



UNA UNIVERSIDAD CON ALMA DE GUERRERO

PROPUESTA DE MÉTODO DE RECONOCIMIENTO DE IMÁGENES PARA LA IDENTIFICACIÓN DEL MELANOMA HUMANO

Tesis para optar por el Título de Ingeniero de Sistemas, que presentan los bachilleres.

AUTORES

Diego Alberto Campos Aquino
Luis Andre Nazareth Mundaca Arriola

ASESOR

Ing. Rosa América Cobeñas Sánchez

CHICLAYO - PERÚ 2016





UNA UNIVERSIDAD CON ALMA DE GUERRERO

PROPUESTA DE MÉTODO DE RECONOCIMIENTO DE IMÁGENES PARA LA IDENTIFICACIÓN DEL MELANOMA HUMANO

Tesis para optar por el Título de Ingeniero de Sistemas, que presentan los bachilleres.

AUTORES

Diego Alberto Campos Aquino
Luis Andre Nazareth Mundaca Arriola

ASESOR

Ing. Rosa América Cobeñas Sánchez

CHICLAYO - PERÚ 2016



Título de Tesis

**PROPUESTA DE MÉTODO DE RECONOCIMIENTO DE
IMÁGENES PARA LA IDENTIFICACIÓN DEL MELANOMA
HUMANO**

Aprobación del proyecto

Ing. Rosa América Cobeñas Sánchez
Asesor especialista

Dr. Jorge Gutiérrez Gutiérrez
Presidente del jurado de tesis

Ing. Rosa América Cobeñas Sánchez
Secretario del jurado de tesis

Ing. Heber Iván Mejía Cabrera
Vocal del jurado de tesis

Bach. Diego Alberto Campos Aquino
Autor

Bach. Luis Andre Nazareth Mundaca Arriola
Autor



DEDICATORIA

A mis padres quienes me dieron vida,
educación, apoyo, consejos y siempre
estuvieron brindándome su amor incondicional.

A mis compañeros de estudio, a mis docentes y
amigos, quienes ayudaron directa e
indirectamente para poder realizar esta tesis.

A todos ellos se los agradezco desde el fondo
de mi alma. Para todos ellos hago esta
dedicatoria.

Diego Campos Aquino

DEDICATORIA

A mis padres.

A mis padres por ser el pilar fundamental en todo lo que soy, en toda mi educación, tanto académica, como de la vida, por su incondicional apoyo perfectamente mantenido a través del tiempo.

Todo este trabajo ha sido posible gracias a ellos.

A mis maestros y amigos.

Por su gran apoyo y motivación para la culminación de nuestros estudios profesionales y para la elaboración de esta tesis

Luis Andre Mundaca Arriola



INDICE

RESUMEN.....	ix
ABSTRACT.....	x
I. PROBLEMA DE INVESTIGACIÓN	11
1.1 Situación Problemática.....	12
1.2. Formulación del problema.....	15
1.3. Delimitación de la Investigación	15
1.4. Justificación e Importancia.....	15
1.5. Limitaciones de la Investigación	16
1.6. Objetivos	16
II. MARCO TEORICO.....	17
2.1. Antecedentes de la Investigación	18
2.2. Estado del Arte.....	22
2.3. Bases Teórico científicas	25
III. MARCO METODOLOGICO.....	69
3.1. Tipo y diseño de la investigación:	70
3.1. Población y Muestra.....	70
3.2. Hipótesis	71
3.3. Variables	71
3.5. Operacionalización.....	71
3.6. Métodos, Técnicas e Instrumentos de Recolección de Datos	73
3.7. Procedimiento para la Recolección de Datos	74
IV. ANALISIS E INTERPRETACION DE LOS RESULTADOS	77
V. DESARROLLO DE LA PROPUESTA.....	96
VI. CONCLUSIONES	112
VII. ANEXOS	116
REFERENCIAS	141



INDICE DE FIGURAS

Figura 1 Diagrama de flujo del algoritmo propuesto.....	14
Figura 2 Diagrama de Flujo del sistema para detectar lesiones.....	19
Figura 3 Arquitectura del sistema de diagnóstico de cáncer de piel (Melanoma)	20
Figura 4 Diagrama de flujo del algoritmo.....	21
Figura 5 Melanoma Benigno a la izquierda y maligno a la derecha.	26
Figura 6 Melanoma Benigno a la izquierda y maligno a la derecha.	26
Figura 7 Melanoma Benigno a la izquierda y maligno a la derecha.	26
Figura 8 Melanoma Benigno a la izquierda y maligno a la derecha.	27
Figura 9 Melanoma Benigno a la izquierda y maligno a la derecha.	27
Figura 10 Conceptos relacionados con la calidad de imagen	30
Figura 11 Una montaña (a) y sus 64 contenedores del histograma nivel gris (b)	37
Figura 12 Implementación del Filtro Mediano.....	39
Figura 13 Típico proceso de análisis de una imagen	42
Figura 14 Resultado de Canny Edge detection para dos diferentes imágenes con valores altos y bajos: 50 y 10 respectivamente.	43
Figura 15 A la izquierda un botón y a la derecha su histograma.....	45
Figura 16 Resultado de reconocimiento de rostros producidos por Rowley,	47
Figura 17 Ejemplo de regiones de imagen correspondientes a (a) clase A y (b) clase B.	48
Figura 18 Gráfico de la media con respecto al valor de la desviación estándar para un número de diferentes imágenes procedentes de clase A (o) Y la clase B (+). En este caso, una línea recta separa las dos clases.....	49
Figura 19 Diagrama del Descriptor de Color	52
Figura 20 División de la imagen en regiones.....	53
Figura 21 Selección del Color más representativo de la imagen.....	54
Figura 22 Dominio espacial y frecuencial DCT.....	55
Figura 23 Exploración en Zigzag.....	56
Figura 24 Ejemplo de estimación de densidad (a) distancia dada entre un pixel a el centroide vecino. (b) Actualización de descriptor vectorial usando distancias de (a)	58
Figura 25 Ejemplo del algoritmo kNN.....	60
Figura 26 Red de dos capas	64
Figura 27 Ejemplo de Validación Cruzada (Cross-Validation).....	67
Figura 28 Grafico de Porcentaje de Aciertos y Errores por Clasificador	81
Figura 29 Grafico de Precisión Detallada por Clasificador	81
Figura 30 Grafico de Porcentaje de Aciertos y Errores por Clasificador	83
Figura 31 Grafico de Precisión Detallada por Clasificador	83
Figura 32 Grafico de Porcentaje de Aciertos y Errores por Clasificador	84
Figura 33 Grafico de Precisión Detallada por Clasificador	85



Figura 34 Grafico de Porcentaje de Aciertos y Errores por Clasificador	86
Figura 35 Grafico de Precisión Detallada por Clasificador	87
Figura 36 Grafico de acierto y errores por cada tipo de clasificador	89
Figura 37 Pruebas de efectividad del clasificador	90
Figura 38 Número de aciertos y errores por clasificador	92
Figura 39 Pruebas de efectividad por clasificador	92
Figura 40 Cuadro comparativo de métodos implementados	93
Figura 41 Diferencias entre Metodos Implementados	94
Figura 42 Flujo del Algoritmo de la investigación	97
Figura 43 Muestra de imagenes del dataset	98
Figura 44 A la izquierda la imagen de un melanoma, a la derecha la misma imagen, pero en escala de grises.....	99
Figura 45 A la izquierda la imagen de un no melanoma, a la derecha la misma imagen, pero en escala de grises.....	99
Figura 46 Estructura de la red neuronal empleada	106
Figura 47 Primer nodo de la capa Oculta (Nodo 2)	108
Figura 48 Segundo nodo de la capa Oculta (Nodo 3)	108
Figura 49 Tercer nodo de la capa Oculta (Nodo 4)	108
Figura 50 Cuarto nodo de la capa Oculta (Nodo 5).....	109
Figura 51 Quinto nodo de la capa Oculta (Nodo 6).....	109
Figura 52 Primer nodo de la capa de Salida (Nodo 0)	109
Figura 53 Segundo nodo de la capa de Salida (Nodo 1).....	110
Figura 54 Método de Validación Cruzada	110
Figura 55 Resultados de las pruebas con redes neuronales.....	111
Figura 56 Diagrama de clases del software implementado.....	111



RESUMEN

La presente investigación presenta un nuevo método para la identificación del melanoma humano usando técnicas de visión computacional; para la implementación se utilizó el lenguaje de programación C++; las características son basadas en GLCM (gray level Co-occurrence matrix) y como clasificador se usó el perceptrón multicapa (MLP), el porcentaje de precisión alcanzado es de 71.4286%, la validación se realizó usando el método cross-over con 10 iteraciones (10-folds).

La realidad problemática responde al hecho de que existen muchas investigaciones que presentan un método para utilizar para identificar el melanoma humano en imágenes haciendo uso de diversas técnicas, por lo que se presenta un nuevo método que funcione mejor bajo nuestra realidad.

PALABRAS CLAVE:

Inteligencia Artificial, Visión Computacional, Reconocimiento de Imágenes, melanoma, cáncer.



ABSTRACT

This research presents a novel method for the identification of human melanoma using computer vision techniques; to implement the C ++ programming language is used; the features are based on GLCM (gray level Co-Occurrence matrix) and classifier neural network type Multilayer Perceptron (MLP) was used, the percentage accuracy reached is 71.4286%, validation was performed using the cross-over method with 10 iterations (10-folds).

The problem actually responds to the fact that there are many investigations that present a method to use to identify the human melanoma on images using various techniques, so a new method that works best under our reality is presented in this document.

KEYWORDS:

Artificial Intelligence, Computer Vision, Image Recognition, Melanoma, Cancer.

I. PROBLEMA DE INVESTIGACIÓN

1.1 Situación Problemática.

El cáncer constituye una de las principales causas de muerte en el mundo. En los últimos años la carga de enfermedad por el cáncer se ha incrementado progresivamente con un mayor impacto en las poblaciones de menores recursos.

La piel humana es una superficie compleja, con geometría a fina escala, lo que hace su aspecto mucho más difícil de modelar, además, las condiciones de visión e iluminación afectan en gran medida su aspecto. La piel puede fácilmente ser examinada con los ojos; sin embargo, aspectos particulares de la piel son mejor evaluados por métodos de imagen no-invasivos.

El cáncer de piel es uno de los más comunes y uno de los tipos más mortales de cáncer entre la población de piel clara. Entre los diferentes tipos de cáncer de piel, melanoma es la forma más grave. Si se reconoce y trata temprano, es casi siempre curable, pero si no lo es, el cáncer puede avanzar y extenderse a otras partes del cuerpo, donde se vuelve difícil de tratar y puede ser fatal. Aunque es no es el más común de los cánceres de piel, es el que causa la mayor parte de las muertes, por esto es crucial identificarlo en una etapa temprana.

Características clínicas de lesiones pigmentadas que sugieren la presencia de melanoma son conocidas como el ABCD del melanoma; A de asimetría, B de bordes irregulares, C de variación de color, textura y D por el diámetro mayor a 6mm.



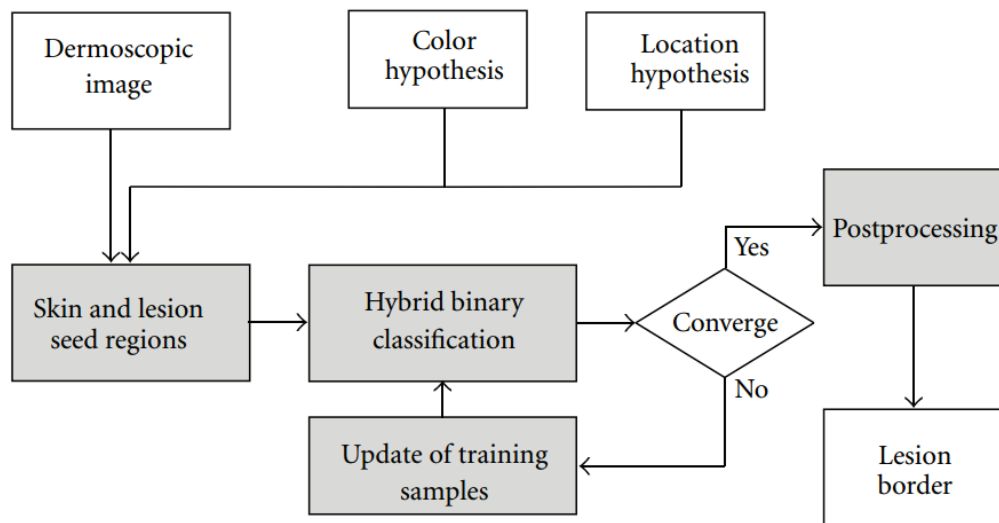
Los criterios ABCD desarrollados por la Sociedad Americana de Cáncer proporcionan un punto de partida para el médico y una guía fácil de recordar por el paciente para su uso en un auto-examen. El uso de estos criterios sirve para identificar el melanoma. (Razmjooy, Somayeh Mousavi, Soleymani, & Khotbesara, 2012)

Existen diferentes tipos de investigaciones sobre el uso de técnicas de visión computacional que tratan de automatizar la detección de cáncer de piel, se explicarán algunas a continuación:

En la investigación “**Automatic Registration Of Images Of Pigmented Skin Lesions**”, se describe un algoritmo para el registro de imágenes digitales de múltiples lesiones en la piel automáticamente. El algoritmo comprende cuatro etapas: (1) generar mapas de lesión; (2) identificar coincidencia inicial lesiones; (3) seleccionar cuatro coincidencias de lesiones angulares; (4) transformar una imagen. Las pruebas de rendimiento utilizando imágenes de ambas lesiones cutáneas simuladas y reales muestran que la adecuación y registro tienen éxito bajo imágenes en condiciones reales.

En la investigación “**Automatic Segmentation of Dermoscopic Images by Iterative Classification**”, se presenta un novedoso enfoque automático para la segmentación de las lesiones de la piel que es particularmente adecuado para el análisis de imágenes dermatoscópicas, Se usan las regiones de semillas como muestra de formación inicial, y para la segmentación de la lesión se utiliza la estrategia de clasificación híbrida (lineal y cuadrática).

Figura 1 Diagrama de flujo del algoritmo propuesto



Fuente: (Zortea M., Olav S., Schopf T., Kirchesh H., Godtliebsen F., 2011)

En la investigación “**Border Detection of Skin Cancer Cells with Fractal Dimension**”, Se propone un medidor de distancia radial y compacidad de las células de la frontera con la irregularidad de las lesiones pigmentadas de la piel que se basan en el potencial de la célula. El estudio del melanoma humano se lleva a cabo en un ambiente controlado, utilizando la geometría fractal como el método de análisis.

Se propone usar los métodos de método Box-conteo (DB) y el método de la salchicha (DS), ya que los datos experimentales demuestran que toda la población no solo posee dimensión fractal sino que también revelan diferentes bordes irregulares.

Las técnicas explicadas anteriormente son algunas de varias investigaciones realizadas, se sabe que estas tienen un buen funcionamiento en la detección del melanoma, por lo que es difícil saber cuál escoger.



Ante esta situación planteamos hacer una comparativa de las técnicas usadas para la detección de melanomas y así identificar cual es el que mayor efectividad tiene en la detección de melanomas en imágenes.

1.2. Formulación del problema

¿Qué método de reconocimiento de imágenes permiten la detección del melanoma humano de forma más efectiva?

1.3. Delimitación de la Investigación

El presente proyecto limita su aplicación al uso de imágenes de un dataset proporcionado por www.dermquest.com (página conformada por un grupo de dermatólogos internacionales de gran prestigio con intereses en la investigación académica) que contengan imágenes de melanomas y no melanomas; así como las técnicas a comparar serán recolectadas de papers de investigación relacionadas a la detección de melanoma humano en imágenes.

1.4. Justificación e Importancia

En el ámbito médico se presenta un esfuerzo para la detección temprana del melanoma humano, aumentando la posibilidad de obtener un diagnóstico rápido y con una efectividad aceptable.

En el ámbito tecnológico podría ayudar a sentar las bases para conocer que técnicas y métodos de visión computacional permiten la detección de enfermedades de la piel humana.

De esta manera, se estaría contribuyendo con el desarrollo de sistemas inteligentes que servirían de apoyo en la medicina para diagnosticar zonas cutáneas que posiblemente sean melanomas.

1.5. Limitaciones de la Investigación

Esta investigación se ve limitada al uso de 206 imágenes descargadas de una base de datos de imágenes médicas, dado que no se ha tenido acceso imágenes de casos de melanoma humano de nuestra comunidad, de dicho número de imágenes, han sido retiradas aquellas que no poseen las condiciones adecuadas para su análisis, reduciendo el número total a 189 imágenes; además cabe decir que el tamaño de resolución de las mismas varía en cada una de las imágenes.

Las técnicas que se compararon, fueron desarrolladas en OpenCV, durante el cual se ha encontrado que los resultados obtenidos son muy diferentes a aquellas que son obtenidos utilizando otras herramientas (MatLab).

1.6. Objetivos

1.6.1. Objetivo General

Formular un método utilizando las técnicas de reconocimiento de imágenes con mayor efectividad en la detección de melanomas humanos

Objetivos Específico

- a. Identificar las características que permiten detectar el melanoma humano.
- b. Analizar técnicas de reconocimiento de imágenes.
- c. Implementar técnicas de reconocimiento de imágenes.
- d. Identificar de acuerdo a los resultados, las técnicas con mayor efectividad.
- e. Integrar las técnicas en un método.

II. MARCO TEORICO

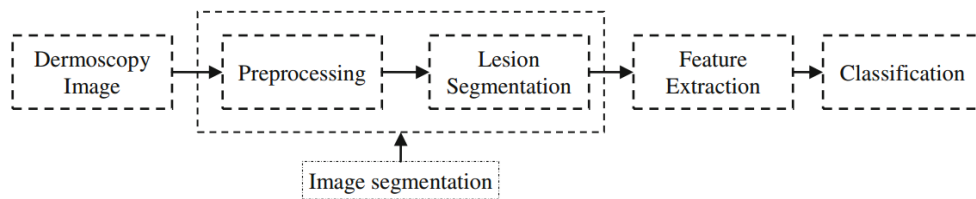
En esta etapa se recopiló y analizó información que sirve como base para el desarrollo del proyecto que deseamos elaborar. Nos ayuda a conocer la problemática a la que nos enfrentamos al realizar nuestra investigación y analizar las posibles soluciones que se plantean para este proyecto. Los artículos presentados en este capítulo son muy importantes ya que muestran un panorama muy amplio sobre las técnicas de visión computacional empleadas en el reconocimiento del melanoma humano en imágenes.

2.1. Antecedentes de la Investigación

N. Razmjoo, B. Somayeh Mousavi, Fazlollah Soleymani y M. Hosseini Khotbesara, “A computer aided diagnosis system for malignant melanomas”. El objetivo de esta investigación es brindar un eficiente método para la segmentación de melanomas malignos. Este método primero elimina los vellos cutáneos y escalas usando un detector de bordes. También se usan operaciones morfológicas para centrarse en un área de la imagen donde haya una existencia potencial de bordes de melanoma, luego se usan las distribuciones de textura con una nueva característica conocida como AIBQ para proporcionar una buena discriminación en las lesiones de la piel para la extracción de características, las características seleccionadas se aplican a un soporte de máquinas de vectores para clasificar los melanomas malignos y benignos, se obtuvo el 95% de precisión, 3% de falsos positivos y 2% de falsos rechazados.



Figura 2 Diagrama de Flujo del sistema para detectar lesiones



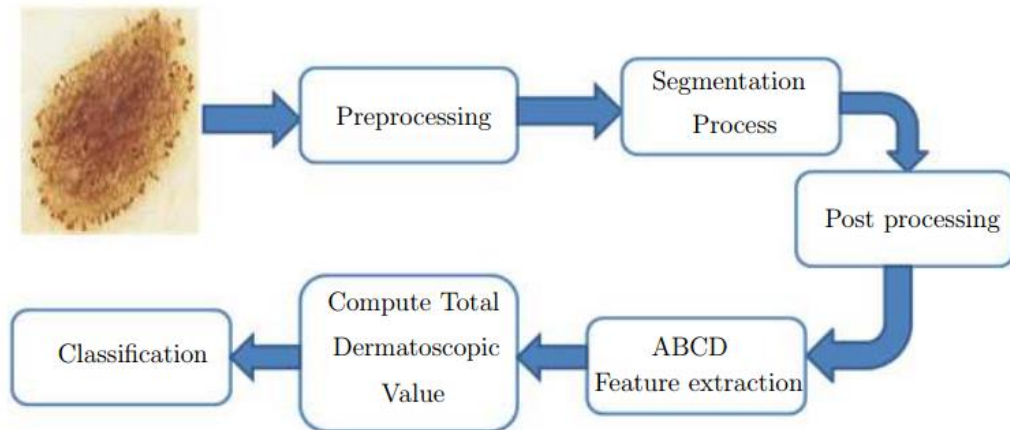
Fuente: (Razmjooy N., Somayeh B., Soleymani F., Hosseini M., 2012)

Esta investigación sólo presenta un nuevo método de segmentación de imágenes para la ayuda de detección de melanoma, a diferencia de la presente investigación que clasificará los resultados de los descriptores extraídos.

Nadia Smaoui y Souhir Bessassi, “A developed system for melanoma diagnosis”. Se busca desarrollar una interfaz que ayude a los dermatólogos en la fase de diagnóstico de melanomas, este trabajo se basa en una combinación de un método de segmentación y un método de análisis y su objetivo es mejorar estos dos métodos con el fin de desarrollar una interfaz que puede ayudar a los dermatólogos en la fase de diagnóstico. Como un primer paso, una secuencia de preprocesamiento se implementa para eliminar el ruido y las estructuras no deseadas de la imagen. Entonces, un enfoque automático de segmentación localiza la lesión de la piel. El siguiente paso es la extracción de características seguido de la regla ABCD para hacer el diagnóstico a través del cálculo de la puntuación TDV. En esta investigación, los tres diagnósticos que se utilizan son melanoma, sospechoso, y lesión cutánea benigna. El experimento utiliza 40 imágenes

que contienen cáncer de piel melanoma sospechoso. Basado en el experimento, la precisión del sistema es 92%, lo que refleja su viabilidad.

Figura 3 Arquitectura del sistema de diagnóstico de cáncer de piel (Melanoma)



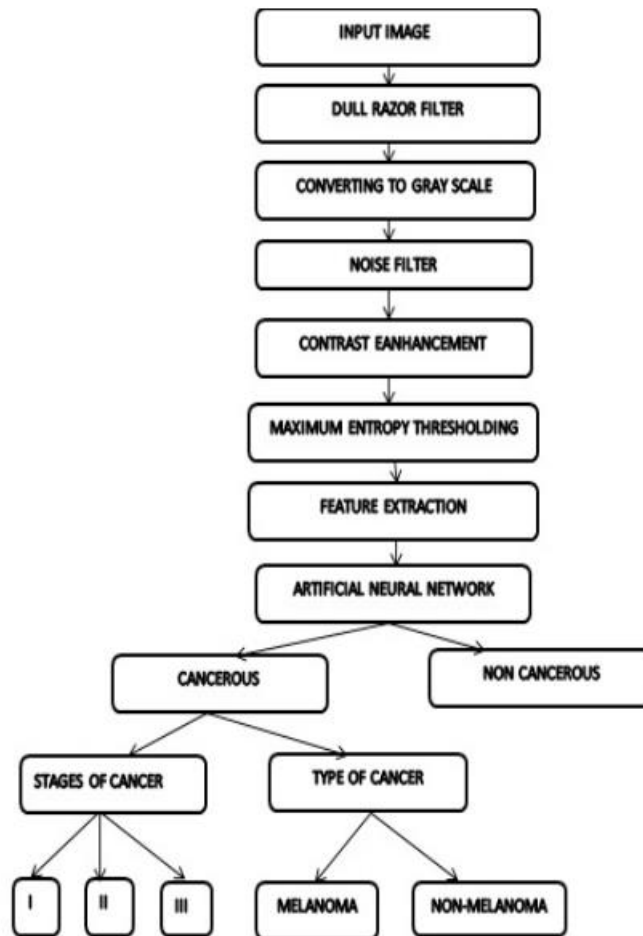
Fuente: (Smaoui N., Bessassi S., 2013)

A diferencia de la presente investigación, Nadia Smaoui y Souhir Bessassi, realizan pre-procesamiento en sus imágenes a clasificar.

