



INSTITUTO POLITÉCNICO NACIONAL
UNIDAD PROFESIONAL INTERDISCIPLINARIA EN
INGENIERÍA Y TECNOLOGÍAS AVANZADAS

UPIITA

Trabajo Terminal II

PARTE 2

**“PROTOTIPO PARA LA DETECCIÓN DE
CÁNCER DE MAMA BASADO EN
CLASIFICACIÓN DE TERMOGRAMAS Y
DATOS CLÍNICOS”**

Autores:

Jan Andrei Merino González

Ricardo Niño de Rivera Barrón

Que para obtener el título de

“Ingeniero en Biónica”

Presenta el alumno:

Ricardo Niño de Rivera Barrón

Asesor:

Dr. Álvaro Anzueto Ríos



Ciudad de México a 31 de diciembre de 2020



INSTITUTO POLITÉCNICO NACIONAL
UNIDAD PROFESIONAL INTERDISCIPLINARIA EN
INGENIERÍA Y TECNOLOGÍAS AVANZADAS
UPIITA

Trabajo Terminal II

PARTE 2
“PROTOTIPO PARA LA DETECCIÓN DE CÁNCER
DE MAMA BASADO EN CLASIFICACIÓN DE
TERMOGRAMAS Y DATOS CLÍNICOS”

Autores:

Jan Andrei Merino González

Ricardo Niño de Rivera Barrón

Que para obtener el título de

“Ingeniero en Biónica”

Presenta el alumno:

Ricardo Niño de Rivera Barrón

Dra. Blanca Tovar Corona
Presidente

Dr. Álvaro Anzueto Ríos
Asesor

M. en E. Cuauhtémoc Valaguez Velázquez
Secretario



M. en C. Rafael Santiago Godoy
Suplente

Ciudad de México a 31 de diciembre de 2020

Agradecimiento

A MIS PADRES

AL INSTITUTO POLITÉCNICO NACIONAL

A MIS PROFESORES

Índice

Índice de Figuras	7
Índice de Tablas.....	10
Resumen	11
Palabras clave:.....	11
Abstract.....	12
Key words:.....	12
Capítulo 1 Introducción.....	13
1.1 Antecedentes	14
1.2 Últimos avances	15
Capítulo 2 Objetivos	17
2.1 General.....	17
2.2 Específicos	17
Capítulo 3 Planteamiento del problema	18
Capítulo 4 Justificación de la investigación	19
Capítulo 5 Marco Teórico	20
5.1 Anatomía y Fisiología de la Glándula Mamaria.....	20
5.1.1 Mama	20
5.1.2 Localización	20
5.1.3 Forma	20
5.1.4 Estructura	20
5.1.5 Vascularización.....	21
5.1.6 Inervación.....	22
5.2 Cáncer de mama.....	22
5.2.1 Factores de riesgo del cáncer de mama	23
5.3 Procesamiento digital de imágenes.....	23
5.3.1 Visión artificial	24
5.4 Inteligencia Artificial.....	32
5.4.1 Aprendizaje Máquina o Automático.....	32
5.4.2 Aprendizaje profundo (Redes Neuronales Convolucionales).....	52
5.5 Correlación lineal.....	56

5.5.1 Coeficiente de correlación de Pearson	56
5.5.2 Coeficiente de Correlación de Spearman.	57
5.6 Matriz de confusión.....	58
5.6.1 Derivaciones de la matriz de confusión (índices de validación).....	58
5.6.2 Curva ROC.....	60
5.7 Termografía infrarroja	61
5.7.1 Aplicaciones del infrarrojo en la medicina.....	62
5.7.2 Ventajas de la termografía	64
Capítulo 6 Desarrollo.....	65
6.1 Construcción de Conjuntos de Entrenamiento y validación	66
6.2 Transformación lineal.....	67
6.3 Modelo por caracterización bilateral	68
6.3.1 Segmentación automática.....	68
6.3.2 Caracterización.....	73
6.3.3 PCA.....	73
6.3.4 Reconocedores.....	74
6.4 Modelo a partir de datos clínicos	75
6.5 Modelo con base en Red Neuronal Convolucional	75
6.5.1 Red Neuronal Artificial Convolucional EfficientNet	75
6.5.2 Red Neuronal Artificial Convolucional ResNet50	76
6.5.3 Red Neuronal Artificial Convolucional GoogleNet	76
Capítulo 7 Resultados.....	78
7.1 Resultados de Segmentación.	78
7.1.1 Segmentación con base en parábolas	78
7.1.2 Segmentación con base en plano sagital	79
7.2 Resultado de reconocedores con base en caracterización bilateral	79
7.3 Análisis, minería de datos y reconocedor a partir de datos clínicos	81
7.3.1 Resultados de reconocedores para datos clínicos	82
7.4 Redes Neuronales Convolucionales.....	84
7.5 Implementación del modelo en IBM Watson	88
7.5.1 Sustitución de la función de activación “swish” por “ReLU”	89
7.5.2 Resultados EfficientNet con modificación de función <i>swish</i>	90

7.5.3 Reentrenando EfficienNet y Resultados obtenidos (EfficientNet IBM).....	91
7.6 Comparativo de Parte 1 y Parte 2 de este trabajo.....	95
Capítulo 8 Prototipo final	97
8.1 Interfaz Gráfica de Usuario (GUI).....	98
8.2 Diagrama de la aplicación,.....	101
Capítulo 9 Conclusiones	103
Capítulo 10 Sugerencias para investigaciones futuras	104
Bibliografía	106
ANEXOS	114
ANEXO A.....	115
ANEXO B.....	132
ANEXO C.....	141
ANEXO D.....	176
ANEXO E	180
ANEXO F	199
ANEXO G.....	206
ANEXO H.....	224
ANEXO I.....	231
ANEXO J.....	237
ANEXO K.....	240
ANEXO L	313
ANEXO M.....	406
ANEXO N	476
ANEXO O	517
ANEXO P.....	556
ANEXO Q	605
ANEXO R.....	617
ANEXO S	623
ANEXO T	655
ANEXO U	681
ANEXO V.....	689
ANEXO W	726
ANEXO X.....	781

ANEXO Y	824
ANEXO Z	870

Índice de Figuras

- Figura 1. Estructura de la glándula mamaria.
- Figura 2. Cuadrantes de la glándula mamaria.
- Figura 3. Etapas de un sistema de visión artificial.
- Figura 4. Proceso de adquisición de una señal física y su transformación a una señal digital.
- Figura 5. Etapas de las propuestas de segmentación
- Figura 6. Función sigmoidea.
- Figura 7. Comportamiento de la función de hipótesis con respecto al valor de 'y'.
- Figura 8. Función Hinge Loss para el primer término de la función de costos.
- Figura 9. Función Hinge Loss para el segundo término de la función de costos.
- Figura 10. Estructura de una neurona biológica.
- Figura 11. Estructura de una neurona perceptrón.
- Figura 12. Estructura de una red neuronal de segunda generación.
- Figura 13. Descripción de un volumen de entrada y/o resultante de una convolución.
- Figura 14. Volumen de salida después de aplicar un filtro en una arquitectura convolucional.
- Figura 15. Arquitectura LeNet5.
- Figura 16. Salida de un filtro Max pooling 2X2.
- Figura 17. Comparativa entre una función rectificadora y una Softplus.
- Figura 18. Ejemplo de un espacio ROC, la línea roja representa los clasificadores con rendimiento similar al azar (caso B).
- Figura 19. Termograma en pseudocolor.
- Figura 20. Termograma en escala de grises.
- Figura 21. Espectro de la luz.
- Figura 22. Cámara infrarroja T650sc.
- Figura 23. Línea recta aproximada mediante regresión lineal.
- Figura 24. Modelo de caracterización bilateral.
- Figura 25. Segmentación por parábolas.
- Figura 26. Histogramas de diferentes termogramas aleatorios.
- Figura 27. Histograma de un termograma.
- Figura 28. Histograma umbralizado por método de Otsu.
- Figura 29. Proceso de binarización para separación de objeto con el fondo.

Figura 30. Reducción de perfil.

Figura 31. Eliminación de bordes superiores.

Figura 32. Resultado de operación SOBEL.

Figura 33. Bordes de segunda binarización por Otsu

Figura 34. Delimitación de regiones de interés.

Figura 35. Segmentación con plano sagital.

Figura 36. Diagrama de dispersión de imágenes de entrenamiento.

Figura 37. Diagrama de dispersión de imágenes de validación.

Figura 38. Resultados de segmentación de pares de mama (individuos).

Figura 39. Resultados de segmentación por mama (individuos).

Figura 40. Curvas ROC Regresión Logística.

Figura 41. Curvas ROC Red Neuronal.

Figura 42. Curvas de entrenamiento de la Red Neuronal.

Figura 43. Descripción de datos clínicos.

Figura 44. Árbol de decisión.

Figura 45. ROC Regresión Logística.

Figura 46. ROC Red Neuronal.

Figura 47. ROC EfficientNet.

Figura 48.-Curvas de entrenamiento EfficientNet.

Figura 49. ROC ResNet50.

Figura 50. Curvas de entrenamiento ResNet50.

Figura 51. ROC GoogleNet.

Figura 52. Curvas de entrenamiento GoogleNet.

Figura 53. ROC EfficientNet Modificada con pesos de EfficientNet "Original".

Figura 54. ROC EfficientNet modificada con 150 iteraciones.

Figura 55. Curvas de entrenamiento EfficientNet modificada 150 iteraciones.

Figura 56. ROC EfficientNet modificada IBM final.

Figura 57. Curvas de entrenamiento EfficientNet modificada IBM final.

Figura 58. Diagrama GUI.

Figura 59. GUI Usuario sube archivo.

Figura 60. GUI Avance de procesamiento de termograma.

Figura 61. GUI 2 Avance de procesamiento de termograma.

Figura 62. Resultado.

Figura 63. Resultado cuando la aplicación se encuentra en mantenimiento.

Figura 64. Diagrama de la Aplicación WEB.

Índice de Tablas

Tabla 1. Descriptores estadísticos de primer orden.

Tabla 2. Descriptores de segundo orden obtenidos a partir de la matriz de co-ocurrencia.

Tabla 3. Ejemplo de Matriz de Confusión.

Tabla 4. Derivaciones de la matriz de confusión.

Tabla 5. Especificaciones de las cámaras T659sc y T630sc del fabricante FLIR.

Tabla 6. Conjuntos de entrenamiento y validación.

Tabla 7. Resultados de reconocedores por asimetría estadística Entrenamiento.

Tabla 8. Resultados de reconocedores por asimetría estadística Validación.

Tabla 9. Reconocedores para datos clínicos Entrenamiento.

Tabla 10. Reconocedores para datos clínicos Validación.

Tabla 11. Resultados Convolucionales Entrenamiento.

Tabla 12. Resultados Convolucionales Validación.

Tabla 13. Resultados de EfficientNet modificada con pesos entrenados en EfficientNet original Conjunto de Entrenamiento.

Tabla 14. Resultados de EfficientNet modificada con pesos entrenados en EfficientNet original Conjunto de Validación.

Tabla 15. Resultados de EfficientNet modificada 150 iteraciones Conjunto de Entrenamiento.

Tabla 16. Resultados de EfficientNet modificada 150 iteraciones Conjunto de Validación.

Tabla 17. Resultados de EfficientNet modificada IBM final Conjunto de Entrenamiento.

Tabla 18. Resultados de EfficientNet modificada IBM final Conjunto de Validación.

Tabla 19. Comparativo modelos Conjunto de Entrenamiento.

Tabla 20. Comparativo modelos Conjunto de Validación.

Resumen

El presente trabajo describe el diseño y construcción de un prototipo de software inteligente capaz de discriminar termogramas de la región mamaria con el objetivo de encontrar los que presentan cáncer de mama.

Para este prototipo se diseñaron y construyeron dos sistemas de segmentación automática de región mamaria. Además, se evaluaron diferentes técnicas de aprendizaje automático. La técnica que demostró el mejor rendimiento se implementó en el prototipo final.

A diferencia del primer trabajo, esta segunda parte recopila una mayor cantidad de pacientes con mayor número de termogramas por paciente. Así mismo se proponen nuevos modelos matemáticos los cuales abordan un enfoque distinto a la de la primera propuesta sin modificar los objetivos del protocolo original.

Palabras clave:

Termografía; Visión Artificial; Inteligencia Artificial; Aprendizaje Automático.

Abstract

The present work describes the design and construction of an intelligent software prototype capable of discriminating thermograms of the mammary region in order to find those with breast cancer.

For this prototype, two automatic mammary region segmentation systems were designed and built. In addition, different machine learning techniques were evaluated. The technique that demonstrated the best performance was implemented in the final prototype.

Unlike the first work, this second part collects a greater number of patients with a greater number of thermograms per patient. Likewise, new mathematical models are proposed which address a different approach to that of the first proposal without modifying the objectives of the original protocol.

Key words:

Thermography; Artificial Vision; Artificial Intelligence; Machine Learning.

Capítulo 1 Introducción

Los sistemas inteligentes con base en la emulación de estructuras biológicas no han sido un tema nuevo en el área médica. [1] Sin embargo, en los últimos años ha existido un interés renovado por el desarrollo de nuevos sistemas inteligentes para el diagnóstico, pronóstico y tratamiento de enfermedades debido al constante avance en la capacidad de procesamiento computacional. Lo que ha permitido desarrollar aplicaciones de software cada vez más robustas y eficientes ante las diferentes problemáticas que enfrenta la medicina. [2]

En este trabajo se formula una propuesta de prototipo de software con la finalidad de encontrar mujeres con indicios de lesiones cancerígenas en la región mamaria a partir de la interpretación automática de termogramas, mediante el uso de inteligencia artificial. Particularmente en este trabajo se comparan diferentes técnicas de aprendizaje automático, algunas de ellas basadas en la teoría de funcionamiento de neuronas biológicas.

Datos de la Organización Mundial de la Salud indican que esta enfermedad es la principal causa de muerte en mujeres mayores de 20 años a nivel mundial. [3] Por lo que es un reto para la comunidad científica desarrollar nuevas técnicas de detección temprana.

Actualmente, las técnicas utilizadas para la detección temprana de esta enfermedad no satisfacen la detección oportuna de nuevos casos con la eficacia y eficiencia necesarias para la disminución de la mortalidad. Esto es debido a distintos factores como lo son: cuestiones económicas, aspectos culturales, ausencia de recursos tecnológicos y humanos, entre otros. Por lo anterior, presentar propuestas que se adecuen a la realidad socioeconómica y cultural de regiones en vías de desarrollo como América Latina y, en específico, México, es un área de oportunidad en la cual este trabajo aporta una alternativa de solución.

Con base en distintos trabajos presentados en este texto, la tendencia mundial para el diagnóstico de distintas enfermedades apunta al desarrollo de soluciones escalables tecnológica y socialmente, adecuadas a la coyuntura de las regiones en vías de desarrollo. Además, existe una tendencia que marca el camino hacia el desarrollo de sistemas inteligentes que complementen (e inclusive sustituyan gradualmente) la intervención del ser humano en alguna parte del proceso de diagnóstico (o en todo). Esto con el objetivo de minimizar lo mejor posible el error humano, así como eficientizar el proceso de prevención y detección de enfermedades que potencialmente causarían estragos al individuo y al sistema de salud.

Por lo anteriormente expuesto, la propuesta de este trabajo se centra en el diseño y construcción de un prototipo de software inteligente capaz de detectar lesiones cancerígenas en termogramas de forma no invasiva. Esto es posible debido a que la termografía, al ser una técnica de medición de temperatura corporal mediante la radiación emitida por el cuerpo, permite obtener termogramas sin emisión radiológica o sometimiento a estrés mecánico, lo que representa un riesgo nulo para el paciente y el operador. Esto último, aunado al sistema inteligente de interpretación, constituyen una solución alternativa a la problemática de detección temprana.

1.1 Antecedentes

En los años setenta, Mozkowitz realizó un estudio donde mostraba la inviabilidad de la termografía como herramienta diagnóstica al presentar resultados que apuntaban a su deficiente comportamiento en comparación con las técnicas de análisis por imagen de aquel entonces. [4] Sin embargo, otros investigadores demostraron que la metodología de Mozkowitz presentaba deficiencias. [5] A principios del siglo XXI, el avance en la tecnología termográfica incentivó a investigadores en el área a realizar estudios combinando esta tecnología con el análisis por computadora, dando como resultado prototipos con respuestas estadísticamente comparables a otras técnicas de imagenología diagnóstica.

El avance tecnológico ha dado como resultado sensores infrarrojos cada vez más sensibles que permiten realizar mediciones precisas y con mayor resolución, lo cual ha resultado en diversos estudios que plantean la utilidad de la termografía como herramienta en la detección de cáncer mamario, muchos de estos estudios demostrando su utilidad para entregar imágenes con suficiente información para realizar métodos diagnósticos efectivos. [6][7][8]

La detección de termogramas a partir de asimetría parte del principio de la existencia de mayor actividad fisiológica en el sector del organismo con lesiones cancerígenas (para motivos de este trabajo, en alguno de los senos) debido a la actividad fisiológica anormal. Esta asimetría es posible medirla mediante la diferencia aritmética de los descriptores estadísticos o temperaturas de regiones puntuales en ambos lados. Sobre este paradigma, existen diversos estudios que han evaluado la viabilidad de esta forma de discriminación de termogramas.

