

Instituto Tecnológico de Estudios Superiores de Monterrey

Campus Monterrey



**Tecnológico
de Monterrey**

Procesamiento de imágenes médicas para el diagnóstico (Gpo 602)

Evidencia: Diagnóstico por Computadora

Nombres de estudiantes:

Matrícula:

Luis Ángel González Castillo A01571007

José Antonio Palacios Dorbecker A01177930

Daniel Agustín Aguilar López A01735969

Carlos Andrés Luis Martínez A00833514

31 de Mayo de 2024. Monterrey, Nuevo León

Índice

Introducción	3
Objetivos	3
Marco teórico	3
Materiales y Métodos	5
Materiales	5
Métodos	5
Preprocesamiento y segmentación de Máquina (MATLAB)	5
Entrenamiento de Máquina (R Studio)	7
Resultados y Discusión	8
MATLAB	8
Resultados de segmentación	8
Reporte de características Extraídas	11
Rstudio	12
Conclusiones	17
Luis Angel González Castillo (A01571007)	17
José Antonio Palacios Dorbecker (A01177930)	17
Daniel Agustín Aguilar López (A01735969)	18
Carlos Andrés Luis Martínez (A00833514)	18
Referencias	20
Anexos	21
Códigos de MATLAB	21
Código 1 “LoadAndProcessQuantitateImage”	21
Código 2 “ProcessMelanomaImages”	26
Código de R Segmentación	27
GitHub Link	35

Diagnóstico por Computadora

Introducción

El melanoma es una forma agresiva de cáncer de piel que puede ser fatal si no se detecta a tiempo. La identificación precisa y temprana es crucial. Este reporte evalúa y compara el desempeño de dos métodos de aprendizaje automático para clasificar melanoma y otras lesiones cutáneas.

El aprendizaje por computadora (Machine Learning) es importante para el diagnóstico médico porque puede analizar grandes volúmenes de datos y detectar patrones sutiles que a menudo pasan desapercibidos para el ojo humano, resultando en diagnósticos más precisos. Además, reduce el sesgo humano al proporcionar evaluaciones objetivas basadas en datos, lo que lleva a una mayor consistencia y confiabilidad en los diagnósticos.

Utilizamos una base de datos de imágenes de lesiones cutáneas para realizar la segmentación y extracción de características de señal, forma, textura y dimensiones fractales. MATLAB se empleará para la segmentación de imágenes y la extracción de características, RStudio se utilizará para entrenar y evaluar los modelos de aprendizaje automático, utilizando su potencial para análisis estadístico y machine learning. Los modelos que entrenaremos incluyen K-Nearest Neighbors (KNN), Naive Bayes (NB), Support Vector Machine (SVM) y Least Absolute Shrinkage and Selection Operator (LASSO). Se empleará la validación cruzada para evaluar el desempeño de cada modelo. Los resultados se analizarán en términos de sensibilidad, especificidad, F1 score y curvas ROC.

Objetivos

Evaluar y reportar el desempeño de la correcta clasificación de Melanoma de dos métodos de Aprendizaje por Computadora.

Marco teórico

La segmentación de imágenes es un proceso fundamental en el campo del procesamiento de imágenes digitales, con el objetivo de dividir una imagen en sus componentes constituyentes o regiones de interés. Esta técnica se utiliza en una amplia gama de aplicaciones, como el análisis médico en la identificación de tumores, análisis de tejidos, segmentación de órganos, etc.

(González et al, 2016)

Machine Learning es un campo en evolución de algoritmos computacionales diseñados para emular la inteligencia humana mediante el aprendizaje del entorno

circundante. Se aplican con éxito en diversos campos como reconocimiento de patrones, visión por computadora, finanzas, entre otros. Los algoritmos se utilizan cada vez más para optimizar y automatizar procesos complejos, lo que mejora tanto la seguridad como la eficacia de la práctica y, en última instancia, conduce a mejores resultados para los pacientes.

(El Naqa I et al, 2015)

El melanoma es un tipo de cáncer de piel que se desarrolla a partir de células productoras de pigmento llamadas melanocitos. Los nevus, comúnmente conocidos como lunares, son crecimientos benignos en la piel que contienen un agrupamiento de células de pigmento. Por otro lado, la queratosis seborreica es una lesión cutánea benigna, comúnmente conocida como verruga senil, que se caracteriza por un crecimiento excesivo de células de la epidermis. Cada uno de estos tiene características distintivas en términos de apariencia y comportamiento clínico, lo que los hace importantes de distinguir en la práctica dermatológica.

(Tschandl et al, 2018)

La cross-validation es una técnica que divide los datos en múltiples subconjuntos para entrenar y evaluar un modelo varias veces, promediando los resultados para obtener una estimación más confiable del rendimiento del modelo.

El K-Nearest Neighbors (KNN) clasifica una muestra basándose en los “k puntos” de datos más cercanos en el espacio de características, asignando la clase más común entre ellos. El Naive Bayes (NB) es un clasificador probabilístico que asume la independencia entre características y aplica el teorema de Bayes para asignar probabilidades a las clases. La Support Vector Machine (SVM) busca el hiperplano óptimo que maximiza el margen entre clases, separando los datos de manera efectiva en espacios de alta dimensión. El Least Absolute Shrinkage and Selection Operator (LASSO) es un método de regresión que realiza selección de características y regularización, penalizando la suma de los valores absolutos de los coeficientes para mejorar la precisión y reducir el sobre ajuste.

(Nti et al, 2021) (Tougui et al, 2021)

La matriz de confusión y las métricas son herramientas utilizadas para evaluar el rendimiento de un modelo de clasificación. La sensibilidad es la probabilidad de clasificar correctamente individuos que son positivos según la prueba, mientras que la especificidad es la probabilidad de clasificar correctamente individuos que son negativos. El F1 Score es una medida que combina precisión y sensibilidad, útil para encontrar un equilibrio entre ambas métricas. Las curvas ROC son gráficos que muestran la relación entre la sensibilidad y la tasa de falsos positivos a diferentes umbrales de decisión. El área bajo la curva (AUC) es una medida agregada del rendimiento del modelo, donde un AUC de 1.0 indica un modelo perfecto y un AUC de 0.5 indica un rendimiento no mejor que el azar.

(López et al, 1998)

Las características de señal como media, desviación estándar, sesgo y kurtosis proporcionan información sobre la distribución de intensidades de píxeles en una imagen. Las características de forma como volumen y superficie describen la geometría de los objetos en una imagen. Las características de textura, como las características GLCM y Haralick, capturan la información sobre la variación de intensidades en una región de interés. Las dimensiones fractales describen la complejidad estructural de una imagen en diferentes escalas.

(Sachdeva et al, 2013)

Materiales y Métodos

Materiales

- *MATLAB* 2023b (Desarrollador: *MathWorks*) (versión: R2023Bb, Update 1, 64-bit)
- *RStudio* (Desarrollador: *Posit Software, PBC*) (Versión 2023.12.1+402)
- Bases de datos de Melanoma, Lesiones Nevus, Queratosis Seborreica. (Provistas por nuestro profesor de la materia, José Gerardo Tamez-Peña)

Métodos

Preprocesamiento y segmentación de Máquina (MATLAB)

La metodología seguida para realizar el preprocesamiento y segmentación en MATLAB en el estudio de melanoma y otras lesiones cutáneas se puede describir de la siguiente manera (Revisar Anexo Códigos de MATLAB):

Segmentación de Imágenes:

1. Se define una ruta específica para las imágenes de melanoma, nevus y queratosis seborreica.
2. Se inicializan las variables `allquantitativeLesionData` y `allquantitativeControlData` para almacenar los datos de las lesiones y los controles.
3. Se obtiene una lista de archivos .jpg en la carpeta especificada.
4. Se recorre cada archivo de imagen en la lista y se carga y procesa la imagen utilizando la función `loadAndProcessQuantitateImage`.
5. Se generan archivos CSV separados para las características de las lesiones y los controles de melanoma, nevus y queratosis seborreica.

Extracción de Características:

1. Se utilizan técnicas avanzadas de segmentación de imágenes para identificar regiones de interés.
2. Se extraen características de señal, forma, textura y dimensiones fractales de las imágenes segmentadas.
3. Las características de señal incluyen media, desviación estándar, sesgo y kurtosis.
4. Las características de forma incluyen volumen y superficie de los objetos en la imagen.
5. Las características de textura se obtienen a través de GLCM y Haralick.
6. Las dimensiones fractales describen la complejidad estructural de la imagen en diferentes escalas.

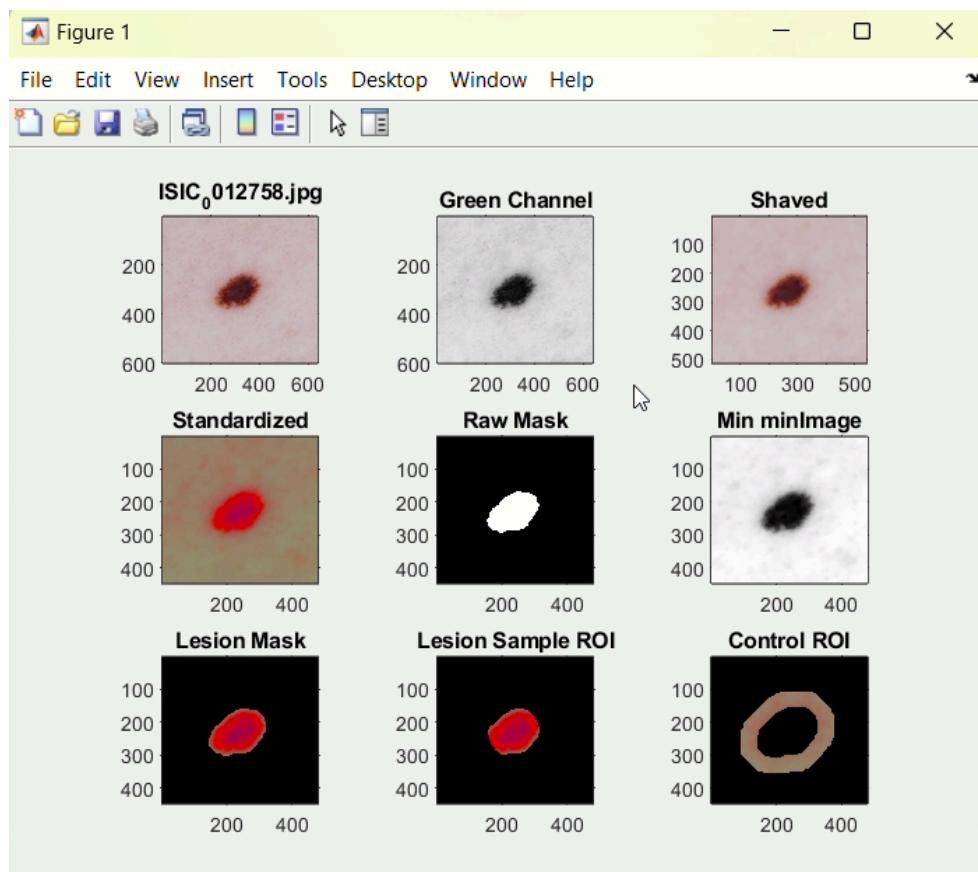


Figura 1. Ejemplo de una segmentación considerada como “buena”.