Sarika Choudhari, Seema Biday, “**Artificial neural network for skin cancer detection**”, Se propone una modalidad de detección de melanomas humanos basada en redes neuronales, hacen uso de DULLRAZOR para la eliminación de ruido en la imagen, específicamente de los vellos, luego hacen el proceso de segmentación de la imagen, una vez hecho esto se hace una extracción de características usando GLCM y la clasificación usando redes neuronales, esto clasifica los datos dados en cancerosos y no cancerosos, los cancerosos a la vez son clasificados como melanoma y no melanoma.

Se hizo las pruebas con 30 imágenes y se obtuvo el 86.66% de precisión.

Figura 4 Diagrama de flujo del algoritmo



Fuente: (Choudhari S., Biday S., 2012)

Esta investigación al igual que la presente, muestra un método de reconocimiento para la detección de melanoma en imágenes; pero con la diferencia que pretende indicar el estado del cáncer (Fase I, II o III) en sus resultados obtenidos.

Todas estas investigaciones nos ayudan a tener una visión general sobre los avances que existentes en el reconocimiento de melanoma humano automatizado.



2.2. Estado del Arte

Durante los años 90 **Raymond G. White, Douglas A. Perednia y Robert A. Schowengerdt** realizaron la investigación titulada “**Automated Feature Detection In Digital Images Of Skin**” en la cual se presenta un nuevo enfoque para la detección de características cutáneas, centrándose en la detección de lesiones pigmentadas en imágenes digitales. Para calcular el área y el perímetro de las lesiones utilizaron el filtro Laplacian-of-Gaussian (LOG) junto a un esquema de interpolación de subpixel que, aunque es eficaz, resulta muy costoso en términos de tiempo y recursos computacionales. Se pudo clasificar las lesiones con sensibilidades medias en exceso de 75%, lo que significa que el valor predictivo positivo es de más del 60% utilizando el algoritmo K-NN.

Del mismo modo, con el uso de Fisher de MDA se pudo alcanzar sensibilidades medias de más de 95% y la media de valor predictivo positivo de más de 35%. La precisión puede ser mejorada mediante el control cuidadoso de la iluminación y la formación de un algoritmo de clasificación separado para los diferentes tipos de piel.

Luego en 1997 **L. Xu, M. Jackowski, A. Goshtasby, D. Roseman, S.Bines, C. Yu, A. Dhawan, A. Huntley** desarrollaron la investigación “**Segmentation Of Skin Cancer Images**” donde se presenta un método de 3 pasos para la segmentación de imágenes de cáncer de piel y otras lesiones pigmentadas. El primer paso es el preprocesamiento: primero una imagen de color es transformada en una imagen de intensidad de tal manera que la intensidad en un píxel muestra la distancia color de ese



pixel con el color del fondo. El color del fondo es tomado para ser el color promedio de píxeles en las pequeñas ventanas en las cuatro esquinas de la imagen; el segundo paso es denominada la segmentación inicial: el valor de umbral se determina a partir de la intensidad media de píxeles de alto gradiente en la imagen de intensidad obtenido. Este valor de umbral se utiliza para encontrar límites aproximados en la lesión; el tercer paso llamado Refinamiento de Region: una región límite es refinada usando la información del borde de la imagen. Esto implica la inicialización de una curva elástica cerrada en un límite aproximado, y la reducción y la ampliación de ésta para encajar en los bordes en sus vecinos.

Más adelante, específicamente en 2006 **Ilias Maglogiannis, Dimitrios I. Kosmopoulos** desarrollaron la investigación titulada **Computational vision systems for the detection of malignant melanoma** donde se explica que para la detección de melanomas es necesario tomar en cuenta las características de asimetría, bordes y color. Agruparon las imágenes de melanoma en 3 clases, VGP-Vertical Growth Phase, RGP-Radial Growth Phase, DSP-Dysplastic; sobre ellos utilizaron redes neuronales y Maquinas de Soporte Vectorial (SVMs) llegando a la conclusión de que las SVM son las que mejor desempeño tuvieron en la detección de melanoma maligno.

En la 2008, **Rajvi Parikh, Dr Hitesh Shah**, autores de la investigación titulada **A Survey on Computer Vision Based Diagnosis for Skin Lesion Detection** presentaron su estudio en el cual explican que, el primer paso para la detección de melanoma en imágenes es la eliminación



de ruido, estos autores utilizaron el software Dullrazor para realizar esta acción; el paso siguiente es la extracción de características, para detectar la asimetría usaron una ecuación para calcular el índice de asimetría, para detectar la irregularidad de bordes calcularon el Índice de Compacidad, Dimensión Fractal, Borde Asimetría de Borde y Transición de Pigmentación, para detectar la variación de colores se propone calcular la homogeneidad de colores, finalmente de acuerdo a los valores anteriormente obtenidos se realiza un promedio el cual indicará si es una herida simple, sospecho de melanoma o melanoma declarado.

En el 2011 **A. HATZIGIDAS, A. PAPASTERGIU, P. TZEKIS** presentaron la investigación **A Novel Algorithm For The Detection Of Melanoma Border** en la cual se presenta un simple, y muy rápido algoritmo que tiene la capacidad de detectar de manera eficaz y automáticamente el borde de un posible melanoma. El primer paso es detectar los bordes de la lesión guiándose del color base de la imagen; el segundo paso es ubicar un punto en centro de la lesión apoyándose en el método Monte Carlo (ingresando los valores anteriormente obtenidos); el tercer paso es ubicar un punto más oscuro que el color base de la imagen en el borde de la lesión empezando a partir del punto anteriormente calculado; el cuarto y final paso es dibujar el borde del melanoma.

Dados los avances de la tecnología móvil, en el 2013 **Teresa Margarita Puebla Iglesias, Alejandro Vladimir Bergues Balart, Romina Robert Pérez y Dionis López Ramos** presentaron su investigación **DIAGNÓSTICO DEL MELANOMA MALIGNO DESDE TELÉFONOS**



INTELIGENTES Y TABLETAS donde explican su método de detección de melanoma en imágenes; en primer lugar se realiza el pre-procesamiento y segmentación de la lesión; luego toca el momento de extraer las características guiándose del nemotécnico ABCD (asimetría, bordes, color y diámetro); finalmente para dar el diagnóstico de la lesión utilizaron el algoritmo KNN (el de los 5 vecinos).

2.3. Bases Teórico científicas

Melanoma

Concepto

(Foundation, The Skin Cancer, s.f.) La forma de cáncer de piel más peligrosa. En fase avanzada, puede ser una enfermedad grave; si se extiende a los órganos internos, puede poner en peligro la vida. Afortunadamente, el melanoma raras veces ataca sin avisar. Hay que estar alerta porque si se detecta pronto, casi siempre tiene cura.

Los lunares, las manchas marrones y los crecimientos de la piel suelen ser, aunque no siempre, benignos. Si alguien tiene más de 100 lunares corre un máximo riesgo de sufrir melanoma. Los primeros signos pueden aparecer en uno o más de estos lunares. Por eso es tan importante conocer muy bien nuestra propia piel. Examinar la piel de todo el cuerpo una vez al mes y que un doctor realice un chequeo médico una vez al año.

Reconocimiento de Melanomas

(Foundation, The Skin Cancer, s.f.) Las primeras cinco letras del alfabeto son una guía que nos avisa de las primeras señales del melanoma.

Asimetría (A): A diferencia del melanoma, este lunar benigno es simétrico.

Si dibujamos una línea en el centro, los dos lados coinciden.



Si se dibuja una línea sobre este lunar, las dos mitades no coinciden, lo cual quiere decir que son asimétricas, una señal de alarma para el melanoma.

Figura 5 Melanoma Benigno a la izquierda y maligno a la derecha.



Imagen recuperada de <http://cancerdepiel.org>

Borde (**B**): Un lunar benigno presenta bordes iguales y regulares, a diferencia del melanoma.

Los bordes de un melanoma precoz suelen no ser iguales. Los extremos pueden estar ondulados o ser irregulares.

Figura 6 Melanoma Benigno a la izquierda y maligno a la derecha.



Imagen recuperada de <http://cancerdepiel.org>

Color (**C**): La mayoría de lunares benignos suelen ser de un único color, en concreto, de una única tonalidad de marrón.

Muchos colores distintos es también otra señal de alerta. Puede haber distintas tonalidades de marrón, color tostado o negro. Un melanoma también puede ser de color rojo, blanco y azul.

Figura 7 Melanoma Benigno a la izquierda y maligno a la derecha.



Imagen recuperada de <http://cancerdepiel.org>

Diámetro (D): Los lunares benignos suelen tener un diámetro más pequeño que los malignos.

Los melanomas suelen tener un diámetro más grande que la goma de un lápiz (1/4 pulgada o 6mm) aunque, cuando se detectan por primera vez, suelen ser más pequeños.

Figura 8 Melanoma Benigno a la izquierda y maligno a la derecha.



Imagen recuperada de <http://cancerdepiel.org>

Evolución (E): Los lunares benignos comunes permanecen invariables con el paso del tiempo. Permanezca alerta si un lunar empieza a cambiar o evolucionar de una u otra forma.

Figura 9 Melanoma Benigno a la izquierda y maligno a la derecha.



Imagen recuperada de <http://cancerdepiel.org>

Imagen

Concepto

(Montabone, 2010) Es la representación visual de un objeto, una persona, o escena producida por un dispositivo óptico tal como un espejo, lentes o una cámara. Esta representación es bidimensional (2D), aunque corresponde a una de las infinitamente muchas proyecciones del mundo real, objeto tridimensional o escena.

Resolución

(Montabone, 2010) La resolución de una imagen es el número de píxeles que van a ser usados para representar la imagen, por ejemplo si la resolución de una imagen es $2,816 \times 2,112$, esto equivale a decir el ancho de la imagen es 2,816 y la altura es 2,112 píxeles, también se podría decir que la resolución de la imagen es de 6 megapíxeles (MP), ya que $2,816 \times 2,112$ es 5.947.392, que es aproximadamente 6 millones de píxeles, o 6MP.

En general, mientras una imagen tenga más cantidad de píxeles se verá mejor, esto es hasta cierto límite donde no hay ganancia de calidad perceptible por el ojo humano. Este límite depende del uso de la imagen.

Color

(Szeliski, 2010) El color es la longitud de onda de la radiación electromagnética entrante. Como se puede ver en el diagrama, la longitud de onda de la radiación electromagnética tiene un amplio espectro de



valores continuos, lo que significa que hay un número infinito de posibles colores en la naturaleza.

El color es muy importante para las imágenes digitales. En un entorno digital, se utilizan un número específico de bits para representar la información. Esto significa que no podemos almacenar un número infinito de colores. Se necesita asignar un cierto número de bits que se utiliza para representar el color, este número se llama profundidad de color.

Profundidad de Color

(Montabone, 2010) La profundidad de color es el número de bits que se utilizan para representar el color de cada píxel en una imagen digital. Cuantos más bits usados, más colores únicos hay en la imagen.

Cuando más colores únicos son usados, la imagen se ve más natural para el ojo humano, hasta el punto de no notar ninguna diferencia al incrementar la profundidad de color.

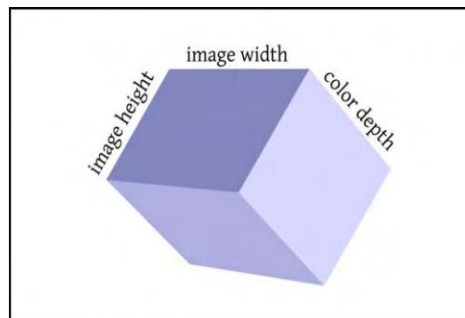
Si se utiliza la profundidad de 1 bit, la imagen solo tendrá 2 colores posibles. Estos 2 colores pueden ser en realidad cualquiera de los colores disponibles para que el ordenador pueda reproducir, pero por lo general en negro y el blanco se utiliza en este escenario.

Si la profundidad se incrementa a 8 bits, se podrá utilizar un máximo de 256 diferentes colores, que se utiliza normalmente para la representación de imágenes en escala de grises o de un conjunto reducido de colores.

El uso de 24 bits de profundidad, le da más de 16 millones de colores únicos. Cuando el aumento de la profundidad de color a más de 24 bits;

la mayoría de los humanos no pueden diferenciar de la original de 24 bits, es por esto que esta profundidad de color a menudo se llama truecolor. Se puede pensar en la profundidad de color como en un componente más de la calidad de la imagen, junto con la altura y ancho de la imagen.

Figura 10 Conceptos relacionados con la calidad de imagen



Fuente: (Montabone, 2010)

Espacios de Color

(Montabone, 2010) Los seres humanos tienen la visión tricromática, es decir que en términos simples significa que la visión humana se basa en tres receptores, uno para el rojo, otro para el verde y otro para el azul. De la misma manera hay muchos espacios de color diferentes, o modelos, que por lo general tienen tres o cuatro canales.

RGB

(Montabone, 2010) Uno de los espacios de color más utilizados es el RGB, este tiene 3 canales, el rojo, el azul y el verde. Cada espacio de color se crea usando una combinación de estos tres.

La combinación de colores se hace en base a la mezcla aditiva de color, que es cómo funciona la luz. Comenzando con una superficie de color

negro, la luz se proyecta sobre ella, obteniendo así diferentes colores. Por ejemplo, si usted tiene y se les proyecta con la misma intensidad o el brillo en la misma superficie de color negro, la luz reflejada aparecerá como amarillo.

En el caso de una imagen digital en el espacio de color RGB, cada píxel se define como (r, g, b) donde r , g , y b son los valores de cada canal. Los valores posibles para cada canal van desde 0 (sin luz con ese color) al número de valores posibles para el canal menos uno. Por ejemplo, si la profundidad de color es de 24 bits, entonces cada uno de los tres canales tiene 8 bits de profundidad, por lo tanto, cada canal tiene valores que van desde 0 hasta 255.

En ese escenario, un color rojo puro está representado por $(255, 0, 0)$, ya que necesita la máxima intensidad en el canal rojo, y no hay luz de los canales verdes o azules. De la misma manera, un puro amarillo está representado por $(255, 255, 0)$ porque el rojo y el verde sumados dan como resultado el amarillo.

HSV

(Montabone, 2010) HSV es el estándar para hue, saturación y valor. En este color de espacio, el hue representa el color primario, la saturación y valor representa la sombra específica para esa tonalidad. El uso principal de este espacio de color para la creación de una mejor interacción con los seres humanos, debido a nuestra tendencia habitual a reconocer diferentes colores (HUE) en diferentes sombras (saturación y valor). Es más natural para los seres humanos seleccionar primero un color sólido y



luego seleccionar un tono específico para ello, como la luz roja o azul oscuro, por ejemplo, que la selección de un valor específico de RGB.

YUV

(Montabone, 2010) Espacio de color YUV se compone del canal luma (Y), que es el brillo de la imagen, y dos canales de crominancia (U y V) que describen el color específico de la imagen. Se ha utilizado principalmente para la codificación de vídeo analógica y la televisión.

Formatos de Imagen

(Montabone, 2010) Existen diferentes formatos para el almacenamiento de los píxeles de una imagen, porque se va a modificar el contenido de los píxeles, es muy importante saber que formato de archivo se utilizará, ya que algunos formatos generan pérdida de información, y otros no.

Formatos con pérdida comprimen la imagen original, la eliminación de ciertos detalles de la misma manera que todavía se vea similar a la vista del ojo humano, pero con un tamaño de archivo más pequeño. Esto significa que algunos datos se pierden en el proceso.

Lo bueno de ellos es que los tamaños de archivo se pueden volver muy pequeña. Lo malo es que, en algunos casos, la imagen pierde demasiada información y no se ve tan bueno como el original. Además, cada vez que se guarda en un formato con pérdida, se repite el proceso, degradando su imagen aún más.

Formatos sin pérdida, por otro lado, codifican la imagen original en una manera de modo que no pierda ningún detalle de la imagen original.



Aun así, pueden comprimir las imágenes, pero no tanto como los formatos con pérdida. Lo bueno de estos formatos es que siempre conservan la calidad total de su imagen original.

Lo malo es que pueden ocupar más espacio en el disco que los que con pérdidas.

JPG

(Montabone, 2010) Es el formato más común en las imágenes digitales, esto soporta la profundidad de color de 24 bits (tres canales de color de 8 bits cada uno).

Casi todas las cámaras digitales son capaces de escribir utilizando este formato, y es ampliamente apoyado en diferentes programas de visualización.

Este produce archivos con tamaño pequeño utilizando la compresión con pérdida. Cada vez que se guarda una imagen en este formato, la calidad se degrada debido a que se realiza una compresión con pérdida.

Este formato también es compatible con una función llamada JPEG progresivo. Esta característica le permite guardar la imagen de forma inteligente para que la información sea ordenada de una manera específica para que diferentes resoluciones de la imagen se estén presentando al usuario hasta que se reciba la imagen completa. Esto es útil para imágenes de gran tamaño que se transfieren a las computadoras a través de una conexión lenta.

GIF



(Montabone, 2010) GIF significa Graphics Interchange Format. Es capaz de almacenar una profundidad de color de los únicos 8 bits (256 colores).

Se ocupa de la transparencia en una forma muy básica (cada píxel es o bien transparente o no). La principal ventaja de este formato de archivo es que puede ser utilizado para guardar animaciones. También, debido a que es un formato antiguo, es muy compatible con las versiones antiguas de navegadores web y otros programas. Utiliza la compresión sin pérdidas.

Este formato se ha utilizado sobre todo para la visualización de gráficos y animaciones en páginas web. Debido a la limitación en el número de colores, no se recomienda el uso de este formato para almacenar fotografías digitales.

PNG

(Montabone, 2010) PNG significa Portable Network Graphics. Es una versión públicamente disponible y utilizable del formato GIF. Puede manejar una profundidad de color de hasta 48 bits (3 canales de color de 16 bits cada uno).

Tiene mucho mejor soporte para la transparencia de GIF; que le permite tener diferentes niveles de transparencia de cada píxel (canal alfa). El soporte para este formato comenzó a crecer desde su origen en 1995, y hoy es ampliamente apoyado. En general, PNG es un formato mejor que GIF; que ofrece una mejor



compresión, se puede almacenar más colores, tiene un mejor apoyo de la transparencia, y así sucesivamente.

TIFF

(Montabone, 2010) TIFF es sinónimo de Tagged Image File Format. Es uno de los formatos de imagen más flexibles. Permite ahorrar más sólo una imagen por archivo, lo que explica por qué es el formato estándar para la digitalización de documentos.

Procesamiento de Imágenes

Operadores de Puntos

(Szeliski, 2010) Los tipos más simples de las transformaciones de procesamiento de imagen son los operadores de puntos, donde el valor de cada píxel de salida depende de sólo el valor correspondiente píxel de entrada (además de, potencialmente, alguna información recopilada globalmente o parámetros). Ejemplos de tales operadores incluyen brillo y ajustes de contraste, así como corrección de color y transformaciones.

Transformación de Pixel

(Szeliski, 2010) Un operador general de procesamiento de imágenes es una función que toma una o más imágenes de entrada y produce una imagen de salida. En el dominio continuo, esto se puede denotar como

$$g(x) = h(f(x)) \quad o \quad g(x) = h(f_0(X), \dots, f_n(X)),$$

Donde x está en el dominio D -dimensional de las funciones (por lo general $D = 2$ para las imágenes) y las funciones f y g operan sobre algún rango, que puede ser escalar o valor vectorial, por ejemplo, para imágenes en color o movimiento 2D.



Transformaciones de Color

(Szeliski, 2010) Mientras que las imágenes en color se pueden tratar como funciones arbitrarias de valores vectoriales o colecciones de bandas independientes, por lo general tiene sentido pensar en ellos como señales altamente correlacionadas con fuertes conexiones con el proceso de formación de la imagen, diseño del sensor y la percepción humana.

Ecualización de Histograma

(WU, ZHOU, & ZHANG, 2010) Es un hecho que el histograma de una imagen de intensidad se encuentra dentro de un rango de datos limitada. Esas imágenes tienen generalmente plano negro o blanco y el fondo.

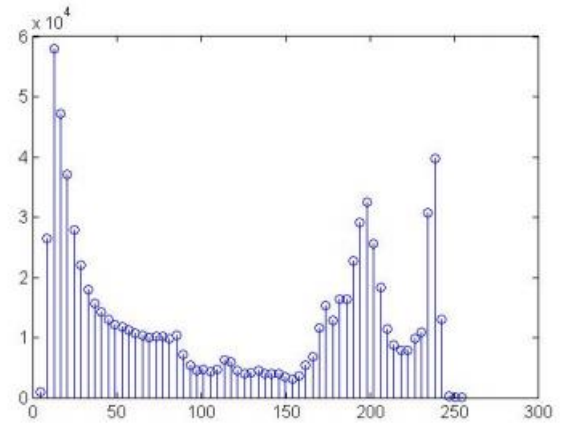
Figura 11 Una montaña (a) y sus 64 contenedores del histograma nivel gris (b) muestra un ejemplo de imagen cuya distribución de intensidad es blanco o negro. Desde Figura 11 Una montaña (a) y sus 64 contenedores del histograma nivel gris (b), podemos ver que una porción muy grande de píxeles cuya intensidad se basa en el rango $[0,50]$ o $[180,255]$. Una porción muy pequeña de píxeles reside en el rango de $[50,180]$. Esto hace que algunos detalles de la imagen apenas visibles, por ejemplo, los árboles en las montañas en la imagen mostrada en Figura 11 Una montaña (a) y sus 64 contenedores del histograma nivel gris (b). Este problema puede resolverse mediante una técnica de histograma de estiramiento llamado ecualización de histograma.



Figura 11 Una montaña (a) y sus 64 contenedores del histograma nivel gris (b)



(a)



(b)

Fuente: (WU, ZHOU, & ZHANG, 2010)

La idea básica de ecualización de histograma es encontrar la transformada de intensidad de tal manera que el histograma de la imagen transformada es uniforme. De las teorías probabilísticas existentes, esto existe como la transformada de intensidad. Supongamos que tenemos una imagen $f(x, y)$, y su histograma $h(i)$, que tienen la función acumulativa de $h(i)$ de la siguiente manera:

$$c(i) = \int_0^i h(t) dt$$

Se puede demostrar que una transformada con variable $y = c(i)$ siguen una distribución uniforme. Por lo tanto, para una imagen de nivel de gris 256, la ecualización de histograma se puede realizar por la siguiente ecuación:



$$t = \frac{256}{n} * c(f(x, y))$$

Donde n es el número total de píxeles en la imagen.

Operaciones básicas de Filtros de imagen

Umbralización

También conocida como thresholding consiste en eliminar los valores superiores o inferiores (poniéndolos a cero) respecto a un valor conocido como umbral.

$$IMG(i, j) = IMA(i, j) * (IMA(i, j) \geq umbral)$$

Binarización

Es una variante de la umbralización que consiste en dejar a 0 todos los píxeles menores de un umbral y a uno aquellos que son iguales o mayores, quedando constituida la imagen final por un conjunto de unos y ceros.

$$IMG(i, j) = (IMA(i, j) \geq umbral)$$

Filtrado

Existen diversos tipos de ruido que se producen por múltiples circunstancias: los sensores, ruidos eléctricos, perturbaciones en el medio de transmisión, efectos térmicos, campos electromagnéticos, etc.

Filtro de Media o de Suavizado

(González Marcos, y otros, 2006) Consiste en sustituir cada píxel por la media aritmética de los puntos que tienen al alrededor (incluido él mismo).