En 2009, Schaefer et al publicaron un algoritmo capaz de discriminar termogramas anómalos a partir del análisis de la asimetría térmica utilizando descriptores estadísticos. [9] Qi et al propusieron el análisis de asimetría mediante la obtención de la media de los valores de los píxeles. [10] Kuruganti et al, en 2002, propusieron un trabajo mejorado del anterior incluyendo entre las características a la media, varianza, asimetría estadística y curtosis de la distribución de píxeles. [11]

Cruz-Ramírez et al, presentaron en 2012 una propuesta de obtención de asimetría con base en mediciones puntuales de regiones del paciente (axilas, bordes inferiores del seno, cuello, entre otros), sin embargo, sus resultados fueron menos eficientes que los trabajos anteriores, aunque el reconocedor propuesto, una red neuronal bayessiana, presentaba mayor complejidad que los anteriores. [12]

Existen también diversos trabajos que proponen técnicas sin necesariamente utilizar un análisis de asimetría para discriminar termogramas. A continuación, se mencionan algunos de ellos.

Acharya et al publicaron en 2010 un reconocedor inteligente basado en una Máquina de Soporte Vectorial, el cual recibía como entradas las características de textura de la imagen. [13] Trabajo similar al propuesto por Milosevic et al en 2014. [14]

Uno de los aspectos en los cuales diversos investigadores han trabajado ha sido en la segmentación automática de las glándulas mamarias a partir de termogramas. Existen diferentes propuestas para realizar esta tarea. En 2001, NG et al propusieron la segmentación automática de las glándulas mamarias utilizando un algoritmo que consistía en la modificación de técnicas comunes de

umbralización y detección de bordes. [15] Estos mismos investigadores propusieron una mejora de su algoritmo en 2006. [16]

En 2004, Scales et al [17] propusieron la siguiente metodología:

1. Remoción manual de los bordes superiores del cuerpo y la cintura.
2. Detección automática de bordes utilizando Canny.
3. Separación de los bordes restantes del cuerpo, izquierdo, derecho y superiores.
4. Detección de las líneas más bajas que definen a los senos.
5. Interpolación de segundo orden de las curvas encontradas en el paso 4.
6. Detección de la región con la mayor curvatura en los bordes izquierdo y derecho del cuerpo.
7. Estimación de las líneas superiores de los senos usando un número de reglas y regiones empíricas del cuerpo.
8. Separación de la región de interés utilizando los límites encontrados en los pasos 3, 6 y 7.

Golestani et al propusieron una segmentación basada en ajustes de nivel analizando numéricamente las superficies y formas del termograma, técnica conocida como *Level Set*. [18]

1.2 Últimos avances

Recientemente se han publicado diversos estudios que combinan sistemas inteligentes con termografía, obteniendo resultados que ponen en entre dicho lo establecido por Mozkowitz.

Pérez et al propuso un proyecto de investigación en Ecuador en el año 2014 para la detección de carcinomas en termogramas utilizando técnicas diversas de clasificación y extracción de características. [19]

Guamán, también en Ecuador, propuso diversos reconocedores a partir de características de textura de la imagen, en específico características propuestas por Haralick en los años setenta. [20]

Uno de los trabajos utilizando Maquinas de Soporte Vectorial fue el propuesto por Wakankar et al. En este trabajo se proponía la extracción de características de color de la imagen. [21]

El uso de redes neuronales artificiales en el reconocimiento de termogramas también ha sido materia de estudio, como ejemplo el estudio de Koay et al, quienes publicaron un trabajo en 2004 sobre el uso de una red neuronal de segunda generación para la resolución de esta tarea. [22] Al igual que el trabajo de Haddadnia et al, quienes compararon el funcionamiento de una red neuronal de segunda generación (con 6 neuronas en la capa oculta) y un algoritmo genético, utilizando como características de entrada para los reconocedores inteligentes: edad del individuo, diferencias de temperatura entre los senos, “patrón térmico”, promedio, varianza, asimetría estadística, curtosis y entropía. [23]

En 2017, Wahab et al, presentaron un trabajo utilizando dos técnicas de segmentación automática en conjunto con una red neuronal de segunda generación para la detección temprana de cáncer de mama utilizando termogramas adquiridos mediante un protocolo de estrés térmico. [24] Sin

embargo, este estudio fue realizado en ratones de laboratorio con intención futura de ser aplicado en humanos.

Araújo propuso en 2014 una técnica de extracción de características a partir de intervalos simbólicos, donde estableció tres clases de salida: enferma, con quistes y sanas. [25] Un estudio similar al anterior fue el propuesto por Okuniewski et al en 2016, sin embargo, difería del anterior al utilizar imágenes separadas obtenidas de dos sensores independientes. [26]

En la India (2016) se presentó un estudio que propuso el uso de un Histograma de Gradientes Orientados (HOG), después la aplicación de una reducción dimensional de las características obtenidas a partir del HOG utilizando un Análisis de Componentes Principales (PCA). En este estudio se compararon los resultados de nueve clasificadores diferentes: árbol de decisión, clasificador discriminante (lineal y cuadrático), clasificador de k-vecinos más cercanos (KNN), clasificador difuso Sugeno, clasificador bayesiano ingenuo, máquina de soporte vectorial, clasificador AdaBoost, red neuronal probabilística y, por último, índice de riesgo de cáncer de mama. [27]

En 2015, Gogoi et al presentaron una propuesta que combinaba diferentes algoritmos inteligentes para el análisis de termogramas con la finalidad de hallar los que presentaban indicios de lesiones cancerígenas. [28]

Oleszkiewicz et al propusieron en 2016 un sistema basado en la clasificación de termogramas utilizando una Máquina de Soporte Vectorial un Optimizador Secuencial de Mínimos. [29]

Neumann et al, de la Universidad de tecnología de Warsaw, propusieron en 2016 un algoritmo SMOTE para la clasificación de termogramas. [30]

Por último, en 2014, investigadores de la India propusieron la utilización de clasificadores neurodifusos para detección de lesiones cancerígenas a partir de termografías mamarias. [31]

Capítulo 2 Objetivos

2.1 General

Detección de cáncer de mama en las etapas I y II, basado en termogramas.

2.2 Específicos

- Diseño y construcción del prototipo de software de adquisición de imágenes termográficas.
- Elaboración del software para la segmentación automática de la mama derecha e izquierda.
- Diseño del software encargado de la clasificación de termogramas.
- Validación del sistema.

Capítulo 3 Planteamiento del problema

Los sistemas de interpretación artificial de imágenes médicas se enfrentan a diferentes problemáticas. La primera, es la adquisición de la imagen de forma adecuada, es decir, que la imagen cuente con las características suficientes para transmitir información acerca del fenómeno capturado en ella.

El segundo problema al que se enfrenta un sistema artificial de interpretación de imágenes es la segmentación y caracterización de las regiones de interés. En otras palabras, definir y separar objetos en una imagen con el fin de obtener las cualidades que lo definen.

El tercer problema es obtener un clasificador que sea capaz de discriminar los diferentes objetos en la imagen. En este trabajo en particular, el problema a resolver es crear un clasificador artificial robusto que sea capaz de alcanzar una tasa de aciertos (exactitud) superior al 80%.

Englobando lo anterior, el problema a resolver por este trabajo es la creación de un sistema clasificador inteligente capaz de recibir un termograma, segmentar y describir los objetos de interés (glándulas mamarias) en dicho termograma y clasificarlo correctamente, en función de su clase de pertenencia.

Capítulo 4 Justificación de la investigación

El desarrollo de sistemas inteligentes, que emulen la capacidad cognitiva del ser humano, es una de las áreas de desarrollo en las cuales el ingeniero en biónica cuenta con el conocimiento técnico y científico para diseñar y desarrollar soluciones.

La inteligencia artificial, el procesamiento de imágenes digitales, el reconocimiento de patrones y, las herramientas estadísticas para el análisis de fenómenos, son algunas de las áreas que el ingeniero en biónica es capaz de implementar para emular la capacidad cognoscitiva de un ser humano en el proceso de toma decisiones.

En medicina, esto no es la excepción, la capacidad de un médico para establecer un diagnóstico es susceptible a ser imitado por un sistema artificial. El que propone el siguiente trabajo, no escapa a la tendencia que existe en diferentes esferas de la investigación actual para la creación de desarrollos científicos y tecnológicos que tengan como objetivo emular, e incluso superar, la capacidad de los médicos para tomar decisiones rutinarias. Esto con el fin de acelerar procesos de interpretación y disminuir el error humano; para el caso particular de este trabajo: interpretar termogramas.

Con base en lo anterior, el desarrollo de un sistema inteligente para la solución de una problemática de salud pública como lo es la detección temprana de cáncer de mama, implica que la capacidad técnica y científica de un ingeniero es aplicada en la solución de una problemática concreta. Solución con potencial de ser implementada en la realidad a corto o mediano plazo.

Capítulo 5 Marco Teórico

5.1 Anatomía y Fisiología de la Glándula Mamaria

5.1.1 Mama

Las mamas son dos relieves ovoides situados en la cara anterior del tórax, una a cada lado de la línea media del cuerpo, con forma y tamaño variables. En el varón se diferencia de la mujer al no desarrollar glándulas mamarias. [32]

En la mujer, las glándulas mamarias comienzan a desarrollarse a partir de la pubertad debido a la influencia de las hormonas ováricas (estrógenos y progesterona). [32]

5.1.2 Localización

Se encuentran localizadas en la parte anterior de los músculos pectoral mayor, región anterior del serrato anterior y región craneal del músculo oblicuo externo del abdomen, proyectándose entre la tercera y séptima costillas, desde el borde lateral del hueso esternón hasta la línea media axilar. [32]

5.1.3 Forma

Presentan una forma cónica, en su vértice se diferencian la aréola mamaria y el pezón. [32]

Existen unas 15 a 20 elevaciones de tamaño reducido denominadas *tubérculos de Morgagni*, las cuales, durante el embarazo y la lactancia aumentan su tamaño y pasan a denominarse *tubérculos de Montgomery*. Estos tubérculos son glándulas sebáceas modificadas de la aréola mamaria. [32]

El pezón morfológicamente se asemeja a un cono truncado y presenta entre 15 a 20 orificios en los cuales desembocan los conductos galactóforos. [32]

5.1.4 Estructura

La glándula mamaria es una estructura semiesférica ubicada por debajo de las estructuras cutáneas y subcutáneas de la mama. Dicha estructura puede presentar dos prolongaciones: una hacia la vaina de los músculos recto abdominales, prolongación llamada inferior, la segunda es superoexterna, hacia la axila, prolongación axilar. [32]

La superficie anterior de la glándula mamaria muestra una serie de crestas, de estructura fibroglandular, que se prolongan hasta la piel constituyendo tabiques, que configuran el ligamento suspensorio de la mama (ligamentos de Cooper), estableciendo compartimentos del tejido adiposo

con una cantidad variable de grasa (importantes como determinantes de signos clínicos y radiológicos en la patología mamaria). El vértice de la mama, constituido por la aréola y el pezón, está exento de estas formaciones adiposas y presenta pequeños haces de fibras musculares lisas. Tras la menopausia, el tejido adiposo de la mama puede reducirse, disminuyendo de volumen y perdiendo la forma semiesférica. [32]

La superficie posterior de la glándula mamaria, ligeramente cóncava, está separada del músculo pectoral mayor por el espacio retromamario, un espacio de deslizamiento que permite el desplazamiento de la glándula sobre la pared torácica. El espacio retromamario está recorrido en sentido cráneo-caudal por la fascia retromamaria, que se extiende desde la clavícula en sentido caudal. La presencia de esta fascia divide el espacio retromamario en un compartimento anterior, que contiene el tejido adiposo retromamario, y un compartimento posterior, que no posee grasa y constituye el verdadero espacio de deslizamiento. [32]

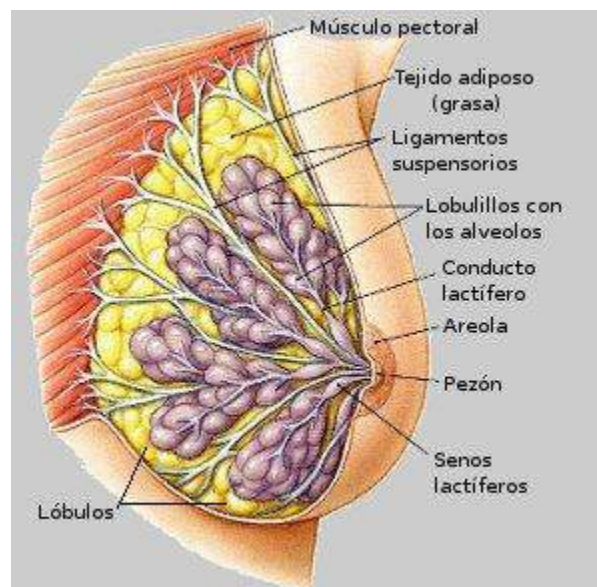


Figura 1. Estructura de la glándula mamaria.

5.1.5 Vascularización.

Arterias. La vascularización arterial procede de las arterias intercostales, de la torácica interna, rama de la subclavia, y de las torácica lateral y acromio-torácica, ramas de la arteria axilar. [32]

Venas. Las venas siguen un trayecto similar e inverso al arterial. [32]

Linfáticos. Los linfáticos de la mama se reúnen en cuadrantes. Los vasos linfáticos de los cuadrantes mediales perforan los primeros espacios intercostales y penetran en el tórax, drenando en los ganglios paraesternales. Los vasos linfáticos de los cuadrantes laterales de la mama desembocan en el grupo anterior de los ganglios axilares (localizados junto al borde inferior del músculo pectoral mayor). Algunos vasos linfáticos comunican con los de la mama contralateral y con los de la pared anterior del abdomen. [32]

5.1.6 Inervación

Desde los nervios intercostales o torácicos, del cuarto al sexto, llegan fibras vegetativas a la mama, sin embargo, su actividad secretora es principalmente hormonodependiente. Además, el pezón está finamente inervado, enviando la información aferente sensitiva a través de los nervios intercostales.

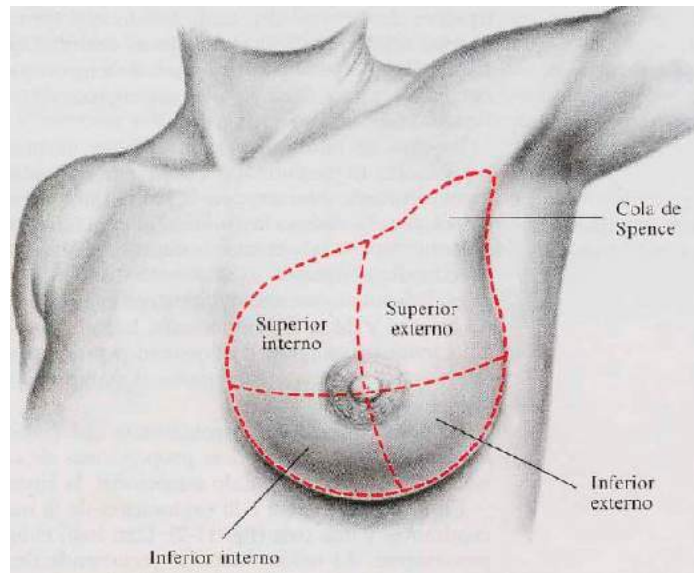


Figura 2. Cuadrantes de la glándula mamaria.

5.2 Cáncer de mama

Cuando en un conjunto de células se presenta el fenómeno de retraso en la apoptosis celular se produce el crecimiento anormal del tejido que éstas conforman. Este crecimiento anormal de tejido es conocido como tumor. El tumor puede ser maligno o benigno, siendo esto en función del comportamiento histopatológico.

El concepto *cáncer de mama* se refiere a una tumoración de carácter maligno con presencia en alguno de los tejidos que constituyen las diferentes estructuras de la glándula mamaria. En términos generales es posible clasificar estos carcinomas en dos grandes tipos: carcinoma ductal infiltrante y carcinoma lobulillar infiltrante. Estos nombres hacen referencia a la estructura anatómica. Para el caso del carcinoma ductal infiltrante, el tumor se localiza en alguno de los diversos conductos que tienen como función transportar leche materna, como los conductos galactóforos. El segundo tipo de carcinoma (lobulillar infiltrante) se ubica en las glándulas especializadas en producción de leche materna (lobulillos o acinos). Existen también otros tipos de cáncer de mama, pero no superan en su conjunto el 10% de los casos registrados. [33]

5.2.1 Factores de riesgo del cáncer de mama

Los factores de riesgo son cualquier rasgo, característica o exposición de un individuo que aumente su probabilidad de sufrir una enfermedad o lesión. [34] Por tanto, el cáncer de mama también presenta diversos factores que aumentan la probabilidad de aparecer en un individuo, los cuales son [33]:

- Tener antecedentes personales de cáncer de mama invasivo, carcinoma ductal in situ (CDIS) o carcinoma lobulillar in situ (CLIS).
- Tener antecedentes personales de enfermedad benigna (no cancerosa) de la mama.
- Tener antecedentes familiares de cáncer de mama en un pariente de primer grado (madre, hija o hermana).
- Presentar cambios heredados en los genes BRCA1 o BRCA2 o en otros genes que aumentan el riesgo de cáncer de mama.
- Presentar tejido de la mama que se observe denso en un mamograma.
- La exposición del tejido mamario al estrógeno elaborado por el cuerpo. Esto puede ser a causa de:
 - Presentar la menstruación en una edad temprana.
 - Dar a luz por primera vez a una edad más avanzada, o nunca haber tenido hijos.
 - Presentar el inicio de la menopausia a una edad tardía.
- Tomar hormonas como estrógenos combinados con progestinas para los síntomas de la menopausia.
- Haber recibido tratamiento con radioterapia dirigida a la mama o el tórax.
- Consumir alcohol.
- Ser obeso.
- Ser de piel blanca.