A continuación, se detallan los conceptos observados en la *Figura 1*.

- ❖ Fuente de imagen “.jpg”

- Esta es la imagen original del melanoma que se va a segmentar. En este caso, se llama ISIC_012758.jpg.
- ❖ Green Channel
 - Esta imagen muestra solo el canal verde de la imagen original. En este caso se trabajó con canales individuales (rojo, verde, azul) para resaltar ciertas características.
- ❖ Shaved
 - Realiza una versión de imagen del cual recorta características de la imagen y rellena con lo que tiene alrededor, en este caso retirar los vellos de los pies para que así no se llegue a confundir un bulto con el melanoma
- ❖ Standardized
 - Esta imagen muestra la versión estandarizada de la imagen original. La estandarización puede implicar normalización de la iluminación, ajuste de contraste u otras técnicas para asegurar que las condiciones de la imagen sean consistentes.
- ❖ Raw Mask
 - Esta es la máscara inicial o cruda del melanoma. Una máscara binaria donde el área blanca representa la región del melanoma detectada y el área negra representa el fondo.
- ❖ Min minImage
 - Esta imagen parece ser el resultado de una operación de minimización o filtro sobre la imagen original, lo que puede ayudar a resaltar características importantes del melanoma.
- ❖ Lesion Mask
 - Esta es la máscara final del melanoma, que ha sido refinada para representar con precisión los bordes de la lesión.
- ❖ Lesion Sample ROI
 - Esta imagen muestra la región de interés (ROI) del melanoma. Aquí, el melanoma está resaltado para su análisis posterior.
- ❖ Control ROI
 - Esta imagen muestra una región de control para comparar con la lesión. La región de control suele ser una parte de la piel sana para realizar comparaciones de características. Comúnmente es el contorno del melanoma.

Entrenamiento de Máquina (R Studio)

La metodología seguida para el entrenamiento de máquina en R Studio en el estudio de melanoma y otras lesiones cutáneas se puede describir de la siguiente manera (Revisar Anexo Código de R Segmentación):

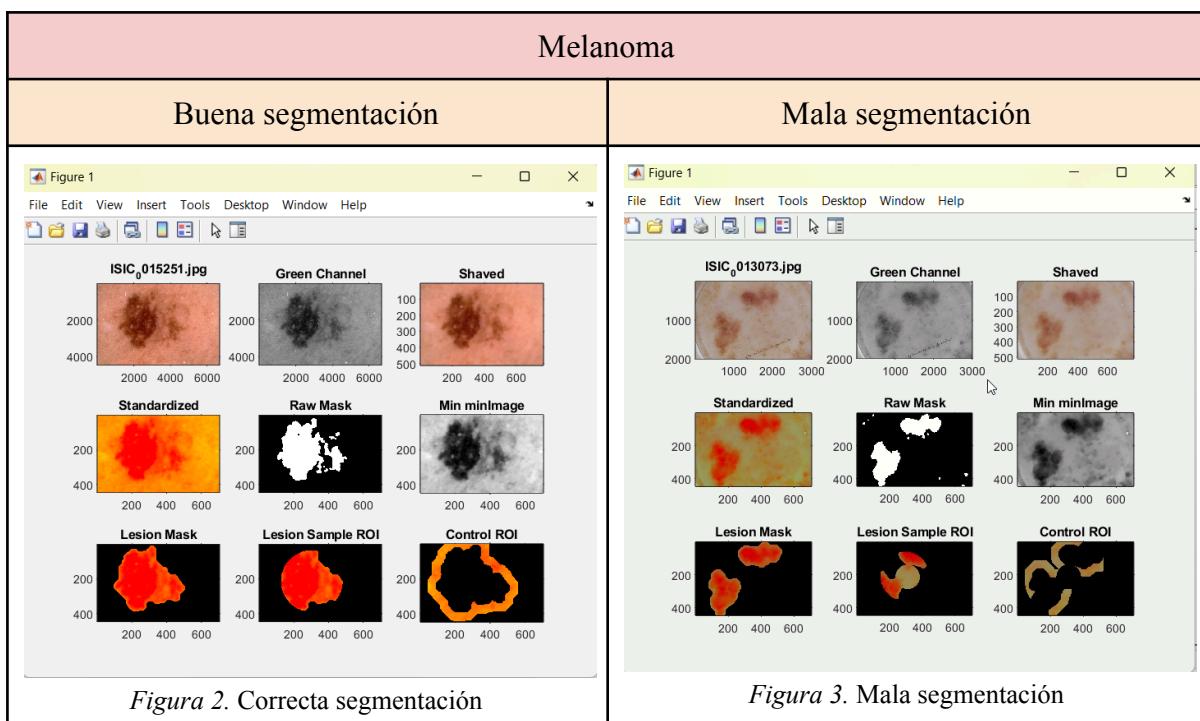
- ❖ Se carga el conjunto de datos de características extraídas de las imágenes de lesiones de piel para el melanoma. Se utilizan librerías como "FRESA.CAD", "e1071" y "readxl" para el análisis y procesamiento de datos.
- ❖ En sí, el acondicionamiento de los datos comienza con un colorado de nombre de columnas, se calculan los promedios y desviaciones estándar.
- ❖ Se divide el conjunto de datos en un set de entrenamiento y un set de test para validar los modelos.
- ❖ Se aplican algoritmos de aprendizaje automático basados en características radiónicas para la clasificación de melanoma y otros tipos de lesiones cutáneas, incluyendo:
 - K-Nearest Neighbors (KNN)
 - Naive Bayes (NB)
 - Support Vector Machine (SVM)
 - Least Absolute Shrinkage and Selection Operator (LASSO)
- ❖ Se utiliza la técnica de cross-validación para evaluar el desempeño de cada modelo y garantizar su generalización.
- ❖ Se ajustan los hiper-parámetros de los modelos para optimizar su rendimiento en la clasificación de las lesiones cutáneas.

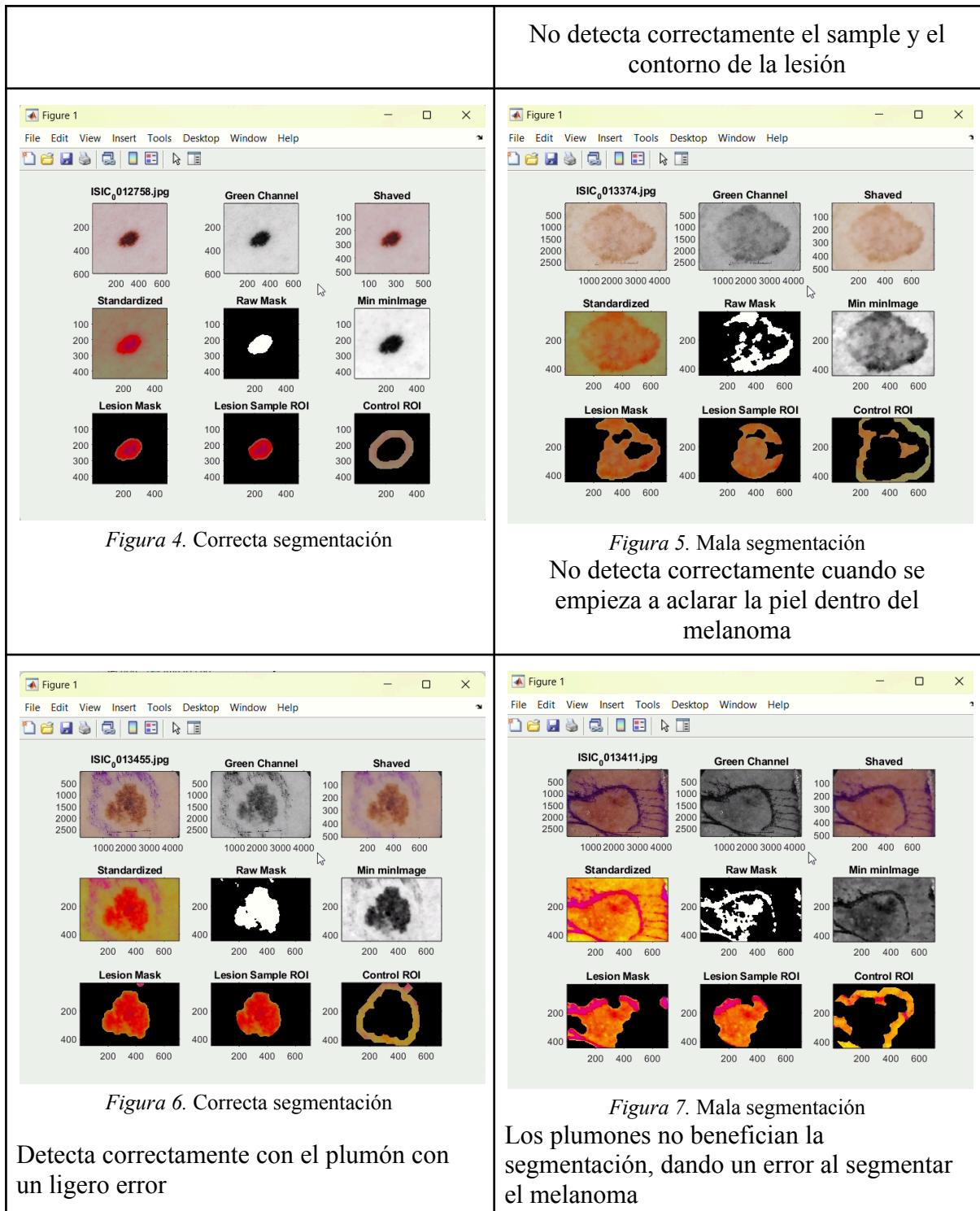
Resultados y Discusión

MATLAB

Resultados de segmentación

i. Ejemplos de buena y mala segmentación





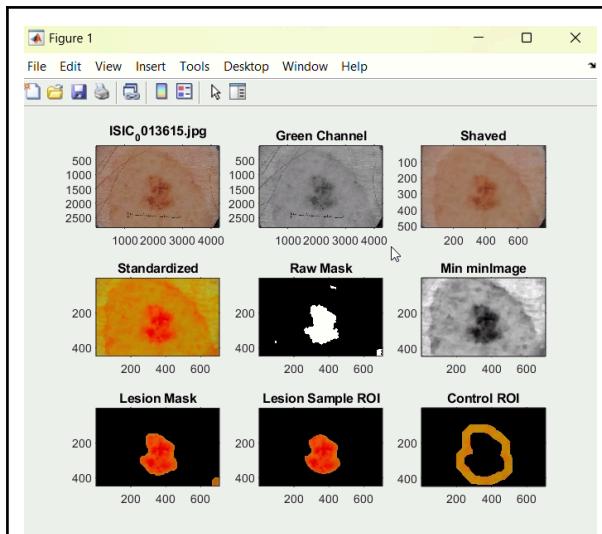


Figura 8. Correcta segmentación

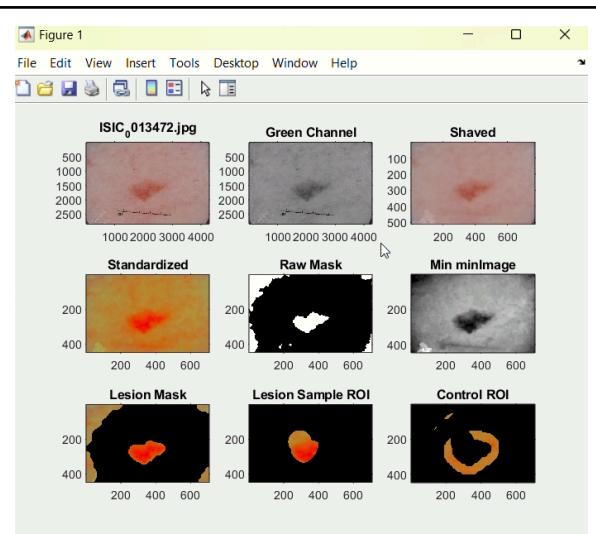


Figura 9. Mala segmentación

No detecta bien el contorno, crea esferas en el Sample ROI.