Se toma una ventana de $N \times N$ puntos, donde el punto a sustituir es el central, se obtiene la media de la suma de todos los valores de los píxeles de la ventana y se sustituye el píxel en la nueva imagen por el valor obtenido.

$$IMB(i, j) = \frac{\sum_{m=-\frac{N}{2}}^{\frac{N}{2}} \sum_{n=-\frac{N}{2}}^{\frac{N}{2}} IMA(i + m, j + n)}{N \cdot N}$$

Filtro de Mediana

(Davies, 2012) La idea aquí es localizar los píxeles en la imagen que tienen valores extremos y por lo tanto, intensidades altamente improbables y hacer caso omiso de sus intensidades reales, sustituyéndolos por los valores más adecuados. Esto es similar a dibujar un gráfico a través de un conjunto de parcelas e ir ignorando las parcelas que, evidentemente, están muy lejos de la mejor forma curva. Una manera obvia de conseguir esto es aplicar un filtro "límite" que previene cualquier píxel que tiene una intensidad fuera del rango de intensidad de sus vecinos.

Figura 12 Implementación del Filtro Mediano



```

for (i = 0; i <= 255; i++) hist[i] = 0;
for all pixels in image do {
    for (m = 0; m <= 8; m++) hist[ P[m] ]++;
    i = 0; sum = 0;
    while (sum < 5) {
        sum = sum + hist[i];
        i = i + 1;
    }
    Q0 = i - 1;
    for (m = 0; m <= 8; m++) hist[ P[m] ] = 0;
}
    
```

Fuente: (Davies, 2012)

(Burger & J. Burge, 2009) Este filtro, reemplaza cualquier valor de pixel por la mediana de los pixeles en la región correspondiente a filtrar R ,

$$I'(u, v) \leftarrow \text{median}\{I(u + i, v + j) | (i, j) \in R\}$$

Transformaciones Morfológicas

Dilatación Binaria

(Davies, 2012) Esta operación consiste en la convolución de una imagen A con algún de kernel (B), que puede tener cualquier forma o tamaño, por lo general un cuadrado o un círculo.

(Universidad Nacional de Quilmes, 2005) Sean A y B conjuntos en Z^2 . La dilatación de A y B , expresada por $A \ominus B$ se define como:

$$A \ominus B = \{z | ((\hat{B})_z \cap A \neq \emptyset)\}$$

Esta ecuación consiste en obtener la reflexión de B sobre su origen y trasladar esta reflexión por z . La dilatación de A por B es entonces el conjunto de todos los desplazamientos, z , tal que la reflexión de



B y A se solapan por al menos un elemento. Teniendo en cuenta lo anterior, la dilatación de A por B también se puede expresar como:

$$A \ominus B = \{z | ((\hat{B})_z \cap A \subseteq A)\}$$

En general, la dilatación aumenta el tamaño de un objeto, La cantidad y la forma en que aumenta el tamaño depende de la elección de un elemento estructural.

Erosión Binaria

Esta operación es la hermana de la dilatación. Lo que esto hace es calcular un mínimo local sobre el área del núcleo.

(González Marcos, y otros, 2006) Dados los conjuntos A y B de Z^2 la erosión denotada por $A \ominus B$, se define por:

$$A \ominus B = \{x \in E^N | x = x + b \in A \text{ para todo } b \text{ en } B\}$$

La erosión, es la función dual de la dilatación (pero no es la inversa, es decir, si se hace una erosión y luego una dilatación, no se queda la imagen como estaba).

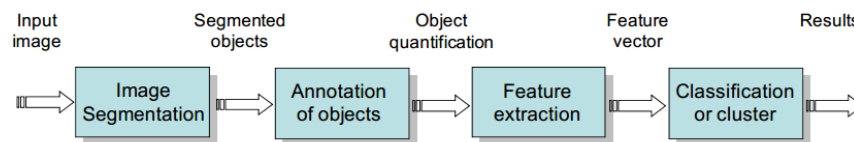
Segmentación de Imágenes

(Zhou, Wu, & Zhang) Segmentación de imagen es un componente fundamental en muchas aplicaciones de visión por computador, y puede ser abordado como un problema de agrupamiento, la segmentación de las imágenes presentadas a un sistema de análisis de imagen depende de manera crítica de la escena para ser detectada, la geometría de la imagen, la configuración, y el



sensor utilizado para transducir la escena en una imagen digital, y en última instancia la salida (meta) deseada del sistema.

Figura 13 Típico proceso de análisis de una imagen



Fuente: (Zhou, Wu, & Zhang)

Filtros Para Detección de Bordes

(Zhou, Wu, & Zhang) Un detector de bordes detecta los límites de un objeto. Estos métodos explotan el hecho de que los valores de intensidad de pixel cambian rápidamente en el límite (borde) de dos regiones. Ejemplos de detectores de bordes son: Canny, Laplacian, Prewitt, Roberts y Sobel.

Sobel

(Zhou, Wu, & Zhang) Es uno de los filtros de borde más útiles y ampliamente disponibles. Se estima primero el gradiente de intensidad en las direcciones horizontal y vertical desde lineal con coeficientes respectivamente.

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \text{ y } \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

La intensidad máxima se calcula como la raíz cuadrada de la suma de los cuadrados de los gradientes horizontales y verticales. La dirección del gradiente máximo también está disponible en el arco tangente de la relación de los gradientes verticales y horizontales.



Canny

(Zhou, Wu, & Zhang) Este detector de bordes es también muy popular, tiene en cuenta la compensación entre la sensibilidad de detección de bordes contra la precisión de la localización del borde.

El detector de borde se compone de cuatro etapas:

- a. Suavizado gaussiano para reducir el ruido y eliminación de pequeños detalles.
- b. Magnitud del gradiente y cálculo de dirección.
- c. Supresión no máxima de gradientes más pequeños por otros más grandes para enfocar la localización borde.
- d. Seguimiento de borde por histéresis

Figura 14 Resultado de Canny Edge detection para dos diferentes imágenes con valores altos y bajos: 50 y 10 respectivamente.

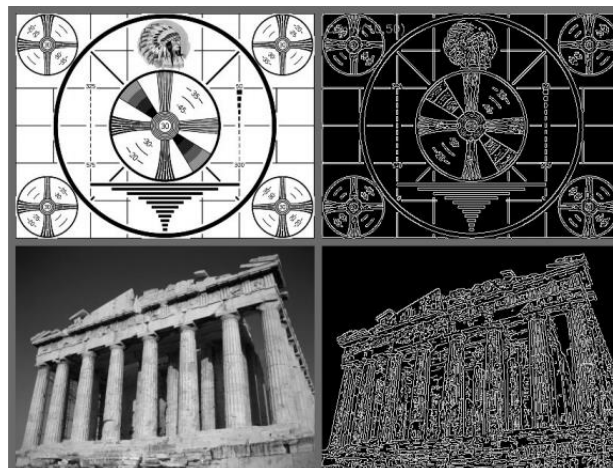


Imagen recuperada de Learning Opencv (Bradski & Kaebler, 2008)

Filtro Gaussiano o Laplaciano

(Zhou, Wu, & Zhang) El Laplaciano de filtro gaussiano encuentra bordes mediante la búsqueda de cruces por cero después de la filtrar la imagen de entrada con un filtro Laplaciano de un filtro gaussiano. Considere la función gaussiana.

$$h(r) = e^{-\frac{r^2}{2\sigma^2}}$$

Donde $r^2 = x^2 + y^2$ y es la desviación estándar.

Esto es una función de suavizado que, de convolución con una imagen, se desenfoca. El grado de desenfoco se determina por el valor de σ . El Laplaciano de esta función (la segunda derivada con respecto a r) es σ :

$$\nabla^2 h(r) = -\left[\frac{r^2 - \sigma^2}{\sigma^4}\right]e^{-\frac{r^2}{2\sigma^2}}$$

Por obvias razones, esta función es llamada "Laplacian of a Gaussian" (LoG). Porque la segunda derivada es una operación lineal. Convolucionar una imagen con una $\nabla^2 h(r)$ es lo mismo que la convolución de la imagen con la función de alisado primero y luego calcular el Laplaciano del resultado.

Técnicas de Segmentación de Imágenes

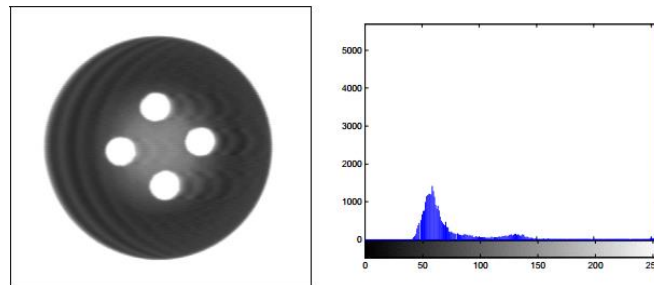
Segmentación basada en Umbralización

(González Marcos, y otros, 2006) El Histograma, como se ha visto en capítulos anteriores, permite ver la frecuencia relativa de aparición de cada nivel de luminancia de una imagen.



Este método consiste en agrupar los píxeles según sus niveles de intensidad luminosa.

Figura 15 A la izquierda un botón y a la derecha su histograma



Fuente: (González Marcos, y otros, 2006)

En la imagen se puede distinguir la montaña correspondiente a los niveles de gris del botón y a la izquierda del todo una línea que indica los niveles de luminosidad del fondo color blanco nieve.

Segmentación basada en Bordes

(González Marcos, y otros, 2006) Esta técnica se basa en los bordes encontrados en la imagen por los detectores. Los métodos de extracción de bordes no pueden ser utilizados como un resultado de segmentación. Se necesita seguir otros pasos para combinar los bordes en cadenas de bordes que corresponden con fronteras de objetos en la imagen.

Segmentación Orientada a Regiones

(González Marcos, y otros, 2006) Los métodos que se examinan en este tipo de segmentación construyen las regiones directamente.

Las regiones resultantes de estos procesos de segmentación deberán ser homogéneas y maximales, donde maximal significa que los criterios de homogeneidad no se cumplirán tras la unión con alguna de sus adyacentes.

(López Bonal, 1995) Dentro de esta aproximación a la segmentación de imágenes, resulta fundamental el concepto de región. Una región es un conjunto conexo de puntos que cumple un cierto predicado de uniformidad.

Desde este punto de vista, si R es la imagen completa podemos considerar la segmentación como el problema de obtener un conjunto de regiones $R_1 \dots R_m$

De manera que:

1. $\bigcup_{\forall i} R_i = R$
2. R_i es conexa $\forall i$
3. $R_i \cap R_j = \emptyset, \forall i \neq j$
4. $P(R_i) = \text{Cierto}, \forall i$
5. $P(R_i \cup R_j) = \text{Falso}, \forall i \neq j$

Donde $P()$ es un predicado lógico definido sobre los puntos de R_i , y que constituye el criterio de uniformidad.

Crecimiento de regiones

En este caso, el procedimiento consiste en agregar pixels a regiones existentes iterativamente, hasta que el predicado sea falso. Para conseguir este objetivo es necesario empezar con un conjunto de puntos aislados llamados “semillas”, a partir de los cuales se formaran las regiones resultantes.



División y fusión de regiones

Este procedimiento es una extensión del anterior, en la que se permite dividir una región en dos o más subregiones, además de poder unir regiones.

Los procedimientos de división y fusión operan normalmente con las regiones directamente y no con pixels como en el caso anterior.

Reconocimiento de Patrones

Detección de Objetos

(Szeliski, 2010) Si se nos da una imagen de analizar, como el retrato de grupo en la figura, podríamos tratar de aplicar un algoritmo de reconocimiento de todas las posibles sub-ventana en esta imagen.

Tales algoritmos son propensos a ser tanto lento y propenso a errores. En cambio, es más eficaz para construir Detectores de propósito especial, cuyo trabajo consiste en encontrar rápidamente probables regiones donde podrían ocurrir objetos particulares.

Figura 16 Resultado de reconocimiento de rostros producidos por

Rowley,

Baluja, y Kanade (1998)

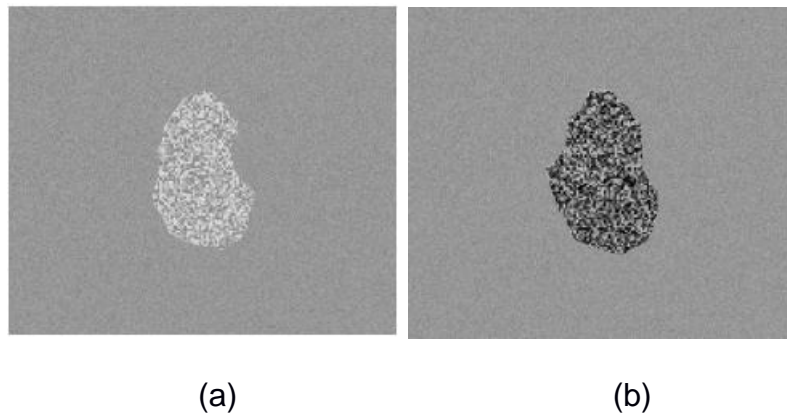


Fuente: (Szeliski, 2010)

Características, vectores de características y clasificadores

(Theodoridis & Koutroumbas, 2009) El primer paso es identificar las cantidades medibles que hacen que estas dos regiones sean distintas unas de otras.

Figura 17 Ejemplo de regiones de imagen correspondientes a (a) clase A y (b) clase B.



Fuente: (Theodoridis & Koutroumbas, 2009)

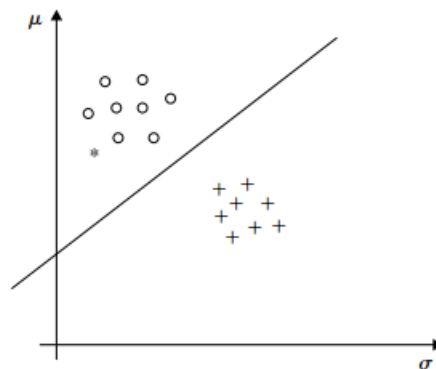
La Figura 17 Ejemplo de regiones de imagen correspondientes a (a) clase A y (b) clase B. muestra dos imágenes, cada una con una región distinta dentro de él. Se podría decir que la región de clase a son resultados de una lesión benigna y la (b) de una maligna (cáncer).

La Figura 19 muestra un gráfico del valor medio de la intensidad en cada región de interés frente a la desviación estándar correspondiente alrededor de esta media.

Cada punto corresponde a una imagen diferente de la base de datos disponible.

Resulta que los patrones de la clase A tienden a extenderse en una zona diferente de los patrones de clase B. La línea recta parece ser un buen candidato para la separación de las dos clases.

Figura 18 Gráfico de la media con respecto al valor de la desviación estándar para un número de diferentes imágenes procedentes de clase A (o) Y la clase B (+). En este caso, una línea recta separa las dos clases.



Fuente: (Theodoridis & Koutroumbas, 2009)

Descriptores en Imágenes

(Boullosa García, 2011) Los descriptores visuales describen las características de los contenidos dispuestos en imágenes o en vídeos. Describen características elementales tales como la forma, el color, la textura o el movimiento, entre otros.

El contenido de una imagen está codificado digitalmente en el valor de cada una de las unidades mínimas de información que la componen llamadas pixels.

De esta manera los pixels representan el nexo de unión entre el contenido abstracto de sus valores y las características propias de una imagen que entendemos como relevantes para el humano. Es por ello que cualquier método de gestión de imágenes basado en su contenido deberá guardar algún tipo de relación o actuar sobre el valor de los mismos.

Idealmente, un descriptor visual debería poseer las siguientes propiedades:

Propiedad	Descripción
Simplicidad	El descriptor debería representar las características extraídas de la imagen de manera clara y sencilla para permitir una fácil interpretación de su contenido.
Repetibilidad	El descriptor generado a partir de una imagen debe ser independiente del momento en el que se genere.
Diferenciabilidad	Dada una imagen, el descriptor generado debe poseer alto grado de discriminación respecto de otras imágenes y al mismo tiempo contener información que permita establecer una relación entre imágenes similares.



<p>Invarianza</p>	<p>Cuando existen deformaciones en la representación de dos imágenes, es deseable que los descriptores que las representan aporten la robustez necesaria para poder relacionarlas aún bajo diferentes transformaciones.</p>
<p>Eficacia</p>	<p>Es deseable que los recursos consumidos para generar el descriptor sean aceptables para poder ser utilizados en aplicaciones con restricciones críticas de espacio y/o tiempo.</p>

Fuente: (Boullosa García, 2011)

Descriptores de Color

El descriptor Color Layout fue diseñado para capturar la distribución espacial del color en una imagen. La representación se basa en los coeficientes de la Transformada Discreta del Coseno (DCT) sobre los valores de las componentes Y, Cb y Cr de la imagen. Esta representación se caracteriza por presentar una resolución invariante respecto del tamaño de la imagen y al mismo tiempo muy compacta.

La creación del descriptor se lleva a cabo mediante un proceso que se divide en las 4 etapas siguientes:



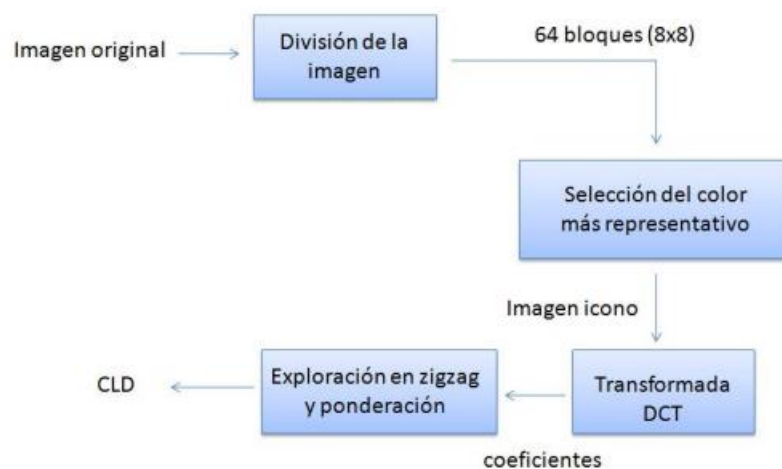
División de la imagen: En la primera de las etapas, la imagen original de entrada se divide mediante una rejilla en diferentes bloques o regiones.

Selección del color más representativo: Para cada uno de los bloques de la cuadrícula se selecciona un único color como representante de cada bloque.

Transformada DCT: Una vez se obtiene el icono de la imagen y tras efectuar una conversión del espacio de color de la imagen original al espacio de color YCbCr, se realiza el cálculo de la DCT de cada una de las tres componentes de color, obteniendo así los llamados coeficientes de la DCT en una matriz.

Exploración en zigzag: En esta última etapa se realiza una exploración en zigzag de los coeficientes de la matriz, con el objetivo de ponderar en mayor medida aquellos relacionados con las bajas frecuencias.

Figura 19 Diagrama del Descriptor de Color



Fuente: (Boullosa García, 2011)



A) División de la imagen

En la primera de las etapas, la imagen original es dividida en 64 (8x8) regiones o bloques mediante una cuadrícula cuyas dimensiones se ajustan al tamaño de la imagen. Cada uno de los bloques tiene unas dimensiones de $\left(\frac{M}{8} \times \frac{N}{8}\right)$, siendo M y N las dimensiones de la imagen original. Mediante esta operación, se consigue la invarianza respecto del tamaño de la imagen original.

Figura 20 División de la imagen en regiones



Fuente: (Boullosa García, 2011)

B) Selección del color más representativo

Una vez la imagen está dividida en bloques de igual tamaño, se identifica para cada uno de ellos el color más representativo. Existen diversos métodos para calcular el color más representativo de cada bloque, siendo la media del color de los píxeles comprendidos en cada bloque el recomendado por el estándar debido a su simplicidad y a que la precisión de la descripción es suficiente.

De este modo se obtiene una imagen de tamaño 8 x 8 con una apariencia borrosa, también llamada “thumbnail”, y cuyo resultado puede observarse en la Figura 3.6. Esta imagen es representada por las 3 matrices de tamaño 8 x 8, donde en cada una de ellas es almacenada una componente de color de los colores representativos de cada bloque.

Figura 21 Selección del Color más representativo de la imagen



Fuente: (Boullosa García, 2011)

Transformada DCT

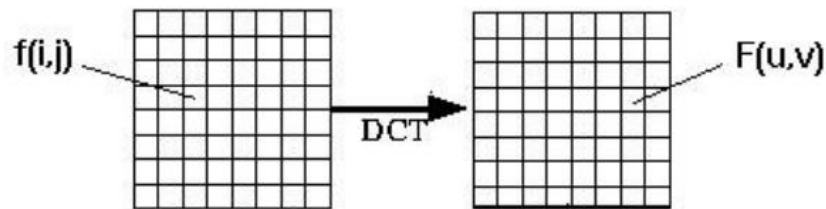
Como paso previo al cálculo de la DCT sobre la matriz de color, se realiza una conversión del espacio de color de la imagen original al espacio de color YCbCr formado por la crominancia Y, la crominancia azul y roja Cb y Cr respectivamente.

La matriz de color es transformada mediante la aplicación de la Transformada Discreta del Coseno (DCT) obteniendo de esta manera 3 grupos de 64 coeficientes. Para calcular la DCT de una matriz 2D se utiliza la siguiente fórmula:

$$F(u, v) = \alpha_u \alpha_v \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} A_{ij} \cos \frac{\pi(2i+1)u}{2M} \cos \frac{\pi(2j+1)v}{2N}, \quad \begin{matrix} 0 \leq u \leq M-1 \\ 0 \leq v \leq N-1 \end{matrix}$$

$$\alpha_u = \begin{cases} \frac{1}{\sqrt{M}} & u = 0 \\ \sqrt{\frac{2}{M}} & 1 \leq u \leq M-1 \end{cases} \quad \alpha_v = \begin{cases} \frac{1}{\sqrt{N}} & v = 0 \\ \sqrt{\frac{2}{N}} & 1 \leq v \leq N-1 \end{cases}$$

Figura 22 Dominio espacial y frecuencial DCT



Fuente: (Boullosa García, 2011)

Donde (i, j) denotan las coordenadas de la matriz de color, (u, v) representan las coordenadas en el dominio transformado y A_{ij} hace referencia a la intensidad del pixel en la posición (i, j) de la matriz de entrada. En la figura 22 se encuentran representados ambos dominios.

Recorrido en Zigzag

En esta última etapa se realiza un recorrido en zigzag por los coeficientes de la DCT de las tres componentes YCbCr obtenidos en la etapa anterior tras haber realizado una cuantificación previa de los mismos. El motivo de este trazado tiene que ver con la ubicación de las componentes de baja

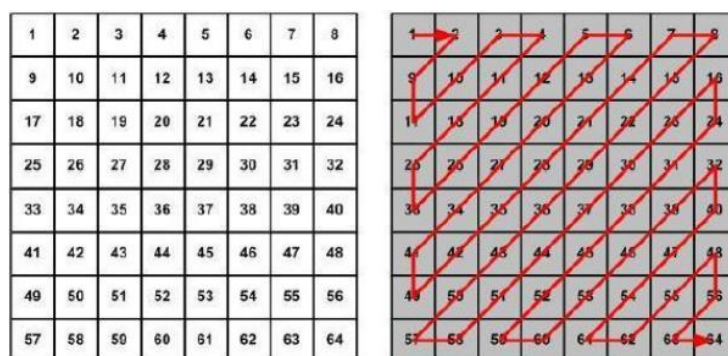
frecuencia de la imagen localizadas en la parte superior izquierda de la matriz transformada lo que significa que la energía de la imagen se concentra en dicha localización.

La manera de proceder con respecto al seguimiento del trazado se puede observar en la Figura 3.8 cuyo comienzo se sitúa en la parte superior izquierda para terminal en la diagonal opuesta.

La creación del descriptor Color Layout se compone por lo tanto de 3 vectores de características que contienen los diferentes coeficientes de luminancia y crominancia, Y, Cb y Cr respectivamente, representados según el orden del recorrido en zigzag.

Según el estándar, por defecto se utilizan tan sólo los 6 primeros coeficientes relativos a la luminancia y los 3 primeros de cada una de las crominancias, aunque la inclusión de un número mayor de coeficientes está sujeta a la precisión o requisitos de la etapa de comparación.

Figura 23 Exploración en Zigzag



Fuente: (Boullosa García, 2011)

Comparación entre descriptores



La comparación entre descriptores tiene como objetivo evaluar el grado de similitud entre dos imágenes mediante el cálculo de la distancia entre ellos.