La probabilidad de presentar cáncer aumenta a medida que el individuo envejece. [2]

En 2013, Gaudet et al publicaron un estudio donde se descubrió que el consumo de tabaco aumenta la probabilidad de padecer cáncer de mama. [35]

5.3 Procesamiento digital de imágenes

El procesamiento digital de imágenes es el conjunto de técnicas destinadas a manipular matrices digitales las cuales están vinculadas a una imagen que puede ser proyectada en la pantalla de un sistema informático. Esta disciplina de la informática requiere el conocimiento de los siguientes conceptos [36]:

Visión por computadora: consiste en la adquisición, procesamiento, clasificación y reconocimiento de imágenes digitales.

Píxel: elemento básico de una imagen.

Imagen: arreglo bidimensional de píxeles con diferente intensidad lumínica (escala de gris). Si la intensidad lumínica de un píxel se representa por n bits, entonces existirán 2^n escalas de gris diferentes.

Color: el color se forma mediante la combinación de los tres colores básicos: rojo, verde y azul (en inglés corresponde a las siglas RGB), y puede expresarse mediante una tripleta de valores de 0 a 1 (R, G, B), donde R, G y B representan las intensidades de cada uno de los tres colores básicos.

Brillo: indica si un área está más o menos iluminada.

Tono: indica si un área parece similar al rojo, amarillo, verde o azul, o a una proporción de ellos.

Luminosidad: es el brillo de una zona respecto a otra.

Croma: indica la coloración de un área respecto al brillo de un blanco de referencia.

Espacio RGB: se basa en la combinación de tres señales de luminancia cromática distinta: rojo, verde y azul. La forma más sencilla de obtener un color específico es determinando la cantidad de color rojo, verde y azul que se requiere combinar para obtener el color deseado, para lo cual se realiza la suma aritmética de las componentes $X = R + G + B$, gráficamente representada por un cubo. En la recta que une el origen con el valor máximo se encuentran ubicados los grises (escala de gris), debido a que sus tres componentes son iguales. Cuando una cámara adquiere una imagen a color, para cada píxel en color se tienen en realidad tres componentes, una para cada uno de los colores básicos (rojo, verde y azul); la ganancia máxima para cada componente corresponde a la longitud de onda de los tres colores básicos.

Mapa de color: el mapa de color es una matriz de $n \times 3$, donde cada región es una tripleta de colores. El primer renglón corresponde al valor mínimo del eje de color y el último renglón al máximo. Al definir diferentes distribuciones de intensidad de los tres colores básicos, se crean diferentes mapas de color. Algunos de los mapas de color predeterminados en Matlab y bibliotecas de lenguajes de programación más comunes son: *hsv*, *cool*, *hot*, *jet*, *gray*, *flag*.

Histograma de una imagen: el histograma de una imagen es una representación del número de píxeles de cierto nivel de gris en función de los niveles de gris.

5.3.1 Visión artificial

V.S. Nalwa ofrece la siguiente definición sobre visión artificial: *“La visión artificial o comprensión de imágenes describe la deducción automática de la estructura y propiedades de un mundo tridimensional, posiblemente dinámico, bien a partir de una o varias imágenes bidimensionales de ese mundo”*. [37] En términos prácticos es posible definir a la visión artificial como una rama de la inteligencia artificial la cual se encarga del desarrollo de sistemas informáticos capaces de interpretar, de manera autónoma y eficaz, una imagen o un conjunto de imágenes con la finalidad de obtener información.

Una representación del proceso general de visión artificial es mostrada en la figura 3. [38]

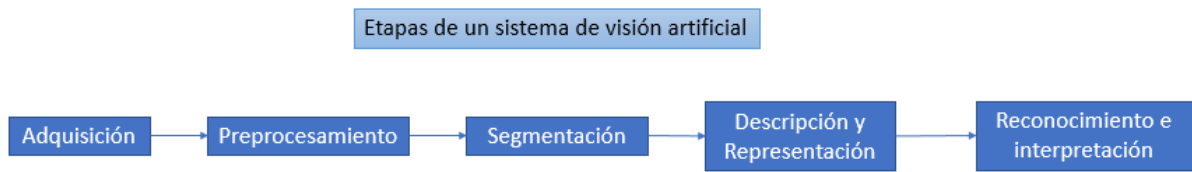


Figura 3. Etapas de un sistema de visión artificial.

5.3.1.1 Adquisición

En esta etapa del proceso el operador del equipo fotográfico capta la imagen por medio de un sensor especializado en el espectro de luz de interés (infrarrojo lejano para este trabajo). Durante este proceso el transductor del equipo fotográfico obtiene la información lumínica del ambiente y la transforma (o transduce) en una señal eléctrica. Al ser el transductor en sí un arreglo matricial de pequeños transductores, éste envía señales eléctricas de los diferentes puntos a la unidad de conversión analógica digital (ADC), la cual se encarga de digitalizar la señal obtenida para su posterior envío al computador. Este proceso también es conocido como digitalización de la imagen (figura 4).

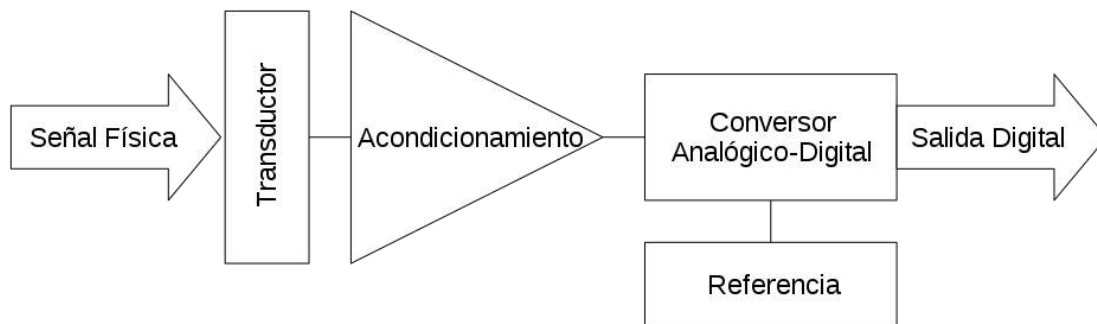


Figura 4. Proceso de adquisición de una señal física y su transformación a una señal digital.

La importancia de esta etapa reside en obtener la mejor imagen posible, es decir, adquirir la mayor cantidad de información sobre el fenómeno a estudiar sin interferencia de ruido externo o minimizando este último. Un ejemplo ilustrador proviene de la imagenología médica, donde existe un sistema de evaluación que establece la probabilidad de presentar lesiones relacionadas con cáncer de mama a partir de la interpretación de las características radiológicas de la imagen por un especialista; este sistema recibe por nombre BI-RADS. Una mala o deficiente adquisición es calificada como BI-RADS 0, lo que implica que la imagen no pueda ser interpretada por lo que es preciso realizar otra adquisición. [39]

5.3.1.2 Preprocesamiento

Posterior a la adquisición y digitalización de la imagen, la mayoría de las veces (esto en función de la aplicación) es necesario realzar, eliminar y/o filtrar regiones y/o características de la imagen con la finalidad de poder eliminar ruido y/o resaltar elementos de interés, de tal manera que se efficienticen los pasos ulteriores de procesamiento.

Existen diversos métodos utilizados en el preprocesamiento tales como manejo de contraste y niveles de gris, disminución de ruido, suavizado, realces de bordes, filtrado, interpolación y ajustes de falso color. [40]

5.3.1.3 Segmentación

Esta etapa consiste en localizar y delimitar regiones de interés en la imagen, es decir, el conjunto de píxeles que conformen dicha región u objeto. Para mejor exposición de los alcances del presente trabajo, se proponen dos procesos de segmentación automática (figura 5), con el objetivo de separar al individuo del fondo, encontrar bordes que aporten información sobre la delimitación de la región de las mamas en el individuo y, por consecuencia, segmentar las regiones de interés que son requeridas para el cumplimiento de los objetivos establecidos.

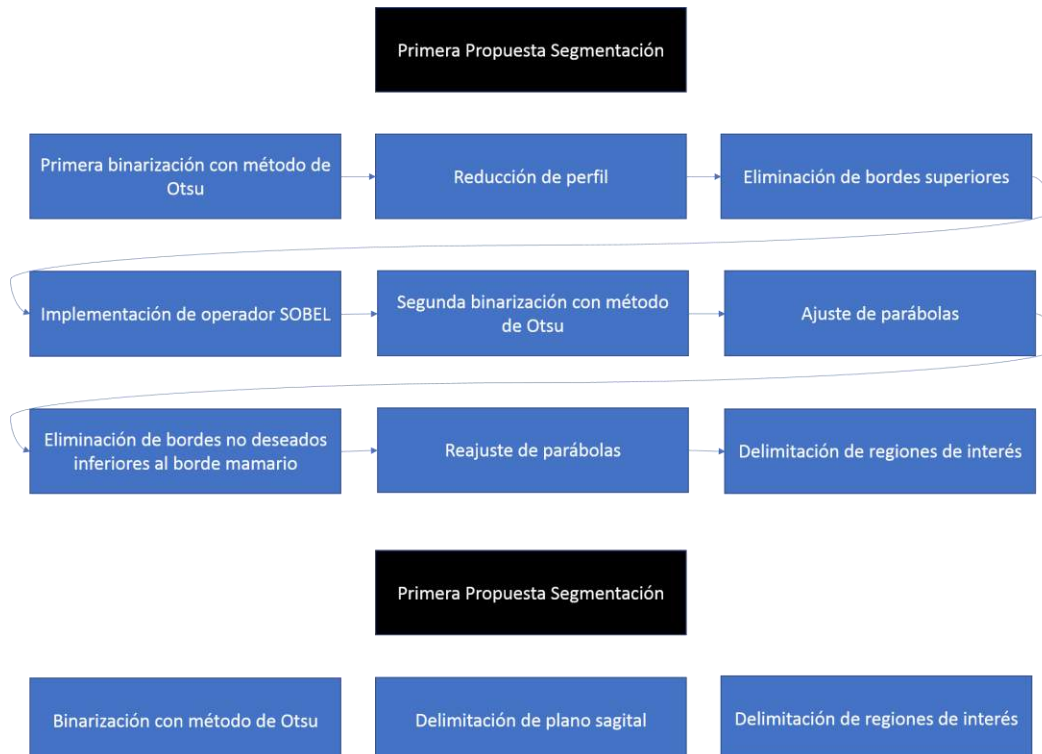


Figura 5. Etapas de las propuestas de segmentación.

5.3.1.4 Umbralización mediante método de Otsu

La umbralización es la separación de objetos o regiones en una imagen digital en escala de grises, utilizando para esto un valor determinado (umbral) como referencia.

Es posible definir la umbralización de la siguiente forma:

$$g(x, y) = \begin{cases} G_O & \text{si } f(x, y) \geq T \\ G_B & \text{si } f(x, y) < T \end{cases}$$

Donde $f(x, y)$ representa la imagen en función de sus coordenadas x, y ; $g(x, y)$ es la imagen resultante “imagen umbralizada”; G_O representa el valor por el que se sustituirá al objeto y G_B es el valor por el cual se sustituirá al fondo. Cuando G_O y G_B reciben los valores de ‘1’ y ‘0’, respectivamente, el proceso recibe el nombre de binarización. Finalmente, T , representa el umbral.

La selección del valor T no es una tarea trivial. En la actualidad, existen diferentes métodos para obtener dicho valor, sin embargo, en el año de 1979, Otsu N. propuso un método de umbralización con base en la acumulación de clases de grises. [41]

El método consiste en encontrar un valor umbral óptimo k^* , situación que ocurre cuando la varianza entre clases de grises es máxima, dicha varianza está dada por la siguiente ecuación:

$$\sigma_B^2 = \frac{[\mu_T \omega(k) - \mu(k)]^2}{\omega(k)[1 - \omega(k)]}$$

La ecuación anterior contiene los siguientes componentes:

Valor medio de la imagen $\mu_T = \mu(L) = \sum_{i=1}^L i p_i$

Momento acumulativo de primer orden $\mu(k) = \sum_{i=1}^k i p_i$

Momento acumulativo de orden cero $\omega(k) = \sum_{i=1}^k p_i$

Donde L representa el número de niveles de gris en la imagen y p_i es la probabilidad de aparición del píxel i .

Analizando la ecuación que define a la varianza entre clases de grises σ_B^2 , es posible determinar que cuando el error cuadrado entre el valor medio de la imagen, escalado por el momento acumulativo de orden cero, y el momento acumulativo de primer orden sea grande, la varianza tenderá al máximo. Desde otro punto de vista, es indispensable encontrar un k óptimo (k^*), el cual cumpla con la siguiente condición:

$$\sigma_B^2(k^*) = \max \sigma_B^2(k); 1 < k < L$$

5.3.1.5 Operador SOBEL

El operador Sobel realiza dos convoluciones con el objetivo de “realzar” bordes verticales y horizontales mediante una aproximación discreta del gradiente de cada punto de la imagen, dando

como resultado una matriz de valores continuos. El resultado de esta operación es una matriz donde cada punto es la norma del vector gradiente en el punto. Estos valores no son discretos ni binarios sino continuos que se interpretan como la intensidad con la que ese valor “representa” a un borde (probabilidad de pertenecer a un borde). Por tanto, mientras mayor sea el gradiente más abrupto será el cambio de tonalidad de un punto a otro (tonalidad de gris) lo que implica que los puntos con mayor “intensidad” resultantes de esta operación son “con razonable probabilidad” miembros de un borde de la imagen. [42]

En la mayoría de las imágenes que se analizan en este trabajo los bordes inferiores de las mamas resaltan con mayor “intensidad”, por lo que se presume que es posible obtener estos bordes utilizando la operación propuesta.

5.3.1.6 Regresión por mínimos cuadrados

El método de regresión por mínimos cuadrados es una técnica de análisis numérico empleada para el ajuste de un conjunto de pares ordenados a funciones lineales de orden n . Mediante este método es posible ajustar una serie de puntos representados por un borde en particular de una imagen a una función lineal de segundo grado.

Sea $\{(x_k, y_k)\}_{k=1}^n$ un conjunto de n puntos en el plano real, y sea $\{f_j(x)\}_{j=1}^m$ una base de m funciones linealmente independientes en un espacio de funciones. Se desea encontrar una función $f(x)$ que sea combinación lineal de las funciones base, de modo que $f(x_k) \approx y_k$, esto es:

$$f(x) = \sum_{j=1}^m c_j f_j(x)$$

Por tanto, se trata de hallar los m coeficientes c_j que hagan que la función aproximante $f(x)$ dé la mejor aproximación para los puntos dados (x_k, y_k) . El criterio de “mejor aproximación” puede variar, pero en general se basa en aquél que minimice una “acumulación” del error individual (en cada punto) sobre el conjunto total. En primer lugar, el error (con signo positivo o negativo) de la función $f(x)$ en un solo punto, (x_k, y_k) , se define como:

$$e_k = y_k - f(x_k)$$

Sin embargo, se intenta medir y minimizar el error en el conjunto de aproximación, $\{(x_k, y_k)\}_{k=1}^n$. La aproximación por mínimos cuadrados emplea el error cuadrático medio para tal fin, siendo este definido como:

$$E_c(f) = \frac{\sum_{k=1}^n (e_k)^2}{n}$$

Para minimizar el error anteriormente definido, se parte del hecho que la función f debe poder describirse como una combinación lineal de una base de funciones. Los coeficientes de la combinación lineal serán los parámetros que queremos determinar. Por ejemplo, supongamos que f es una función cuadrática, lo que quiere decir que es una combinación lineal, $f(x) = ax^2 + bx +$

c , de las funciones $f_1(x) = x^2$, $f_2(x) = x$ y $f_3(x) = 1$ ($m = 3$ para este ejemplo), y que se pretende determinar los valores de los coeficientes: a, b, c de modo que minimicen la suma (S) de los cuadrados de los residuos:

$$S = \sum_{i=1}^n (y_i - f(x_i))^2 = \sum_{i=1}^n (y_i - ax_i^2 - bx_i - c)^2$$

Lo anterior explica el nombre de mínimos cuadrados. A las funciones que multiplican a los coeficientes buscados, que en este caso son: x^2 , x y 1 , se les conoce con el nombre de funciones base de la aproximación, y pueden ser funciones cualesquiera. [43]

5.3.1.7 Descripción

El proceso de descripción implica la caracterización de las regiones de interés a evaluar. Para tal fin, se presenta a continuación el enfoque teórico que regirá al presente trabajo. Se caracterizará cada región de interés mediante descriptores estadísticos de primero y segundo orden, los cuales están son calculados a partir del histograma (primer orden) y la matriz de co-ocurrencia (segundo orden) de las regiones de interés.