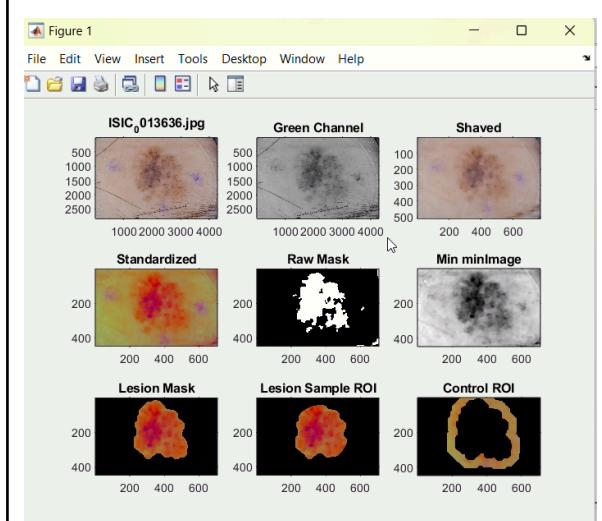


Figura 10. Correcta segmentación
Complejidad bien detectada

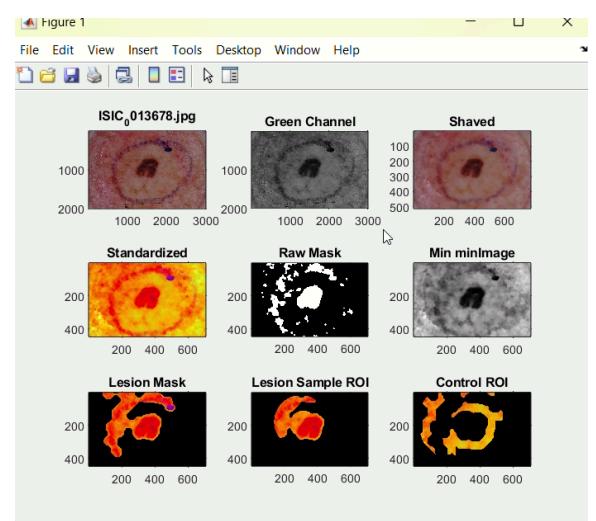


Figura 11. Mala segmentación
Detecta en el raw mask algunas incorrectas y aparte seleccionada con marcador como melanoma

Tabla 1. Segmentación buena y mala

Definimos una buena segmentación cuando la forma en la que segmenta la imagen es correcta y remarca solo aquello que es parte del melanoma y define el control zonas específicas que no cuentan con el melanoma para establecer un control de la imagen y de esa forma una comparativa. Por otro lado, definimos una mala segmentación cuando uno de estos rasgos no se cumple, o ya sea que uno de los factores no los haya cumplido correctamente, como por ejemplo la Figura 11, en la cual considera la máscara de la lesión aquello que está marcado con plumón y no es un melanoma.

Reporte de características Extraídas

A continuación, se detallan las características obtenidas del

i. Señal: Media, standard deviation, skewness y kurtosis

- “temp_mean”: Valor medio de intensidad de la lesión.
- “masked_m2”: Desviación estándar de los valores de intensidad de la lesión.
- “masked_m3”: Asimetría de los valores de intensidad de la lesión.
- “masked_m4”: Kurtosis de los valores de intensidad de la lesión.
- entropy: Entropía de los valores de intensidad de la lesión.
- cov: Coeficiente de variación.
- q90cov: Coeficiente de variación basado en cuantiles.

ii. Forma: Volumen, superficie, etc.

- “log(1+volume)”: Logaritmo del volumen de la lesión.
- “log(1+surface)”: Logaritmo del área de superficie de la lesión.
- “compactness”: Compacidad de la lesión.
- “gradsum/volume”: Magnitud media del gradiente por unidad de volumen.
- “gradsum2/gradsum”: Relación de gradientes de diferentes núcleos de convolución.

iii. Textura: GLCM y Haralick features

- “GLCM_stats_1.Contrast”: Contraste del primer GLCM.
- “GLCM_stats_1.Correlation”: Correlación del primer GLCM.
- “GLCM_stats_1.Energy”: Energía del primer GLCM.
- “GLCM_stats_1.Homogeneity”: Homogeneidad del primer GLCM.
- “GLCM_stats_2.Contrast”: Contraste del segundo GLCM.
- “GLCM_stats_2.Correlation”: Correlación del segundo GLCM.
- “GLCM_stats_2.Energy”: Energía del segundo GLCM.
- “GLCM_stats_2.Homogeneity”: Homogeneidad del segundo GLCM.
- “GLCM_stats_3.Contrast”: Contraste del tercer GLCM.
- “GLCM_stats_3.Correlation”: Correlación del tercer GLCM.
- “GLCM_stats_3.Energy”: Energía del tercer GLCM.
- “GLCM_stats_3.Homogeneity”: Homogeneidad del tercer GLCM.
- “GLCM_stats_4.Contrast”: Contraste del cuarto GLCM.
- “GLCM_stats_4.Correlation”: Correlación del cuarto GLCM.
- “GLCM_stats_4.Energy”: Energía del cuarto GLCM.
- “GLCM_stats_4.Homogeneity”: Homogeneidad del cuarto GLCM.
- “meanGLCMSlope (Correlation)": Pendiente media de la correlación de GLCM.
- “meanGLCMSlope (Contrast)": Pendiente media del contraste de GLCM.
- “meanGLCMSlope (Energy)": Pendiente media de la energía de GLCM.

- “meanGLCMSlope (Homogeneity)”: Pendiente media de la homogeneidad de GLCM.

iv. Fractal dimensions

- “compactness”: Compacidad de la lesión.
- “gradsum/volume”: Magnitud media del gradiente por unidad de volumen.
- “gradsum2/gradsum”: Relación de gradientes de diferentes núcleos de convolución.

Rstudio

Como método de visualización de la estructura de los datos obtenidos de las características. Esto con el fin de identificación de patrones, en la *Figura 12.*, las filas se colocaron las diferentes lesiones cargadas como imágenes con sus respectivas características en las columnas. Adicionalmente, en el lado izquierdo, podemos observar que las filas de las lesiones se marcaron con colores rojos y azules, donde el azul representa el 0 y una lesión no melanoma, y el rojo, siendo el opuesto, de melanoma identificado con valor 1. Adicionalmente, se colocaron árboles de dendrograma, que representan el ‘clustering’ jerárquico donde lesiones similares se agrupan, y de igual manera, características similares se agrupan.

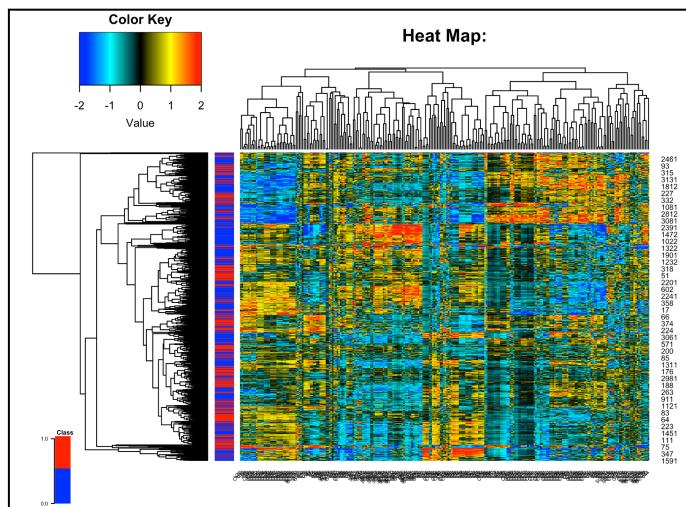


Figura 12. Resumen visual de distribución, lesiones (filas), y características (columnas).

Utilizando este mapa de calor, se puede observar cómo es que se muestran los diferentes tipos de lesiones, y como es que dependiendo del color, se puede relacionar la importancia de dicha característica para su clasificación. También se muestran filas con colores diferentes, lo cual representa un valor atípico o una lesión con características únicas.

Como siguiente, se realizaron gráficas ROC (Curva de Característica Operativa del Receptor), donde se utilizaron para visualizar el rendimiento de los diferentes modelos de

clasificación binaria en diferentes umbrales que se utilizaron. En sí, dentro de estos, la línea negra muestra la capacidad del modelo LASSO para clasificar entre dos clases (positiva y negativa) en varios umbrales. Cuanto más cerca esté la curva de la esquina superior izquierda, mejor es el modelo para discriminar entre las clases, que es lo que se busca.

Dentro de las siguientes *Figuras* (13, 14, 15, 16);

- Se presenta un AUC (área bajo la curva), la cual indica el rendimiento general del modelo. Varía de 0.5 (adivinanza aleatoria) a 1.0 (clasificación perfecta).
- Se presenta un BACC (Precisión Equilibrada), la cual es el promedio de sensibilidad (TPR) y la especificidad (TNR). Como en el caso de la *Figura 16.*, un valor de BACC de 0.900 indica una buena precisión general.
- Se presenta una matriz de confusión, donde dentro de esta, se encuentran los valores por área de la cantidad de Verdaderos Positivos (TP), Falsos Positivos (FP), Verdaderos Negativos (TN) y Falsos Negativos (FN), que predijeron los modelos.
- Se presentaron métricas adicionales:
 - TPR (Tasa de Verdaderos Positivos/Sensibilidad): Proporción de verdaderos positivos correctamente identificados (0.930).
 - TNR (Tasa de Verdaderos Negativos/Especificidad): Proporción de verdaderos negativos correctamente identificados (0.869).
 - ACC (Precisión): Proporción general de clasificaciones correctas (0.881).
 - F1 Score: Media armónica de la precisión y la recuperación (0.755).
 - BER (Tasa de Error Equilibrada): Promedio de la tasa de error en la clase positiva y la tasa de error en la clase negativa (0.100).