La función de comparación recogida en el estándar es básicamente una suma ponderada de diferencias cuadráticas entre las componentes de ambos descriptores y que es representada por la siguiente fórmula:

$$D_{CLD} = \sqrt{\sum_i w_i^Y (Y_i - Y'_i)^2} + \sqrt{\sum_i w_i^{Cb} (Cb_i - Cb'_i)^2} + \sqrt{\sum_i w_i^{Cr} (Cr_i - Cr'_i)^2}$$

Donde (i, j, k) representan los coeficientes de las diferentes componentes (Y, Cb, Cr) respectivamente y w_i^Y , w_i^{Cb} , w_i^{Cr} son los pesos elegidos para establecer las contribuciones de cada componente en la métrica. Los pesos, así como el número de coeficientes pueden variar dependiendo del rendimiento alcanzado en el proceso de comparación o de la importancia que se quiera dar a una u otra componente.

Observando la fórmula sobre la métrica de comparación se puede observar que dos imágenes comparten más similitudes cuanto más pequeño sea el valor de la distancia, resultando la misma imagen cuando el valor es 0.

Descriptores de Forma

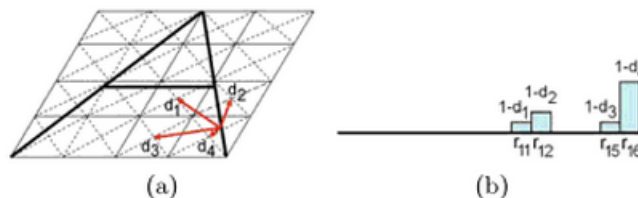
Los descriptores de forma se utilizan para describir un objeto valiéndose de las características que se pueden obtener según su forma, lo cual es posible gracias al uso de distintas técnicas de detección de bordes. Para esto generalmente, se utilizan características de bajo nivel debido a que estas pueden ser extraídas automáticamente de una imagen sin tener previamente información sobre la forma del objeto.



Blurred Shape Model

(Almazán, Valvery, & Fornés, 2011) Dado un conjunto de puntos que forman la forma de un símbolo particular, cada punto contribuye para calcular la DESCRIPTOR BSM. Esto se hace dividiendo la imagen dada en una cuadrícula $N \times N$ con sub-regiones de igual tamaño (células). Luego, cada célula recibe los votos de los ubicados en las células adyacentes. De este modo, cada píxel contribuye a medida densidad de su celular sub-región, y sus queridos vecinos. Esta contribución se pondera de acuerdo a la distancia entre el punto y el centro de gravedad de la célula receptora del voto.

Figura 24 Ejemplo de estimación de densidad (a) distancia dada entre un pixel a el centroide vecino. (b) Actualización de descriptor vectorial usando distancias de (a)



Fuente: (Almazán, Valvery, & Fornés, 2011)

Square Model of Shape Matrix

(Almazán, Valvery, & Fornés, 2011) Modelo cuadrada de matriz de forma, también llamado descriptor de cuadrícula, se construye de acuerdo con el siguiente algoritmo: para la forma S , construir un cuadrado centrado en el centro de gravedad G de S . El tamaño de cada lado es igual a $2L$, donde L es la máxima distancia euclidiana de G a un punto M en el límite de la



forma Point M se encuentra en el centro de un lado y GM es perpendicular a un lado.

Se divide la plaza en sub-cuadrados $N \times N$ y denotan $S_{kj}, k, j = 1, \dots, N$ los sub-cuadrados de la cuadrícula. Se define la matriz de la forma:

$$B_{kj} = \begin{cases} 1 & \Leftrightarrow u(S_{kj} \cap S) \geq u(S_{kj})/2 \\ 0 & : \text{ otros} \end{cases}$$

Gray Level Co-Occurrence Matrix

(Qing Shi & Jeon, 2006) Gray Level Co-Occurrence Matrix, describe la co-ocurrencia de los diferentes niveles de gris en alguna posición espacial especificada en una imagen. La matriz de co-ocurrencia de nivel de gris de la imagen natural tiende a ser distribuida en diagonal porque los niveles de gris de los píxeles vecinos en imágenes natural son a menudo altamente correlacionados. Después de la incrustación de datos, sin embargo, la alta concentración a lo largo de la diagonal principal de la matriz de co-ocurrencia de nivel de gris se extiende debido a que las altas correlaciones entre los píxeles de la imagen original se han reducido.

(Haralick, 2005) Para poder obtener las características de textura, se utilizan los siguientes descriptores de textura:

Contraste:

$$Co = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} (i-j)^2 p(i,j)$$

Correlación:

$$C = \frac{\sum_{i=0}^{G-1} \sum_{j=0}^{G-1} ij p_{d\theta} - \mu_x \mu_y}{\sigma_x \sigma_y}$$



Diferencia Inversa:

$$Pm = \max_{ij} p(i, j)$$

Energía:

$$E = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} [p(i, j)]^2$$

Entropía:

$$En = - \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} p(i, j) \log(2) p(i, j)$$

Homogeneidad:

$$H = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \frac{p(i, j)}{(1 + |i + j|)}$$

Momento de Diferencias orden K:

$$MD = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} (i - j)^k p(i, j)$$

Algoritmos de Clasificación

KNN

La técnica de vecino más cercano (NN) es muy simple, altamente eficiente y eficaz en el campo de reconocimiento de patrones, categorización de texto, reconocimiento de objetos, etc. Su sencillez es su principal ventaja, pero no pueden ser ignorados sus inconvenientes. La complejidad del cálculo y el requisito de memoria también deben importar. (Nitin & Vandana, 2010)

Figura 25 Ejemplo del algoritmo kNN



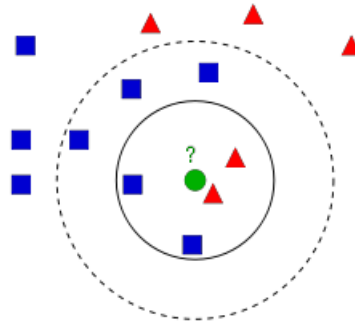


Imagen recuperada de <http://docs.opencv.org/>

En contraste con otros algoritmos de aprendizaje supervisado, K-NN no genera un modelo fruto del aprendizaje con datos de entrenamiento, sino que el aprendizaje sucede en el mismo momento en el que se prueban los datos de test. A este tipo de algoritmos se les llama mecanismos de aprendizaje perezoso. Se basa en la suposición de que los prototipos más cercanos tienen una probabilidad a posteriori similar.

Si $K_i(X)$ es el número de muestras de la clase a tomar presentes en los k vecinos más próximos al punto X , esta regla puede expresarse como:

$$d(X) = w_c \text{ si } K_c(X) = \max_{i=1 \rightarrow j} K_i(K)$$

Redes Bayesianas

(Felgaer, 2005) Las redes bayesianas o probabilísticas se fundamentan en la teoría de la probabilidad y combinan la potencia del teorema de Bayes con la expresividad semántica de los grafos dirigidos.

El teorema de Bayes expresa la probabilidad condicional de un evento aleatorio A si conocemos el valor de B .

$$P(A_i | B) = \frac{P(A_i) * P(B | A_i)}{P(B)}$$



Un criterio que se ha utilizado para describir la estructura de una red bayesiana atiende al grado en que sus elementos son “visibles” para el usuario (Edwards, 1998; Edwards y Fasolo, 2001). En este sentido una red bayesiana sería una estructura compuesta por cuatro niveles. En el nivel superior, una red bayesiana ser a un conjunto de variables representadas por nodos y un conjunto de flechas que relacionan estas variables en términos de influencia. En un nivel inferior estarían los niveles o estados, también conocidos como espacio de estados (Nadkarni y Shenoy, 2001, 2004), que pueden asumir cada una de las variables del modelo. En tercer lugar, tendríamos un conjunto de funciones de probabilidad condicional, una para cada nodo, donde se representaría la probabilidad de ocurrencia de cada estado de la variable condicionado a los posibles valores de las variables que determinan el valor de la variable. Por último, en el nivel más subordinado estarían un conjunto de algoritmos que permitirían que la red recalculase las probabilidades asignadas a cada uno de sus niveles cuando conocemos alguna evidencia sobre el modelo. Sin embargo, la descripción más frecuente de una red bayesiana se basa en dos elementos, una dimensión cualitativa y otra cuantitativa (p. e. Cowell et al., 1999; arbolino y Taroni, 2002; Nadkarni y Shenoy, 2001, 2004; Martínez y Rodríguez, 2003). Los fundamentos teóricos de estos dos elementos o dimensiones se sustentan en dos grandes pilares de la modelización matemática, la teoría de grafos y la teoría de la probabilidad (Ríos, 1995 Dimensión Cualitativa Una de las ventajas más importantes de las redes bayesianas es que pueden representar tanto el aspecto



cuantitativo de un problema como su aspecto cualitativo. El soporte teórico de la dimensión cualitativa en las redes bayesianas lo aporta la teoría de grafos. La teoría de grafos trata de crear modelos gráficos (grafos) que representen los elementos del problema en un sentido holista y fue introducida por Euler para dar solución al problema de los puentes de Königsberg (Harary, 1969; Ríos, 1995). La teoría de grafos se ha utilizado, por ejemplo, para la solución de problemas en la búsqueda de moléculas isomórficas, en el diseño de circuitos electrónicos e incluso en la representación de estructuras empresariales (Harary, 1969; Ríos, 1995; Ronald, 1988); en el campo de la Psicología fue Lewin quien usó la teoría de grafos para construir sus teorías sobre el espacio vital y los vectores psicológicos (Harary, 1969; Hothersall, 1995/1997).

Una red bayesiana es un grafo y aunque la definición de grafo, y la terminología que lo acompaña, varía en función de los autores (p. e. Harary, 1969; Ronald, 1988; Spirtes, Glymour, y Scheines, 2000; Tutte, 1984; Xiang, 2002); podemos definirlo como un par $G = (V, E)$, donde V es un conjunto finito de vértices, nodos o variables y E es un subconjunto del conjunto $V \times V$ de pares ordenados de vértices llamados enlaces o aristas. Además, una red bayesiana es un tipo particular de gráfico que se denomina grafo dirigido acíclico. Dirigido hace referencia a que los enlaces entre los vértices del grafo están orientados. Así, si los dos pares ordenados (A, B) y (B, A) pertenecen a G diremos que existe un enlace no dirigido entre A y B y se representa como $A \sim B$. También podemos decir que A y B son vecinos o que son adyacentes. El conjunto de vecinos

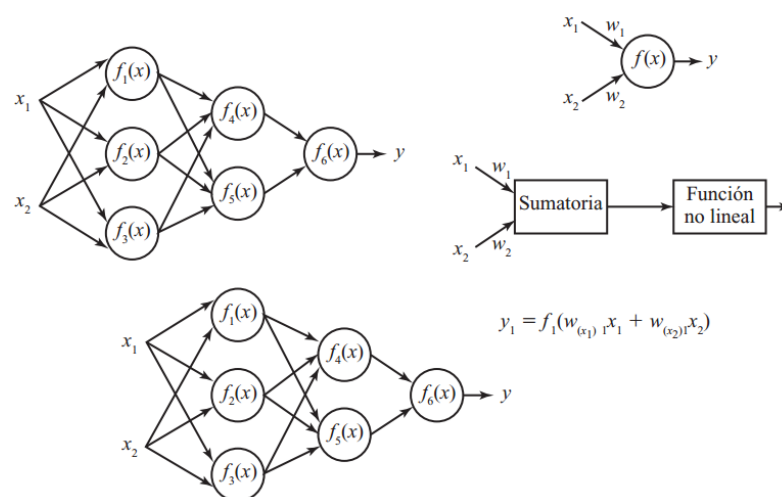


del vértice A es denotado por $Ady(A)$. Sin embargo, si $(A, B) \in E$ pero $(B, A) \notin E$, diremos que hay un enlace dirigido o arco entre los nodos y lo representaremos como $A \rightarrow B$. También podremos decir que A es padre o la variable madre de B que B es hijo o la variable madre de A. Acíclico se refiere a que no pueden existir ciclos o loops en el grafo, esto es; que si comenzamos a recorrer un camino dirigido desde un nodo, nunca podríamos regresar al punto de partida.

Redes neuronales artificiales

(Cruz, 2010) El algoritmo Backpropagation para redes multicapa es una generalización del algoritmo de mínimos cuadrados. Ambos algoritmos realizan su labor de actualización de pesos y ganancias con base en el error medio cuadrático. La red Backpropagation trabaja bajo aprendizaje supervisado y por tanto necesita un conjunto de instrucciones de entrenamiento que le describa cada salida y su valor de salida esperado.

Figura 26 Red de dos capas



Fuente: (Cruz, 2010)



Determinación de Arquitectura Optima

Uno de los principales problemas de la red Neuronal es la definición de su arquitectura, que incluye las funciones de transferencia y salida a utilizar y la topología de la red. Para garantizar el aprendizaje, las funciones de transferencia y salida deben ser continuas y derivables en todos los puntos, lo que lleva a la selección mayoritaria de funciones de transferencia sigmoideas y funciones de salida lineales. En cuanto a la topología de la red, la determinación del número de capas que la integran y del número de neuronas ocultas a incluir en cada capa resulta una tarea mucho más compleja, que incide directamente en la capacidad de generalización del modelo. Dado que toda red neuronal dispone necesariamente de una capa de entrada, que recibe los estímulos externos, el problema se limita a establecer el número y extensión de las capas ocultas. Si bien se ha demostrado que la propiedad de aproximador universal de funciones de la red Neuronal requiere un máximo de dos capas ocultas, en la mayoría de los casos una única capa oculta resulta suficiente para conseguir óptimos resultados. LIPPMAN (1987) considera que redes con una única capa oculta resultan suficientes para resolver problemas arbitrariamente complejos, siempre que la capa oculta incluya al menos tres veces el número de nodos de entrada. Por su parte, HECHT-NIELSEN (1990) y LIPPMAN (1987) aplican una extensión del teorema de Kolmogorov para demostrar que una red con una única capa oculta integrada por $2N + 1$ neuronas y con funciones de transferencia continuas, no lineales y monótonamente crecientes, resulta suficiente



para computar cualquier función continua de N variables de entrada. Para determinar el número de neuronas ocultas de cada capa suelen utilizarse reglas ad hoc que, aunque no resultan matemáticamente justificables, han demostrado un buen comportamiento en diversas aplicaciones prácticas. Entre las reglas más conocidas destacan:

Regla de la pirámide geométrica

La regla de la pirámide geométrica se basa en la suposición de que el número de neuronas de la capa oculta ha de ser inferior al total de variables de entrada, pero superior al número de variables de salida (BLUM, 1992, p. 60; MASTERS, 1993, p. 176). Se considera que el número de neuronas en cada capa sigue una progresión geométrica, tal que, para una red con una única capa oculta, el número de neuronas intermedias debe ser próximo a $\sqrt{N \cdot M}$, siendo N el número de variables de entrada y M el total de neuronas de salida.

Regla de la capa oculta-capla entrada

Según esta regla, el número de neuronas ocultas está relacionado con el número de neuronas de entrada (SWINGLER, 1996, p. 53; BERRY y LINOFF, 1997, p. 323). En particular, suele aplicarse la regla 2 x 1, de forma que el número de neuronas ocultas no puede ser superior al doble del número de variables de entrada.

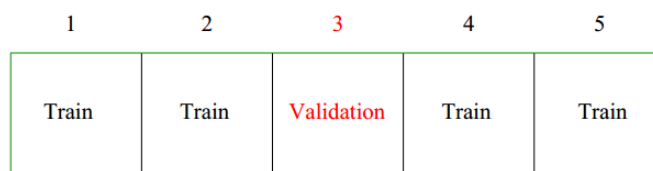
Modelo de Validación Cruzada

(Microsoft, s.f.) El modelo de validación cruzada se puede usar para probar un modelo de clasificación o regresión no entrenado. Los resultados pueden ayudar a interpretar la calidad del conjunto de datos y comprender si el modelo es susceptible de variaciones en los datos.

El modelo de validación cruzada toma un modelo de clasificación o regresión inexperto y un conjunto de datos de entrenamiento establecidos como una entrada. Devuelve resultados de la evaluación de veces validación cruzada, así como el conjunto de datos anotado.

(Hastie, Tibshirani, & Friendman) Lo ideal sería que, si tuviéramos suficientes datos, tendríamos dejar de lado un conjunto de validación y utilización para evaluar el rendimiento de nuestro modelo de predicción. Puesto que los datos son a menudo escaso, esto por lo general no es posible. Para finura el problema, validación cruzada K veces utiliza parte de los datos disponibles para ajustar el modelo, y un diferente parte para probarlo. Dividimos los datos en K partes aproximadamente iguales de tamaño; por ejemplo, cuando $K = 5$, el escenario se ve así:

Figura 27 Ejemplo de Validación Cruzada (Cross-Validation)



Fuente: (Hastie, Tibshirani, & Friendman)



Para la parte de orden k (tercera arriba), encajamos el modelo al otro $K - 1$ piezas de los datos y se calcula el error de predicción del modelo ajustado cuando la predicción de la k -ésima parte de los datos. Hacemos esto para $k = 1, 2, \dots, K$ y combinar las estimaciones de K de error de predicción.

A continuación, la estimación de validación cruzada de error de predicción es

$$CV(\hat{f}) = \frac{1}{N} \sum_{i=1}^N L(y_i, \hat{f}^{-k(i)}(X_i))$$

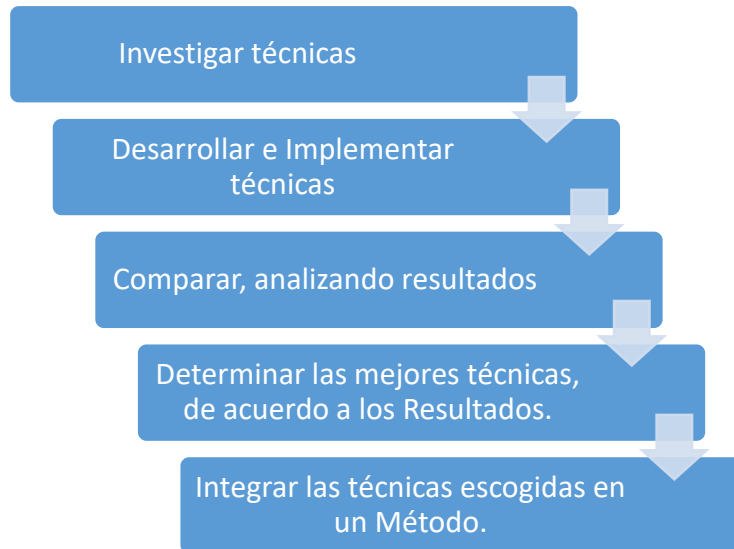
III. MARCO METODOLOGICO

3.1. Tipo y diseño de la investigación:

Tipo de Investigación.

Tecnológico – Cuasi Experimental

Diseño de la Investigación.



Fuente: Elaboración propia

3.1. Población y Muestra

Población

Nuestra población son todas las imágenes de melanomas y no melanomas existentes. La población es infinita.

Muestra

La muestra es intencional, que por conveniencia para la investigación está conformada por un dataset de imágenes contenedoras de 206 imágenes de melanomas y no melanomas; dicho dataset es gratuito y descargado de www.dermquest.com (página conformada por un grupo de dermatólogos internacionales de gran prestigio con intereses en la investigación académica).

3.2. Hipótesis

Al comparar las técnicas de reconocimiento de imágenes, se podrá formar un método que brinde mayor efectividad en la detección de melanomas.

3.3. Variables

Variable Independiente

Método propuesto de reconocimiento de imágenes.

Variable Dependiente

Efectividad en la identificación de melanomas humanos en imágenes.

3.5. Operacionalización

Variable Independiente	Dimensiones	Indicadores	Descripción
Método propuesto de reconocimiento de imágenes.	Procesamiento de Imágenes	Numero de Algoritmos de Procesamiento Empleados	Cantidad de algoritmos de procesamiento que intervienen en el análisis
	Muestras	Cantidad de Imágenes Empleadas	Cantidad de Imágenes de Muestra para el Entrenamiento del Sistema
		Numero de Descriptores	Cantidad de descriptores utilizados para extraer las características.



Variable Dependiente	Dimensiones	Indicadores	Descripción
Efectividad en la identificación de melanomas humanos en imágenes.	Clasificación	Numero de Imágenes Correctamente Clasificadas	Cantidad de imágenes cuya determinación y clasificación corresponden a la realidad
		Numero de Imágenes Erróneamente Clasificadas	Cantidad de imágenes cuya determinación y clasificación no corresponden a la realidad
		Probabilidad de Acierto Global	Porcentaje de aciertos dado por el número de imágenes correctamente clasificadas sobre el total de la muestra



3.6. Métodos, Técnicas e Instrumentos de Recolección de Datos

3.6.1. Métodos de Recolección de Datos

Los métodos de investigación que se utilizarán en este estudio son: observación, análisis, de síntesis y experimental.

Observación: Son las características que podemos realizar, así como asesores y jurado calificador de la presente investigación.

Análisis: Dado que se va a descomponer el objeto de estudio en sus partes para conocer sus riesgos y propiedades.

Experimental: La investigación se ejecutará a partir de una situación real de un problema, abordándose en la implementación de métodos de visión computacional en la cual fundamenta la elaboración y verificación de la hipótesis.

Síntesis: Porque una vez analizado el problema planteado y los métodos de visión artificial a implementar, se plantea a desarrollar una solución bajo los métodos seleccionados

3.6.2. Técnicas de Recolección de Datos

Análisis Documental: Consiste en extraer la información de los diferentes documentos, libros, papers, etc. Los cuales presentan una serie de teorías, formulas, métodos para dar solución al problema planteado. Todo esto servirá para determinar el modelo a seguir, diseñar el algoritmo requerido y finalmente analizar los resultados obtenidos después de haber aplicado los algoritmos seleccionados.



Observación: Es el registro visual de lo que ocurre en una situación real, clasificando los acontecimientos pertinentes de acuerdo con algún esquema previsto y según el problema que se estudia. Es preciso estar alerta para discriminar adecuadamente todo este conjunto posible de informaciones.

3.7. Procedimiento para la Recolección de Datos

a) Análisis documental: El proceso llevado a cabo para realizar la recolección de datos mediante el análisis documental en la investigación es el siguiente:

1. Recopilar imágenes que contengan al melanoma humano, dichas imágenes deberán ser descargadas de un dataset proporcionado por www.dermquest.com (página conformada por un grupo de dermatólogos internacionales de gran prestigio con intereses en la investigación académica).
2. Investigar las técnicas de reconocimiento de imágenes utilizadas para detectar el melanoma; desarrolladas y propuestas en investigaciones anteriormente realizadas.
3. Investigar los métodos de visión artificial que servirán para cumplir con el objetivo del proyecto.

b) Observación: Una vez implementados las técnicas de reconocimiento de imágenes utilizadas para detectar el melanoma, el siguiente paso será evaluar el comportamiento de las técnicas implementadas.



3.8. Análisis Estadístico e Interpretación de los Datos

Precisión

Es la proporción de muestras correctamente diagnosticadas del total de muestras evaluadas.

$$P = \frac{|\#Taciertos \cap \#Tmuestras|}{|\#Tmuestras|}$$

Revocación

Es la proporción de muestras correctamente diagnosticadas para cada tipo de imagen (melanoma y no melanoma).

$$R = \frac{|\#Taciertos \cap \#Tmuestras|}{|\#Taciertos|}$$

Valor-F

Nos permite corregir el error de la distancia, en los casos en los que la revocación (R) y la precisión (P) se compensan, dando así, una medida de precisión al algoritmo

$$F = 2 * \frac{\text{Precisión} * \text{Revocación}}{\text{Precisión} + \text{Revocación}}$$

3.9. Principios Éticos

Los criterios éticos que se respetan en el presente proyecto de tesis es el Código Deontológico del Colegio de Ingenieros del Perú en su Capítulo III “Faltas Contra la Ética Profesional y Sanciones” y su Sub Capítulo II “De la Relación con El Público” en su Artículo 106 expresa: Los ingenieros, al explicar su trabajo, méritos o emitir opiniones sobre temas de ingeniería, actuarán con seriedad y convicción, cuidando de no crear conflictos de intereses, esforzándose por ampliar el conocimiento del público a cerca de la ingeniería y de los servicios que presta a la sociedad.