5.3.1.7.1 Descriptores estadísticos de primer orden

Los descriptores estadísticos de primer orden son parámetros (que en este trabajo se emplearán como características) obtenidos del histograma de la imagen. Un histograma $H(i)$ representa la frecuencia de aparición de un valor en específico en un conjunto de eventos determinado, para el caso particular de una imagen, i representa el valor de cada píxel, el cual está en función de la profundidad de color de la imagen (8 bits, 12 bits, 16 bits, etc.). Si la frecuencia de aparición de cada posible valor de i es “normalizada” utilizando como denominador la cantidad total de posibles valores de grises se estaría ante un histograma que describiría una función de probabilidad $p(i)$, es decir, la probabilidad de aparición del valor específico de un píxel en la imagen definida como $p(i) = H(i)/T$, donde T es el número total posibles valores de un píxel en la imagen.

Tabla 1. Descriptores estadísticos de primer orden.

Nombre	Definición
Media	$\mu = \sum_{i=0}^{L-1} ip(i)$
Desviación estándar	$\sigma = \sum_{i=0}^{L-1} (i - \mu)^2 p(i)$

Asimetría	$\frac{\sum_{i=0}^{L-1}(i - \mu)^3 p(i)}{\sigma^3}$
Curtosis	$\frac{\sum_{i=0}^{L-1}(i - \mu)^4 p(i)}{\sigma^4}$
Energía	$\sum_{i=0}^{L-1} p^2(i)$
Entropía	$\sum_{i=0}^{L-1} \log_2(p(i)) p(i)$
Moda	Es el dato que aparece con mayor frecuencia
Mediana	Datos menores o iguales que la mediana presentarán el 50% de los datos, y los que sean mayores representarán el otro 50%
Máximo	Es el valor que tiene la mayor magnitud
Rango	Es la diferencia entre el máximo y el mínimo

Nota: L representa la cantidad de valores que puede adquirir un píxel.

5.3.1.7.2 Descriptores estadísticos de segundo orden

Los descriptores estadísticos de segundo orden hacen referencia a un conjunto de valores obtenidos a partir del tratamiento estadístico de matrices de co-ocurrencia en imágenes digitales. Estas características fueron propuestas por Haralick et al en 1979, llamándolas *características de textura*. [44] Las expresiones matemáticas de estas características se definen en la tabla 2.

La matriz de co-ocurrencia es un arreglo matricial que representa la probabilidad conjunta de que dos píxeles tengan un valor de intensidad i y j , respectivamente, a una distancia d , en una dirección determinada.

Tabla 2. Descriptores de segundo orden obtenidos a partir de la matriz de co-ocurrencia.

Nombre	Definición
Segundo momento angular, uniformidad o energía	$f_1 = \sum_i \sum_j p(i, j)^2$
Contraste	$f_2 = \sum_{n=0}^{N-1} n^2 \left\{ \sum_{i=1}^N \sum_{ i-j =n}^N p(i, j) \right\}$
Correlación	$f_3 = \frac{\sum_i \sum_j (ij) p(i, j) - \mu_x \mu_y}{\sigma_x \sigma_y}$
Varianza	$f_4 = \sum_i \sum_j (i - \mu)^2 p(i, j)$
Entropía	$f_5 = \sum_i \sum_j p(i, j) \log\{p(i, j)\}$
Varianza de la diferencia	$f_6 = \sum_{k=0}^{N-1} \left[k - \sum_{l=0}^{N-1} l p_{x-y}(l) \right]^2 p_{x-y}(k)$
Homogeneidad	$f_7 = \sum_i \sum_j \frac{1}{1 + i - j } p(i, j)$
<p>$p(i, j)$ es el valor del elemento (i, j) en la matriz de co-ocurrencia; $\sum_i = \sum_{i=1}^N$; $\sum_j = \sum_{j=1}^N$</p> <p>N es la dimensión de la matriz de co-ocurrencia ($N \times N$). μ es la media de μ_x y μ_y</p> $p_x(i) = \sum_{j=1}^N p(i, j); p_y(j) = \sum_{i=1}^N p(i, j); \mu_x = \sum_{i=1}^N i p_x(i); \mu_y = \sum_{j=1}^N j p_y(j)$ $\sigma_x = \sqrt{\sum_{i=1}^N p_x(i) (i - \mu_x)^2}; \sigma_y = \sqrt{\sum_{j=1}^N p_y(j) (j - \mu_y)^2}; Q(i, j) = \sum_{k=1}^N \frac{p(i, k) p(j, k)}{p_x(i) p_y(j)}$ $p_{x+y}(k) = \sum_{i=1}^N \sum_{\substack{j=1 \\ i+j=k}}^N p(i, j); k = 2, 3, \dots, 2N; p_{x-y}(k) = \sum_{i=1}^N \sum_{\substack{j=1 \\ i-j =k}}^N p(i, j); k = 0, 1, \dots, N-1$	

$$HXY_1 = - \sum_{i=1}^N \sum_{j=1}^N p(i,j) \log\{p_x(i)p_y(j)\}; HXY_2 = - \sum_{i=1}^N \sum_{j=1}^N p_x(i)p_y(j) \log\{p_x(x)p_y(j)\}$$

HX y HY son las entropías de p_x y p_y , respectivamente.

5.4 Inteligencia Artificial.

La inteligencia artificial es una rama de la informática que estudia la creación de sistemas que sean capaces de resolver problemas de forma autónoma, eficiente y eficaz, lo que implica minimizar la probabilidad de fracaso en la realización de la tarea e imitando la capacidad cognitiva de un ser humano. Sin embargo, no existe un consenso sobre la definición de este concepto debido a la divergencia de opiniones con respecto a la definición de inteligencia. [45] En 1950, Alan Turing publicó su artículo "Computing Machinery and Intelligence", en el cual planteó la pregunta "¿Pueden las máquinas pensar?". Este trabajo constituyó un hito en el desarrollo formal de sistemas inteligentes en ciencia de la computación, siendo el referente moderno más importante para el desarrollo posterior de esta disciplina. [46]

Actualmente existen diferentes desarrollos que utilizan la inteligencia artificial, tales como: detección de material no deseado en bandejas de correo electrónico, búsqueda de información por medio de imágenes en buscadores web, automóviles autónomos, reconocedores de voz y, en medicina, algunos desarrollos en diagnóstico de imagen como el propuesto por Yuzheng Wu et al [47], quienes propusieron una herramienta de diagnóstico mediante redes neuronales artificiales para la discriminación de mastogramas.

5.4.1 Aprendizaje Máquina o Automático.

En los últimos años ha surgido un interés renovado y creciente por el desarrollo de una rama de la Inteligencia Artificial llamada Aprendizaje Automático (Machine Learning), como consecuencia al desarrollo constante de mejores sistemas de procesamiento informático y la interconectividad que ofrecen las nuevas tecnologías en telecomunicaciones, lo cual propicia el desarrollo de nuevas áreas del conocimiento en el tratamiento de grandes cantidades de datos (Big Data), y nuevas aplicaciones en Ciencia de Datos (Data Science) así como en el estudio y desarrollo de soluciones con base en Reconocimiento de Patrones (Pattern Recognition) y Minería de Datos (Data Mining). [48]

El Aprendizaje Automático consiste en "entrenar" a un sistema artificial de tal manera que sea capaz de predecir una salida a partir de un conjunto de características, proceso que es conocido como "aprendizaje". [49]

El desarrollo de sistemas de aprendizaje automático no es un tema nuevo en ciencia de la computación ya que existen antecedentes de la aplicación del concepto en modelos matemáticos desde la década de los sesentas. Como ejemplo es posible citar a una idealización del funcionamiento de la neurona biológica. Donde se partía del supuesto de que una neurona es, en última instancia, la suma de distintas entradas ponderadas. Y la forma en que la neurona pondera estas entradas sería consecuencia de un proceso de "aprendizaje" el cual fue simplificado utilizando un sistema iterativo de penalización. Este modelo fue nombrado por Rosenblatt como *Perceptrón*,

el cual es reconocido como la primera idealización matemática de una neurona biológica en una aplicación computacional. [50]

5.4.1.1 Tipos de aprendizaje automático

El aprendizaje automático se divide en dos grandes áreas: aprendizaje supervisado y aprendizaje no supervisado. La primera consiste en encontrar una función de hipótesis predictiva obtenida a partir de diferentes ejemplos de tal manera que el sistema conoce también la salida. La segunda consiste en encontrar una relación entre datos, de tal manera que se eliminen redundancias en el conjunto o se creen agrupamientos de clases. [51]

Cabe mencionar que existen otras técnicas de aprendizaje como son las de aprendizaje semi-supervisado, por reforzamiento, transducción y multitarea. Sin embargo, no serán objeto de este trabajo.

5.4.1.2 Aprendizaje supervisado

Existen dos tipos de problemas que enfrenta el aprendizaje supervisado: problemas de regresión y problemas de clasificación. [52]

Los problemas de regresión son aquellos que implican la elaboración de una función predictiva o función de hipótesis con valores de salida continuos, por ejemplo, predecir los precios de una casa en un área determinada. La técnica más común para la resolución de este tipo de problemas es la regresión lineal, la cual fue propuesta por la ciencia estadística. [53]

Por otra parte, existen los problemas de clasificación, los cuales requieren la creación de funciones de hipótesis que tengan como salida valores discretos, estableciendo clases de salida. [52] Algunas técnicas desarrolladas para resolverlos son: Maquinas de Soporte Vectorial [54], Redes Neuronales Artificiales de Primera Generación (Perceptrón) [50], Redes Neuronales Artificiales de Segunda Generación (Aprendizaje Backpropagation o Regla de Aprendizaje Delta) [55], Redes Neuronales de Tercera Generación (Recurrentes) [56], Regresión Logística [57]; algunas de las cuales serán descritas en este trabajo.

5.4.1.2.1 Decision Tree

Este modelo no lineal se construye a partir de las características. En un plano 2D con dos características, el modelo serían una serie de líneas HORIZONTALES y VERTICALES las cuales definen particiones donde se agrupan los puntos que representan a una u otra clase. El objetivo es encontrar la cantidad óptima de particiones que nos permitan discriminar “lo mejor posible” las clases de salida. [58] No es propiamente un modelo que implemente un algoritmo de aprendizaje automático ya que el diseñador predefine el peso que se le otorgará a cada atributo.

Los pasos del algoritmo para alcanzar el mejor modelo son:

1. Escoger un atributo (característica) de la base de datos que sea el “más prometedor” para separar a las clases de salida.
2. Calcular el peso que este atributo seleccionado tiene al dividir los datos.
3. Dividir los datos con base al mejor valor (umbral) del atributo.
4. Ir nuevamente al paso uno y repetir para el resto de los atributos.

5.4.1.2.2 Random Forest

Este modelo es un conjunto de árboles de decisión que se implementan en subconjuntos del conjunto de entrenamiento. El promedio del rendimiento de cada uno de estos árboles es el valor resultante. [59] Al igual que el modelo anterior no implementa un algoritmo de aprendizaje automático.

5.4.1.2.3 Regresión logística.

La técnica de regresión logística fue desarrollada en el año de 1958 por David Cox [57]. Consiste en la creación de un modelo binario de clasificación el cual es empleado para estimar la probabilidad de pertenencia de un elemento a una clase determinada (0,1; verdadero, falso).

5.4.1.2.3.1 Clasificación binaria en la Regresión Logística

La clasificación binaria se refiere a que un sistema tiene únicamente dos valores de salida (0,1). Por lo tanto, es posible definir una variable $y \in \{0,1\}$, la cual representa las clases de salida, para este trabajo en particular 0 es la clase “negativa” y 1 la clase “positiva”.

Por lo tanto, nuestra función de hipótesis o función predictiva $h_{\theta}(x)$ debe satisfacer la siguiente condición $0 \leq h_{\theta}(x) \leq 1$. Siendo x el vector característico de un evento o elemento particular del cual se desea conocer su clase (negativa o positiva).

La regresión lineal define a $h_{\theta} = \theta^T x$, siendo θ el vector de pesos obtenido después de un entrenamiento del algoritmo con m ejemplos. Sin embargo, esta función de hipótesis entrega valores continuos cualesquiera, en otras palabras, no cumple con la restricción que anteriormente describimos para la regresión logística. Por lo anterior, se presenta el concepto de función sigmoidea, la cual definimos como:

$$g(z) = \frac{1}{1 + e^{-z}}$$

Esta función tiene la propiedad de presentar un intervalo de salida (0,1), apreciable gráficamente en la figura 6.

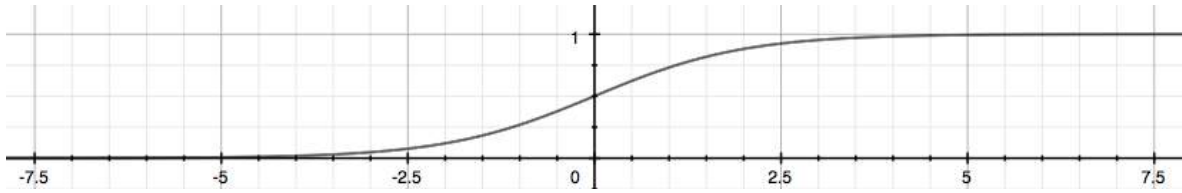


Figura 6. Función sigmoidea.

En consecuencia, definimos $z = \theta^T x$ estableciendo entonces $h_\theta(x) = g(\theta^T x)$

Con lo anterior, $h_\theta(x)$ proporcionará la probabilidad de pertenencia de x a la clase 1 o “positiva”.

Por ejemplo, si $h_\theta(x) = 0.65$ entonces x tiene un 65% de probabilidad de pertenecer a la clase 1.

Formalmente definimos la probabilidad de que ocurra cada evento de la siguiente manera:

$$h_\theta(x) = P(y = 1|x; \theta) = 1 - P(y = 0|x; \theta)$$

$$P(y = 0|x; \theta) + P(y = 1|x; \theta) = 1$$

5.4.1.2.3.2 Frontera de Decisión

Con la finalidad de establecer una frontera de decisión definimos lo siguiente:

$$h_\theta(x) \geq 0.5 \rightarrow y = 1$$

$$h_\theta(x) < 0.5 \rightarrow y = 0$$

De esta forma la función logística g dará como salida un valor mayor o igual a 0.5 cuando su entrada z sea mayor o igual a 0.

Recordando:

$$z = 0, e^0 = 1 \Rightarrow g(z) = 1/2$$

$$z \rightarrow \infty, e^{-\infty} \rightarrow 0 \Rightarrow g(z) = 1$$

$$z \rightarrow -\infty, e^{\infty} \rightarrow \infty \Rightarrow g(z) = 0$$

Si la entrada de g es $\theta^T x$, entonces significa:

$$h_\theta = g(\theta^T x) \geq 0.5$$

$$\text{cuando } \theta^T x \geq 0$$

De lo anterior podemos decir:

$$\theta^T x \geq 0 \Rightarrow y = 1$$

$$\theta^T x < 0 \Rightarrow y = 0$$

De lo anterior se establece que la frontera de decisión es la línea que separa el área donde $y = 0$ y $y = 1$. Siendo ésta, consecuencia de la función de hipótesis.

Un ejemplo de lo anterior es:

$$\theta = \begin{bmatrix} 5 \\ -1 \\ 0 \end{bmatrix}$$

$$y = 1 \text{ si } 5 + (-1)x_1 + 0x_2 \geq 0$$

$$5 - x_1 \geq 0$$

$$-x_1 \geq -5$$

$$x_1 \leq 5$$

En este caso, la frontera de decisión es una línea recta vertical localizada en el plano donde $x_1 = 5$, y todo lo que se encuentre a la izquierda pertenece a la clase $y = 1$, mientras que todo lo que se encuentre a la derecha pertenece a $y = 0$.

5.4.1.2.3.3 Función de Costo

La función de costos de un algoritmo de aprendizaje se refiere a la función encargada de medir la eficiencia del aprendizaje de éste. Para el caso particular de la regresión logística definimos a la función de costos $J(\theta)$ como:

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m \text{Cost}(h_{\theta}(x^{(i)}), y^{(i)})$$

$$\text{Cost}(h_{\theta}(x), y) = -\ln(h_{\theta}(x)) \quad \text{si } y = 1$$

$$\text{Cost}(h_{\theta}(x), y) = -\ln(1 - h_{\theta}(x)) \quad \text{si } y = 0$$

$x^{(i)}$: el ejemplo i -ésimo.

$y^{(i)}$: la clase del ejemplo i -ésimo.

Donde $1 \leq i \leq m$, siendo m el número de ejemplos empleados en el entrenamiento.