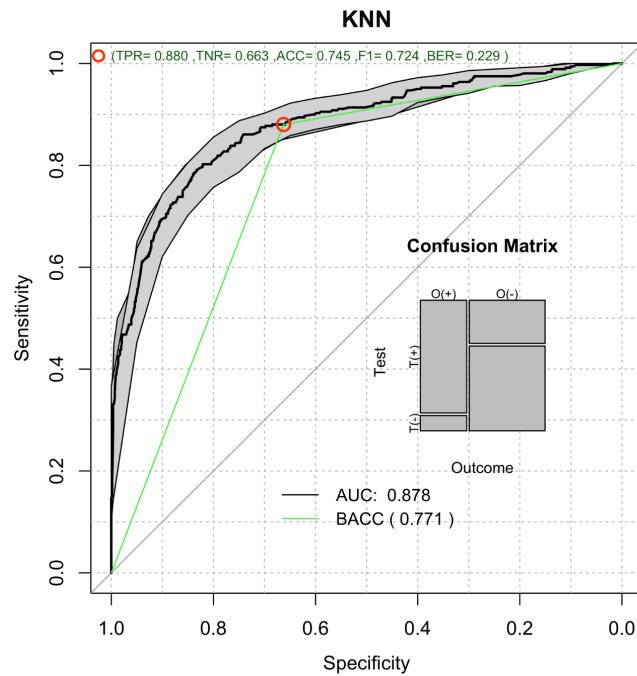


Figura 13. Gráfico KNN y ROC

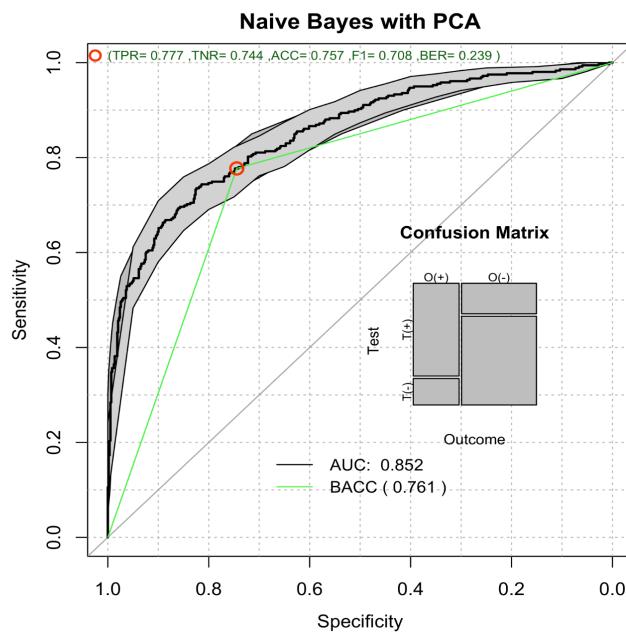


Figura 14. Gráfico NB y ROC

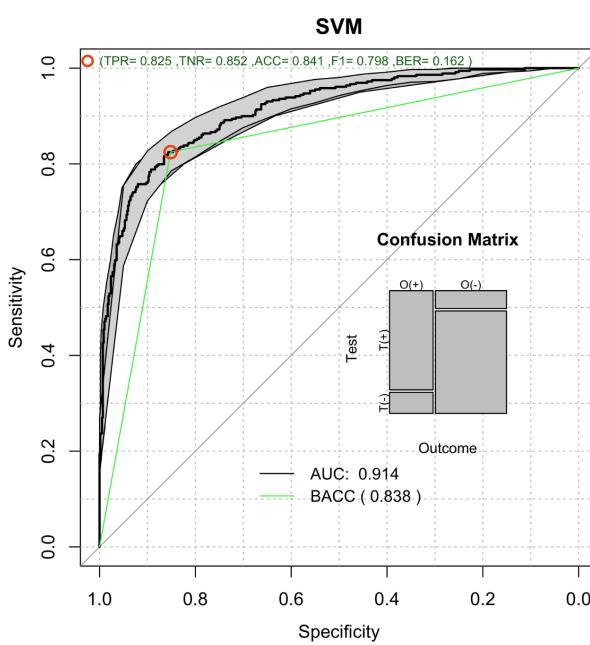


Figura 15. Gráfico SVM y ROC

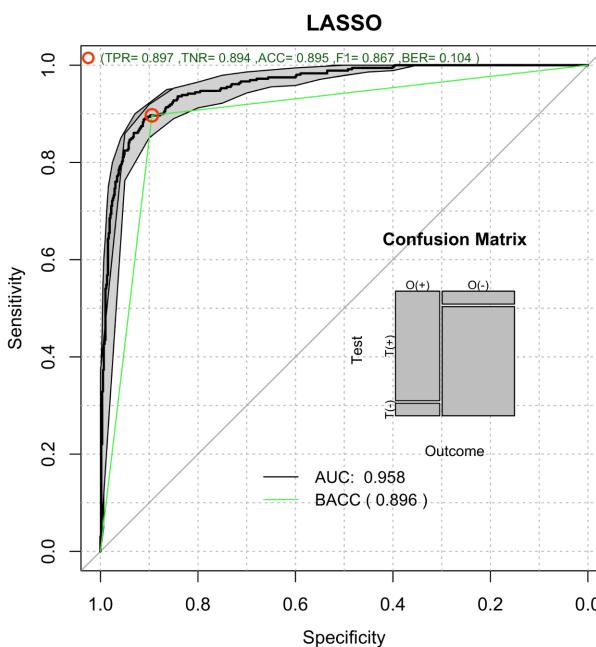


Figura 16. Gráfico LASSO

Las curvas ROC representan la relación entre la sensibilidad (TPR - True Positive Rate) y la especificidad (TNR - True Negative Rate) para cada modelo (KNN, NB, SVM, LASSO). Al estar cerca de la parte superior izquierda, indica que los modelos tienen una alta capacidad de discriminar entre las clases positivas y negativas, lo cual es deseable en problemas de clasificación. Cuanto más cerca estén las curvas ROC del punto (0,1) en el gráfico, mejor será el rendimiento del modelo en términos de sensibilidad y especificidad.

La proximidad de las curvas ROC a la esquina superior izquierda junto con la correcta clasificación en las matrices de confusión sugiere que los modelos (KNN, NB, SVM, LASSO, este último siendo el de mejor desempeño) tienen un buen rendimiento en la clasificación de melanoma y otras lesiones cutáneas. La alta sensibilidad y especificidad indican que los modelos son capaces de identificar correctamente tanto los casos positivos como los negativos, lo cual es crucial en aplicaciones médicas donde la precisión es vital.

Los resultados de las curvas ROC y las matrices de confusión sugieren que los modelos entrenados en R Studio (KNN, NB, SVM, LASSO) son efectivos en la clasificación de melanoma y otras lesiones cutáneas, mostrando un buen equilibrio entre sensibilidad y especificidad.

Categoría	KNN	Naive Bayes (NB)	SVM	LASSO
TPR (Tasa de Verdaderos Positivos/Sensibilidad)	0.880	0.777	0.825	0.897
TNR (Tasa de Verdaderos Negativos/Especificidad)	0.663	0.744	0.852	0.894
ACC (Precisión)	0.745	0.757	0.841	0.895
F1 (Media armónica de la precisión y la recuperación)	0.724	0.708	0.798	0.867
BER (Tasa de Error Equilibrada)	0.229	0.239	0.162	0.104

Tabla 2. Comparación de categorías de desempeño

Se hizo uso de una gráfica de barras para demostrar la frecuencia con la que cada característica fue seleccionada durante la validación cruzada del modelo LASSO, con sus perspectivas etiquetas para cada barra. De esta forma, el gráfico ayuda a visualizar qué características son más importantes o más frecuentes seleccionadas por el modelo visto en la figura 17.

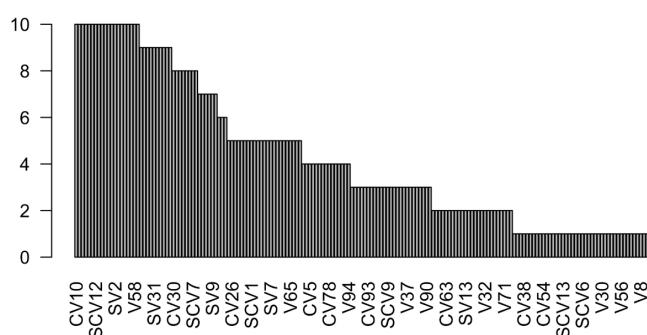


Figura 17. Gráfica de barras de características más importantes/frecuentes por el modelo

Este histograma permite identificar cuáles son las características clave que más contribuyen al rendimiento del modelo y subsecuentemente reducir la dimensionalidad del conjunto de datos al centrarse solo en estas características del conjunto de datos al centrarse solo en estas características para mejorar la interpretabilidad del modelo.

Dentro de la *Figura 17*, se graficaron las características mayormente utilizadas en el modelo LASSO y se logra apreciar un mayor uso de las características CV10, SCV12 y SV2, y como las características que se consideran menos importantes/frecuentes serían la CV38, CV54, SCV6, V30, V56, V8.

Conclusiones

Luis Angel González Castillo (A01571007)

El uso de técnicas avanzadas de procesamiento de imágenes y algoritmos de aprendizaje automático, como K-Nearest Neighbors (KNN), Naive Bayes (NB), Support Vector Machines (SVM) y LASSO, ha demostrado ser efectivo en la clasificación precisa de melanoma y otras lesiones cutáneas. Los modelos entrenados en R Studio muestran un buen equilibrio entre sensibilidad y especificidad, lo cual es importante para aplicaciones médicas. La alta sensibilidad y especificidad indican que los modelos identifican correctamente tanto los casos positivos como los negativos, lo cual es fundamental para la detección temprana y precisa del melanoma.

El análisis de curvas ROC, matrices de confusión y métricas de desempeño como la sensibilidad, especificidad, precisión y F1 score respalda la eficacia de estos modelos. La combinación de la extracción de características de las imágenes médicas y el uso de algoritmos de aprendizaje automático mejora los diagnósticos médicos y la detección temprana de enfermedades cutáneas. El reporte destaca el potencial del aprendizaje automático en el diagnóstico de enfermedades dermatológicas, mostrando como estas técnicas podrían ser útiles en el ámbito profesional.

José Antonio Palacios Dorbecker (A01177930)

Este estudio ha evaluado y comparado el desempeño de métodos de aprendizaje automático en la clasificación de melanoma y otras lesiones cutáneas. Utilizando técnicas de segmentación de imágenes y extracción de características.