3.10. Criterios de Rigor Científico

Consentimiento informado.

Los Tesistas deberán estar de acuerdo con ser informantes y conocer sus derechos y responsabilidades durante la investigación.

Manejo de Riesgos.

La investigación requiere de una eficiencia y no de un beneficio personal para realizar una investigación consistente.

Observación Participante.

La participación de los Tesistas requiere una responsabilidad ética por los efectos y consecuencias que pueden surgir durante la investigación



IV. ANALISIS E INTERPRETACION DE LOS RESULTADOS

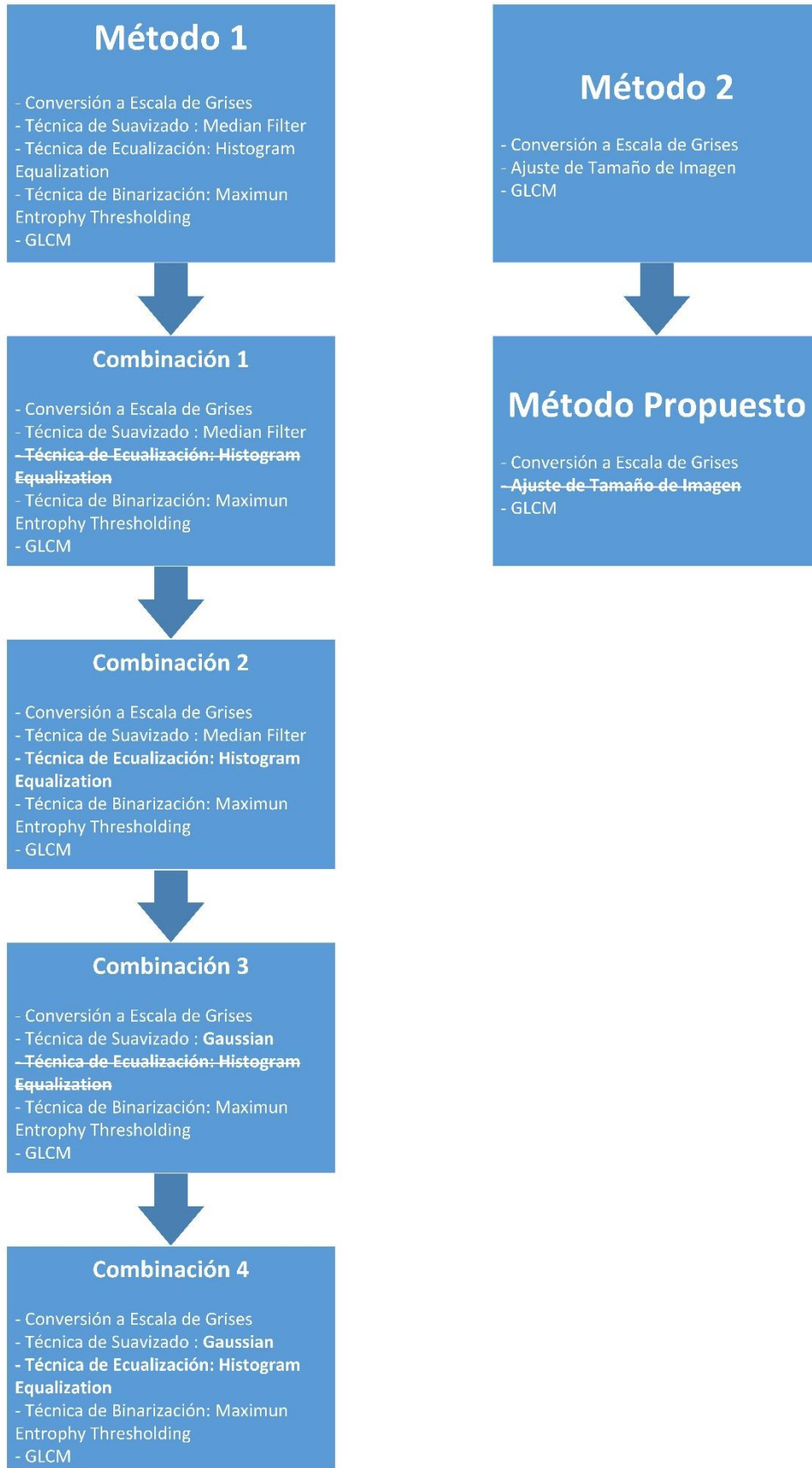
En este capítulo se procederá a explicar los métodos (propuestos en investigaciones anteriormente desarrolladas) de reconocimiento de melanoma, así como la interpretación de los resultados; seguidamente se presentará el método propuesto por los tesisistas, junto a la interpretación de sus resultados. Se utilizaron un total de 189 imágenes entre imágenes que contienen melanoma e imágenes que no contienen melanoma.

El número de imágenes clasificado es:

Imágenes	Total
Melanoma	110
No Melanoma	79
Total	189

En el siguiente diagrama de bloques se indican las combinaciones de cada uno de los métodos desarrollados, así como sus técnicas empleadas.

Diagrama de Bloques de los Métodos de Reconocimiento de Imágenes Implementados para el Reconocimiento del Melanoma Humano en Imágenes



PRIMER METODO:

Extraído de la investigación titulada “Artificial Neural Network for SkinCancer Detection”, cuyos autores son Sarika Choudhari y Seema Biday.

Combinacion 1:**Técnicas Utilizadas:****Conversión a Escala de Grises**

Técnica de Suavizado: MedianFilter

Técnica de Binarización: Maximum Entrophy Thresholding

Extracción de características: Para reconocer si es melanoma o no, se extrajeron las siguientes características: Mean, Desviación Estándar, Skweness, Kurtosis, Contraste Promedio, Correlación, Energy y Homogeneidad.

Resultados:

Para el reconocimiento de imágenes se usó Redes Neuronales Artificiales con 8 neuronas en la capa de entrada, 12 neuronas en la capa oculta y 2 neuronas en la capa de salida, luego se usó KNN con $k = 1$, y finalmente Bayes; además del uso de Validación Cruzada tomando 10 iteraciones para las pruebas.

Los resultados obtenidos fueron:

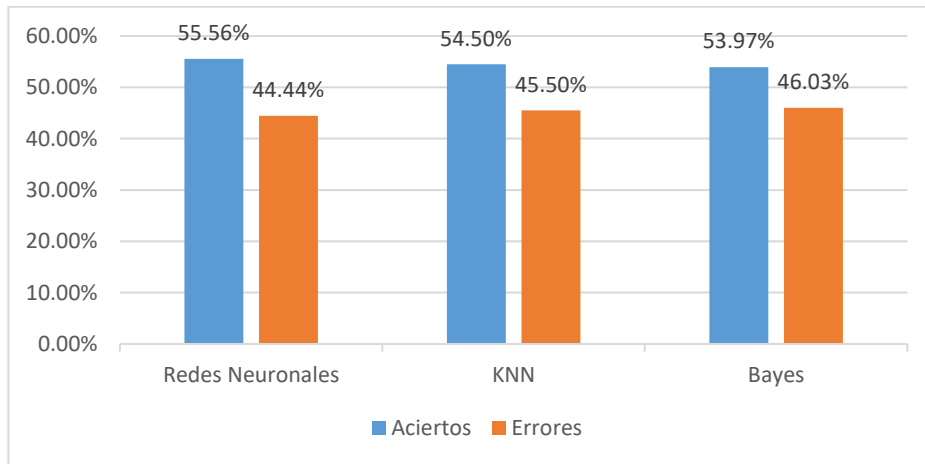


Figura 28 Grafico de Porcentaje de Aciertos y Errores por Clasificador

En el grafico se puede identificar que, utilizando Redes Neuronales, se obtiene un mayor porcentaje de acierto, siendo 55.5556% el porcentaje de acierto y 44.4444% el porcentaje de Error, frente a knn cuyos resultados son 54.4974% y 45.5026% respectivamente y en comparación a Bayes cuyos resultados son 53.9683% y 46.0317% respectivamente.

A continuación, se presenta un gráfico con la precisión detallada por clases:

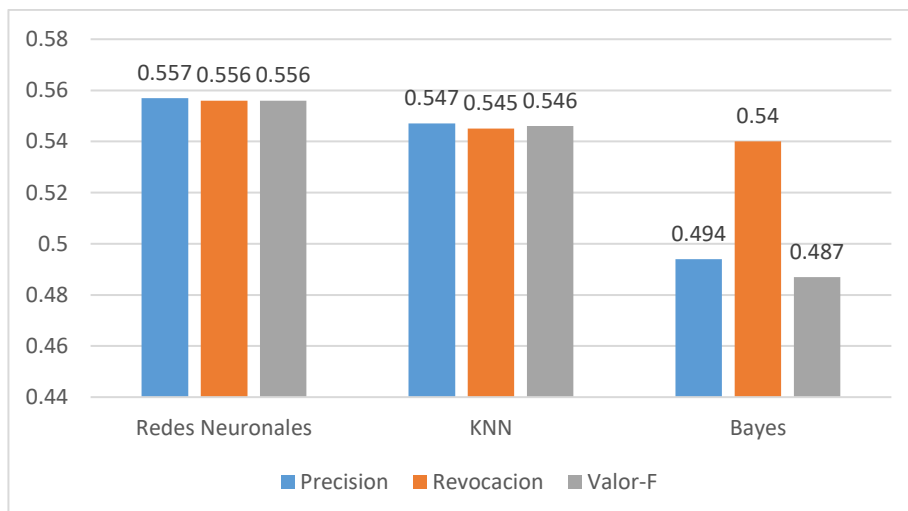


Figura 29 Grafico de Precisión Detallada por Clasificador



Numero de Descriptores Extraídos por Imagen: 9 descriptores.

Numero Algoritmos Empleados: 9 descriptores.

Combinación 2:

Técnicas Utilizadas:

Conversión a Escala de Grises

Técnica de Suavizado: MedianFilter

Técnica de Ecuilización: Histogram Equalization

Técnica de Binarización: Maximum Entrophy Thresholding

Extracción de características: Para reconocer si es melanoma o no, se extrajeron las siguientes características: Mean, Desviación Estándar, Skweness, Kurtosis, Contraste Promedio, Correlación, Energy y Homogeneidad.

Resultados:

Para el reconocimiento de imágenes se usó Redes Neuronales Artificiales con 8 neuronas en la capa de entrada, 12 neuronas en la capa oculta y 2 neuronas en la capa de salida, luego se usó KNN con $k = 1$, y finalmente Bayes; además del uso de Validación Cruzada tomando 10 iteraciones para las pruebas.

Los resultados obtenidos fueron:

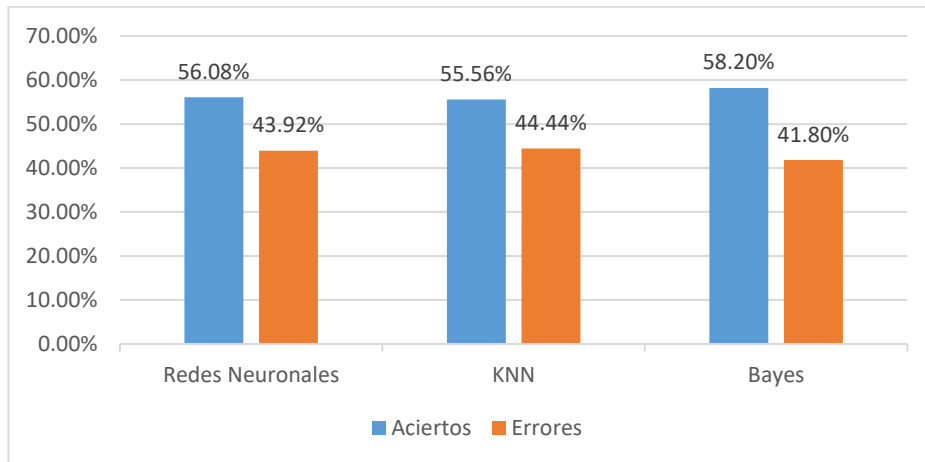


Figura 30 Grafico de Porcentaje de Aciertos y Errores por Clasificador

En el grafico se puede identificar que, utilizando Redes Neuronales, se obtiene un mayor porcentaje de acierto, siendo 56.0847% el porcentaje de acierto y 43.9153% el porcentaje de Error, frente a knn cuyos resultados son 55.5556% y 44.4444% respectivamente y en comparación a Bayes cuyos resultados son 58.2011% y 41.7989% respectivamente.

A continuación, se presenta un gráfico con la precisión detallada por clases:

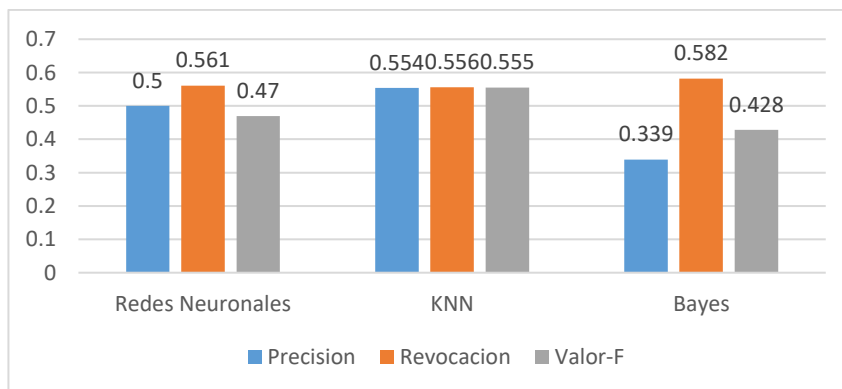


Figura 31 Grafico de Precisión Detallada por Clasificador

Numero de Descriptores Extraídos por Imagen: 9 descriptores.

Numero Algoritmos Empleados: 10 descriptores.



Combinación 3:

Técnicas Utilizadas:

Conversión a Escala de Grises

Técnica de Suavizado: Gaussian

Técnica de Binarización: Maximum Entrophy Thresholding

Extracción de características: Para reconocer si es melanoma o no, se extrajeron las siguientes características: Mean, Desviación Estándar, Skweness, Kurtosis, Contraste Promedio, Correlación, Energy y Homogeneidad.

Resultados:

Para el reconocimiento de imágenes se usó Redes Neuronales Artificiales con 8 neuronas en la capa de entrada, 12 neuronas en la capa oculta y 2 neuronas en la capa de salida, luego se usó KNN con $k = 1$, y finalmente Bayes; además del uso de Validación Cruzada tomando 10 iteraciones para las pruebas.

Los resultados obtenidos fueron:

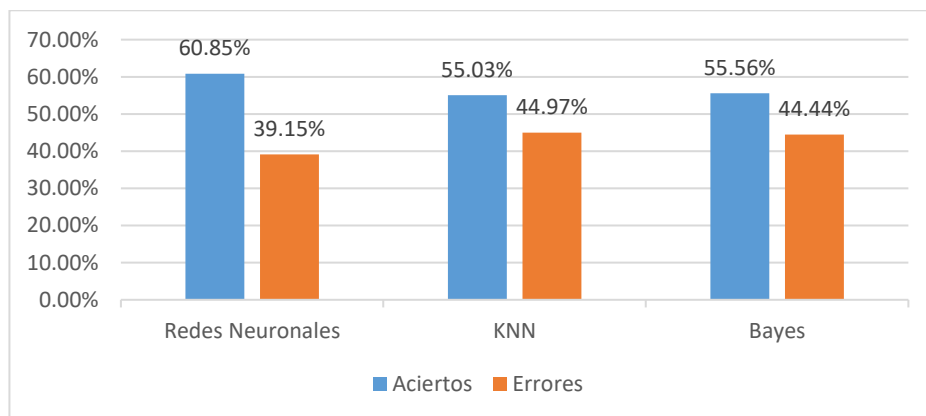


Figura 32 Grafico de Porcentaje de Aciertos y Errores por Clasificador



En el grafico se puede identificar que, utilizando Redes Neuronales, se obtiene un mayor porcentaje de acierto, siendo 60.8466% el porcentaje de acierto y 60.8466% el porcentaje de Error, frente a knn cuyos resultados son 55.0265% y 44.9735% respectivamente y en comparación a Bayes cuyos resultados son 55.5556% y 44.4444% respectivamente.

A continuación, se presenta un gráfico con la precisión detallada por clases:

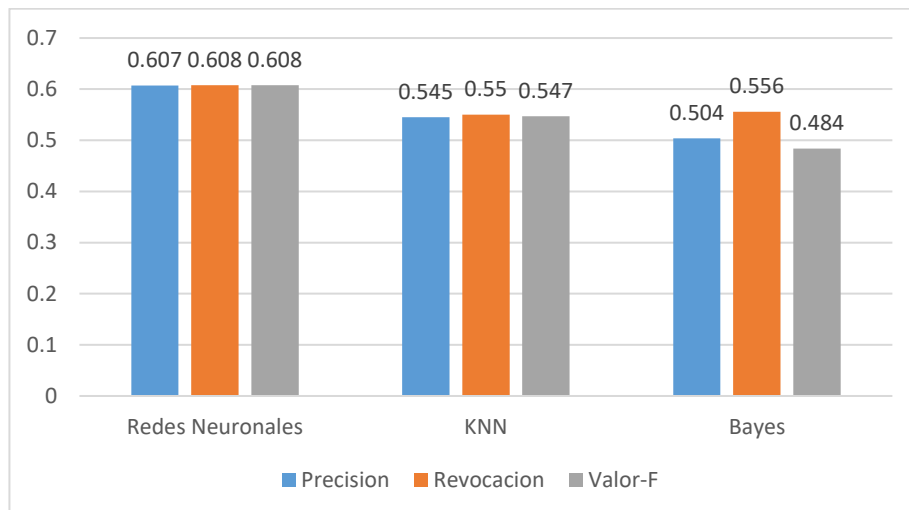


Figura 33 Grafico de Precisión Detallada por Clasificador

Numero de Descriptores Extraídos por Imagen: 9 descriptores.

Numero Algoritmos Empleados: 9 descriptores.

Combinación 4:

Técnicas Utilizadas:

Conversión a Escala de Grises

Técnica de Suavizado: Gaussian

Técnica de Equalización: Histogram Equalization



Técnica de Binarización: Maximum Entrophy Thresholding

Extracción de características: Para reconocer si es melanoma o no, se extrajeron las siguientes características: Mean, Desviación Estándar, Skweness, Kurtosis, Contraste Promedio, Correlación, Energy y Homogeneidad.

Resultados:

Para el reconocimiento de imágenes se usó Redes Neuronales Artificiales con 8 neuronas en la capa de entrada, 12 neuronas en la capa oculta y 2 neuronas en la capa de salida, luego se usó KNN con $k = 1$, y finalmente Bayes; además del uso de Validación Cruzada tomando 10 iteraciones para las pruebas.

Los resultados obtenidos fueron:

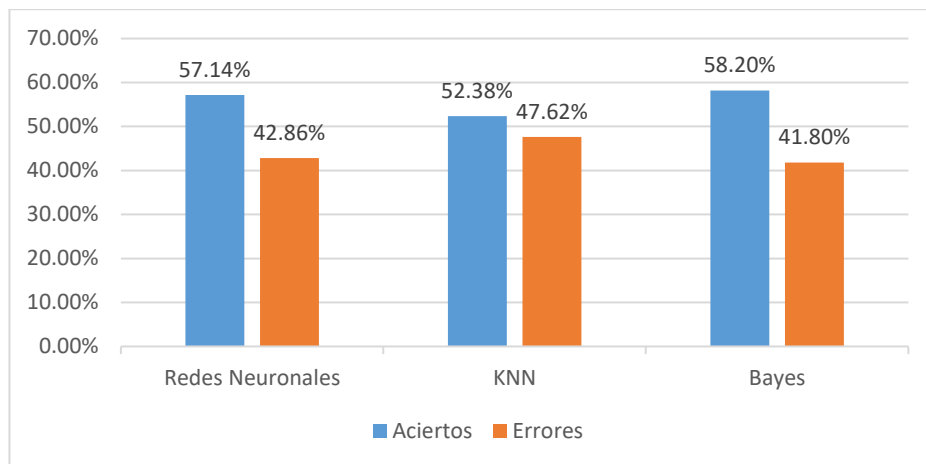


Figura 34 Grafico de Porcentaje de Aciertos y Errores por Clasificador

En el grafico se puede identificar que, utilizando Redes Neuronales, se obtiene un mayor porcentaje de acierto, siendo 57,1429% el porcentaje de acierto y 42,8571 % el porcentaje de Error, frente a knn cuyos resultados son 52,381% y 47,619% respectivamente y en



comparación a Bayes cuyos resultados son 58,2011% y 41,7989% respectivamente.

A continuación, se presenta un gráfico con la precisión detallada por clases:

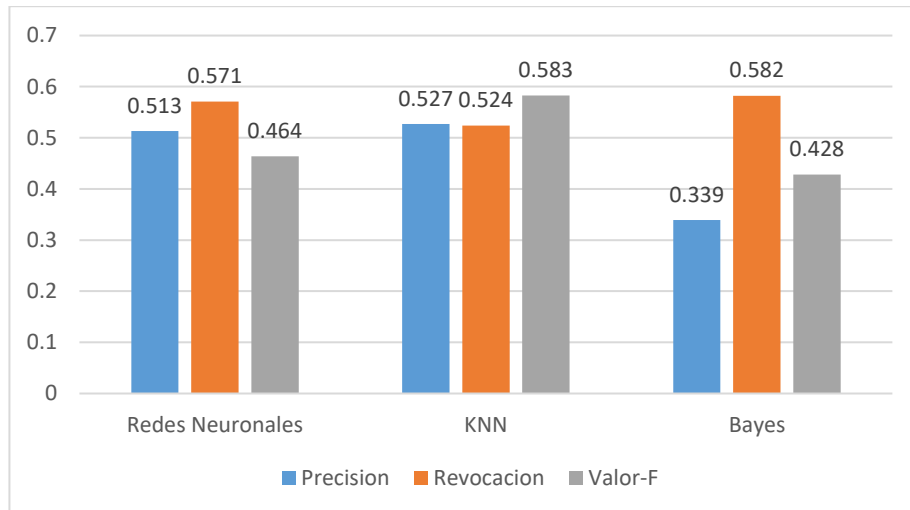


Figura 35 Grafico de Precisión Detallada por Clasificador

Numero de Descriptores Extraídos por Imagen: 9 descriptores.

Numero Algoritmos Empleados: 10 descriptores.

DISCUSIÓN:

La decisión de identificar las imágenes que contienen melanoma o no, utilizando los descriptores anteriormente explicados queda a criterio de los autores de la investigación usada para replicar el método.

El porcentaje de efectividad según la investigación, varía a los resultados obtenidos por los autores, dado que las imágenes usadas no son las mismas que emplearon los autores originales de la investigación usada para replicar el método.

Para la programación de la extracción de características de este método se usó el lenguaje C++ junto a la librería OpenCV 2.4.11 y para el reconocimiento y evaluación de resultados se usó la librería WEKA, tomando su función MultiLayerPerceptron. Al extraer los descriptores de cada imagen, se pudo observar una gran variación en los valores calculados matemáticamente, se concluyó que la razón era que estos descriptores durante esta investigación estaban siendo calculados manualmente utilizando una fórmula matemática junto a OpenCV contrariamente a la investigación de los autores originales los cuales utilizaron la herramienta MatLab durante todo su proyecto.

SEGUNDO METODO:

Extraído de la investigación titulada “Automatic Detection of Melanoma Skin Cancer using Texture Analysis”, cuyos autores son Mariam A.Sheha, Mai S.Mabrouk y Amr Sharawy.

Técnicas Utilizadas:

Conversión a Escala de Grises

Ajuste de tamaño de la imagen

Extracción de características: Para reconocer si es melanoma o no se extrajeron las siguientes características: homogeneidad, contraste, correlación, energía, entropía, clustershade, clustertendency, probabilidad máxima, información de correlación 1, información de correlación 2.