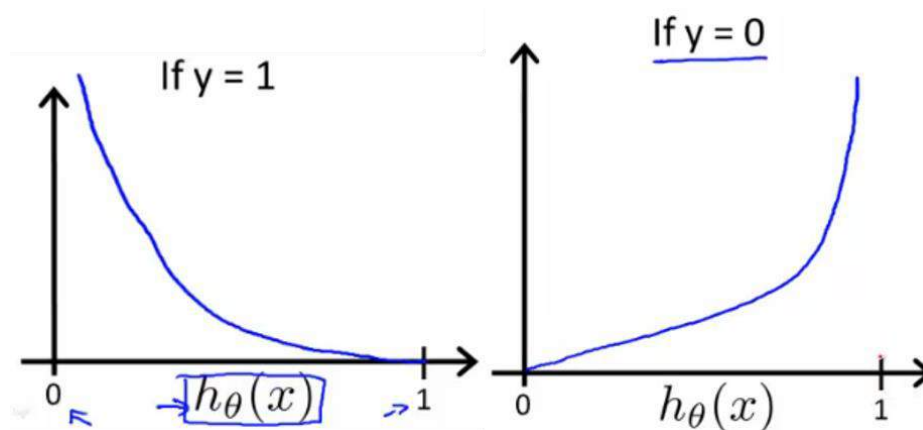


Figura 7. Comportamiento de la función de hipótesis con respecto al valor de 'y'.

Cuando la salida de la función de hipótesis esté alejada de y , entonces la salida de la función de costos será grande. Si la función de hipótesis es igual a y , entonces nuestro costo es 0.

$$Cost(h_{\theta}(x), y) = 0 \text{ si } h_{\theta}(x) = y$$

$$Cost(h_{\theta}(x), y) \rightarrow \infty \text{ si } y = 0 \text{ \& } h_{\theta}(x) \rightarrow 1$$

$$Cost(h_{\theta}(x), y) = 0 \text{ si } y = 1 \text{ \& } h_{\theta}(x) \rightarrow 0$$

Si la respuesta y es 0, entonces la función de costos será 0 si la salida de la función de hipótesis también es 0. Si la salida de la función de hipótesis tiende a 1, entonces la salida de la función de costos tenderá a infinito.

Si la respuesta y es 1, entonces la función de costos será 0 si la salida de la función de hipótesis también es 1. Si la salida de la función de hipótesis tiende a 0, entonces la salida de la función de costos tenderá a infinito.

Función de costos simplificada y gradiente de descenso.

Es posible simplificar los dos casos condicionales de la función de costos a un solo caso:

$$Cost(h_{\theta}(x), y) = -y \ln(h_{\theta}(x)) - (1 - y) \ln(1 - h_{\theta}(x))$$

Nótese que cuando y es igual a 1, entonces el segundo termino $(1 - y) \log(1 - h_{\theta}(x))$ será cero y no afectará el resultado. Si y es igual a cero, entonces el primer termino $-y \log(h_{\theta}(x))$ será cero y no afectará el resultado.

Podemos escribir la función de costos de la siguiente manera:

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^m [y^{(i)} \log(h_{\theta}(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_{\theta}(x^{(i)}))]$$

La implementación vectorizada de esta expresión es:

$$h = g(X\theta)$$

$$J(\theta) = \frac{1}{m} \cdot (-y^T \ln(h) - (1 - y)^T \log(1 - h))$$

5.4.1.2.3.4 Descenso por gradiente

El descenso por gradiente es la cantidad calculada a partir de la función de costos la cual es sustraída al vector de pesos θ de forma iterativa, con la finalidad de minimizar la función de costos, lo que eventualmente dará como resultado una función de hipótesis optimizada en el entrenamiento.

De forma formal se define el algoritmo como:

Repeat {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

$$\}$$

Es posible sustituir la derivada utilizando de la siguiente forma:

$$\text{Repeat } \{$$

$$\theta_j := \theta_j - \frac{\alpha}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

$$\}$$

Donde j se encuentra entre los valores 1 y n , donde n es el número de características

La implementación vectorizada del algoritmo anterior es:

$$\theta := \theta - \frac{\alpha}{m} X^T (g(X\theta) - y)$$

5.4.1.2.3.4.1 Derivada parcial de $J(\theta)$

Para calcular la derivada parcial de $J(\theta)$ primero es necesario el cálculo de la derivada parcial de la función sigmoidea.

$$\begin{aligned} \sigma(x)' &= \left(\frac{1}{1+e^{-x}} \right)' = \frac{-(1+e^{-x})'}{(1+e^{-x})^2} = \frac{-1' - (e^{-x})'}{(1+e^{-x})^2} = \frac{0 - (-x)'(e^{-x})}{(1+e^{-x})^2} = \frac{-(-1)(e^{-x})}{(1+e^{-x})^2} \\ &= \frac{e^{-x}}{(1+e^{-x})^2} = \left(\frac{1}{1+e^{-x}} \right) \left(\frac{e^{-x}}{1+e^{-x}} \right) = \sigma(x) \left(\frac{1+e^{-x}}{1+e^{-x}} - \frac{1}{1+e^{-x}} \right) \\ &= \sigma(x)(1 - \sigma(x)) \end{aligned}$$

Con este resultado es posible encontrar derivada parcial de $J(\theta)$:

$$\begin{aligned} \frac{\partial}{\partial \theta_j} J(\theta) &= \frac{\partial}{\partial \theta_j} \frac{-1}{m} \sum_{i=1}^m [y^{(i)} \log(h_{\theta}(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_{\theta}(x^{(i)}))] \\ &= -\frac{1}{m} \sum_{i=1}^m \left[y^{(i)} \frac{\partial}{\partial \theta_j} \log(h_{\theta}(x^{(i)})) + (1 - y^{(i)}) \frac{\partial}{\partial \theta_j} \log(1 - h_{\theta}(x^{(i)})) \right] \\ &= -\frac{1}{m} \sum_{i=1}^m \left[\frac{y^{(i)} \frac{\partial}{\partial \theta_j} h_{\theta}(x^{(i)})}{h_{\theta}(x^{(i)})} + \frac{(1 - y^{(i)}) \frac{\partial}{\partial \theta_j} (1 - h_{\theta}(x^{(i)}))}{1 - h_{\theta}(x^{(i)})} \right] \end{aligned}$$

$$\begin{aligned}
&= -\frac{1}{m} \sum_{i=1}^m \left[\frac{y^{(i)} \frac{\partial}{\partial \theta_j} \sigma(\theta^T x^{(i)})}{h_\theta(x^{(i)})} + \frac{(1 - y^{(i)}) \frac{\partial}{\partial \theta_j} (1 - \sigma(\theta^T x^{(i)}))}{1 - h_\theta(x^{(i)})} \right] \\
&= -\frac{1}{m} \sum_{i=1}^m \left[\frac{y^{(i)} \sigma(\theta^T x^{(i)}) (1 - \sigma(\theta^T x^{(i)})) \frac{\partial}{\partial \theta_j} \theta^T x^{(i)}}{h_\theta(x^{(i)})} \right. \\
&\quad \left. + \frac{-(1 - y^{(i)}) \sigma(\theta^T x^{(i)}) (1 - \sigma(\theta^T x^{(i)})) \frac{\partial}{\partial \theta_j} \theta^T x^{(i)}}{1 - h_\theta(x^{(i)})} \right] \\
&= -\frac{1}{m} \sum_{i=1}^m \left[\frac{y^{(i)} h_\theta(x^{(i)}) (1 - h_\theta(x^{(i)})) \frac{\partial}{\partial \theta_j} \theta^T x^{(i)}}{h_\theta(x^{(i)})} \right. \\
&\quad \left. - \frac{(1 - y^{(i)}) h_\theta(x^{(i)}) (1 - h_\theta(x^{(i)})) \frac{\partial}{\partial \theta_j} \theta^T x^{(i)}}{1 - h_\theta(x^{(i)})} \right] \\
&= -\frac{1}{m} \sum_{i=1}^m [y^{(i)} (1 - h_\theta(x^{(i)})) x_j^{(i)} - (1 - y^{(i)}) h_\theta(x^{(i)}) x_j^{(i)}] \\
&= -\frac{1}{m} \sum_{i=1}^m [y^{(i)} (1 - h_\theta(x^{(i)})) - (1 - y^{(i)}) h_\theta(x^{(i)})] x_j^{(i)} \\
&= -\frac{1}{m} \sum_{i=1}^m [y^{(i)} - y^{(i)} h_\theta(x^{(i)}) - h_\theta(x^{(i)}) + y^{(i)} h_\theta(x^{(i)})] x_j^{(i)} \\
&= -\frac{1}{m} \sum_{i=1}^m [y^{(i)} - h_\theta(x^{(i)})] x_j^{(i)} \\
&= \frac{1}{m} \sum_{i=1}^m [h_\theta(x^{(i)}) - y^{(i)}] x_j^{(i)}
\end{aligned}$$

La versión vectorizada:

$$\nabla J(\theta) = \frac{1}{m} \cdot X^T \cdot (g(X \cdot \theta) - y)$$

5.4.1.2.4 Máquina de Soporte Vectorial

La máquina de soporte vectorial fue desarrollada por Vladimir Vapnick y su equipo de desarrollo de laboratorio en AT&T en 1995. [54] La descripción del funcionamiento de esta técnica de aprendizaje automático es descrita a continuación.

5.4.1.2.4.1 Objetivo de optimización

La Máquina de Soporte Vectorial (SVM) es uno de los algoritmos de aprendizaje supervisado más utilizados en la investigación sobre generación de modelos predictivos en la actualidad.

Recordando, en la Regresión Logística utilizábamos las siguientes reglas:

si $y = 1$, entonces $h_{\theta}(x) \approx 1$ & $\theta^T x \gg 0$

si $y = 0$, entonces $h_{\theta}(x) \approx 0$ & $\theta^T x \ll 0$

La función de costos de la Regresión Logística es:

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m -y^{(i)} \ln(h_{\theta}(x^{(i)})) - (1 - y^{(i)}) \ln(1 - h_{\theta}(x^{(i)}))$$
$$= \frac{1}{m} \sum_{i=1}^m -y^{(i)} \ln\left(\frac{1}{1 + e^{-\theta^T x^{(i)}}}\right) - (1 - y^{(i)}) \ln\left(1 - \frac{1}{1 + e^{-\theta^T x^{(i)}}}\right)$$

Para construir la SVM, es preciso modificar el primer término de la función de costos $-\ln(h_{\theta}(x)) = -\ln\left(\frac{1}{1 + e^{-\theta^T x}}\right)$ de tal manera que cuando $\theta^T x$ (en adelante, se referirá a este término como z) sea mayor que 1, la salida sea 0. Además, para valores de z menores que 1, se utilizará una línea recta decreciente en lugar de la curva sigmoidea. En la literatura esto es llamado función *hinge loss* o función de pérdida de bisagra. [60]

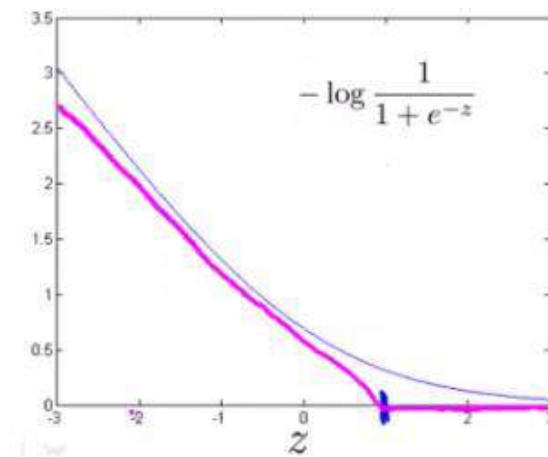


Figura 8. Función Hinge Loss para el primer término de la función de costos.

De forma similar, se modifica el segundo término de la función de costos $-\ln(1 - h_\theta) = -\ln\left(1 - \frac{1}{1+e^{-\theta^T x}}\right)$, de tal manera que cuando z sea menor o igual a -1 , la salida sea 0. De igual forma se modifica para que con valores de z mayores que -1 , se utilice una línea recta creciente en lugar de la curva sigmoidea.

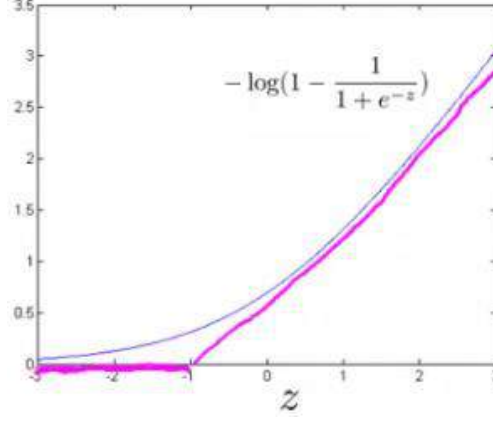


Figura 9. Función Hinge Loss para el segundo término de la función de costos.

Denominamos estos términos modificados como $cost_1(z)$ y $cost_0(z)$ (respectivamente, nótese que el $cost_1(z)$ es el costo para el clasificador cuando $y = 1$, y $cost_0(z)$ es el costo para el clasificador cuando $y = 0$), y es posible definirlos (dónde k es una constante arbitraria de la magnitud de la inclinación de la línea recta):

$$z = \theta^T x$$

$$cost_0 = \max(0, k(1 + z))$$

$$cost_1 = \max(0, k(1 - z))$$

Recordando la función de costo regularizada de la regresión logística:

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m y^{(i)} (-\ln(h_\theta(x^{(i)}))) + (1 - y^{(i)}) (-\ln(1 - h_\theta(x^{(i)}))) + \frac{\lambda}{2m} \sum_{j=1}^n \theta_j^2$$

Donde $\frac{\lambda}{2m} \sum_{j=1}^n \theta_j^2$ recibe el nombre de término de regularización y es utilizado para ajustar la función de costos con el fin de evitar un sobreajuste o subajuste de la función de hipótesis resultante del entrenamiento. λ es la tasa de regularización y n es el número de características, el cual es equivalente al tamaño del vector θ .

La función anterior es posible transformarla en la función de costo de la máquina de soporte vectorial sustituyendo los valores de $cost_0(z)$ y $cost_1(z)$:

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m y^{(i)} cost_1(\theta^T x^{(i)}) + (1 - y^{(i)}) cost_0(\theta^T x^{(i)}) + \frac{\lambda}{2m} \sum_{j=1}^n \theta_j^2$$

Es posible modificar esta ecuación multiplicándola por el término m (en consecuencia, removiendo el factor m en el denominador). Nótese que esto no afecta la optimización, ya que se multiplica la función de costo por una constante positiva.

$$J(\theta) = \sum_{i=1}^m y^{(i)} \text{cost}_1(\theta^T x^{(i)}) + (1 - y^{(i)}) \text{cost}_0(\theta^T x^{(i)}) + \frac{\lambda}{2} \sum_{j=1}^n \theta_j^2$$

Además, la convención en la SVM dicta utilizar el factor C en lugar de λ , de tal forma que:

$$J(\theta) = C \sum_{i=1}^m y^{(i)} \text{cost}_1(\theta^T x^{(i)}) + (1 - y^{(i)}) \text{cost}_0(\theta^T x^{(i)}) + \frac{1}{2} \sum_{j=1}^n \theta_j^2$$

Esto es equivalente a multiplicar la ecuación por $C = \frac{1}{\lambda}$, lo que resulta en los mismos valores cuando es optimizada. Ahora, cuando se desea regularizar más (es decir, reducir el sobreajuste), se decrementa C , y cuando se desea regularizar menos (esto es, reducir el subajuste), se incrementa C .

Finalmente se observa que la función de hipótesis de la SVM no es interpretada como la probabilidad de y de ser 1 o 0 (como ocurre para la función de hipótesis de la regresión logística). En lugar de esto, la salida es 1 o 0. En términos técnicos es una función discriminante.

$$h_{\theta}(x) = \begin{cases} 1 & \text{if } \theta^T x \geq 0 \\ 0 & \text{otro caso} \end{cases}$$

5.4.1.2.4.2 Kernel

El Kernel de una máquina de soporte vectorial se refiere a la función a la cual se intentará aproximar al conjunto de datos la cual puede ser lineal, polinomial, etc.

Dado x , calcular nuevas características en función de su proximidad a los puntos de referencia $l^{(1)}, l^{(2)}, l^{(3)}$.

Para realizar lo anterior, se debe encontrar la “similitud” de x y algún punto de referencia $l^{(i)}$:

$$f_i = \text{similitud}(x, l^{(i)}) = e^{-\frac{\|x - l^{(i)}\|^2}{2\sigma^2}}$$

Esta función “similitud” es llamada Kernel Gaussiano. Es un ejemplo específico de un kernel.

La función de similitud puede también ser escrita como a continuación se muestra:

$$f_i = \text{similitud}(x, l^{(i)}) = e^{-\frac{\sum_{j=1}^n (x_j - l_j^{(i)})^2}{2\sigma^2}}$$

Existen un par de propiedades de la función similitud:

$$\text{Si } x \approx l^{(i)}, \text{ entonces } f_i = e^{-\frac{\approx 0^2}{2\sigma^2}} \approx 1$$

$$\text{Si } x \text{ es distante de } l^{(i)}, \text{ entonces } f_i = e^{-\frac{(\text{número grande})^2}{2\sigma^2}} \approx 0$$

En otras palabras, si x y el punto de referencia son cercanos, entonces la similaridad será cercana a 1, y si x y el punto de referencia son lejanos uno con respecto al otro, la similaridad será cercana a 0.

Cada punto de referencia otorga las características de la hipótesis:

$$l^{(1)} \rightarrow f_1$$

$$l^{(2)} \rightarrow f_2$$

$$l^{(3)} \rightarrow f_3$$

...

$$h_\theta = \theta_1 f_1 + \theta_2 f_2 + \theta_3 f_3 + \dots$$

σ^2 es un parámetro del Kernel Gaussiano.