Los resultados demostraron que los modelos evaluados, especialmente el modelo LASSO, presentaron un desempeño robusto en la clasificación de melanoma, con una alta sensibilidad y especificidad, lo que indica una capacidad notable para diferenciar entre melanoma y otras lesiones cutáneas. Esto es consistente con la literatura previa, que destaca la capacidad del aprendizaje automático para mejorar los diagnósticos médicos mediante la identificación de patrones sutiles en grandes volúmenes de datos (El Naqa et al., 2015) (González et al., 2016).

La validación cruzada utilizada para evaluar los modelos asegura que los resultados sean generalizables y no dependan de un único conjunto de datos de entrenamiento, mejorando así la confiabilidad de los modelos propuestos (Nti et al., 2021). Los métodos de regularización como LASSO, que penalizan la complejidad del modelo, también ayudan a prevenir el sobreajuste, lo cual es crucial en aplicaciones médicas donde la precisión es vital.

La implementación de modelos en entornos clínicos podría mejorar significativamente la detección temprana y precisa de melanoma, contribuyendo a mejores resultados para los pacientes.

Daniel Agustín Aguilar López (A01735969)

El documento presenta una evaluación detallada de varios métodos de aprendizaje automático aplicados a la clasificación de melanoma y otras lesiones cutáneas, utilizando bases de datos de imágenes segmentadas y características extraídas. Los modelos evaluados incluyen K-Nearest Neighbors (KNN) que clasifica una muestra basándose en los “k puntos” de datos más cercanos en el espacio de características, Naive Bayes (NB) que asume la independencia entre características y aplica el teorema de Bayes, Support Vector Machines (SVM) que busca el hiperplano óptimo que maximiza el margen entre clases y Least Absolute Shrinkage and Selection Operator (LASSO) que es un método de regresión que realiza selección de características y regularización (Nti et al, 2021). Los resultados se analizaron en términos de sensibilidad, especificidad, F1 score y curvas ROC, demostrando que la combinación de técnicas avanzadas de procesamiento de imágenes y algoritmos de aprendizaje automático puede mejorar significativamente la detección temprana y precisa del melanoma.

El código proporcionado en el documento permite la extracción y comparación de características cuantitativas de las lesiones, facilitando la clasificación precisa y el análisis de datos. Este código incluye análisis en R y MATLAB, lo que permite una amplia gama de análisis de datos y visualización de resultados. Además, se destacan las métricas calculadas como la media, la entropía, la compacidad y las estadísticas de la matriz de co-ocurrencia de niveles de gris (GLCM), que son cruciales para la caracterización de las lesiones cutáneas.

Carlos Andrés Luis Martínez (A00833514)

En conclusión, este estudio confirma la efectividad de los modelos de aprendizaje automático, específicamente KNN, Naive Bayes, SVM y LASSO, en la clasificación de melanomas y otras lesiones cutáneas. Utilizando técnicas avanzadas de segmentación de imágenes y extracción de características, se logró una alta precisión en la identificación de bordes de lesiones y la eliminación de artefactos, lo cual es fundamental para la correcta extracción de características.

Los resultados muestran que los modelos evaluados lograron una alta sensibilidad y especificidad, indicando una notable capacidad para distinguir entre melanomas y otras

lesiones cutáneas. Esto es consistente con estudios previos que destacan la capacidad del aprendizaje automático para mejorar los diagnósticos médicos mediante la identificación de patrones sutiles en grandes conjuntos de datos. La validación cruzada empleada asegura que los resultados obtenidos sean confiables.

Referencias

Rafael C. Gonzalez, Richard E. Woods (2016).Procesamiento De Imagenes Digitales ,
<https://archive.org/details/ProcesamientoDeImagenesDigitales2daEdicionRafaelC.GonzalezRichardE.Woods>

El Naqa I, Murphy MJ (2015) Machine learning in radiation oncology: theory and applications. https://doi.org/10.1007/978-3-319-18305-3_1 What is Machine Learning? Springer, Switzerland

Tschandl, P., Rosendahl, C., & Kittler, H. (2018). The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific Data*, 5, 180161. <https://doi.org/10.1038/sdata.2018.161>

Nti, Isaac & Nyarko-Boateng, Owusu & Aning, Justice. (2021). Performance of Machine Learning Algorithms with Different K Values in K-fold Cross-Validation. *International Journal of Information Technology and Computer Science*. 6. 61-71. 10.5815/ijites.2021.06.05.

Tougui, I., Jilbab, A., & Mhamdi, J. E. (2021). Impact of the Choice of Cross-Validation Techniques on the Results of Machine Learning-Based Diagnostic Applications. *Healthcare informatics research*, 27(3), 189–199. <https://doi.org/10.4258/hir.2021.27.3.189>

López de Ullibarri Galparsoro, I., & Pita Fernández, S. (1998). Curvas ROC. *Cad Aten Primaria*, 5(4), 229-235.

Sachdeva, J., Kumar, V., Gupta, I. et al. Segmentation, Feature Extraction, and Multiclass Brain Tumor Classification. *J Digit Imaging* 26, 1141–1150 (2013).
<https://doi.org/10.1007/s10278-013-9600-0>

Anexos

Códigos de MATLAB

Código 1 “LoadAndProcessQuantitateImage”

```
%% Structural Elements
filterDisk30 = strel('disk',3);
filterDisk0 = strel('disk',5);
filterDisk = strel('disk',25);
largeDisk = strel('disk',50);
filterDisk2 = strel('disk',7);
filterDisk3 = strel('disk',10);
numGLCMBins = 40;
%% Filename
quantitativeLesionData = [];
quantitativeControlData = [];
imagename = strcat(melanomopath,imageFile);
%% Image Loading
f=imread(imagename);
figure(1)
subplot(3,3,1)
imshow(f)
title(imageFile);
subplot(3,3,2)
imshow(f(:,:,2),[])
title("Green Channel");
Gmedian = median(double(f(:,:,2))/255.0,'all');
sz = size(f);
f=double(imresize(f,[512,512*sz(2)/sz(1)]))/255;
f = imclose(f,filterDisk0);
f = imopen(f,filterDisk0);
subplot(3,3,3)
imshow(f,[])
title("Shaved");
f = imcrop(f,[32 32 (512*sz(2)/sz(1)-64) 447 ]);
fcp = f;
RImage = f(:,:,1);
GImage = f(:,:,2);
BImage = f(:,:,3);
Gcenter = imcrop(GImage,[32 32 (448*sz(2)/sz(1)-64) 383 ]);
Ggain = 1.00/Gmedian;
f(:,:,1) = (RImage-GImage)*Ggain + 0.5;
f(:,:,2) = (GImage-Gmedian)*Ggain + 0.5;
f(:,:,3) = (BImage-RImage)*Ggain + 0.5;
fcenter = imcrop(fcp,[32 32 (448*sz(2)/sz(1)-64) 383 ]);
for channel = 1:3
    fcp(:,:,channel) = fcp(:,:,channel)-min(min(fcenter(:,:,channel)));
    fcp(:,:,channel) = fcp(:,:,channel)./max(max(fcp(:,:,channel)));
end
minImage = min(fcp,[],3);
figure(1)
subplot(3,3,4)
imshow(f,[])
title("Standardized");
maskthr = 0.75*(median(minImage,'all') + graythresh(minImage))/2.0;
mask = minImage <= maskthr;
subplot(3,3,5)
imshow(mask,[])
title('Raw Mask');
maskMorph = imdilate(mask,filterDisk3);
maskMorph = imclose(maskMorph,filterDisk3);
maskMorph = imopen(maskMorph,filterDisk);
finalmask = imclose(maskMorph,filterDisk);
%imshow(finalmask,[])
%title('Clean Mask');
finalmaskD = imdilate(maskMorph,filterDisk2);
controlmask = imdilate(finalmaskD,filterDisk);
controlmask = imdilate(controlmask,filterDisk) & not(finalmaskD);
%%
subplot(3,3,6)
imshow(minImage,[0,max(max(minImage))])
title('Min minImage');
subplot(3,3,7)
```

```

lesionimage = f.*finalmask ;
imshow(lesionimage,[]);
title('Lesion Mask');
templateRadius = sum(sum(finalmask))/3;
templateRadius2 = sum(sum(finalmask))/12;
sz = size(finalmask);
[X,Y] = meshgrid(1:sz(2),1:sz(1));
templateMx = 0.7*mean(X(finalmask)) + 0.3*sz(2)/2;
templateMy = 0.7*mean(Y(finalmask)) + 0.3*sz(1)/2;
templateMx2 = 0.5*mean(X(finalmask)) + 0.5*sz(2)/2;
templateMy2 = 0.5*mean(Y(finalmask)) + 0.5*sz(1)/2;
finalmask = finalmask & (((X - templateMx).^2 + (Y - templateMy).^2) <
templateRadius);
finalmask = imopen(finalmask,filterDisk3);
finalmask = imclose(finalmask,filterDisk);
if (templateRadius2 < 900 )
    templateRadius2 = 900;
end
finalmask = finalmask | (((X - templateMx2).^2 + (Y - templateMy2).^2) <
templateRadius2);
controlmask = controlmask & not(finalmask) &
imdilate(imdilate(finalmask,largeDisk),largeDisk);
lesionimage = f.*finalmask ;
subplot(3,3,8)
imshow(lesionimage,[]);
title('Lesion Sample ROI');
controlimage = f.*controlmask;
subplot(3,3,9)
imshow(controlimage)
title('Control ROI');
%% Extract Lesion features
dy = [-1 -2 -1; 0 0 0; 1 2 1]/8;
dx = dy';
dy2 = [-1 -2 -1; 0 0 0; 0 0 0; 0 0 0; 1 2 1]/16;
dx2 = dy2';
maskarea = sum(sum(finalmask));
for channel = 1:3

    LesionData = f(:,:,channel);
    minv = min(min(LesionData));
    adLesionData = LesionData - minv;

    % imshow(LesionData,[]);
    histoData = reshape(LesionData(finalmask),1,[]);
    temp_mean = mean(histoData);
    temp_m2 = moment(histoData,2);
    temp_m3 = moment(histoData,3);
    temp_m4 = moment(histoData,4);

    masked_m2 = sqrt(temp_m2);
    masked_m3 = power(abs(temp_m3),1/3).*sign(temp_m3);
    masked_m4 = power(temp_m4,1/4);
    dq = quantile(histoData,[0.01,0.05,0.25,0.5,0.75,0.95,0.99]);
    cov = log(10000.0*masked_m2 + 1.0);
    q90cov = log(10000.0*(dq(6)-dq(2)) + 1.0);
    if (abs(temp_mean) > 0)
        cov = log(10000.0*masked_m2/abs(temp_mean) + 1.0);
    end
    if (abs(dq(4)) > 0)
        q90cov = log(10000.0*(dq(6)-dq(2))/abs(dq(4)) + 1.00);
    end
    [N,edges] = histcounts(histoData,32);
    N = N/sum(N);
    N = N.*log(N);
    TF = isnan(N);
    N(TF) = 0;
    entropy = -sum(N);

    quantitativeLesionData =
[quantitativeLesionData,temp_mean,masked_m2,masked_m3,masked_m4,entropy,cov,q90cov];

volume = sum(sum(adLesionData(finalmask)));
grad = (abs(conv2(LesionData,dx,'same')) + abs(conv2(LesionData,dy,'same')))/2;
grad2 = (abs(conv2(LesionData,dx2,'same')) +
abs(conv2(LesionData,dy2,'same')))/2;
gradsum = sum(sum(grad(finalmask)));