Resultados:

Para el reconocimiento se usó una red neuronal multicapa (1 capa de entrada, 1 oculta y 1 de salida), con 10 neuronas en la capa de entrada, 10 neuronas en la capa oculta y 2 neurona en la capa de salida, luego se usó KNN con $k = 1$, y finalmente Bayes; además del uso de Validación Cruzada tomando 10 iteraciones para las pruebas.

Los resultados obtenidos fueron:

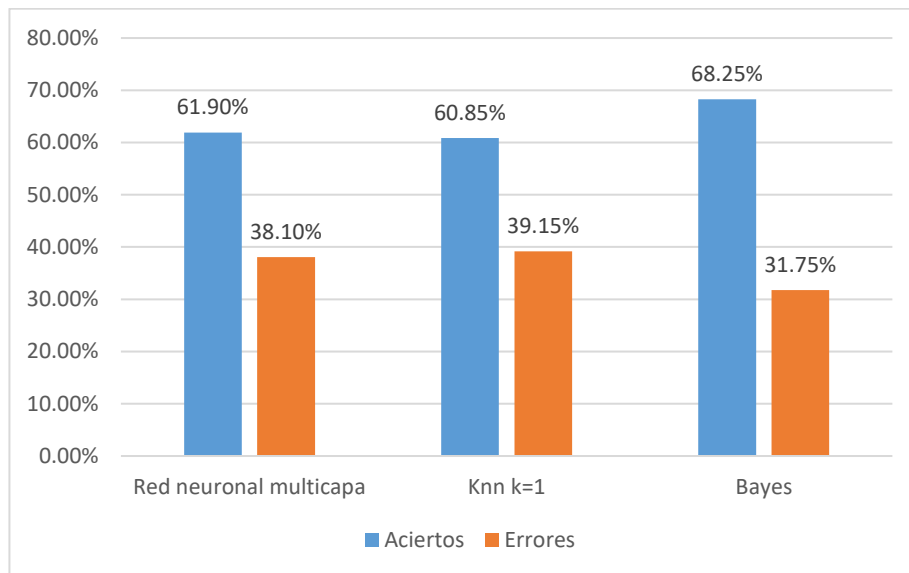


Figura 36 Grafico de acierto y errores por cada tipo de clasificador

Del gráfico se puede interpretar que la red bayesiana tiene mayor porcentaje de imágenes correctamente clasificadas con un 68.25%, superando el valor de la red de neuronal que tiene 61.90%, luego Knn ($k = 1$) con un porcentaje muy bajo de 60.85%.



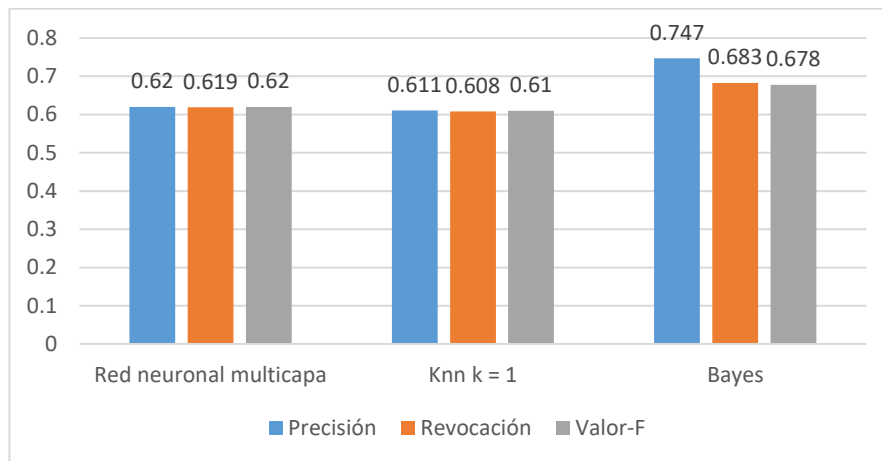


Figura 37 Pruebas de efectividad del clasificador

Numero de Descriptores Extraídos por Imagen: 10 descriptores.

DISCUSIÓN:

La decisión de identificar las imágenes que contienen melanoma o no, con los descriptores anteriormente explicados queda a criterio de los autores de la investigación usada para replicar el método.

En la investigación se propone usar un análisis de textura basado en GLCM. El porcentaje de efectividad según la investigación, varia a los resultados obtenidos por los autores, dado que las imágenes usadas no son las mismas que emplearon los autores originales de la investigación usada para replicar el método.

Para la programación de la extracción de características de este método se usó el lenguaje C++ junto a la librería OpenCV 2.4.11, y para el reconocimiento y evaluación de resultados se usó la librería WEKA, tomando su función MultiLayerPerceptron.

TERCER METODO (METODO PROPUESTO):

El siguiente es el método desarrollado y propuesto por los tesisistas, el cual está basado en su gran mayoría del segundo método anteriormente explicado, extraído de la investigación titulada “Automatic Detection of Melanoma Skin Cancer using Texture Analysis”, cuyos autores son Mariam A.Sheha, Mai S.Mabrouk y Amr Sharawy.

A diferencia del segundo método, en éste; se evita la parte de redimensionado de imagen, y se reajusta el número de neuronas en la capa oculta de la Red Neuronal.

Técnicas Utilizadas:

Conversión a Escala de Grises

Extracción de características: Para reconocer si es melanoma o no se extrajeron las siguientes características: homogeneidad, contraste, correlación, energía, entropía, clustershade, clustertendency, probabilidad máxima, información de correlación 1, información de correlación 2.

Resultados:

Para el reconocimiento se usó una red neuronal multicapa (1 capa de entrada, 1 oculta y 1 de salida), con 10 neuronas en la capa de entrada, 5 neuronas en la capa oculta y 2 neuronas en la capa de salida, luego

se usó KNN con $k = 1$, y finalmente Bayes; además del uso de Validación Cruzada tomando 10 iteraciones para las pruebas.

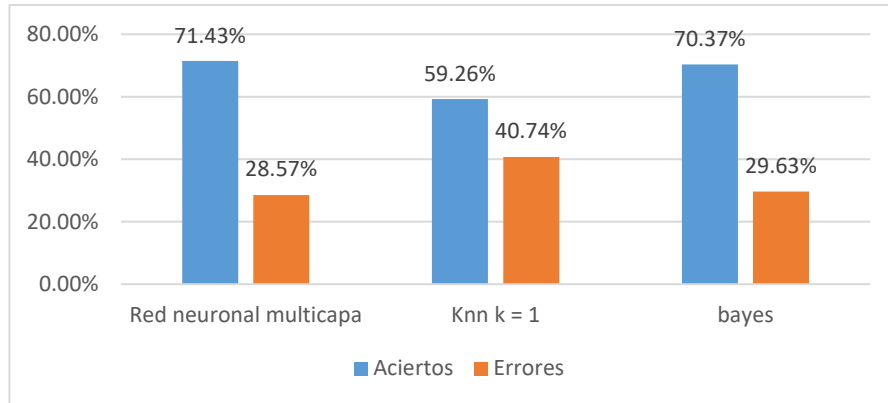


Figura 38 Número de aciertos y errores por clasificador

Del gráfico se puede interpretar que la red neuronal multicapa tiene mayor porcentaje de imágenes correctamente clasificadas con un 71.43%, superando el valor de la red de bayes que tiene 70.37%, luego Knn ($k = 1$) con un porcentaje muy bajo de 59.26%.

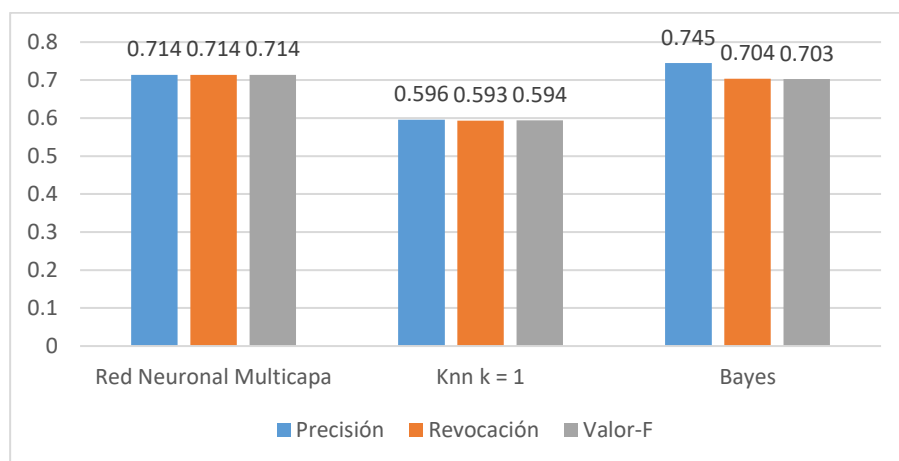


Figura 39 Pruebas de efectividad por clasificador

Tiempo de Reconocimiento:

La aplicación utilizó 18834 milisegundos para su ejecución total.



Numero de Descriptores Extraídos por Imagen: 10 descriptores.

Discusión:

Para la programación de este método se usó el lenguaje C++ junto a la librería OpenCV 2.4.11 y para el reconocimiento y evaluación de resultados se usó la librería WEKA, tomando su función MultiLayerPerceptron.

Se omitió utilizar la Técnica de Ajuste de Tamaño de Imagen y en la Red Neuronal se redujo el número de neuronas ocultas de 10 a 5 neuronas, ya que se detectó a Juicio de Experto que estas 2 características del método disminuían el porcentaje de efectividad

Tabla de Comparación

Para realizar la comparación se escogieron aquellas combinaciones que, utilizando Redes Neuronales, brindaban mayor efectividad en el reconocimiento de melanoma humano en imágenes.

Figura 40 Cuadro comparativo de métodos implementados

	METODO 1 – Combinacion 3	METODO 2	METODO 3
Técnicas Empleadas	Conversión a Escala de Grises	Conversión a Escala de Grises	Conversión a Escala de Grises



	<p>Técnica de Suavizado: Gaussiano</p> <p>Técnica de Binarización: Maximum Entrophy Thresholding</p> <p>Extracción de características: Mean, Desviación Estándar, Skweness, Kurtosis, Contraste Promedio, Correlación, Energy y Homogeneidad.</p>	<p>Ajuste de tamaño de la imagen</p> <p>Extracción de características: Homogeneidad, contraste, correlación, energía, entropía, clustershade, clustertendency, probabilidad máxima, información de correlación 1, información de correlación 2.</p>	<p>Extracción de características: Homogeneidad, contraste, correlación, energía, entropía, clustershade, clustertendency, probabilidad máxima, información de correlación 1, información de correlación 2.</p>
Porcentaje de Efectividad	60,8466%	61.9048 %	71.4286 %

Fuente : Elaboración propia

Esta tabla indica que el Tercer Método desarrollado por los tesisas, alcanza un mayor porcentaje de efectividad frente a el primer y segundo método, siendo la diferencia de 14.8148% y 9.5238% respectivamente.

Figura 41 Diferencias entre Metodos Implementados

	Metodo 1	Metodo 2	Metodo 3
	Combinacion 3		(Propuesto)



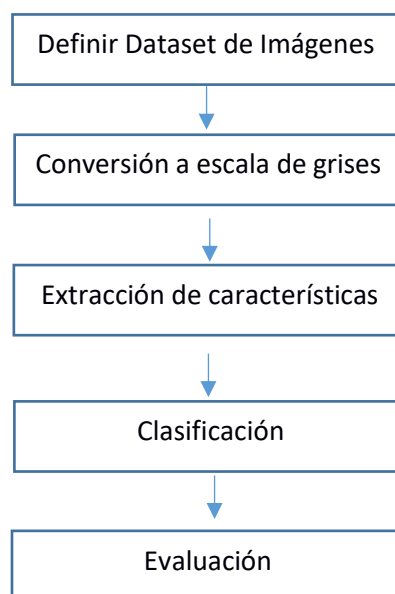
Principal Característica	Hace uso de segmentación de imágenes utilizando el Método de Maxima Entropía	Evita el uso de Segmentación de Imágenes	Evita el uso de Segmentación y Ajuste de Tamaño de Imágenes
# de Tecnicas	5	3	2
% de Imágenes Correctamente Clasificadas	60,8466%	61.9048 %	71.4286 %
% de Imágenes Incorrectamente Clasificadas	39.1534%	38.0952%	28.5714%

V. DESARROLLO DE LA PROPUESTA

El sistema comienza con la lectura de las imágenes, separadas en 2 carpetas (melanomas y no melanomas), luego procesa cada imagen, convirtiéndola a escala de grises para luego extraer las características GLCM, luego de haber procesado todas las imágenes, los datos resultantes se pasan a una red neuronal y se evalúa su precisión mediante el método de validación cruzada.

Para la elaboración del software, se ha utilizado el lenguaje de programación C++, en el entorno de desarrollo QT Creator, y con el depurador CDB.

Figura 42 Flujo del Algoritmo de la investigación



Fuente: Elaboración propia

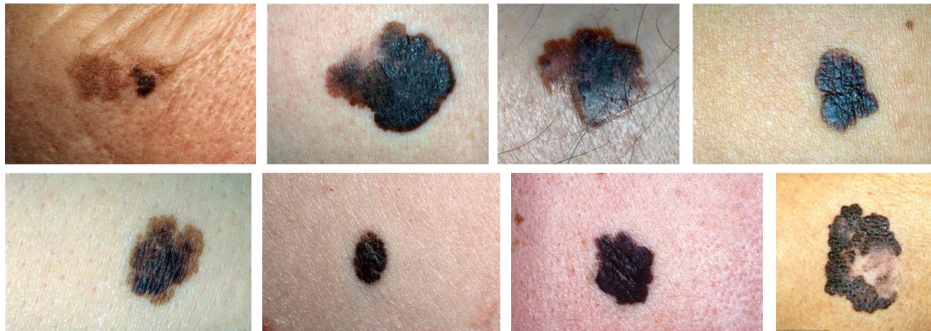


1. Definir Dataset de Imágenes

Las imágenes han sido extraídas de un dataset de imágenes contenedoras de 206 imágenes de melanomas y no melanomas; dicho dataset ha sido descargado de www.dermquest.com (página conformada por un grupo de dermatólogos internacionales de gran prestigio con intereses en la investigación académica), de estas 206 imágenes se han eliminado algunas que tienen demasiada complejidad.

Figura 43 Muestra de imagenes del dataset

Melanomas



No Melanomas



Fuente: Elaboración propia

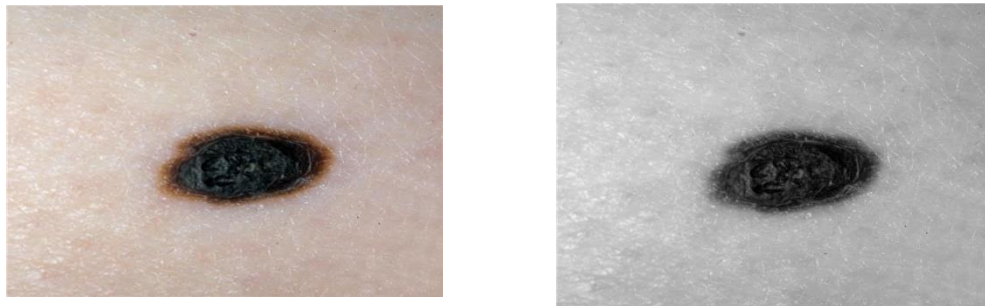
2. Conversión a escala de grises

En esta etapa la imagen se convierte a escala de grises, para poder así extraer las características GLCM

Figura 44 A la izquierda la imagen de un melanoma, a la derecha la misma imagen, pero en escala de grises.



Figura 45 A la izquierda la imagen de un no melanoma, a la derecha la misma imagen, pero en escala de grises.



Fuente: Elaboración propia.

Para realizar la conversión a escala de grises se tiene en cuenta la siguiente formula:

$$Y = 0.299 * R + 0.587 * G + 0.114 * B$$

Donde para cada pixel su equivalente en escala de gris será el promedio de la suma del valor de los canales de color Red, Green and Blue

Extracción de características

Luego de someter al procesamiento adecuado de la imagen se continua a extraer las características de la imagen.

El objetivo de la extracción de características de una imagen es generar una representación numérica precisa y compacta. Un vector de características ideal muestra poca variación entre los objetos de la misma clase y la variación entre objetos de diferentes clases. Las características pueden ser definidas como atributos inherentes de un objeto, mientras que estas características distinguen el objeto desde otra similar.

Se tuvieron en cuenta las siguientes características de GLCM:

Homogeneidad, contraste, correlación, energía, entropía, cluster shade, cluster tendency, maximum probability, information measure of correlation 1, information measure of correlation 2.

Para poder obtener las características de textura, se utilizan los siguientes descriptores de textura:

Contraste:

$$C_o = \sum_{i,j=0}^{N-1} P_{i,j}(i - j)^2$$

Cuando i y j son iguales, la celda está en la diagonal $(ij) = 0$, estos valores representan pixeles enteramente similares a sus vecinos, por lo que se les da un peso de 0, si i y j se diferencian por uno hay un



pequeño cambio, y el peso es 1, si i y j se diferencian por 2, el contraste aumenta y su peso es 4, el peso continua incrementándose exponencialmente como (ij) aumente, este valor es multiplicado por el valor del pixel (P_{ij}).

Correlación:

$$\sum_{i,j=0}^{N-1} P_{i,j} \frac{(i - \mu_i) - (j - \mu_j)}{\sqrt{(\sigma_x^2)(\sigma_j^2)}}$$

Cuando un área de imagen es completamente uniforme, la varianza GLCM es 0, al igual que la imagen varianza de primer orden es 0. Como resultado, el denominador de la ecuación de correlación se convierte en 0, y la correlación se vuelve indefinida. La correlación es una medida de la dependencia lineal de nivel de gris entre los píxeles en las posiciones especificadas respecto a la otra.

Máxima probabilidad:

$$Pm = \max_{ij} p(i, j)$$

Esta simple medida registra el valor P_{ij} más grande encontrado a lo largo de la imagen.

Energía:

$$E = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [p(i, j)]^2$$

Siguiendo la formula planteada, se deduce que los valores constantes máximos o uniformidad en la distribución periódica de nivel de gris,



formarán energía máxima de textura. La energía es 1 si la imagen es constante.

Entropía:

$$\sum_{i,j=0}^{N-1} p(i,j) \log(-\ln P_{i,j})$$

La entropía mide la pérdida de información o mensaje en una señal transmitida y también mide la información de la imagen. Nota: El término $P * \ln (P)$ se maximiza donde su derivada con respecto a P es 0.

Homogeneidad:

$$\sum_{i,j=0}^{N-1} \frac{p(i,j)}{1 + (i - j)^2}$$

El resultado de esta fórmula es la suma de la división del valor del píxel en la posición (i,j) dividido entre la suma de el cuadrado de la diferencia de i con j , sumado con 1.

Cluster Shade:

$$CS = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \{i + j - \mu_x - \mu_y\}^3 \times P(i,j)$$

Esta es la medida de asimetría de la matriz y se considera para medir los conceptos percentuales de uniformidad. El valor $j + \mu_i$ se calcula y se almacena para la primera zona de la imagen, y se actualiza posteriormente moviendo los píxeles de los vecinos. Cuando el valor de Cluster-shade es alto la imagen es asimétrica.

Cluster Prominence:





$$CP = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \{i + j - \mu_x - \mu_y\}^4 \times P(i,j)$$



Cluster prominence es también una medida de asimetría. Cuando el valor cluster-prominence es alto la imagen es menos simétrica. Además cuando el valor de cluster-prominence es bajo, hay un pico en la matriz GLCM alrededor de los valores medios.





A continuación se presentan 2 cuadros con ejemplos de los resultados obtenidos, clasificados por melanomas y no melanomas.

Muestra de Imágenes Contenedoras de Melanoma

Muestra	Homogeneidad	Contraste	Correlación	Energía	Entropía	ClusterShade	Cluster Tendency	Maximum Probability	Correlation Info 1	Correlation Info 2
	0.237901	57.7172	0.94055	0.00111936	7.48725	-116769	1883.99	0.00308698	-0.00808742	0.952878
	0.0701357	406.063	0.838226	0.000268974	8.75342	-732680	4614.06	0.000683032	-0.00260806	0.779843
	0.197585	75.4519	0.971776	0.000464631	8.17471	-684843	5271.27	0.00116549	-0.00870388	0.97018
	0.125114	113.639	0.975841	0.000278253	8.61945	-1.68645e+006	9293.96	0.000745127	-0.00673624	0.952392

Fuente: Elaboración propia.

Muestra de Imágenes contenedoras de No Melanoma

Muestra	Homogeneidad	Contraste	Correlación	Energía	Entropía	ClusterShade	Cluster Tendency	Maximum Probability	Correlation Info 1	Correlation Info 2
	0.155188	101.749	0.956298	0.000338947	8.49182	-196300	4554.69	0.00108637	-0.00826694	0.958507
	0.106986	249.69	0.854472	0.000326711	8.6727	-62700.8	3181.83	0.000955261	-0.00551382	0.876219
	0.219617	60.9848	0.971629	0.000663174	7.92482	-413426	4238.1	0.00195852	-0.00981924	0.975652
	0.161526	128.004	0.925284	0.000748153	7.82254	-547907	3298.42	0.00246857	-0.00503699	0.902837

Fuente: Elaboración propia.

3. Clasificación

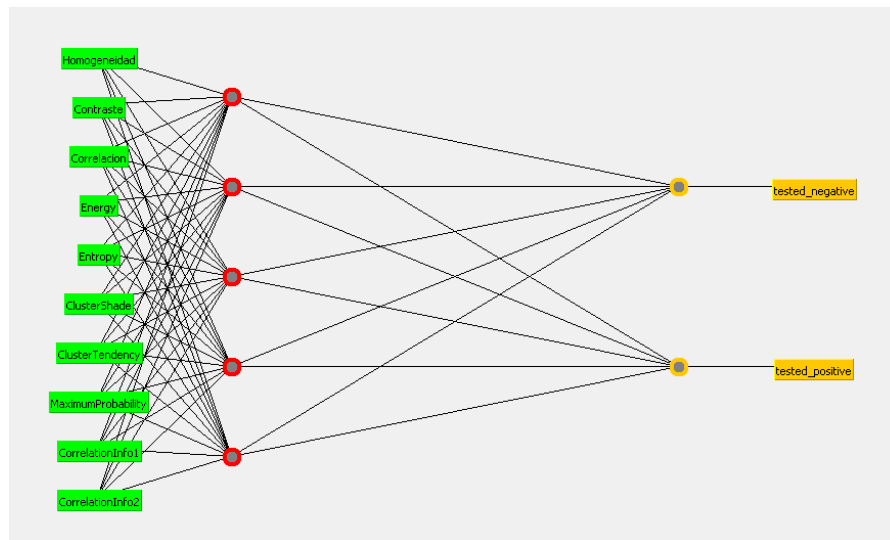
Para la clasificación de las muestras obtenidas en base a las características extraídas por los descriptores se ha escogido como clasificador a la red neuronal perceptrón multicapa, teniendo en cuenta las siguientes características:

El número de neuronas de entrada es igual al número características extraídas por imagen, dichas características han sido extraídas dado que son las necesarias para realizar un análisis por textura en una imagen.

Red neuronal multicapa: con una estructura de 1 capa de entrada (10 neuronas), 1 capa oculta (5 neuronas) y 1 capa de salida (2 neuronas).

Para determinar el número de neuronas ocultas además de seguir la Regla de la capa oculta-capa entrada, la cual indica que el número de neuronas ocultas no debe ser mayor al doble del número de neuronas de la capa de entrada; se realizaron pruebas con distintos números de neuronas ocultas hasta llegar al número actual.

Figura 46 Estructura de la red neuronal empleada



Fuente: Elaboración propia.