Una manera de obtener los puntos de referencia en los lugares exactos es colocarlos en lugares de los ejemplos de entrenamiento. Esto nos da m puntos de referencia, con un punto de referencia por ejemplo de entrenamiento.

Dado el ejemplo x :

$$f_1 = \text{similaridad}(x, l^{(1)}), f_2 = \text{similaridad}(x, l^{(2)}), f_3 = \text{similaridad}(x, l^{(3)}), \text{ etcétera.}$$

Lo anterior genera un “vector de características”, f_i de todas las características del ejemplo $x^{(i)}$. Establecemos $f_0 = 1$ que corresponde a θ_0 . Por consiguiente, dado el ejemplo de entrenamiento $x^{(i)}$:

$$x^{(i)} \rightarrow \begin{bmatrix} f_1^{(i)} = \text{similaridad}(x^{(i)}, l^{(1)}) \\ f_2^{(i)} = \text{similaridad}(x^{(i)}, l^{(2)}) \\ \vdots \\ f_m^{(i)} = \text{similaridad}(x^{(i)}, l^{(m)}) \end{bmatrix}$$

Para obtener los parámetros θ podemos usar un algoritmo de minimización del SVM pero con $f^{(i)}$ sustituyendo a $x^{(i)}$:

$$\min_{\theta} \left(C \sum_{i=1}^m [y^{(i)} \text{cost}_1(\theta^T f^{(i)}) + (1 - y^{(i)}) \text{cost}_0(\theta^T f^{(i)})] + \frac{1}{2} \sum_{j=1}^n \theta_j^2 \right)$$

5.4.1.2.5 Redes Neuronales Artificiales

Las redes neuronales artificiales en este trabajo están inspiradas en el trabajo de Hinton et al, quienes desarrollaron en el año de 1985 una regla de aprendizaje llamada *regla delta*. [55] Esta

nueva técnica de aprendizaje marco un hito en el desarrollo de sistemas inteligentes, iniciando así la segunda generación de redes neuronales artificiales.

5.4.1.2.5.1 Hipótesis no lineal

Realizar la regresión lineal con un complejo conjunto de datos con varias características es difícil de hacer. Supongamos que se desea crear una función de hipótesis de tres características, incluidos todos los términos cuadráticos:

$$g(\theta_0 + \theta_1 x_1^2 + \theta_2 x_1 x_2 + \theta_3 x_1 x_2 + \theta_4 x_2^2 + \theta_5 x_2 x_3 + \theta_6 x_3^2)$$

Lo anterior resulta en seis características. La manera para calcular cuantas características para todos los términos polinomiales es la combinación: $(n + r - 1)! r! (n - 1)!$. En este caso particular se toman todas las combinaciones posibles de dos elementos de tres características $(3 + 2 - 1)! (2! \cdot (3 - 1)!) = 4! 4 = 6$.

Para 100 características, si se desea encontrar (hacer realizar, obtener) los términos cuadráticos obtendríamos $(100 + 2 - 1)! (2 \cdot (100 - 1)) = 5050$ resultando en nuevas características.

Es posible aproximar el número de nuevas características cuando se calculan todos de los términos cuadráticos de la siguiente forma $O(n^2/2)$. Y si se desea incluir todos los términos cuadráticos en la hipótesis, las características crecerían asintóticamente en $O(n^3)$. Estos son crecimientos pronunciados, por lo que a medida que aumenta el número de características, el número de características cuadráticas y cúbicas aumentan rápidamente y eventualmente se convierte en una función de hipótesis impráctica para ser utilizada en un sistema computacional.

Ejemplo: supóngase que se tiene un conjunto de entrenamiento fotografías en blanco y negro de tamaño 50 x 50 píxeles, y si la meta es clasificar cuales son las fotografías que contienen carros. Entonces el tamaño de las características por cada ejemplo es $n = 2500$, si utilizamos cada píxel como característica.

Ahora, continuando con el ejemplo, supóngase que se desea realizara una función de hipótesis con términos cuadráticos. El número aproximado de nuevas características será $(2500^2/2) = 3125000$, lo cual es impráctico.

Las redes neuronales artificiales ofrecen una alternativa para realizar funciones de hipótesis compleja con una gran cantidad de características.

5.4.1.2.5.2 Neurona biológica

Para ilustrar el funcionamiento de una neurona artificial es necesario recordar la estructura y funcionamiento de una neurona biológica.

Las neuronas son un tipo de células del sistema nervioso cuya principal propiedad es la excitabilidad eléctrica de su membrana plasmática. Están especializadas en la recepción de estímulos y conducción del impulso nervioso (en forma de potencial de acción) entre ellas o con otros tipos celulares como, por ejemplo, las fibras musculares de la placa motora. También son altamente diferenciadas y la mayoría no se dividen una vez alcanzan la madurez; no obstante, una minoría sí lo hace. [61]

Las neuronas presentan unas características morfológicas típicas que sustentan sus funciones: un cuerpo celular, llamado soma o “pericarion” central; una o varias prolongaciones cortas que generalmente transmiten impulsos hacia el soma celular, denominadas dendritas; y una prolongación larga, denominada axón, que conduce los impulsos desde el soma hacia otra neurona. [62]

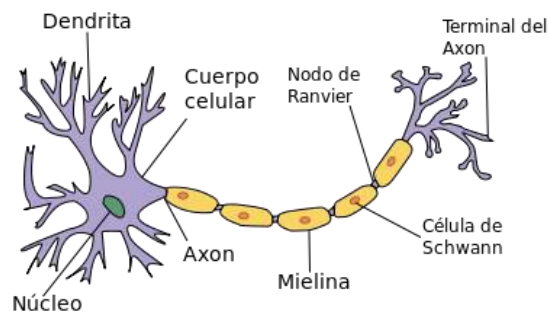


Figura 10. Estructura de una neurona biológica.

5.4.1.2.5.3 Neurona artificial

Las redes neuronales artificiales son imitaciones limitadas de una neurona biológica. En los últimos años ha existido un interés renovado debido a los avances del hardware computacional. [63]

El primer investigador en imitar el funcionamiento básico de una neurona en un sistema informático fue Rosenblatt, quien creó la neurona *Perceptrón* en 1958. [50] Este modelo se basaba en el funcionamiento teórico de la transmisión de información en una neurona biológica. Sin embargo, en 1969, Minsky y Papert demostraron que este modelo no es capaz de resolver problemas donde el conjunto de entrenamiento presentaba ejemplos con características redundantes o con dependencia lineal. [64]

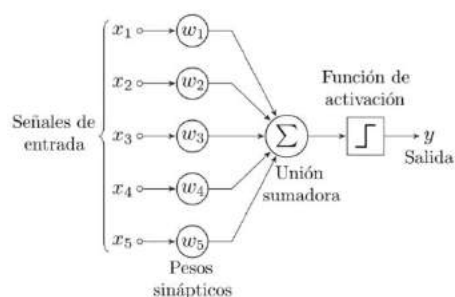


Figura 11. Estructura de una neurona perceptrón.

Sin embargo, en la década de los ochenta surgió la segunda generación de redes neuronales artificiales, la cual propuso una solución a las cuestiones planteadas por Minsky y Papert en 1969. [55] Las neuronas de segunda generación lograron demostrar (a partir de la regla delta de aprendizaje) la posibilidad de construir arquitecturas de redes neuronales con varias capas (capas ocultas) que fuesen capaces de aprender soluciones a problemas con redundancia en las características (dependencia lineal). Un ejemplo de este tipo de redes se muestra en la figura 12.

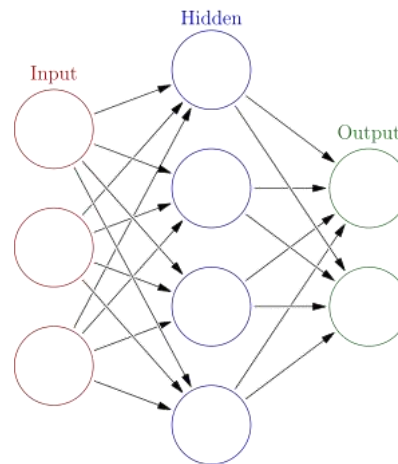


Figura 12. Estructura de una red neuronal de segunda generación.

5.4.1.2.5.4 Modelo de representación

Las neuronas artificiales son básicamente unidades computacionales con entradas (dendritas) y salidas (axones). En el modelo teórico, las dendritas son las entradas de nuestras características de un ejemplo $x^{(i)}$, específicamente las dendritas son representadas por las entradas $x_1^{(i)}, \dots, x_n^{(i)}$. En este modelo la característica x_0 es llamada “unidad de oscilación”.

Las redes neuronales artificiales utilizan una función logística de clasificación $\frac{1}{1+e^{-\theta^T x}}$. La cuál es llamada comúnmente función de activación. Para los parámetros “theta” se utiliza el nombre “pesos”.

Visualmente, una representación simple sería de la forma:

$$\begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} \rightarrow [] \rightarrow h_{\theta}(x)$$

La red neuronal básica está integrada de por lo menos 3 capas, los nodos de entrada (capa 1) se dirigen a otros nodos internos (capa 2), y finalmente son computados por la capa de salida (capa 3). En capas más complejas, las capas internas entre la 1 y la 2 son llamadas “capas ocultas”.

Las capas ocultas o intermedias contienen unidades de activación o intermedias denominadas por la letra a .

$a_i^{(j)}$ = unidad de activación i en la capa j.

$\theta^{(j)}$ = matriz de pesos entre la capa j hacia la capa j + 1.

Si se tiene una capa oculta, se observaría lo siguiente:

$$\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \rightarrow \begin{bmatrix} a_1^{(2)} \\ a_2^{(2)} \\ a_3^{(2)} \end{bmatrix} \rightarrow h_\theta(x)$$

Los valores de cada uno de los nodos de activación son obtenidos de la forma siguiente:

$$a_1^{(2)} = g \left(\theta_{10}^{(1)} x_0 + \theta_{11}^{(1)} x_1 + \theta_{12}^{(1)} x_2 + \theta_{13}^{(1)} x_3 \right)$$

$$a_2^{(2)} = g \left(\theta_{20}^{(1)} x_0 + \theta_{21}^{(1)} x_1 + \theta_{22}^{(1)} x_2 + \theta_{23}^{(1)} x_3 \right)$$

$$a_3^{(2)} = g \left(\theta_{30}^{(1)} x_0 + \theta_{31}^{(1)} x_1 + \theta_{32}^{(1)} x_2 + \theta_{33}^{(1)} x_3 \right)$$

$$h_\theta(x) = a_1^{(3)} = g \left(\theta_{10}^{(2)} a_0^{(2)} + \theta_{11}^{(2)} a_1^{(2)} + \theta_{12}^{(2)} a_2^{(2)} + \theta_{13}^{(2)} a_3^{(2)} \right)$$

Esto significa que se computan los nodos de activación utilizando una matriz de parámetros 3X4. Se aplica cada fila de los parámetros de las entradas para obtener el valor para cada nodo de activación. La hipótesis de salida es la función logística aplicada a la suma de los valores de los nodos de activación, los cuales son multiplicados por otra matriz de parámetros $\theta^{(2)}$ conteniendo los pesos para la segunda capa de nodos.

Cada capa tiene su propia matriz de pesos $\theta^{(j)}$.

La dimensión de estas matrices de pesos es determinada de la siguiente forma:

Si la red tiene s_j unidades en la capa j y s_{j+1} unidades en la capa j + 1, entonces $\theta^{(j)}$ será de las dimensiones $s_{j+1} \times (s_j + 1)$.

El +1 viene de la adición en $\theta^{(j)}$ de los "nodos de oscilación", x_0 y $\theta_0^{(j)}$. En otras palabras, los nodos de salida no incluyen a los nodos de oscilación mientras lo de entrada sí.

Ejemplo: la capa 1 tiene 2 nodos de entrada y una capa 2 tiene 4 nodos de activación. La dimensión de $\theta^{(1)}$ es 4X3 donde $s_j = 2$ y $s_{j+1} = 4$, tal que $s_{j+1} \times (s_j + 1) = 4 \times 3$.

Para implementar la versión vectorizada de las funciones presentadas anteriormente es necesario definir una nueva variable $z_k^{(j)}$ que engloba los parámetros dentro de la función g. En el ejemplo previo si se reemplaza la variable z para todos los parámetros obtendríamos:

$$a_1^{(2)} = g(z_1^{(2)})$$

$$a_2^{(2)} = g(z_2^{(2)})$$

$$a_3^{(2)} = g(z_3^{(2)})$$

En otras palabras, para la capa $j = 2$ y el nodo k , la variable z será:

$$z_k^{(2)} = \theta_{k,0}^{(1)}x_0 + \theta_{k,1}^{(1)}x_1 + \dots + \theta_{k,n}^{(1)}x_n$$

La representación vectorial de x y z^j es:

$$x = \begin{bmatrix} x_0 \\ x_1 \\ \dots \\ x_n \end{bmatrix} \quad z^{(j)} = \begin{bmatrix} z_1^{(j)} \\ z_2^{(j)} \\ \dots \\ z_n^{(j)} \end{bmatrix}$$

Estableciendo $x = a^{(1)}$, es posible reescribir la ecuación de la siguiente forma:

$$z^{(j)} = \theta^{(j-1)}a^{(j-1)}$$

Multiplicar la matriz $\theta^{(j-1)}$ con dimensiones $s_j \times (n+1)$ (donde s_j es el número de nodos de activación) por el vector $a^{(j-1)}$ con tamaño $(n+1)$, resulta en el vector $z^{(j)}$ de tamaño s_j .

Para obtener el vector de los nodos de activación para la capa j como sigue:

$$a^{(j)} = g(z^{(j)})$$

Es posible añadir una unidad de oscilación (igual a 1) a la capa j . Esto será el elemento $a_0^{(j)}$ y será igual a 1.

Para calcular la función de hipótesis final, podemos computar otro vector z :

$$z^{(j+1)} = \theta^{(j)}a^{(j)}$$

La última matriz theta debe tener únicamente un renglón de tal manera que el resultado sea un escalar. Entonces el resultado final es:

$$h_\theta(x) = a^{(j+1)} = g(z^{(j+1)})$$

Nótese que, en este último paso, entre la capa j y la capa $j+1$, se realiza el mismo paso que en la regresión logística.

Añadir más capas ocultas y elementos a estas capas permiten realizar modelos predictivos con hipótesis complejas no lineales.

5.4.1.2.5.5 Función de costo

Para definir la función de costos de una red neuronal es necesario definir algunas variables:

- a) L = número total de capas en la red.
- b) s_l = número de unidades (no contando la unidad de oscilación) en la capa l .
- c) k = número de unidades de salida / clases.

Recordando que en las redes neuronales es posible tener múltiples nodos de salida, es posible denotar a $h_{\theta}(\theta)_k$ como la función de hipótesis que resulta en la k^{th} salida.

La función de costos para las redes neuronales es una generalización de la utilizada para la regresión logística.

Para la regresión logística tenemos que:

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^m \left[y^{(i)} \ln(x^{(i)}) + (1 - y^{(i)}) \ln(1 - h_{\theta}(x^{(i)})) \right] + \frac{\lambda}{2m} \sum_{j=1}^n \theta_j^2$$

Para las redes neuronales, es ligeramente más complicado:

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^m \sum_{k=1}^K \left[y_k^{(i)} \ln \left(\left(h_{\theta}(x^{(i)}) \right)_k \right) + (1 - y_k^{(i)}) \ln \left(1 - \left(h_{\theta}(x^{(i)}) \right)_k \right) \right] + \frac{\lambda}{2m} \sum_{l=1}^{L-1} \sum_{i=1}^{s_l} \sum_{j=1}^{s_{l+1}} \left(\theta_{j,i}^{(l)} \right)^2$$

5.4.1.2.5.6 Algoritmo Backpropagation

Backpropagation es el término que surgió junto con las redes neuronales de segunda generación y consiste en el método de minimización de la función de costo. De forma similar al algoritmo de descenso por gradiente la meta es alcanzar:

$$\min_{\theta} J(\theta)$$

Esto es, se desea minimizar la función J usando un conjunto óptimo de parámetros theta.

La derivada parcial de J :

$$\frac{\partial}{\partial \theta_{i,j}^{(l)}} J(\theta)$$

En este algoritmo de entrenamiento (minimización de la función de costos) calculamos cada nodo:

$\delta_j^{(l)}$ = "error" del nodo j en la capa l

Recordando que $a_j^{(l)}$ es el nodo de activación j en la capa l .

Para la última capa, podemos calcular el vector de valores delta con:

$$\delta^{(L)} = a^{(L)} - y$$

Donde L es el número total de capas y $a^{(L)}$ es el vector de salida de las unidades de activación para la última capa. Por lo tanto, los "valores de error" para la última capa son las diferencias de los resultados actuales en la última capa y la salida correcta en y .

Para obtener los valores delta de las capas antes de la última capa, se utiliza la ecuación:

$$\delta^{(l)} = \left((\theta^{(l)})^T \delta^{(l+1)} \right) .* g'(z^{(l)})$$

Los valores delta de la capa l son calculados por la multiplicación de los valores delta en la siguiente capa con la matriz theta de la capa l . Entonces se multiplica el elemento por la función g' , la cual es la derivada de la función de activación g evaluada con la entrada de valores dada por $z^{(l)}$.