```

```

gradsum2 = sum(sum(grad2(finalmask)));
surface = 128*gradsum + maskarea;
compactness = 512*volume/surface^(3/2);
quantitativeLesionData =
[quantitativeLesionData,log(1+volume),log(1+surface),compactness,gradsum/volume,gradsum2/gra
dsum];
%%%%%% Temperature GLCM %%%%%%
glcm_1 =
graycomatrix(LesionData(finalmask),'NumLevels',numGLCMBins,'GrayLimits',[-1.0,1.50],'Offset
',[0 2; -2 2; -2 0; -2 -2],'Symmetric',true);
glcm_2 =
graycomatrix(LesionData(finalmask),'NumLevels',numGLCMBins,'GrayLimits',[-1.0,1.50],'Offset
',[0 4; -3 3; -4 0; -3 -3],'Symmetric',true);
glcm_3 =
graycomatrix(LesionData(finalmask),'NumLevels',numGLCMBins,'GrayLimits',[-1.0,1.50],'Offset
',[0 8; -6 6; -8 0; -6 -6],'Symmetric',true);
glcm_4 =
graycomatrix(LesionData(finalmask),'NumLevels',numGLCMBins,'GrayLimits',[-1.0,1.50],'Offset
',[0 16; -11 11; -16 0; -11 -11],'Symmetric',true);
glcm_1 = sum(glcm_1,3);
glcm_2 = sum(glcm_2,3);
glcm_3 = sum(glcm_3,3);
glcm_4 = sum(glcm_4,3);
GLCM_stats_1 = graycoprops(glcm_1);
GLCM_stats_2 = graycoprops(glcm_2);
GLCM_stats_3 = graycoprops(glcm_3);
GLCM_stats_4 = graycoprops(glcm_4);
quantitativeLesionData =
[quantitativeLesionData,GLCM_stats_1.Congruence,GLCM_stats_1.Correlation,GLCM_stats_1.Energy,G
LCM_stats_1.Homogeneity];
quantitativeLesionData =
[quantitativeLesionData,GLCM_stats_2.Congruence,GLCM_stats_2.Correlation,GLCM_stats_2.Energy,G
LCM_stats_2.Homogeneity];
quantitativeLesionData =
[quantitativeLesionData,GLCM_stats_3.Congruence,GLCM_stats_3.Correlation,GLCM_stats_3.Energy,G
LCM_stats_3.Homogeneity];
quantitativeLesionData =
[quantitativeLesionData,GLCM_stats_4.Congruence,GLCM_stats_4.Correlation,GLCM_stats_4.Energy,G
LCM_stats_4.Homogeneity];
GLCMSlope =
[abs(log(1/2*(1.001-GLCM_stats_1.Congruence)/(1.001-GLCM_stats_2.Congruence)))/log(2)];
GLCMSlope =
[GLCMSlope,abs(log(1/2*(1.001-GLCM_stats_2.Congruence)/(1.001-GLCM_stats_3.Congruence)))/
log(2)];
GLCMSlope =
[GLCMSlope,abs(log(1/4*(1.001-GLCM_stats_1.Congruence)/(1.001-GLCM_stats_3.Congruence)))/
log(4)];
GLCMSlope =
[GLCMSlope,abs(log(1/2*(1.001-GLCM_stats_3.Congruence)/(1.001-GLCM_stats_4.Congruence)))/
log(2)];
GLCMSlope =
[GLCMSlope,abs(log(1/4*(1.001-GLCM_stats_2.Congruence)/(1.001-GLCM_stats_4.Congruence)))/
log(4)];
GLCMSlope =
[GLCMSlope,abs(log(1/8*(1.001-GLCM_stats_1.Congruence)/(1.001-GLCM_stats_4.Congruence)))/
log(8)];
meanGLCMSlope = mean(GLCMSlope);
quantitativeLesionData = [quantitativeLesionData,meanGLCMSlope];
GLCMSlope = [abs(log(1/2*GLCM_stats_1.Congruence/GLCM_stats_2.Congruence))/log(2)];
GLCMSlope =
[GLCMSlope,abs(log(1/2*GLCM_stats_2.Congruence/GLCM_stats_3.Congruence))/log(2)];
GLCMSlope =
[GLCMSlope,abs(log(1/4*GLCM_stats_1.Congruence/GLCM_stats_3.Congruence))/log(4)];
GLCMSlope =
[GLCMSlope,abs(log(1/2*GLCM_stats_3.Congruence/GLCM_stats_4.Congruence))/log(2)];
GLCMSlope =
[GLCMSlope,abs(log(1/4*GLCM_stats_2.Congruence/GLCM_stats_4.Congruence))/log(4)];
GLCMSlope =
[GLCMSlope,abs(log(1/8*GLCM_stats_1.Congruence/GLCM_stats_4.Congruence))/log(8)];
meanGLCMSlope = mean(GLCMSlope);
quantitativeLesionData = [quantitativeLesionData,meanGLCMSlope];
GLCMSlope = [abs(log(2*GLCM_stats_1.Energy/GLCM_stats_2.Energy))/log(2)];
GLCMSlope =
[GLCMSlope,abs(log(2*GLCM_stats_2.Energy/GLCM_stats_3.Energy))/log(2)];
GLCMSlope =
[GLCMSlope,abs(log(4*GLCM_stats_1.Energy/GLCM_stats_3.Energy))/log(4)];

```

```

GLCMSlope =
[GLCMSlope,abs(log(2*GLCM_stats_3.Energy/GLCM_stats_4.Energy))/log(2)];
GLCMSlope =
[GLCMSlope,abs(log(4*GLCM_stats_2.Energy/GLCM_stats_4.Energy))/log(4)];
GLCMSlope =
[GLCMSlope,abs(log(8*GLCM_stats_1.Energy/GLCM_stats_4.Energy))/log(8));
meanGLCMSlope = mean(GLCMSlope);
quantitativeLesionData = [quantitativeLesionData,meanGLCMSlope];
GLCMSlope =
[abs(log(1/2*(1.001-GLCM_stats_1.Homogeneity)/(1.001-GLCM_stats_2.Homogeneity)))/log(2)];
GLCMSlope =
[GLCMSlope,abs(log(1/2*(1.001-GLCM_stats_2.Homogeneity)/(1.001-GLCM_stats_3.Homogeneity)))/log(2)];
GLCMSlope =
[GLCMSlope,abs(log(1/4*(1.001-GLCM_stats_1.Homogeneity)/(1.001-GLCM_stats_3.Homogeneity)))/log(4)];
GLCMSlope =
[GLCMSlope,abs(log(1/2*(1.001-GLCM_stats_3.Homogeneity)/(1.001-GLCM_stats_4.Homogeneity)))/log(2)];
GLCMSlope =
[GLCMSlope,abs(log(1/4*(1.001-GLCM_stats_2.Homogeneity)/(1.001-GLCM_stats_4.Homogeneity)))/log(4)];
GLCMSlope =
[GLCMSlope,abs(log(1/8*(1.001-GLCM_stats_1.Homogeneity)/(1.001-GLCM_stats_4.Homogeneity)))/log(8)];
meanGLCMSlope = mean(GLCMSlope);
quantitativeLesionData = [quantitativeLesionData,meanGLCMSlope];
end
quantitativeLesionData = [quantitativeLesionData,Ggain,Gmedian];
allquantitativeLesionData = [allquantitativeLesionData;quantitativeLesionData];
%% Extract Control features
maskarea = sum(sum(controlmask));
for channel = 1:3

    ControlData = f(:,:,:,channel);
    minv = min(min(ControlData));
    aControlData = ControlData - minv;
%    imshow(aControlData,[]);
    histoData = reshape(ControlData(controlmask),1,[ ]);
    temp_mean = mean(histoData);
    temp_m2 = moment(histoData,2);
    temp_m3 = moment(histoData,3);
    temp_m4 = moment(histoData,4);

    masked_m2 = sqrt(moment(histoData,2));
    masked_m3 = moment(histoData,3);
    masked_m3 = power(abs(masked_m3),1/3).*sign(masked_m3);
    masked_m4 = power(moment(histoData,4),1/4);
    dq = quantile(histoData,[0.01,0.05,0.25,0.5,0.75,0.95,0.99]);
    cov=0;
    q90cov=0;
    if (abs(temp_mean) > 0)
        cov = log(10000.0*masked_m2/abs(temp_mean) + 1.0);
    end
    if (abs(dq(4)) > 0)
        q90cov = log(10000.0*(dq(6)-dq(2))/abs(dq(4)) + 1.0);
    end
    [N,edges] = histcounts(histoData,32);
    N = N/sum(N);
    N = N.*log(N);
    TF = isnan(N);
    N(TF) = 0;
    entropy = -sum(N);

    quantitativeControlData =
[quantitativeControlData,temp_mean,masked_m2,masked_m3,masked_m4,entropy,cov,q90cov];
    volume = sum(sum(aControlData(controlmask)));
    grad = (abs(conv2(ControlData,dx,'same')) +
abs(conv2(ControlData,dy,'same')))/2;
    grad2 = (abs(conv2(ControlData,dx2,'same')) +
abs(conv2(ControlData,dy2,'same')))/2;
    gradsum = sum(sum(grad(controlmask)));
    gradsum2 = sum(sum(grad2(controlmask)));
    surface = 128*gradsum + maskarea;
    compactness = 512*volume/surface^(3/2);