Normalmente una neurona recibe muchas entradas, como se puede apreciar en el gráfico, cada neurona de la capa oculta recibe 10 entradas, cada una de estas entradas tiene un peso (también llamado peso sináptico), estos se van ajustando en el proceso de aprendizaje hasta conseguir los valores adecuados. Una vez que ha finalizado el proceso de aprendizaje, los pesos finales pueden ser utilizados para posteriores evaluaciones, así evitando el tiempo del entrenamiento de la red. En cada nodo se aplica una función de activación de tipo sigmoideal definida por la función:

$$f(X) = \frac{1}{1 + \exp(-x)}$$

La Formula general para el cálculo de la salida es :

$$S_i^{(j)} = X^{(j-1)} * \omega_i^{(j)}$$

Las capas Intermedias u ocultas están definidas por la formula :

$$\varepsilon_i^{(j)} = (d - f) \left[\frac{df}{dS_1^{(j+1)}} \frac{dS_1^{(j+1)}}{dS_1^{(j)}} + \dots + \frac{df}{dS_{m(j+1)}^{(j+1)}} \frac{dS_{m(j+1)}^{(j+1)}}{dS_1^{(j)}} \right]$$



La regla genérica para el calculo de pesos es :

$$\omega_i^{(j)} = \omega_i^{(j)} + C_i^{(j)} \epsilon_i^{(j)} X^{(j-1)}$$

Los pesos finales de cada nodo para la red percetrón multicapa se describen a continuación:

Figura 47 Primer nodo de la capa Oculta (Nodo 2)

Entradas	Pesos
Threshold	2.2493469124557333
Homogeneidad	7.94195593884816
Contraste	9.575667941632165
Correlacion	-5.343624710527782
Energy	-2.6496016754125953
Entropy	-3.6914052323896174
ClusterShade	-4.071543446787338
ClusterTendency	-1.173557556399002
MaximumProbability	-2.363903940896615
CorrelationInfo1	2.02070519065106
CorrelationInfo2	1.9961557362617017

Fuente: Elaboración propia

Figura 48 Segundo nodo de la capa Oculta (Nodo 3)

Entradas	Pesos
Threshold	-0.5177855561513812
Homogeneidad	-3.949933002267905
Contraste	3.2493451540353315
Correlacion	3.2479446090005455
Energy	-0.6847571025103435
Entropy	2.2761035064304687
ClusterShade	-0.5319090100590332
ClusterTendency	1.7420950594484008
MaximumProbability	-2.1827107101072682
CorrelationInfo1	-0.2276014333267621
CorrelationInfo2	-0.47860308076072905

Fuente: Elaboración propia

Figura 49 Tercer nodo de la capa Oculta (Nodo 4)

Entradas	Pesos
Threshold	-2.8646123007085733



Homogeneidad	1.8544563931256037
Contraste	-7.822839155584398
Correlacion	11.90325030561907
Energy	0.9624398022032487
Entropy	-2.7444550807356967
ClusterShade	-2.8303817227990122
ClusterTendency	7.332945619777911
MaximumProbability	-2.2671125730791655
CorrelationInfo1	-3.2045339499280976
CorrelationInfo2	-5.977516509028067

Figura 50 Cuarto nodo de la capa Oculta (Nodo 5)

Entradas	Pesos
Threshold	0.7760173795676261
Homogeneidad	-1.027289005485402
Contraste	2.2047321390910746
Correlacion	-0.7853830651478403
Energy	-1.174787283741341
Entropy	0.972796559506289
ClusterShade	-1.1901509791431342
ClusterTendency	8.91356553302307
MaximumProbability	-3.4464349229472324
CorrelationInfo1	-5.92339988967316
CorrelationInfo2	-4.530654407238607

Fuente: Elaboración propia.

Figura 51 Quinto nodo de la capa Oculta (Nodo 6)

Entradas	Pesos
Threshold	5.9715324251779744
Homogeneidad	-6.228533972942408
Contraste	-2.702537757167148
Correlacion	-4.221441707710835
Energy	-4.98820099250036
Entropy	-4.53189572042234
ClusterShade	-3.2506239848248226
ClusterTendency	-2.7513589956610778
MaximumProbability	-2.7018274027661615
CorrelationInfo1	5.202107512002328
CorrelationInfo2	-4.630568013347345

Fuente: Elaboración propia.

Figura 52 Primer nodo de la capa de Salida (Nodo 0)



Entradas	Pesos
Threshold	5.9715324251779744
Nodo 2	-6.228533972942408
Nodo 3	-2.702537757167148
Nodo 4	-4.221441707710835
Nodo 5	-4.98820099250036
Nodo 6	-3.2506239848248226

Fuente: Elaboración propia.

Figura 53 Segundo nodo de la capa de Salida (Nodo 1)

Entradas	Pesos
Threshold	-5.971475262771781
Nodo 2	6.228349811229796
Nodo 3	2.7025146120711874
Nodo 4	4.221404573785881
Nodo 5	4.988141595759148
Nodo 6	3.2505975864183103

Fuente: Elaboración propia.

Evaluación

Para esta sección se utilizó la validación cruzada de 10 iteraciones, donde el número total de imágenes (189 imágenes), es dividido en 10 grupos, por lo tanto, se han creado 9 grupos de 18 imágenes y un grupo de 9 imágenes. Finalmente, para cada iteración es escogido cualquier grupo de imágenes para la validación y el resto de grupos de imágenes para el entrenamiento.

Figura 54 Método de Validación Cruzada

1	2	3	4	5
Train	Train	Validation	Train	Train

Fuente: (Hastie, Tibshirani, & Friendman)



Resultados de las pruebas con redes neuronales

Correctamente clasificadas: 135 (71.4286%)

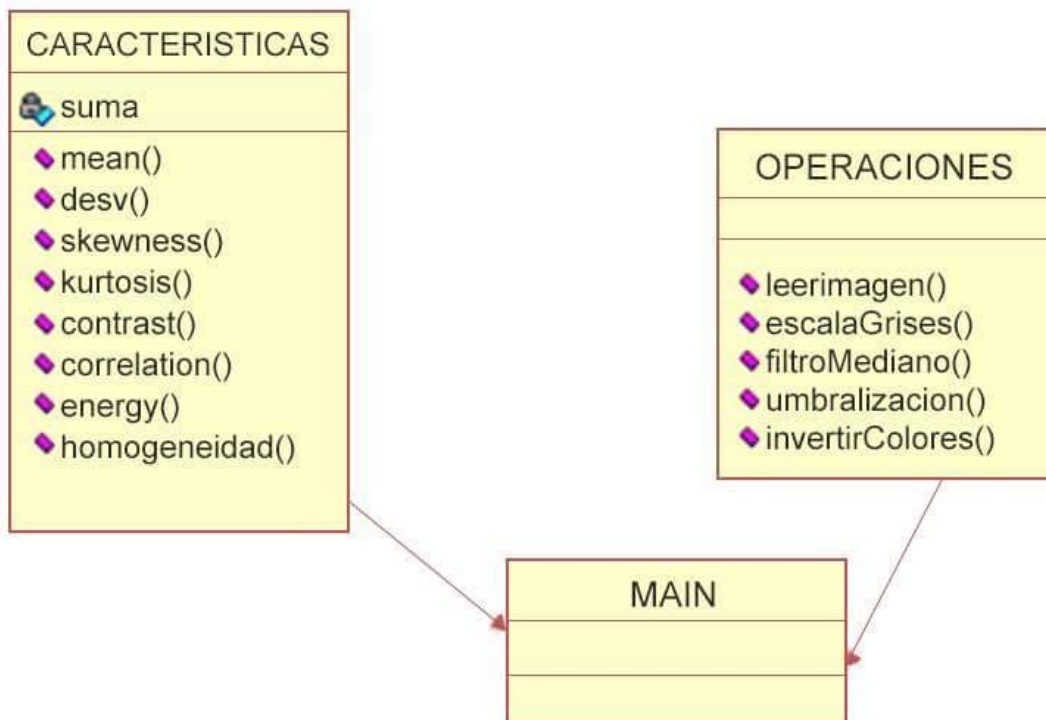
Incorrectamente clasificadas : 54 (28.5714%)

Figura 55 Resultados de las pruebas con redes neuronales

	Precisión	Valor F	Revocación	Clase
	0.658	0.658	0.658	No Melanomas
	0.7555	0.755	0.755	Melanomas
Promedio	0.714	0.714	0.714	

Fuente: Elaboración propia

Figura 56 Diagrama de clases del software implementado



VI. CONCLUSIONES

6.1 Conclusiones

- a. Se identificó que las características para detectar el melanoma son la asimetría, a diferencia del lunar benigno, el melanoma es asimétrico, esto se verifica trazando una línea vertical al centro de la lesión, otra característica es el borde, el melanoma presenta bordes irregulares, otra característica es el color, a diferencia de los lunares benignos que son de un solo color, los melanomas pueden ser de color rojo, blanco y azul, otra característica es el diámetro, a diferencia de los lunares benignos, los melanomas tienen un diámetro mayor a 6mm, y la evolución, que es la variación del tamaño del lunar en el tiempo.
- b. Se analizó distintas técnicas de procesamiento de imágenes, las cuales fueron: conversión a escalas escala de grises; redimensionado de imagen; inversión de colores; filtros (Gaussiano, Mediano); operaciones Morfológicas (Cierre, Dilatación, Erosión); ecualización de histograma para ajuste de contraste; Binarización (Otsu y Manual), de los cuales el método 1 con técnicas de conversión a escala de grises, filtro mediano y la técnica de binarización de máxima entropía, obtiene un 55.56% de acierto, en la segunda combinación, se unen las técnicas usadas anteriormente con la ecualización de histograma con un porcentaje de acierto de 56.08%, la tercera combinación utiliza las técnicas de conversión a escala de grises, filtro gaussiano y la técnica de binarización de máxima entropía, con un porcentaje de acierto de 60.84%, la cuarta combinación para este método, consiste en la unión

de la tercera combinación con la técnica de ecualización de histograma, el cual obtiene un porcentaje de acierto de 57.14%.

Todas estas combinaciones de técnicas no alcanzan el porcentaje de precisión del segundo ni tercer método los cuales fueron de 61.90% y 71.4286% respectivamente, por lo que se asume que el último método es el que contiene una mejor combinación de técnicas.

- c. Se implementó distintas técnicas de procesamiento de imágenes, las cuales fueron: conversión a escalas escala de grises; redimensionado de imagen; inversión de colores; filtros (Gaussiano, Mediano); operaciones Morfológicas (Cierre, Dilatación, Erosión); ecualización de histograma para ajuste de contraste; Binarización (Otsu y Manual), de las técnicas implementadas se ha identificado que la unión de la técnica de escala de grises con la extracción de características usando descriptores de GLCM logra un porcentaje de 71.4286% .
- d. Se identificó que las técnicas que brindan mayor efectividad son: conversión a escala de grises junto a la extracción de características usando GLCM, las cuales indican un porcentaje de efectividad de 71.4286%, a diferencia de los otros 2 métodos desarrollados que tienen un promedio de 59.2593% de efectividad.
- e. Se integraron las siguientes técnicas: conversión a escala de grises junto a la extracción de características usando GLCM; las cuales se implementaron en un método propuesto obteniendo un porcentaje de precisión del 71.4286 %.

6.2 Recomendaciones

Realizar pruebas de efectividad en el reconocimiento de melanomas en imágenes utilizando otro tipo de descriptores (forma, color y/o tamaño).

Realizar pruebas con imágenes obtenidas bajo situaciones controladas y con tamaño y resoluciones idénticas.

VII. ANEXOS

Código fuente del método propuesto

Características.h

```
1  #ifndef CARACTERISTICAS_H
2  #define CARACTERISTICAS_H
3
4
5  #include <iostream>
6  #include "opencv2/opencv.hpp"
7  #include "opencv2/highgui/highgui.hpp"
8  #include "opencv2/imgproc/imgproc.hpp"
9  #include "opencv/cvaux.h"
10
11 using namespace cv;
12 using namespace std;
13
14 class Caracteristicas
15 {
16 private:
17     double suma = 0.0;
18
19 public:
20     Caracteristicas();
21     double mean(Mat src);
22
23     double desv(Mat src);
24
25     double skewness(Mat src);
26     double skewness(Mat src, double mean, double desv);
27
28     double kurtosis(Mat src);
29     double kurtosis(Mat src, double mean, double desviacion);
30
31
32     double contrast(Mat src);
33     double correlation(Mat src);
34     double energy(Mat src);
35     double homogeneidad(Mat src);
36     //void glcm(Mat &img);
37     vector<double> Caracteristicas::glcmCV(Mat src);
38
39 };
40
41 #endif // CARACTERISTICAS_H
42
```



Caracteristicas.cpp

```
1  #include "caracteristicas.h"
2
3  Caracteristicas::Caracteristicas()
4  {
5
6  }
7
8  double Caracteristicas::mean(Mat src)
9  {
10
11     cv::Scalar mean,stddev;
12     meanStdDev(src,mean,stddev,cv::Mat());
13     return mean.val[0];
14
15 }
16
17 double Caracteristicas::desv(Mat src)
18 {
19     cv::Scalar mean,stddev;
20     meanStdDev(src,mean,stddev,cv::Mat());
21     return stddev.val[0];
22
23 }
24
25
26 double Caracteristicas::skewness(Mat src)
27 {
28     this->suma = 0.0;
29     double meanValue = mean(src);
30     double desvValue = desv(src);
31
32     for (int j = 0; j < src.rows; j++) {
33         uchar* data = src.ptr<uchar>(j);
34         for (int i = 0; i < src.cols; i++) {
35             int valor = data[i];
36             double superior = pow((valor - meanValue), 3);
37             double inferior = pow(desvValue, 3);
38             double div = superior / inferior;
39             this->suma = this->suma + div;
40         }
41     }
42     return this->suma / (src.rows * src.cols);
43
44 }
```

```

48 double Caracteristicas::skewness(Mat src, double mean, double desv)
49 {
50     this->suma = 0.0;
51     for (int j = 0; j < src.rows; j++) {
52         uchar* data = src.ptr<uchar>(j);
53         for (int i = 0; i < src.cols; i++) {
54             int valor = data[i];
55             double superior = pow((valor - mean), 3);
56             double inferior = pow(desv, 3);
57             double div = superior / inferior;
58             this->suma = this->suma + div;
59         }
60     }
61     return this->suma / (src.rows * src.cols);
62 }
63
64 double Caracteristicas::kurtosis(Mat src)
65 {
66     this->suma = 0.0;
67     double meanValue = mean(src);
68     double desvValue = desv(src);
69
70     for (int j = 0; j < src.rows; j++) {
71         uchar* data = src.ptr<uchar>(j);
72         for (int i = 0; i < src.cols; i++) {
73             int valor = data[i];
74             double superior = pow((valor - meanValue), 4);
75             double inferior = pow(desvValue, 4);
76             double div = (superior / inferior) - 3;
77             this->suma = this->suma + div;
78         }
79     }
80     return this->suma / (src.rows * src.cols);
81
82 }
83 double Caracteristicas::kurtosis(Mat src, double mean, double desv)
84 {
85     this->suma = 0.0;
86
87
88     for (int j = 0; j < src.rows; j++) {
89         uchar* data = src.ptr<uchar>(j);
90         for (int i = 0; i < src.cols; i++) {
91             int valor = data[i];
92             double superior = pow((valor - mean), 4);
93             double inferior = pow(desv, 4);
94             double div = (superior / inferior) - 3;

```



```
95         this->suma = this->suma + div;
96     }
97 }
98 return this->suma / (src.rows * src.cols);
99
100 }
101
102 double Caracteristicas::contrast(Mat srcc)
103 {
104     IplImage* src=cvCloneImage(&(IplImage)srcc);
105     CvGLCM* glcm;
106     glcm = cvCreateGLCM(src, 2, NULL,4, CV_GLCM_OPTIMIZATION_NONE);
107     cvCreateGLCMDescriptors(glcm, CV_GLCMDESC_OPTIMIZATION_ALLOWDOUBLENES);
108     double contraste = cvGetGLCMDescriptor(glcm, 0, CV_GLCMDESC_CONTRAST);
109     //cvReleaseGLCM(&glcm);
110     cvReleaseImage(&src);
111     return contraste;
112 }
113
114
115 double Caracteristicas::energy(Mat srcc)
116 {
117     IplImage* src=cvCloneImage(&(IplImage)srcc);
118     CvGLCM* glcm;
119     glcm = cvCreateGLCM(src, 2, NULL,4, CV_GLCM_OPTIMIZATION_NONE);
120     cvCreateGLCMDescriptors(glcm, CV_GLCMDESC_OPTIMIZATION_ALLOWDOUBLENES);
121     double energia = cvGetGLCMDescriptor(glcm, 0, CV_GLCMDESC_ENERGY);
122     //cvReleaseGLCM(&glcm);
123     cvReleaseImage(&src);
124     return energia;
125 }
126
127 }
128
129 double Caracteristicas::correlation(Mat srcc)
130 {
131     IplImage* src=cvCloneImage(&(IplImage)srcc);
132     CvGLCM* glcm;
133     glcm = cvCreateGLCM(src, 2, NULL,4, CV_GLCM_OPTIMIZATION_NONE);
134     cvCreateGLCMDescriptors(glcm, CV_GLCMDESC_OPTIMIZATION_ALLOWDOUBLENES);
135     double correlation = cvGetGLCMDescriptor(glcm, 0, CV_GLCMDESC_CORRELATION);
136     cvReleaseImage(&src);
137     return correlation;
138 }
139
140 }
141
```




```
142 double Caracteristicas::homogeneidad(Mat srcc)
143 {
144     IplImage* src = cvCloneImage(&(IplImage)srcc);
145     CvGLCM* glcm;
146     glcm = cvCreateGLCM(src, 2, NULL,4, CV_GLCM_OPTIMIZATION_NONE);
147     cvCreateGLCMDescriptors(glcm, CV_GLCMDESC_OPTIMIZATION_ALLOWDOUBLENEST);
148     double homogeneity = cvGetGLCMDescriptor(glcm, 0, CV_GLCMDESC_HOMOGENITY);
149     //cvReleaseGLCM(&glcm);
150     cvReleaseImage(&src);
151     return homogeneity;
152 }
153
154 vector<double> Caracteristicas::glcmCV(Mat srcc)
155 {
156     vector<double> retorno;//vector de características
157     IplImage* src = cvCloneImage(&(IplImage)srcc);
158     CvGLCM* glcm;
159     glcm = cvCreateGLCM(src, 2, NULL,4, CV_GLCM_OPTIMIZATION_NONE);
160     cvCreateGLCMDescriptors(glcm, CV_GLCMDESC_OPTIMIZATION_ALLOWDOUBLENEST);
161     retorno.push_back(cvGetGLCMDescriptor(glcm, 0, CV_GLCMDESC_HOMOGENITY));
162     retorno.push_back(cvGetGLCMDescriptor(glcm, 0, CV_GLCMDESC_CONTRAST));
163     retorno.push_back(cvGetGLCMDescriptor(glcm, 0, CV_GLCMDESC_CORRELATION));
164     retorno.push_back(cvGetGLCMDescriptor(glcm, 0, CV_GLCMDESC_ENERGY));
165     retorno.push_back(cvGetGLCMDescriptor(glcm, 0, CV_GLCMDESC_ENTROPY));
166     retorno.push_back(cvGetGLCMDescriptor(glcm, 0, CV_GLCMDESC_CLUSTERSHADE));
167     retorno.push_back(cvGetGLCMDescriptor(glcm, 0, CV_GLCMDESC_CLUSTERTENDENCY));
168     retorno.push_back(cvGetGLCMDescriptor(glcm, 0, CV_GLCMDESC_MAXIMUMPROBABILITY));
169     retorno.push_back(cvGetGLCMDescriptor(glcm, 0, CV_GLCMDESC_CORRELATIONINFO1));
170     retorno.push_back(cvGetGLCMDescriptor(glcm, 0, CV_GLCMDESC_CORRELATIONINFO2));
171     cvReleaseImage(&src);
172     return retorno;
173 }
174
```



Operaciones.h

```
1 #ifndef OPERACIONES_H
2 #define OPERACIONES_H
3
4 #include <iostream>
5 #include "opencv2/opencv.hpp"
6 #include "opencv2/highgui/highgui.hpp"
7 #include "opencv2/imgproc/imgproc.hpp"
8 #include "opencv2/ml/ml.hpp"
9
10 using namespace cv;
11 using namespace std;
12
13 class Operaciones
14 {
15
16 public:
17
18     Operaciones();
19     Mat leerImagen(String ruta);
20     Mat escalaGris(Mat src);
21     Mat filtroMediano(Mat src, int tamaño);
22     Mat umbralizacion(Mat src, int min = 0, int max = 255, int tipo = CV_THRESH_OTSU);
23     Mat invertirColores(Mat src);
24
25     float evaluate(cv::Mat& predicted, cv::Mat& actual);
26     cv::Mat labelData(cv::Mat points, bool e);
27     //void plot_binary(cv::Mat& data, cv::Mat& classes, string name);
28     void plot_binary(cv::Mat& data, cv::Mat& classes, string name, int size = 200);
29     //CvANN_MLP mlp(cv::Mat& trainingData, cv::Mat& trainingClasses);
30     CvANN_MLP mlp(cv::Mat& trainingData, cv::Mat& trainingClasses, cv::Mat& testData, cv::Mat& testClasses);
31 };
32
33 #endif // OPERACIONES_H
34
```



Operaciones.cpp

```
1  #include "operaciones.h"
2
3  Operaciones::Operaciones()
4  {
5
6  }
7  Mat Operaciones::leerImagen(String ruta)
8  {
9      return imread(ruta);
10 }
11 Mat Operaciones::escalaGris(Mat src)
12 {
13     cvtColor(src,src,CV_RGB2GRAY);
14     return src;
15 }
16 Mat Operaciones::filtroMediano(Mat src,int tamano)
17 {
18     medianBlur(src,src,tamano);
19     return src;
20 }
21 Mat Operaciones::invertirColores(Mat src)
22 {
23     bitwise_not(src,src);
24     return src;
25 }
26 Mat Operaciones::umbralizacion(Mat src,int min,int max,int tipo)
27 {
28     threshold(src, src, min,max ,tipo);
29     return src;
30 }
31 |
```



Main.cpp

```
#include "mainwindow.h"
#include "ui_mainwindow.h"

#include "operaciones.h"
#include "caracteristicas.h"
#include <QDir>
#include <QString>
#include <QVector>
#include <iostream>
#include <ctime>
#include <sys/timeb.h>
#include <QFileDialog>
#include <QMessageBox>
#include <QProcess>
#include <fstream>
using namespace std;

MainWindow::MainWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::MainWindow) {}

MainWindow::~MainWindow() {}

void MainWindow::procesar()
{
    QString rutaMelanomas = ui->txtMelanomas->text().trimmed() ;
    QString rutaNoMelanomas = ui->txtNoMelanomas->text().trimmed() ;
    if ( rutaMelanomas != "" && rutaNoMelanomas != "" )
    {
        QDir mel(rutaMelanomas);
        QDir nomel(rutaNoMelanomas);
        if(mel.exists() && nomel.exists()){
            ofstream fs("C:\\Users\\Diego Campos\\Desktop\\glcmmethod.arff");
            fs << "@relation pima melanoma" << endl;
        }
    }
}
```



```

fs << "@attribute 'Homogeneidad' real" << endl;
fs << "@attribute 'Contraste' real" << endl;
fs << "@attribute 'Correlacion' real" << endl;
fs << "@attribute 'Energy' real" << endl;
fs << "@attribute 'Entropy' real" << endl;
fs << "@attribute 'ClusterShade' real" << endl;
fs << "@attribute 'ClusterTendency' real" << endl;
fs << "@attribute 'MaximumProbability' real" << endl;
fs << "@attribute 'CorrelationInfo1' real" << endl;
fs << "@attribute 'CorrelationInfo2' real" << endl;
fs << "@attribute 'class' {tested_negative, tested_positive}" << endl;
fs << "@data" << endl;
clock_t startC, finishC;
struct timeb startT, finishT;
unsigned int seconds, milliseconds;

startC = clock();
ftime(&startT);
// QApplication a(argc, argv);
// return a.exec();
/// [+] ENTRENAMIENTO
///     [+] MELANOMAS
///     [+] NOMELANOMAS
///

Operaciones op;
Caracteristicas carac;
double contraste,mean,desv,correlacion,energy,kurtosis,skewness,
homogeneidad;
//vector<Mat> imagenes;
Mat trainingData(0,10,CV_32FC1);
Mat trainingClasses(0,1,CV_32FC1);
Mat testData(0,9,CV_32FC1);

/// [+] ENTRENAMIENTO
///     [+] MELANOMAS
///     [+] NOMELANOMAS
///

Operaciones op;
Caracteristicas carac;
double contraste,mean,desv,correlacion,energy,kurtosis,skewness,
homogeneidad;
//vector<Mat> imagenes;
Mat trainingData(0,10,CV_32FC1);
Mat trainingClasses(0,1,CV_32FC1);
Mat testData(0,9,CV_32FC1);
Mat testClasses(0,1,CV_32FC1);
int limitTestData = 2;

QVector<QString> dirs; //array de carpetas
float e = -1.0f; // -1 : melanoma 1: nomelanoma

dirs.append(rutaMelanomas);
dirs.append(rutaNoMelanomas);

for (int var = 0; var < dirs.size(); ++var) { //RECORRER LAS CARPETAS

int a = 0;//contador de imagenes
if ( var == 0)
{
//cout << "LECTURA DE MELANOMAS ";
}
else
{