La función g prima puede ser escrita:

$$g'(z^{(l)}) = a^{(l)} .* (1 - a^{(l)})$$

Esto puede ser demostrado de la siguiente forma:

$$g(z) = \frac{1}{1 + e^{-z}}$$

$$\frac{\partial g(z)}{\partial z} = - \left(\frac{1}{1 + e^{-z}} \right)^2 \frac{\partial}{\partial z} (1 + e^{-z})$$

Entonces la ecuación completa de los nodos internos es:

$$\delta^{(l)} = \left((\theta^{(l)})^T \delta^{(l+1)} \right) .* a^{(l)} .* (1 - a^{(l)})$$

Es posible calcular los términos de la derivada parcial multiplicando los valores de activación y los valores de error de cada ejemplo de entrenamiento t :

$$\frac{\partial J(\theta)}{\partial \theta_{i,j}^{(l)}} = \frac{1}{m} \sum_{t=1}^m a_j^{(t)(l)} \delta_i^{(t)(l+1)}$$

Con lo anterior es posible describir el algoritmo backpropagation.

Dado el conjunto de entrenamiento $\{(x^{(1)}, y^{(1)}) \dots (x^{(m)}, y^{(m)})\}$

- Establecer $\Delta_{i,j}^{(l)} := 0$ para todo (l, i, j)

Para el ejemplo de entrenamiento $t = 1$ a m :

- Establecer $a^{(1)} := x^{(t)}$
- Utilizar la propagación hacia adelante para calcular $a^{(l)}$ para $l = 2, 3, \dots, L$
- Utilizar $y^{(t)}$, para calcular $\delta^{(L)} = a^{(L)} - y^{(t)}$
- Calcular $\delta^{(L-1)}, \delta^{(L-2)}, \dots, \delta^{(2)}$ usando $\delta^{(l)} = \left((\theta^{(l)})^T \delta^{(l+1)} \right) .* a^{(l)} .* (1 - a^{(l)})$
- $\Delta_{i,j}^{(l)} := \Delta_{i,j}^{(l)} + a_j^{(l)} \delta_i^{(l+1)}$ o con la vectorización $\Delta^{(l)} := \Delta^{(l)} + \delta^{(l+1)} (a^{(l)})^T$
- $D_{i,j}^{(l)} := \frac{1}{m} \left(\Delta_{i,j}^{(l)} + \lambda \theta_{i,j}^{(l)} \right)$ si $j \neq 0$
- $D_{i,j}^{(l)} := \frac{1}{m} \Delta_{i,j}^{(l)}$ si $j = 1$

Matriz delta mayúscula es utilizada como un “acumulador” para agregar valores a medida que se avanza y eventualmente calcular la derivada parcial.

Los términos de la derivada parcial que se buscan:

$$D_{i,j}^{(l)} = \frac{\partial J(\theta)}{\partial \theta_{i,j}^{(l)}}$$

5.4.1.3 Aprendizaje no supervisado

Las técnicas de aprendizaje no supervisado son utilizadas en conjuntos de datos donde no se conoce explícitamente la salida de cada ejemplo. Siendo utilizado para realizar tareas de agrupamiento de datos con el fin de definir clases en los datos o para reducir dimensionalmente a los mismos con el objetivo de eliminar dependencias lineales o no lineales.

5.4.1.3.1 Análisis de Componentes Principales (PCA)

Dadas dos características, x_1 y x_2 , se desea encontrar una línea que describa ambas características a la vez. Por lo tanto, trazamos estas características dentro de esta línea de tal forma que obtengamos una nueva característica. [65]

Lo anterior es posible para tres características, proyectándolas en un plano.

La meta del PCA es reducir el promedio de todas las distancias de cada característica con respecto a la línea de proyección (error de proyección).

Reducir de dos dimensiones a una dimensión es necesario encontrar una dirección (un vector $u^{(1)} \in R^n$) sobre los cuales al proyectar los datos se minimice el error de proyección.

El caso general es:

Reducir de n -dimensiones a k -dimensiones: encontrar los k vectores $u^{(1)}, u^{(2)}, \dots, u^{(k)}$ sobre los cuales al proyectar los datos se minimice el error de proyección.

Antes de aplicar PCA a un conjunto de datos se requiere un paso previo de preprocesamiento:

- Dado un conjunto de entrenamiento: $x^{(1)}, x^{(2)}, \dots, x^{(m)}$.
- Se preprocesa (escalando las características o normalizándolas):

$$\mu_j = \frac{1}{m} \sum_{i=1}^m x_j^{(i)}$$

- Se reemplaza cada $x_j^{(i)}$ con $x_j^{(i)} - \mu_j$
- Si diferentes características se encuentran en diferentes escalas (por ejemplo, x_1 =tamaño de una casa, x_2 =número de cuartos), es necesario escalar las características con la finalidad de tener un intervalo de valores comparable.

Es posible definir específicamente que el promedio para reducir un conjunto de datos de dos dimensiones a una dimensión de la siguiente forma:

$$\Sigma = \frac{1}{m} \sum_{i=1}^m (x^{(i)})(x^{(i)})^T$$

Los valores de z son todos números reales y son las proyecciones de las características dentro de $u^{(1)}$.

Por consiguiente, el PCA tiene dos tareas: descifrar $u^{(1)}, \dots, u^{(k)}$ y también encontrar z_1, z_2, \dots, z_m .

5.4.2 Aprendizaje profundo (Redes Neuronales Convolucionales)

Las redes neuronales convolucionales son una variedad de redes neuronales de segunda generación utilizadas en diversos estudios para la clasificación, detección y localización de objetos en imágenes. Este tipo de arquitectura fue propuesta por primera vez en 1998 por LeCunn et al para la clasificación de imágenes con dígitos manuscritos. [66]

En el área médica también es conocida la utilización de este tipo de modelos en diversas aplicaciones para la detección de diferentes padecimientos como cáncer de pulmón y anomalías en el pecho. [67][68]

5.4.2.1 Historia

Entre 1950 y 1960 Hubel y Wiesel demostraron que la corteza cerebral de los gatos y de los monos respondían a pequeñas regiones de neuronas que además se sobreponían a otras regiones formando una profunda e intrincada estructura a la que nombraron campo receptor visual. Aunado a este descubrimiento propusieron modelos para la detección de patrones con futuras aplicaciones en visión artificial. [69][70][71][72][73][74][76]

Inspirado por los descubrimientos anteriores, Kunihiko Fukushima propuso en 1980 la implementación de un nuevo paradigma conocido en inglés como *neocognition*. De esta forma se estableció la primera propuesta de red convolucional con “capas convolucionales” o filtros y “unidades de submuestreo”. Las primeras capas realizan operaciones sobre porciones de la imagen con la finalidad de transformarla en una nueva, lo que se conoce actualmente como “convolución”. Por su parte las “unidades de submuestreo” realizaban el promedio de un área específica de las salidas de una capa con la finalidad de reducirla dimensionalmente (*Average pooling*). [77] Posteriormente Weng et al. propusieron diferentes “unidades de submuestreo”, las cuales se encargarían de hallar el valor máximo del arreglo (*Max-pooling*) en lugar del promedio. [78]

Fue hasta 1998 que LeCunn et al propusieron la primera Red Neuronal Convolutiva moderna que aprendía utilizando el paradigma de regla delta o backpropagation consiguiendo identificar imágenes con números manuscritos. Esta arquitectura posteriormente fue conocida como LeNet5. [79]

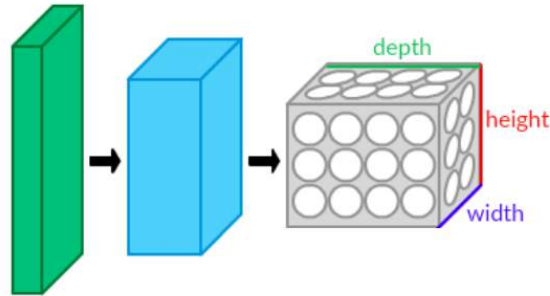


Figura 13 Descripción de un volumen de entrada y/o resultante de una convolución

5.4.2.2 Bloques

La arquitectura de una Red Neuronal Convolucional (RNC) está formada por un apilamiento de distintas capas que transforman el volumen de entrada en un volumen de salida a través de una función diferenciable. Existen pocos tipos de capas comúnmente utilizadas. Las cuáles describiremos a continuación. [80]

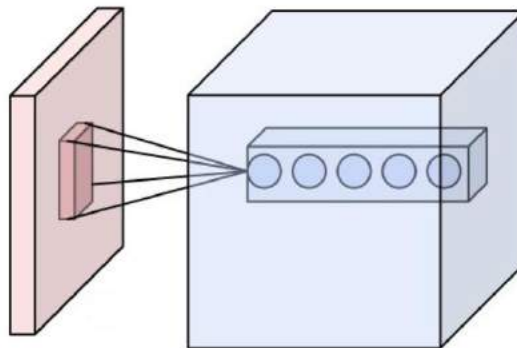


Figura 14 Volumen de salida después de aplicar un filtro en una arquitectura convolucional

5.4.2.3 Conectividad local

Cuando se lidia con entradas con alta dimensionalidad, como las imágenes digitales, es poco práctico conectar las neuronas con todas las neuronas del volumen previo porque tal arquitectura de red no toma en cuenta la estructura espacial de los datos. Las redes convolucionales aprovechan la correlación espacial local para realizar un escaso patrón de conectividad local entre neuronas de capas adyacente: cada neurona conectada está conectada a solamente una pequeña región del volumen de entrada. [79]

El alcance de esta conectividad es un hiperparámetro llamado campo receptivo de la neurona. Las conexiones son locales en el espacio (a lo alto y ancho), pero siempre a lo largo de la profundidad

del volumen entero de entrada. De tal forma que la arquitectura se asegura de que los filtros aprendidos produzcan la respuesta más fuerte a un patrón de entrada espacialmente local. [79]

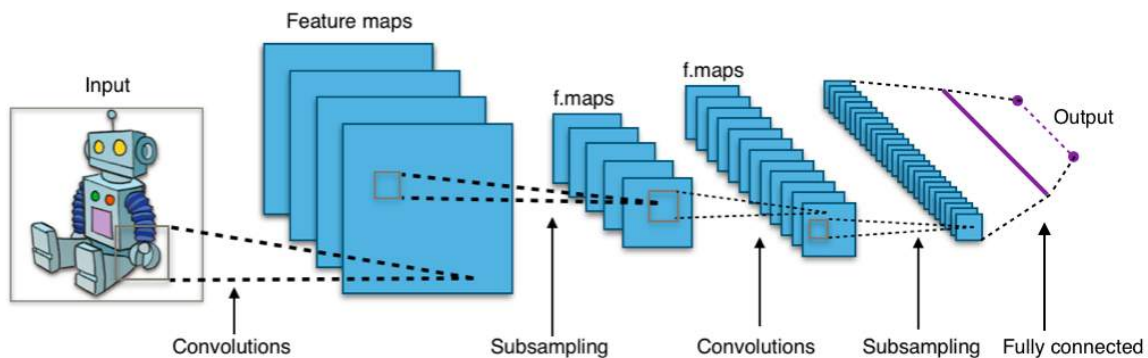


Figura 15 Arquitectura LeNet5

5.4.2.4 Arreglo espacial

Tres hiperparámetros controlan el tamaño del volumen de salida de la capa convolucional: la profundidad, el paso y el “cero relleno” (*zero padding*).

La profundidad del volumen de salida controla el número de neuronas en la capa que se conecta a la misma región del volumen de entrada. Estas neuronas aprenden a activarse para diferentes características en la entrada. Por ejemplo, si la primera capa convolucional toma la imagen en bruto como entrada, entonces diferentes neuronas a lo largo de la dimensión de la profundidad pueden activarse en presencia de varios bordes orientados o manchas de color. [79]

El paso (*stride*) controla cómo se asignan las columnas de profundidad alrededor de las dimensiones espaciales (ancho y alto). Cuando el *stride* es 1, entonces movemos los filtros un pixel a la vez, esto conduce a un alto solapamiento entre los campos receptivos y las columnas, y también a lo largo de los volúmenes de salida. Cuando el *stride* es 2 entonces los filtros saltan dos pixeles a la vez. De forma similar, para cualquier entero $S > 0$, causa que el filtro se traslade S unidades a la vez por salida. [79]

Algunas veces es conveniente rellenar la entrada con ceros en los bordes del volumen de entrada. El tamaño de este relleno es un tercer hiperparámetro llamado *Padding* (relleno) que proporciona control del tamaño espacial del volumen de salida. En particular, a veces es deseable preservar exactamente el tamaño espacial del volumen de entrada. [79]

El tamaño espacial de volumen puede ser calculado como una función de la salida del volumen de entrada W , el tamaño del campo del kernel de las neuronas de la capa convolucional K , el stride S , y la cantidad de zero padding P utilizados en el borde. La fórmula para calcular cuantas neuronas “se ajustan” en un volumen dado es:

$$\frac{W - K + 2P}{S} + 1$$

Si este número no es un entero, entonces los “*strides*” son incorrectos y las neuronas no se pueden colocar en mosaico para que se ajusten en el volumen de entrada de forma simétrica, en general, establecer un *zero padding* $P = \frac{K-1}{2}$ cuando el *stride* es $S = 1$ asegura que el volumen de entrada y el volumen de salida tendrán el mismo tamaño espacial, sin embargo, no es siempre completamente necesario usar todas las neuronas de la capa previa. [79]

5.4.2.5 Pooling layer

Otro concepto importante de las RCN’s es *pooling* (agrupación), el cual es una forma de submuestreo no lineal. Existen varias funciones no lineales que implementan el *pooling* entre las cuales está el *max pooling*.

Intuitivamente, la localización exacta de una característica es menos importante que su localización aproximada con respecto a otras características. Está es la idea detrás del uso del *pooling* en redes neuronales convolucionales. Las capas *pooling* funcionan para reducir de forma progresiva el tamaño espacial de la representación, para reducir el número de parámetros, la cantidad de memoria utilizada, el costo computacional y, por lo tanto, controlar el sobreajuste. [79]

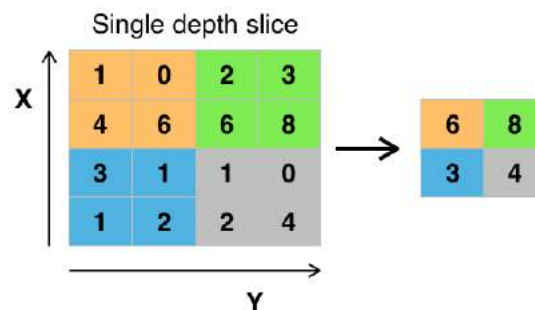


Figura 16 Salida de un filtro Max pooling 2X2

5.4.2.6 Capa ReLU

ReLU significa Unidad de Rectificado Lineal por sus siglas en inglés, definida como $f(x) = \max(0, x)$. Esto Incrementa las propiedades no lineales de la función de decisión y de la red en general. [80]

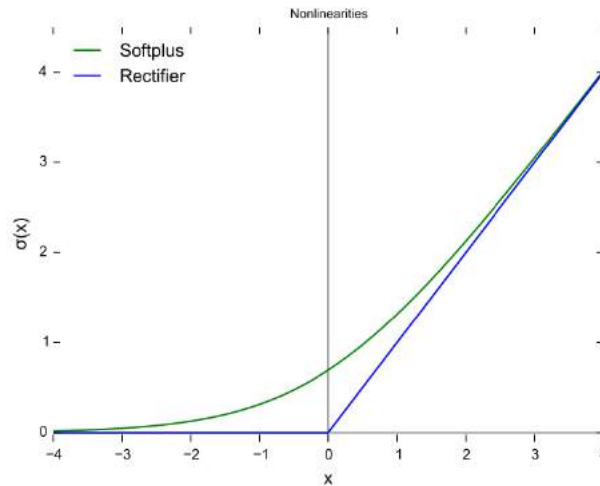


Figura 17 Comparativa entre una función rectificadora y una Softplus

5.4.2.7 Capa completamente conectada

La última capa de la red neuronal convolucional recibe el nombre de *capa completamente conectada*. Las neuronas en la capa completamente conectada tienen conexiones con todas las activaciones de la capa previa como sucede en las capas de las redes neuronales no convolucionales. [79]

5.5 Correlación lineal

La correlación indica la relación lineal y proporcionalidad entre dos variables. Para los fines de este trabajo se abordará el coeficiente de correlación de Pearson y el Coeficiente de correlación de Spearman.

5.5.1 Coeficiente de correlación de Pearson

En estadística, el coeficiente de correlación de Pearson es una medida de la relación lineal entre dos variables aleatorias cuantitativas. A diferencia de la covarianza, la correlación de Pearson es independiente de la escala de medida de las variables.