```

```

quantitativeControlData =
[quantitativeControlData, log(1.0+volume+1), log(1.0+surface), compactness, gradsum/volume, grads
um2/gradsum];

%%%%%%%%%%%%% GLCM %%%%%%%%%%%%%%
glcm_1 =
graycomatrix(ControlData(controlmask), 'NumLevels', numGLCMBins, 'GrayLimits', [-1.0, 1.50], 'Off
set', [0 2; -2 2; -2 0; -2 -2], 'Symmetric', true);
glcm_2 =
graycomatrix(ControlData(controlmask), 'NumLevels', numGLCMBins, 'GrayLimits', [-1.0, 1.50], 'Off
set', [0 4; -3 3; -4 0; -3 -3], 'Symmetric', true);
glcm_3 =
graycomatrix(ControlData(controlmask), 'NumLevels', numGLCMBins, 'GrayLimits', [-1.0, 1.50], 'Off
set', [0 8; -6 6; -8 0; -6 -6], 'Symmetric', true);
glcm_4 =
graycomatrix(ControlData(controlmask), 'NumLevels', numGLCMBins, 'GrayLimits', [-1.0, 1.50], 'Off
set', [0 16; -11 11; -16 0; -11 -11], 'Symmetric', true);
glcm_1 = sum(glcm_1, 3);
glcm_2 = sum(glcm_2, 3);
glcm_3 = sum(glcm_3, 3);
glcm_4 = sum(glcm_4, 3);
GLCM_stats_1 = graycoprops(glcm_1);
GLCM_stats_2 = graycoprops(glcm_2);
GLCM_stats_3 = graycoprops(glcm_3);
GLCM_stats_4 = graycoprops(glcm_4);
quantitativeControlData =
[quantitativeControlData, GLCM_stats_1.Contrast, GLCM_stats_1.Correlation, GLCM_stats_1.Energy,
GLCM_stats_1.Homogeneity];
quantitativeControlData =
[quantitativeControlData, GLCM_stats_2.Contrast, GLCM_stats_2.Correlation, GLCM_stats_2.Energy,
GLCM_stats_2.Homogeneity];
quantitativeControlData =
[quantitativeControlData, GLCM_stats_3.Contrast, GLCM_stats_3.Correlation, GLCM_stats_3.Energy,
GLCM_stats_3.Homogeneity];
quantitativeControlData =
[quantitativeControlData, GLCM_stats_4.Contrast, GLCM_stats_4.Correlation, GLCM_stats_4.Energy,
GLCM_stats_4.Homogeneity];
GLCMSlope =
[abs(log(1/2*(1.001-GLCM_stats_1.Correlation)/(1.001-GLCM_stats_2.Correlation)))/log(2)];
GLCMSlope =
[GLCMSlope, abs(log(1/2*(1.001-GLCM_stats_2.Correlation)/(1.001-GLCM_stats_3.Correlation)))/
log(2)];
GLCMSlope =
[GLCMSlope, abs(log(1/4*(1.001-GLCM_stats_1.Correlation)/(1.001-GLCM_stats_3.Correlation)))/
log(4)];
GLCMSlope =
[GLCMSlope, abs(log(1/2*(1.001-GLCM_stats_3.Correlation)/(1.001-GLCM_stats_4.Correlation)))/
log(2)];
GLCMSlope =
[GLCMSlope, abs(log(1/4*(1.001-GLCM_stats_2.Correlation)/(1.001-GLCM_stats_4.Correlation)))/
log(4)];
GLCMSlope =
[GLCMSlope, abs(log(1/8*(1.001-GLCM_stats_1.Correlation)/(1.001-GLCM_stats_4.Correlation)))/
log(8)];
meanGLCMSlope = mean(GLCMSlope);
quantitativeControlData = [quantitativeControlData, meanGLCMSlope];
GLCMSlope = [abs(log(1/2*GLCM_stats_1.Contrast/GLCM_stats_2.Contrast))/log(2)];
GLCMSlope =
[GLCMSlope, abs(log(1/2*GLCM_stats_2.Contrast/GLCM_stats_3.Contrast))/log(2)];
GLCMSlope =
[GLCMSlope, abs(log(1/4*GLCM_stats_1.Contrast/GLCM_stats_3.Contrast))/log(4)];
GLCMSlope =
[GLCMSlope, abs(log(1/2*GLCM_stats_3.Contrast/GLCM_stats_4.Contrast))/log(2)];
GLCMSlope =
[GLCMSlope, abs(log(1/4*GLCM_stats_2.Contrast/GLCM_stats_4.Contrast))/log(4)];
GLCMSlope =
[GLCMSlope, abs(log(1/8*GLCM_stats_1.Contrast/GLCM_stats_4.Contrast))/log(8)];
meanGLCMSlope = mean(GLCMSlope);
quantitativeControlData = [quantitativeControlData, meanGLCMSlope];
GLCMSlope = [abs(log(2*GLCM_stats_1.Energy/GLCM_stats_2.Energy))/log(2)];
GLCMSlope =
[GLCMSlope, abs(log(2*GLCM_stats_2.Energy/GLCM_stats_3.Energy))/log(2)];
GLCMSlope =
[GLCMSlope, abs(log(4*GLCM_stats_1.Energy/GLCM_stats_3.Energy))/log(4)];
GLCMSlope =
[GLCMSlope, abs(log(2*GLCM_stats_3.Energy/GLCM_stats_4.Energy))/log(2)];

```

```

GLCMSlope =
[GLCMSlope,abs(log(4*GLCM_stats_2.Energy/GLCM_stats_4.Energy))/log(4)];
GLCMSlope =
[GLCMSlope,abs(log(8*GLCM_stats_1.Energy/GLCM_stats_4.Energy))/log(8)];
meanGLCMSlope = mean(GLCMSlope);
quantitativeControlData = [quantitativeControlData,meanGLCMSlope];
GLCMSlope =
[abs(log(1/2*(1.001-GLCM_stats_1.Homogeneity)/(1.001-GLCM_stats_2.Homogeneity)))/log(2)];
GLCMSlope =
[GLCMSlope,abs(log(1/2*(1.001-GLCM_stats_2.Homogeneity)/(1.001-GLCM_stats_3.Homogeneity)))/log(2)];
GLCMSlope =
[GLCMSlope,abs(log(1/4*(1.001-GLCM_stats_1.Homogeneity)/(1.001-GLCM_stats_3.Homogeneity)))/log(4)];
GLCMSlope =
[GLCMSlope,abs(log(1/2*(1.001-GLCM_stats_3.Homogeneity)/(1.001-GLCM_stats_4.Homogeneity)))/log(2)];
GLCMSlope =
[GLCMSlope,abs(log(1/4*(1.001-GLCM_stats_2.Homogeneity)/(1.001-GLCM_stats_4.Homogeneity)))/log(4)];
GLCMSlope =
[GLCMSlope,abs(log(1/8*(1.001-GLCM_stats_1.Homogeneity)/(1.001-GLCM_stats_4.Homogeneity)))/log(8)];
meanGLCMSlope = mean(GLCMSlope);
quantitativeControlData = [quantitativeControlData,meanGLCMSlope];
end
allquantitativeControlData = [allquantitativeControlData;quantitativeControlData];

```

Código 2 “ProcessMelanomaImages”

Melanoma

```
melanomopath='C:\Users\tibur\OneDrive\Documentos\MATLAB\DataMelanoma\test\melanoma\'  
;  
allquantitativeLesionData = [];  
allquantitativeControlData = [];  
files = dir(fullfile(melanomopath, '*.jpg')); % lista de archivos .jpg en la  
carpeta  
for i = 1:length(files)  
    imageFile = files(i).name;  
    loadAndProcessQuantitateImage;  
  
end  
writematrix(allquantitativeLesionData,'MelanomaLesionFeatures.csv')  
writematrix(allquantitativeControlData,'MelanomaControlFeatures.csv')
```

Nevus

```
melanomopath='C:\Users\tibur\OneDrive\Documentos\MATLAB\DataMelanoma\test\nevus\'  
allquantitativeLesionData = [];  
allquantitativeControlData = [];  
files = dir(fullfile(melanomopath, '*.jpg')); % lista de archivos .jpg en la  
carpeta  
for i = 1:length(files)  
    imageFile = files(i).name;  
    loadAndProcessQuantitateImage;  
  
end  
writematrix(allquantitativeLesionData,'NevusLesionFeatures.csv')  
writematrix(allquantitativeControlData,'NevusControlFeatures.csv')
```

Seborrheic

```
melanomopath='C:\Users\tibur\OneDrive\Documentos\MATLAB\DataMelanoma\test\seborrhei  
c\';  
allquantitativeLesionData = [];  
allquantitativeControlData = [];  
files = dir(fullfile(melanomopath, '*.jpg')); % lista de archivos .jpg en la  
carpeta  
for i = 1:length(files)  
    imageFile = files(i).name;  
    loadAndProcessQuantitateImage;  
  
end  
writematrix(allquantitativeLesionData,'SeborrheicLesionFeatures.csv')  
writematrix(allquantitativeControlData,'SeborrheicControlFeatures.csv')
```

Código de R Segmentación

title: "MelanomavsSeborreic"

author: "Jose Tamez"

date: "4/24/2024"

output:

html_document:

toc: true

```

fig_caption: true
number_sections: true
editor_options:
  chunk_output_type: console
---

{r setup, include=FALSE}
knitr::opts_chunk$set(collapse = TRUE, warning = FALSE, message = FALSE, comment = "#>")

# Melanoma vs Seborreic_Keratosis

{r functions,echo = FALSE }

library("FRESA.CAD")
library("e1071")
library(readxl)

## Loading data sets

{r}

# /Users/joseantoniopalaciosdorbecker/Documents/Carrera/Semestre 6/Materias/Procesamiento/R/

MelanomaFeatures <- read.csv("/Users/joseantoniopalaciosdorbecker/Documents/Carrera/Semestre 6/Materias/Procesamiento/R/Evidencia1/MelanomaLesionFeatures.csv", header=FALSE)

MelanomaControlFeatures <-
read.csv("/Users/joseantoniopalaciosdorbecker/Documents/Carrera/Semestre 6/Materias/Procesamiento/R/Evidencia1/MelanomaControlFeatures.csv", header=FALSE)

SeborrheicFeatures <- read.csv("/Users/joseantoniopalaciosdorbecker/Documents/Carrera/Semestre 6/Materias/Procesamiento/R/Evidencia1/SeborrheicLesionFeatures.csv", header=FALSE)

SeborrheicControlFeatures <-
read.csv("/Users/joseantoniopalaciosdorbecker/Documents/Carrera/Semestre 6/Materias/Procesamiento/R/Evidencia1/SeborrheicControlFeatures.csv", header=FALSE)

NevusFeatures <- read.csv("/Users/joseantoniopalaciosdorbecker/Documents/Carrera/Semestre 6/Materias/Procesamiento/R/Evidencia1/NevusLesionFeatures.csv", header=FALSE)

NevusControlFeatures <- read.csv("/Users/joseantoniopalaciosdorbecker/Documents/Carrera/Semestre 6/Materias/Procesamiento/R/Evidencia1/NevusControlFeatures.csv", header=FALSE)

#LesionFeatureNames <- read_excel("FeatureNames.xlsx")
#ControlFeatureNames <- read_excel("FeatureNames.xlsx", sheet = "Control")