```

```

//cout << "LECTURA DE NO MELANOMAS ";
e = 1.0f;//cambio de folder
}

QDir dir(dirs.at(var));
dir.setNameFilters(QStringList() << "*.png" << "*.jpg");
QStringList fileList = dir.entryList();
Mat dataInput(dir.count(),10,CV_32FC1);//almacen de características de la imagen.
Mat dataTestInput(2,9,CV_32FC1);//almacen de características de la imagen.
// cout << "Inicio dataInput \n" ;
//cout << dataInput.rows;
// cout << "-----";
//Mat dataClasses(dir.count(),1,CV_32FC1);//

string fulluri = "";
vector<double> res;
foreach (QString path, fileList)
{
    fulluri = dirs.at(var).toStdString()+path.toStdString();
    //cout << "RUTA: " << fulluri << endl;
    Mat output_img = imread(fulluri);
    if (output_img.empty()) {
        cerr << "\nERROR: Can't be loaded image" << fulluri << endl;
        EXIT_FAILURE;
    }

    //////////// PREPROCESAMIENTO

    //Mat output_img = img.clone();
    //Se convierte a escala de grises
    output_img = op.escalaGrises(output_img);
    //equalizeHist(output_img,output_img);
    //GaussianBlur(output_img,output_img,Size(3,3),0);
    //Se aplica filtro mediano

    //////////// CARACTERISTICAS
    //carac.glcm(output_img);
    //cout << "-----";
    //cout << " Num. " << a << endl;

    try{
        mean = carac.mean(output_img);
        //cout << mean << ",";

        desv = carac.desv(output_img);
        //cout << desv << ",";

        skewness = carac.skewness(output_img,mean,desv);
        //cout << skewness<< ",";
        kurtosis = carac.kurtosis(output_img,mean,desv);
        //cout << kurtosis << ",";

        res = carac.glcmCV(output_img);
        /*
        if ( e == 0)
        {
            cout << "tested_positive" << endl;
        }else{
            cout << "tested_negative" << endl;
        }
        */
    }catch(std::exception& e)
    {
        std::cout<< e.what() << std::endl;
    }

    }catch(...){
        std::cout << "damn" << std::endl;
    }
}

```



```

//Agregando Resultados al Input de la RN
dataInput.at<float>(a,0)= res.at(0) ;//HOMOGENITY
dataInput.at<float>(a,1)= res.at(1) ;//CONTRAST
dataInput.at<float>(a,2)= res.at(2) ;//CORRELATION
dataInput.at<float>(a,3)= res.at(3) ;//ENERGY
dataInput.at<float>(a,4)= res.at(4) ;//ENTROPIA
dataInput.at<float>(a,5)= res.at(5) ;//CLUSTERSHADE
dataInput.at<float>(a,6)= res.at(6) ;//CLUSTERTENDENCY
dataInput.at<float>(a,7)= res.at(7) ;//MAXIMUMPROBABILITY
dataInput.at<float>(a,8)= res.at(8) ;//CORRELATIONINFO1
dataInput.at<float>(a,9)= res.at(9) ;//CORRELATIONINFO2

fs <<dataInput.at<float>(a,0) << ",";
fs <<dataInput.at<float>(a,1) << ",";
fs <<dataInput.at<float>(a,2) << ",";
fs <<dataInput.at<float>(a,3) << ",";
fs <<dataInput.at<float>(a,4) << ",";
fs <<dataInput.at<float>(a,5) << ",";
fs <<dataInput.at<float>(a,6) << ",";
fs <<dataInput.at<float>(a,7) << ",";
fs <<dataInput.at<float>(a,8) << ",";
fs <<dataInput.at<float>(a,9) << ",";

if ( e == -1.0f)
{
fs << "tested_positive" << endl;
}else
{
fs << "tested_negative" << endl;
}

//dataClasses.at<float>(a,0)=e;

for (int var = 0; var < 10; ++var)
{
//printf("a = %i \n",a);
//printf("dataInput[%i] = %f \n",var,dataInput.at<float>(a,var));
//printf("-----");

//ui->txtArea->append(QString("a = %1 \n").arg(a));
//ui->txtArea->append(QString("dataInput[%1] = %2 \n").arg(var,dataInput.at<float>(a,var)));
//ui->txtArea->append("-----\n");

}

a++;//contador de imagenes

} //FIN DE CARPETA

for (int varr = 0; varr < dataInput.rows; varr++)
{
trainingData.push_back(dataInput.row(varr));
trainingClasses.push_back(e);
}

} // FIN DE LA LECTURA DE LAS CARPETAS
fs.close();
QProcess process;
process.start("java -cp \"C:\\Program Files (x86)\\Weka-3-6\\weka.jar\" weka.classifiers.functions.Multilayer
process.waitForFinished(-1); // will wait forever until finished

QByteArray stdouts = process.readAllStandardOutput();
QByteArray stderrs = process.readAllStandardError();

ui->txtArea->append(QString::fromUtf8( stdouts.data() ));

```



```

        finishC = clock();
        ftime(&finishT);
        seconds = finishT.time - startT.time - 1;
        milliseconds = (1000 - startT.millitm) + finishT.millitm;
        cout << "Time (clock): " << (finishC - startC)/CLOCKS_PER_SEC << endl;
        //cout << "Time (difftime): " << difftime(finishC, startC) << endl;
        cout << "Time (ftime): " << (milliseconds + seconds * 1000) << endl;
    }else{
        showError("Error, ingrese directorios válidos");
    }

    }else{
        showError("Ingrese los directorios de los dataset");
    }
}

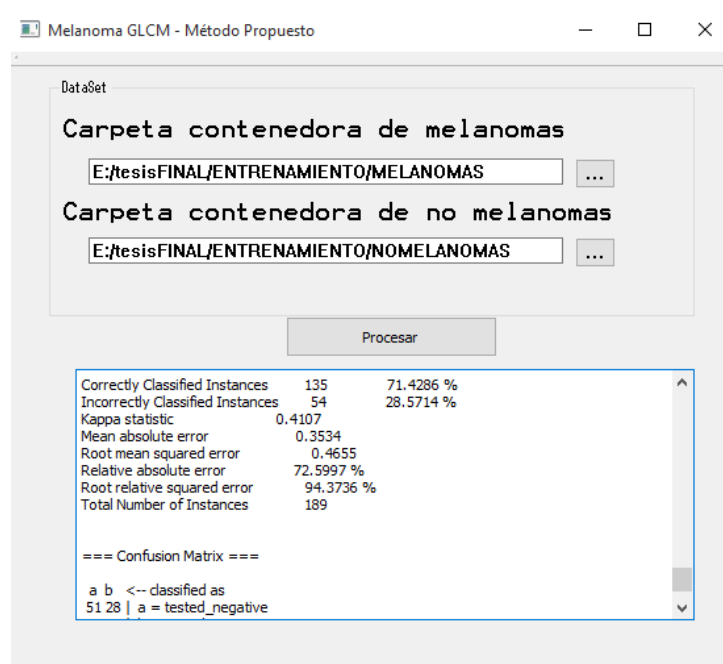
void MainWindow::showError(QString msg2)
{
    QMessageBox msg;
    msg.setText(msg2);
    msg.exec();
}

void MainWindow::on_btnMelanoma_clicked()
{
    QString directorio = QFileDialog::getExistingDirectory(this, tr("Open Directory"),
        "E:\\tesisFINAL\\ENTRENAMIENTO",
        QFileDialog::ShowDirsOnly
        | QFileDialog::DontResolveSymlinks);

    ui->txtMelanomas->setText(directorio);
}

```

Interfaz principal



Resultado de la ejecución

Options: -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H 5

Sigmoid Node 0

Inputs Weights

Threshold 5.9715324251779744

Node 2 -6.228533972942408

Node 3 -2.702537757167148

Node 4 -4.221441707710835

Node 5 -4.98820099250036

Node 6 -3.2506239848248226

Sigmoid Node 1

Inputs Weights

Threshold -5.971475262771781

Node 2 6.228349811229796

Node 3 2.7025146120711874

Node 4 4.221404573785881

Node 5 4.988141595759148

Node 6 3.2505975864183103

Sigmoid Node 2

Inputs Weights

Threshold 2.2493469124557333

Attrib Homogeneidad 7.94195593884816

Attrib Contraste 9.575667941632165

Attrib Correlacion -5.343624710527782

Attrib Energy -2.6496016754125953

Attrib Entropy -3.6914052323896174

Attrib ClusterShade -4.071543446787338

Attrib ClusterTendency -1.173557556399002

Attrib MaximumProbability -2.363903940896615

Attrib CorrelationInfo1 2.02070519065106

Attrib CorrelationInfo2 1.9961557362617017

Sigmoid Node 3

Inputs Weights

Threshold -0.5177855561513812

Attrib Homogeneidad -3.949933002267905

Attrib Contraste 3.2493451540353315

Attrib Correlacion 3.2479446090005455

Attrib Energy -0.6847571025103435

Attrib Entropy 2.2761035064304687

Attrib ClusterShade -0.5319090100590332

Attrib ClusterTendency 1.7420950594484008

Attrib MaximumProbability -2.1827107101072682

Attrib CorrelationInfo1 -0.2276014333267621

Attrib CorrelationInfo2 -0.47860308076072905

Sigmoid Node 4



Inputs Weights

Threshold -2.8646123007085733
Attrib Homogeneidad 1.8544563931256037
Attrib Contraste -7.822839155584398
Attrib Correlacion 11.90325030561907
Attrib Energy 0.9624398022032487
Attrib Entropy -2.7444550807356967
Attrib ClusterShade -2.8303817227990122
Attrib ClusterTendency 7.332945619777911
Attrib MaximumProbability -2.2671125730791655
Attrib CorrelationInfo1 -3.2045339499280976
Attrib CorrelationInfo2 -5.977516509028067
Sigmoid Node 5

Inputs Weights

Threshold 0.7760173795676261
Attrib Homogeneidad -1.027289005485402
Attrib Contraste 2.2047321390910746
Attrib Correlacion -0.7853830651478403
Attrib Energy -1.174787283741341
Attrib Entropy 0.972796559506289
Attrib ClusterShade -1.1901509791431342
Attrib ClusterTendency 8.91356553302307
Attrib MaximumProbability -3.4464349229472324
Attrib CorrelationInfo1 -5.92339988967316
Attrib CorrelationInfo2 -4.530654407238607

Sigmoid Node 6

Inputs Weights

Threshold -0.11387391965754406
Attrib Homogeneidad 0.36940158723770433
Attrib Contraste 6.374322846005856
Attrib Correlacion -0.5554133191581548
Attrib Energy -0.33314905371769615
Attrib Entropy -4.53189572042234
Attrib ClusterShade 1.6710504619781137
Attrib ClusterTendency -2.7513589956610778
Attrib MaximumProbability -2.7018274027661615
Attrib CorrelationInfo1 5.202107512002328
Attrib CorrelationInfo2 -4.630568013347345

Class tested_negative

Input

Node 0

Class tested_positive

Input

Node 1

Time taken to build model: 0.33 seconds

Time taken to test model on training data: 0.01 seconds



==== Error on training data ====

Correctly Classified Instances 149 78.836 %
Incorrectly Classified Instances 40 21.164 %
Kappa statistic 0.5506
Mean absolute error 0.29
Root mean squared error 0.39
Relative absolute error 59.5764 %
Root relative squared error 79.0796 %
Total Number of Instances 189

==== Confusion Matrix ====

a b <-- classified as
50 29 | a = tested_negative
11 99 | b = tested_positive

==== Stratified cross-validation ====

Correctly Classified Instances 135 71.4286 %
Incorrectly Classified Instances 54 28.5714 %
Kappa statistic 0.4107
Mean absolute error 0.3534
Root mean squared error 0.4655
Relative absolute error 72.5997 %
Root relative squared error 94.3736 %
Total Number of Instances 189

==== Confusion Matrix ====

a b <-- classified as
51 28 | a = tested_negative
26 84 | b = tested_positive

Combinaciones de técnicas realizadas

Combinación N° 01

conversión a escala de grises

filtro mediano



binarización de máxima entropía

descriptores : Mean, Desviación Estándar, Skweness, Kurtosis, Contraste

Promedio, Correlación, Energy y Homogeneidad.

porcentaje de acierto 55.56% usando clasificador red neuronal backpropagation

1 capa oculta con 12 neuronas

Combinación N° 02

conversión a escala de grises

filtro mediano

binarización de máxima entropía

descriptores : Mean, Desviación Estándar, Skweness, Kurtosis, Contraste

Promedio, Correlación, Energy y Homogeneidad.

porcentaje de acierto 53.97% usando redes bayesianas con el algoritmo de

busqueda k2 y el estimador simple (Alpha = 0.5)

Combinación N° 03

conversión a escala de grises

filtro mediano

binarización de máxima entropía

descriptores : Mean, Desviación Estándar, Skweness, Kurtosis, Contraste

Promedio, Correlación, Energy y Homogeneidad.

porcentaje de acierto 54.50% usando knn = 1

Combinación N° 04



conversión a escala de grises

ecualización de histograma

filtro mediano

binarización de máxima entropía

descriptores : Mean, Desviación Estándar, Skweness, Kurtosis, Contraste

Promedio, Correlación, Energy y Homogeneidad.

porcentaje de acierto de 56.08% usando clasificador red neuronal

backpropagation 1 capa oculta con 12 neuronas

Combinación N° 05

conversión a escala de grises

ecualización de histograma

filtro mediano

binarización de máxima entropía

descriptores : Mean, Desviación Estándar, Skweness, Kurtosis, Contraste

Promedio, Correlación, Energy y Homogeneidad.

porcentaje de acierto de 58.20% usando el clasificador usando redes

bayesianas con el algoritmo de búsqueda k2 y el estimador simple (Alpha = 0.5)

Combinación N° 06

conversión a escala de grises

ecualización de histograma

filtro mediano



binarización de máxima entropía

descriptores : Mean, Desviación Estándar, Skweness, Kurtosis, Contraste

Promedio, Correlación, Energy y Homogeneidad.

porcentaje de acierto de 55.56% usando clasificador knn = 1

Combinación N° 07

conversión a escala de grises

filtro gaussiano

binarización de máxima entropía

descriptores : Mean, Desviación Estándar, Skweness, Kurtosis, Contraste

Promedio, Correlación, Energy y Homogeneidad.

porcentaje de acierto de 60.85% usando clasificador red neuronal

backpropagation 1 capa oculta con 12 neuronas

Combinación N° 08

conversión a escala de grises

filtro gaussiano

binarización de máxima entropía

descriptores : Mean, Desviación Estándar, Skweness, Kurtosis, Contraste

Promedio, Correlación, Energy y Homogeneidad.

porcentaje de acierto de 55.56% usando como clasificador usando redes

bayesianas con el algoritmo de búsqueda k2 y el estimador simple (Alpha = 0.5)



Combinación N° 09

conversión a escala de grises

filtro gaussiano

binarización de máxima entropía

descriptores : Mean, Desviación Estándar, Skweness, Kurtosis, Contraste

Promedio, Correlación, Energy y Homogeneidad.

porcentaje de acierto de 55.03% usando clasificador knn=1

Combinación N° 10

conversión a escala de grises

ecualización de histograma

filtro gaussiano

binarización de máxima entropía

descriptores : Mean, Desviación Estándar, Skweness, Kurtosis, Contraste

Promedio, Correlación, Energy y Homogeneidad.

porcentaje de acierto de 57.14% usando clasificador red neuronal

backpropagation 1 capa oculta con 5 neuronas

Combinación N° 11

conversión a escala de grises

ecualización de histograma

filtro gaussiano

binarización de máxima entropía



descriptores : Mean, Desviación Estándar, Skweness, Kurtosis, Contraste
Promedio, Correlación, Energy y Homogeneidad.

porcentaje de acierto de 58.20% usando como clasificador usando redes
bayesianas con el algoritmo de busqueda k2 y el estimador simple (Alpha =
0.5)

Combinación N° 12

conversión a escala de grises

ecualización de histograma

filtro gaussiano

binarización de máxima entropía

descriptores : Mean, Desviación Estándar, Skweness, Kurtosis, Contraste
Promedio, Correlación, Energy y Homogeneidad.

porcentaje de acierto de 52.38% con knn = 1

Combinación N° 13

redimensionado de imagen

conversion escala de grises

descriptores: homogeneidad, contraste, correlación, energía, entropía,

clustershade, clustertendency, probabilidad máxima, información de correlación

1, información de correlación 2

porcentaje de acierto de 61.90% usando clasificador red neuronal

backpropagation 1 capa oculta con 5 neuronas



Combinación N° 14

redimensionado de imagen

conversion escala de grises

descriptores: homogeneidad, contraste, correlación, energía, entropía,

clustershade, clustertendency, probabilidad máxima, información de correlación

1, información de correlación 2

porcentaje de acierto de 68.25% usando como clasificador usando redes

bayesianas con el algoritmo de búsqueda k2 y el estimador simple (Alpha =

0.5)

Combinación N° 15

redimensionado de imagen

conversion escala de grises

descriptores: homogeneidad, contraste, correlación, energía, entropía,

clustershade, clustertendency, probabilidad máxima, información de correlación

1, información de correlación 2

porcentaje de acierto de 60.85% usando clasificador knn=1

Combinación N° 16

conversion escala de grises

descriptores: homogeneidad, contraste, correlación, energía, entropía,

clustershade, clustertendency, probabilidad máxima, información de correlación

1, información de correlación 2



porcentaje de acierto de 71.43% usando clasificador red neuronal

backpropagation 1 capa oculta con 5 neuronas

Combinación N° 17

conversion escala de grises

descriptores: homogeneidad, contraste, correlación, energía, entropía,

clustershade, clustertendency, probabilidad máxima, información de correlación

1, información de correlación 2

porcentaje de acierto de 70.37% usando clasificador usando redes bayesianas

con el algoritmo de busqueda k2 y el estimador simple (Alpha = 0.5)

Combinación N° 18

conversion escala de grises

descriptores: homogeneidad, contraste, correlación, energía, entropía,

clustershade, clustertendency, probabilidad máxima, información de correlación

1, información de correlación 2

porcentaje de acierto de 59.26% usando clasificador knn=1

Combinación N° 19

conversion escala de grises

filtro mediano (3)

descriptores: homogeneidad, contraste, correlación, energía, entropía,

clustershade, clustertendency, probabilidad máxima, información de correlación

1, información de correlación 2



porcentaje de acierto de 69.84% usando clasificador red neuronal

backpropagation 1 capa oculta con 5 neuronas

Combinación N° 20

conversion escala de grises

filtro gaussiano (3)

descriptores: homogeneidad, contraste, correlación, energía, entropía,

clustershade, clustertendency, probabilidad máxima, información de correlación

1, información de correlación 2

porcentaje de acierto de 66.14% usando clasificador red neuronal

backpropagation 1 capa oculta con 5 neuronas

Combinación N° 21

conversion escala de grises

filtro gaussiano (5)

descriptores: homogeneidad, contraste, correlación, energía, entropía,

clustershade, clustertendency, probabilidad máxima, información de correlación

1, información de correlación 2

porcentaje de acierto de 64.55% usando clasificador red neuronal

backpropagation 1 capa oculta con 5 neuronas

Combinación N° 22

conversion escala de grises

ecualizacion de histograma



descriptores: homogeneidad, contraste, correlación, energía, entropía,

clustershade, clustertendency, probabilidad máxima, información de correlación

1, información de correlación 2

porcentaje de acierto de 66.14% usando clasificador red neuronal

backpropagation 1 capa oculta con 5 neuronas



REFERENCIAS

- Almazán, J., Valvery, E., & Fornés, A. (2011). *Deforming the Blurred Shape Model for Shape Description and Recognition*. Barcelona, España.
- Boullosa García, Ó. (2011). *Estudio comparativo de descriptores visuales para la detección de escenas cuasi-duplicadas*. Madrid, España.
- Bradski, G., & Kaebler, A. (2008). *Learning OpenCV Computer vision with the opencv library*. O'Reilly.
- Burger, W., & J. Burge, M. (2009). *Digital Image Processing: An Algorithmic Introduction Using Java*. Springer Science & Business Media.
- Choudhari, S., & Biday, S. (2014). *Artificial neural network for skin cancer detection*.
- Cruz, P. P. (2010). *INTELIGENCIA ARTIFICIAL con aplicaciones a la ingeniería*. Mexico D.F: Alfaomega.
- Davies, E. (2012). *Computer and Machine Vision 4th Edition Theory, Algorithms, Practicalities*.
- Felgaer, P. (2005). *Optimización de redes bayesianas basado en técnicas de aprendizaje por inducción*.
- Foundation, The Skin Cancer. (s.f.). *The Skin Cancer Foundation*. Obtenido de <http://cancerdepiel.org/>
- González Marcos, A., Pisón Ascacibar, F., Pernía Espinoza, A., Alba Elías, F., Castejón Limas, M., Ordieres Meré, J., & Vergara González, E. (2006). *Técnicas y Algoritmos Básicos de Visión Artificial*.
- Haralick, R. (2005). *Statistical and Structural Approaches to Texture, Proceedings of the IEEE. 786-804*. IEEE.
- Hastie, T., Tibshirani, R., & Friedman, J. (s.f.). *The Elements of Statistical Learning - Data mining, Inference, and Prediction*. Springer.
- Kountchev, R., & Nakamatsu, K. (2012). *Advances in Reasoning-Based Image Processing Intelligent Systems: Conventional and Intelligent Paradigms*. Springer Science & Business Media.
- López Bonal, M. (1995). *Notas de visión y apuntes sobre ingeniería del software: notas del III Curso de Verano de Informática*.
- Microsoft. (s.f.). *Microsoft*. Obtenido de <https://msdn.microsoft.com/en-us/library/azure/dn905852.aspx>
- Montabone, S. (2010). *Beginning Digital Image Processing*. Apress.
- Nitin, B., & Vandana. (2010). *Survey of Nearest Neighbor Techniques*.
- O. Kuznetsov, S., P. Mandal, D., K. Kundu, M., & K. Pal, S. (2011). *Pattern Recognition and Machine Intelligence*. Springer.



- Qing Shi, Y., & Jeon, B. (2006). *Digital Watermarking: 5th International Workshop, IWDW 2006, Jeju Island, Korea, November 8-10, 2006, Proceedings*. Springer Science & Business Media.
- Razmjoo, N., Somayeh Mousavi, B., Soleymani, F., & Khotbesara, H. (2012). *A computer aided diagnosis system for malignant melanomas*. London.
- Smaoui, N., & Bessassi, S. (2013). *A developed system for melanoma diagnosis*.
- Szeliski, R. (2010). *Computer Vision: Algorithms and Applications*. Springer.
- Theodoridis, S., & Koutroumbas, K. (2009). *Pattern Recognition*. Academic Press.
- Universidad Nacional de Quilmes. (2005). *Operaciones Morfológicas en imágenes Binarias*.
- WU, J., ZHOU, H., & ZHANG, J. (2010). *Digital Image processing: Part I*.
- Zhou, H., Wu, J., & Zhang, J. (s.f.). *Digital Image Processing: Part II*.