering Large-Scale Advanced Metering Infrastructure Data. *IEEE Transactions on Power Delivery, 32*(2), pp. 609–618.

[18] Cristianini, N., & Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press.

[19] Scholkopf, B., & Smola, A.J. (2002). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press.

[20] Department of Computing, Mathematical and Statistical Sciences, University of Namibia (n.d.). Determining the optimal number of folds to use in a K-fold cross-validation: A neural network classification experiment.

[21] Scaler Topics (n.d.). K Fold Cross Validation in Machine Learning. Disponible en: https://www.scaler.com/topics/ [Acceso: 25 noviembre 2024].

[22] Guyon, I., & Elisseeff, A. (2003). An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research*, 3, pp. 1157–1182.

[23] Kohavi, R., & John, G.H. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2), pp. 273–324.

[24] Altmann, A., Toloşi, L., Sander, O., & Lengauer, T. (2010). Permutation importance: a corrected feature importance measure. *Bioinformatics*, 26(10), pp. 1340–1347.

[25] Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1), pp. 5–32.

[26] Polesie, S., Gillstedt, M., Kittler, H., Rinner, C., Tschandl, P., & Paoli, J. (2022). Assessment of melanoma thickness based on dermoscopy images: An open, web-based, international, diagnostic study. *Journal of the European Academy of Dermatology and Venereology*, 36(11), pp. 2002–2007.

[27] Domínguez-Morales, J.P., Hernández-Rodríguez, J.C., Durán-López, L., Conejo-Mir, J., & Pereyra-Rodríguez, J.J. (2024). Melanoma Breslow thickness classification using ensemble-based knowledge distillation with semi-supervised convolutional neural networks. *IEEE Journal of Biomedical and Health Informatics*.

[28] Hernández-Rodríguez, J.C., Durán-López, L., Domínguez-Morales, J.P., Ortiz-Álvarez, J., Conejo-Mir, J., & Pereyra-Rodríguez, J.J. (2023). Prediction of melanoma Breslow thickness using deep transfer learning algorithms. *Clinical and Experimental Dermatology*, llad107.

[29] Sáez, A., Sánchez-Monedero, J., Gutiérrez, P. A., & Hervás-Martínez, C. (2016). Machine learning methods for binary and multiclass classification of melanoma thickness from dermoscopic images. *IEEE Transactions on Medical Imaging*, 35(4), pp. 1036–1045.

ÍNDICE DE CONCEPTOS

No se encuentran entradas de índice.

GLOSARIO

No se encontraron elementos de tabla de autoridades.