```

```

#FeatureNames <- read_excel("FeatureNames.xlsx",sheet = "PerChannel")

## Data Conditioning

{r FRESA Map, results = "hide", dpi=300, fig.height= 6.0, fig.width= 8.0}

op <- par(no.readonly = TRUE)

colnames(MelanomaControlFeatures) <- paste("C",colnames(MelanomaControlFeatures),sep="")
colnames(SeborrheicControlFeatures) <- paste("C",colnames(SeborrheicControlFeatures),sep="")

colnames(NevusControlFeatures) <- paste("C",colnames(NevusControlFeatures),sep="")

sum(is.na(MelanomaFeatures))

sum(is.na(SeborrheicFeatures))

sum(is.na(NevusFeatures))

sum(is.na(MelanomaControlFeatures))

sum(is.na(NevusControlFeatures))

channel1 <- c(1:32)

channel2 <- c(33:64)

channel3 <- c(65:96)

MMelanomaFeatures <- ( MelanomaFeatures[,channel1] +
  MelanomaFeatures[,channel2] +
  MelanomaFeatures[,channel3] )/3.0

SMelanomaFeatures <- ( abs(MelanomaFeatures[,channel1] - MMelanomaFeatures) +
  abs(MelanomaFeatures[,channel2] - MMelanomaFeatures) +
  abs(MelanomaFeatures[,channel3] - MMelanomaFeatures) )/3.0

colnames(MMelanomaFeatures) <- paste("M",colnames(MMelanomaFeatures),sep="")

colnames(SMelanomaFeatures) <- paste("S",colnames(SMelanomaFeatures),sep="")

MelanomaFeatures <-
cbind(MelanomaFeatures,SMelanomaFeatures/(0.01+abs(MMelanomaFeatures)))

MNevusFeatures <- ( NevusFeatures[,channel1] +
  NevusFeatures[,channel2] +
  NevusFeatures[,channel3] )/3.0

```

```

SNevasFeatures <- ( abs(NevusFeatures[,channel1] - MNevusFeatures) +
                      abs(NevusFeatures[,channel2] - MNevusFeatures) +
                      abs(NevusFeatures[,channel3] - MNevusFeatures) )/3.0

colnames(MNevusFeatures) <- paste("M",colnames(MNevusFeatures),sep="")
colnames(SNevusFeatures) <- paste("S",colnames(SNevusFeatures),sep="")

NevusFeatures <- cbind(NevusFeatures,SNevusFeatures/(0.001+abs(MNevusFeatures)))

MSeborrhicFeatures <- ( SeborrheicFeatures[,channel1] +
                           SeborrheicFeatures[,channel2] +
                           SeborrheicFeatures[,channel3] )/3.0

SSeborrhicFeatures <- ( abs(SeborrheicFeatures[,channel1] - MSeborrhicFeatures) +
                           abs(SeborrheicFeatures[,channel2] - MSeborrhicFeatures) +
                           abs(SeborrheicFeatures[,channel3] - MSeborrhicFeatures) )/3.0

colnames(MSeborrheicFeatures) <- paste("M",colnames(MSeborrheicFeatures),sep="")
colnames(SSeborrhicFeatures) <- paste("S",colnames(SSeborrhicFeatures),sep="")

SeborrheicFeatures <-
cbind(SeborrheicFeatures,SSeborrhicFeatures/(0.001+abs(MSeborrheicFeatures)))

MMelanomaControlFeatures <- ( MelanomaControlFeatures[,channel1] +
                               MelanomaControlFeatures[,channel2] +
                               MelanomaControlFeatures[,channel3] )/3.0

SMelanomaControlFeatures <- ( abs(MelanomaControlFeatures[,channel1] - MMelanomaControlFeatures) +
                               abs(MelanomaControlFeatures[,channel2] - MMelanomaControlFeatures) +
                               abs(MelanomaControlFeatures[,channel3] - MMelanomaControlFeatures) )/3.0

colnames(MMelanomaControlFeatures) <- paste("M",colnames(MMelanomaControlFeatures),sep="")
colnames(SMelanomaControlFeatures) <- paste("S",colnames(SMelanomaControlFeatures),sep="")

MelanomaControlFeatures <-
cbind(MelanomaControlFeatures,SMelanomaControlFeatures/(0.001+abs(MMelanomaControlFeatures)))

```

```

MNevusControlFeatures <- ( NevusControlFeatures[,channel1] +
                           NevusControlFeatures[,channel2] +
                           NevusControlFeatures[,channel3] )/3.0

SNevusControlFeatures <- (abs(NevusControlFeatures[,channel1] - MNevusControlFeatures) +
                           abs(NevusControlFeatures[,channel2] - MNevusControlFeatures) +
                           abs(NevusControlFeatures[,channel3] - MNevusControlFeatures))/3.0

colnames(MNevusControlFeatures) <- paste("M",colnames(MNevusControlFeatures),sep="")
colnames(SNevusControlFeatures) <- paste("S",colnames(SNevusControlFeatures),sep="")

NevusControlFeatures <-
cbind(NevusControlFeatures,SNevusControlFeatures/(0.001+abs(MNevusControlFeatures)))

MSeborrheicControlFeatures <- ( SeborrheicControlFeatures[,channel1] +
                           SeborrheicControlFeatures[,channel2] +
                           SeborrheicControlFeatures[,channel3] )/3.0

SSeborrheicControlFeatures <- (abs(SeborrheicControlFeatures[,channel1] - MSeborrheicControlFeatures) +
                           abs(SeborrheicControlFeatures[,channel2] - MSeborrheicControlFeatures) +
                           abs(SeborrheicControlFeatures[,channel3] - MSeborrheicControlFeatures))/3.0

colnames(MSeborrheicControlFeatures) <- paste("M",colnames(MSeborrheicControlFeatures),sep="")
colnames(SSeborrheicControlFeatures) <- paste("S",colnames(SSeborrheicControlFeatures),sep="")

SeborrheicControlFeatures <- cbind(SeborrheicControlFeatures,
                                    SSSeborrheicControlFeatures/(0.001+abs(MSeborrheicControlFeatures)))

CtrDiff <- MelanomaFeatures[,1:ncol(MelanomaControlFeatures)] - MelanomaControlFeatures;
colnames(CtrDiff) <- colnames(MelanomaControlFeatures)

MelanomaFeatures <- cbind(MelanomaFeatures,CtrDiff)

CtrDiff <- NevusFeatures[,1:ncol(MelanomaControlFeatures)] - NevusControlFeatures;
colnames(CtrDiff) <- colnames(NevusControlFeatures)

NevusFeatures <- cbind(NevusFeatures,CtrDiff)

CtrDiff <- SeborrheicFeatures[,1:ncol(MelanomaControlFeatures)] - SeborrheicControlFeatures;
colnames(CtrDiff) <- colnames(SeborrheicControlFeatures)

```

```

SeborrheicFeatures <- cbind(SeborrheicFeatures,CtrDiff)

MelanomaFeatures <- MelanomaFeatures[complete.cases(MelanomaFeatures),]

NevusFeatures <- NevusFeatures[complete.cases(NevusFeatures),]

SeborrheicFeatures <- SeborrheicFeatures[complete.cases(SeborrheicFeatures),]

MelanomaFeatures$Class <- rep(1,nrow(MelanomaFeatures))

NevusFeatures$Class <- rep(0,nrow(NevusFeatures))

SeborrheicFeatures$Class <- rep(0,nrow(SeborrheicFeatures))

MelanomaSeborrheicNevus <- rbind(MelanomaFeatures,SeborrheicFeatures,NevusFeatures)

table(MelanomaSeborrheicNevus$Class)

## The Heatmap

{r, dpi=300, fig.height= 6.0, fig.width= 8.0}

hm <- heatMaps(Outcome = "Class",

  data = MelanomaSeborrheicNevus[1:900,],

  title = "Heat Map:",Scale = TRUE,

  cexRow = 0.75,cexCol = 0.5,srtCol = 45)

par(op)

## Learning Melanoma with KNN

{r, , warning = FALSE}

cvKNN <- randomCV(MelanomaSeborrheicNevus,"Class",

  KNN_method,

  trainFraction = 0.70,

  repetitions = 10,

  classSamplingType = "Ba",

  featureSelectionFunction = univariate_KS,

  featureSelection.control = list(limit= 30), # que nos regrese los 30 mas significativos

  )

# si se entrena balanceado, el punto donde hay cancer o no es de 0.5

```

```

# el criterio de decisión del sistema depende de como fue entrenado el sistema

# es mejor entrenar con todos los datos

#MAD - mean absolute difference

#Area Under Curve (AUC) representa el intervalo de confianza

# BACC - Balance Accuracy : Sensibility + Specificity / 2, area bajo curva verda

## KNN plot performance

{r, results='asis', warning = FALSE, dpi=300, fig.height= 6.0, fig.width= 6.0}

performace <- predictionStats_binary(cvKNN$medianTest,"KNN")

par(op)

pander::pander(performace$aucs)

pander::pander(performace$accc)

pander::pander(performace$CM.analysis)

## SVM

{r}

cvSVM <- randomCV(MelanomaSeborrheicNevus,"Class",

e1071::svm,

asFactor = TRUE,

trainSampleSets= cvKNN$trainSamplesSets,

featureSelectionFunction = cvKNN$selectedFeaturesSet,

probability=TRUE

)

## SVM plot performance

{r , results='asis', dpi=300, fig.height= 6.0, fig.width= 6.0}

performace <- predictionStats_binary(cvSVM$medianTest,"SVM")

par(op)

```

```

pander::pander(performace$aucs)

pander::pander(performace$accc)

pander::pander(performace$CM.analysis)

# curva roc

# sensitivity: Ppred / Preal

# Especificidad = Npred / Neg_real

# circulo rojo es un predictor continuo, curva rock

## Learning Melanoma with Naive Bayes with PCA

{r}

cvNB <- randomCV(MelanomaSeborrheicNevus,"Class",

NAIVE_BAYES,

trainSampleSets= cvKNN$trainSamplesSets,

featureSelectionFunction = cvKNN$selectedFeaturesSet

)

## NB performance

{r, results='asis', dpi=300, fig.height= 6.0, fig.width= 6.0}

performace <- predictionStats_binary(cvNB$medianTest,"Naive Bayes with PCA")

par(op)

pander::pander(performace$aucs)

pander::pander(performace$accc)

pander::pander(performace$CM.analysis)

## Learning Melanoma with LASSO

{r , warning = FALSE}

cvLASSO <- randomCV(MelanomaSeborrheicNevus,"Class",

LASSO_MIN,

trainSampleSets= cvKNN$trainSamplesSets,

family="binomial")

```

```
## LASSO performance

{r , results='asis', warning = FALSE, dpi=300, fig.height= 6.0, fig.width= 6.0}

performace <- predictionStats_binary(cvLASSO$medianTest,"LASSO")

par(op)

pander::pander(performace$aucs)

pander::pander(performace$accc)

pander::pander(performace$CM.analysis)

{r, results='asis', warning = FALSE, dpi=300, fig.height= 4.0, fig.width= 7.0}

barplot(cvLASSO$featureFrequency,las=2)
```

GitHub Link

<https://github.com/luisgzza01571007/EQ6-Procesamiento-de-Imagenes-GPO602.git>