De manera menos formal, podemos definir el coeficiente de correlación de Pearson como un índice que puede utilizarse para medir el grado de relación de dos variables siempre y cuando ambas sean cuantitativas. [81]

En el caso de que se esté estudiando dos variables aleatorias X y Y sobre una población; el coeficiente de correlación de Pearson se simboliza con la letra $\rho_{x,y}$, siendo la expresión que nos permite calcularlo:

$$\rho_{X,Y} = \frac{\sigma_{XY}}{\sigma_X \sigma_Y} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y}$$

Donde:

- σ_{XY} es la covarianza de (X, Y)
- σ_X es la desviación típica de la variable X
- σ_Y es la desviación típica de la variable Y

El valor del índice de correlación varía en el intervalo $[-1, 1]$, indicando el sentido de la relación:

- Si $\rho_{XY} = 1$, existe una correlación positiva perfecta. El índice indica una dependencia total entre las dos variables denominada *relación directa*: cuando una de ellas aumenta, la otra también lo hace en proporción constante.
- Si $0 < \rho_{XY} < 1$, existe una correlación positiva.
- Si $\rho_{XY} = 0$, ni existe relación lineal. Pero esto no necesariamente implica que las variables son independientes: pueden existir todavía relaciones no lineales entre las variables.
- Si $-1 < \rho_{XY} < 0$, existe una correlación negativa.
- Si $\rho_{XY} = -1$, existe una correlación negativa perfecta. El índice indica una dependencia total entre las dos variables llamada *relación inversa*: cuando una de ellas aumenta, la otra disminuye en proporción constante.

5.5.2 Coeficiente de Correlación de Spearman

En estadística, el coeficiente de correlación de Spearman ρ , es una medida de la correlación (la asociación o interdependencia) entre dos variables aleatorias continuas. Para calcular ρ , los datos son ordenados y reemplazados por su respectivo orden. [82]

El estadístico ρ viene dado por la expresión:

$$\rho = 1 - \frac{6 \sum D^2}{N(N^2 - 1)}$$

Donde D es la diferencia entre los correspondientes estadísticos de orden de $x - y$. N es el número de parejas

Se tiene que considerar la existencia de datos idénticos a la hora de ordenarlos, aunque si éstos son pocos, se puede ignorar tal circunstancia.

La interpretación del coeficiente de Spearman es igual que la del coeficiente de correlación de Pearson.

5.6 Matriz de confusión

En el campo de la inteligencia artificial una matriz de confusión es una herramienta que permite la visualización del desempeño de un algoritmo que se emplea en aprendizaje supervisado. Cada columna de la matriz representa el número de predicciones de cada clase, mientras que cada fila representa a las instancias en la clase real. Uno de los beneficios de las matrices de confusión es que facilitan ver si el sistema está confundiendo dos clases. [83]

Si en los datos de entrada el número de muestras de clases diferentes cambia mucho la tasa de error del clasificador no es representativa de lo bien que realiza la tarea el clasificador. Si por ejemplo hay 990 muestras de la clase 1 y sólo 10 de la clase 2, el clasificador puede tener fácilmente un sesgo hacia la clase 1. Si el clasificador clasifica todas las muestras como clase 1 su precisión será del 99%. Esto no significa que sea un buen clasificador, pues tuvo un 100% de error en la clasificación de las muestras de la clase 2.

En la matriz de ejemplo que aparece a continuación, de 8 gatos reales, el sistema predijo que tres eran perros y de seis perros predijo que uno era un conejo y dos eran gatos. A partir de la matriz se puede ver que el sistema tiene problemas distinguiendo entre gatos y perros, pero que puede distinguir razonablemente bien entre conejos y otros animales.

Tabla 3. Ejemplo de Matriz de Confusión.

		Valor Predicho		
		Gato	Perro	Conejo
Valor Real	Gato	5	3	0
	Perro	2	3	1
	Conejo	0	2	11

5.6.1 Derivaciones de la matriz de confusión (índices de validación)

Tabla 4. Derivaciones de la matriz de confusión

Nombre	Definición	Ecuación
Clase positiva	El número de casos reales positivos en los datos.	P
Clase negativa	El número de casos reales negativos en los datos.	N
Verdaderos positivos	Número de elementos de la clase positiva clasificados con éxito.	TP
Verdaderos negativos	Número de elementos de la clase negativa clasificados con éxito.	TN

Falsos positivos o error Tipo I	Número de elementos de la clase negativa clasificados incorrectamente.	FP
Falsos negativos o error Tipo II	Número de elementos de la clase positiva clasificados incorrectamente.	FN
Sensibilidad (Recall) o Tasa de Verdaderos Positivos	Medida de la proporción de elementos positivos que son identificados correctamente como tales.	$TPR = \frac{TP}{TP + FN}$
Especificidad o Tasa de Verdaderos Negativos	Medida de la proporción de elementos negativos que son identificados correctamente como tales.	$TNR = \frac{TN}{TN + FP}$
Precisión o Valor Predictivo Positivo	Proporción de elementos positivos clasificados correctamente con respecto a todos los elementos que se clasifican como positivos.	$PPV = \frac{TP}{TP + FP}$
Valor Predictivo Negativo	Proporción de elementos negativos clasificados correctamente con respecto a todos los elementos que se clasifican como negativos.	$NPV = \frac{TN}{TN + FN}$
Tasa de Falsos Negativos	Proporción de elementos positivos clasificados incorrectamente con respecto a todos los elementos positivos.	$FNR = \frac{FN}{FN + TP}$
Tasa de Falsos Positivos	Proporción de elementos negativos clasificados incorrectamente con respecto a todos los elementos negativos.	$FPR = \frac{FP}{FP + TN}$
Tasa de Descubrimiento Falso	Medida de la proporción de elementos negativos clasificados incorrectamente con respecto a todos los elementos clasificados como positivos.	$FDR = \frac{FP}{FP + TP}$
Tasa de Omisión Falsa	Medida de la proporción de elementos positivos clasificados incorrectamente con respecto a todos los elementos clasificados como negativos.	$FOR = \frac{FN}{FN + TN}$

Exactitud	Describe la proporción de cantidad de casos correctamente clasificados.	$ACC = \frac{TP + TN}{P + N}$
F1 score	Es la media armónica de la precisión y la sensibilidad.	$F_1 = 2 \cdot \frac{PPV \cdot TPR}{PPV + TPR}$

5.6.2 Curva ROC

Una curva ROC (acrónimo de Receiver Operating Characteristic, o Característica Operativa del Receptor) es una representación gráfica de la sensibilidad frente a la especificidad para un sistema clasificador binario según se varía el umbral de discriminación. Otra interpretación de este gráfico es la representación de la razón o ratio de verdaderos positivos (VPR = Razón de Verdaderos Positivos) frente a la razón o ratio de falsos positivos (FPR = Razón de Falsos Positivos) también según se varía el umbral de discriminación (valor a partir del cual decidimos que un caso es un positivo). [84]

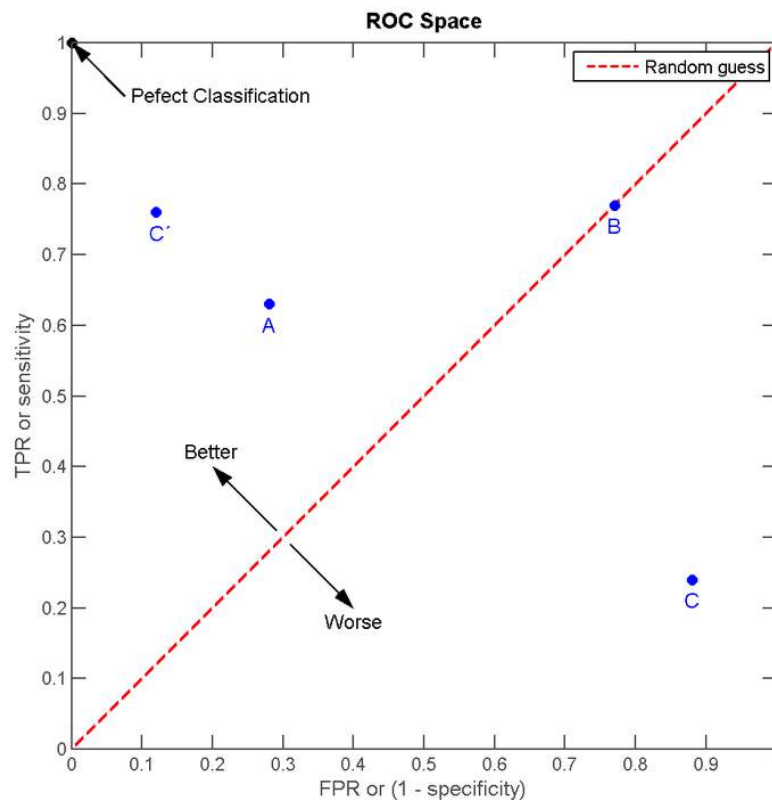


Figura 18. Ejemplo de un espacio ROC, la línea roja representa al azar (caso B).

5.7 Termografía infrarroja

La termografía infrarroja hace referencia a una serie de técnicas las cuales tienen como propósito medir de manera no invasiva la temperatura de un objeto. Esto a partir del principio físico de relación proporcional entre la luz infrarroja emitida y la temperatura superficial del objeto. [85]

La medición es llevada a cabo mediante sensores de luz infrarroja. Un ejemplo de cámaras que realizan mediciones infrarrojas se muestra en la figura 22 (FLIR T650sc). La cámara cuenta con un sensor que en sí mismo es un arreglo matricial de diferentes transductores capaces de obtener mediciones puntuales. Para la comprensión e interpretación de estas mediciones los desarrolladores de esta tecnología utilizan una paleta de colores que están directamente relacionados con la temperatura superficial; a esto se le conoce como pseudocolor. En la figura 19 se expone un ejemplo concreto de pseudocolor, mientras que la ilustración 20 muestra la intensidad lumínica captada por el sensor; representada esta última por una imagen en escala de grises.

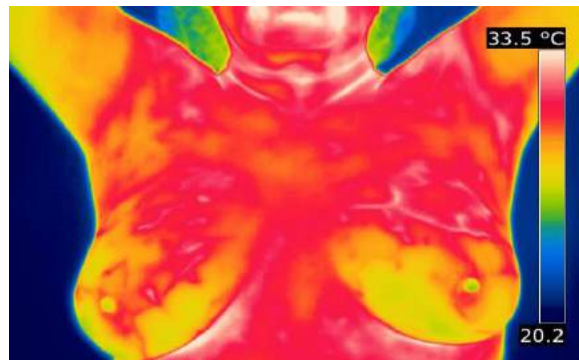


Figura 19. Termograma en pseudocolor.

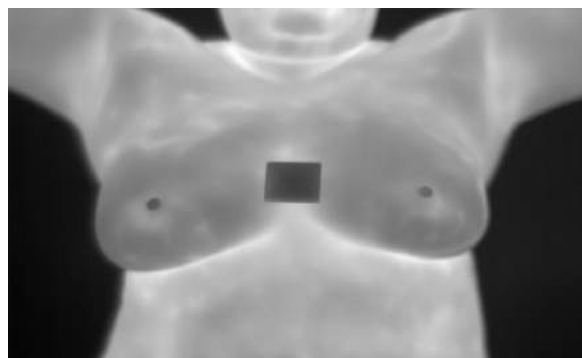


Figura 20. Termograma en escala de grises.

La radiación infrarroja se encuentra en el intervalo de longitud de onda (1mm - 750nm) aproximadamente. Todo cuerpo emite este tipo de radiación por lo cual la termografía es utilizada en diferentes áreas tales como medicina, industria pesada, deportes, agricultura, aplicaciones

militares, astronomía, electrónica, entre otras. El espectro infrarrojo para fines prácticos se puede dividir en diferentes regiones del espectro electromagnético en función de su longitud de onda, las cuales son: onda corta ($1.1 - 2.5\mu\text{m}$), onda media ($2.5 - 7.5\mu\text{m}$), onda larga ($7 - 15\mu\text{m}$) y onda muy larga ($> 15\mu\text{m}$). La figura 21 muestra el intervalo en el que se encuentra la radiación infrarroja con respecto a otras longitudes de onda de la luz. [86]

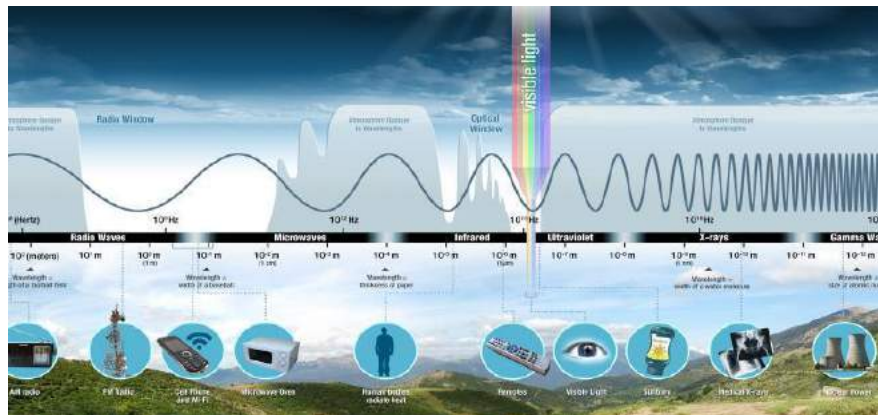


Figura 21. Espectro de la luz.

5.7.1 Aplicaciones del infrarrojo en la medicina

Debido a que la termografía infrarroja detecta temperatura en la superficie de cualquier objeto, en medicina se ha investigado ampliamente su aplicación en diferentes áreas. Lo anterior se debe a que la actividad fisiológica se encuentra directamente relacionada con la variación de temperatura de la región a estudiar; esto se traduce en emisión de luz infrarroja, fenómeno que es medido a través de cámaras especializadas con un nivel de sensibilidad menor a 0.02°C , como la mostrada en la figura 22. [87]



Figura 22. Cámara infrarroja T650sc.

En medicina existen diferentes metodologías para obtener una medición del nivel de temperatura, los cuales son nombradas a continuación.

Termometría por Contacto Eléctrico; esta técnica consiste en realizar la medición de temperatura a partir de un arreglo de termistores, los cuales modifican el valor de su resistencia eléctrica en función de la temperatura. Sin embargo, algunos investigadores han aportado evidencia que apunta a que su uso conlleva a una alteración del flujo de sangre de la región lo cual modifica la medición. [88]

Umbral de Discriminación de Temperatura Cutánea; esta técnica consiste en la medición de temperatura en función de la medición de una pequeña fibra nerviosa. Se ha utilizado con una variación de sensores de fibras de cristal líquido junto con cámaras infrarrojas, haciendo incidir una luz para medir la temperatura de la región. Su aplicación principal es en estudios de pie diabético. [88]

Termografía Mediante Uso de Sensores de Cristal Líquido; esta aplicación consiste en medir la temperatura de una región mediante el uso de sensores de cristal líquido los cuales cambian de color dependiendo la temperatura lo cual forma diferentes patrones de color. Su aplicación se ve limitada debido a su limitada sensibilidad térmica (0.3 – 1°C). [88]

Termograma Infrarrojo (en la literatura también se puede encontrar como IRT); es la más precisa de las metodologías empleadas para la medición con una alta precisión de la temperatura corporal, esto debido a que la tecnología de cámaras infrarrojas ha mejorado de manera significativa en los últimos años y no altera la medición al no colocar sensores superficiales en el objeto de estudio. El principio de medición fue descrito anteriormente. [88]

Algunos ejemplos de aplicación de termografía infrarroja en medicina son:

- Oncología.
- Dolencias.
- Desordenes vasculares.
- Artritis/Reumatismo.
- Neurología.
- Cirugía.
- Oftalmología.
- Viabilidad tisular.
- Desordenes dermatológicos.
- Monitoreo de la eficacia de medicamentos y tratamientos.
- Tiroides.
- Odontología.
- Estudio respiratorio.
- Medicina deportiva y rehabilitación.

5.7.2 Ventajas de la termografía

La termografía es capaz de detectar lesiones provocadas por cambios fisiológicos, del metabolismo o el flujo de sangre antes que otras técnicas logren hacerlo. [89] En particular, para el cáncer de seno, la mastografía y la termografía no son herramientas de detección *per se* de lesiones cancerígenas, sin embargo, son capaces de detectar cambios en la estructura anatómica y comportamiento fisiológico de los tejidos respectivamente; lo cual es posible relacionar con una probabilidad de lesión cancerígena. [90] [91] Estudios anteriores han demostrado que la termografía tiene una alta probabilidad de encontrar lesiones precursoras de cáncer. [92] Complementariamente, estos estudios han encontrado que la termografía tiene la habilidad de encontrar tumores pequeños que difícilmente pueden ser detectados por palpación. [93]

En 2016, investigadores de la India realizaron un estudio comparativo donde demostraron que la termografía es capaz de detectar lesiones no visibles mediante imagen mastográfica. [94] En el mismo sentido, se publicaron diversos estudios que demuestran la capacidad de la termografía de equiparar sus resultados, en términos estadísticos, a los métodos convencionales de imagenología médica para la detección de lesiones cancerígenas en la glándula mamaria. [95] [96] [97]

Capítulo 6 Desarrollo

En este capítulo se describen las diferentes etapas del prototipo de software, desde la adquisición hasta la clasificación artificial de los termogramas.

La adquisición de los termogramas se realizó mediante el protocolo presentado en el anexo W. En esta adquisición se obtuvieron seis imágenes por voluntaria, tres imágenes obtenidas de forma basal o sin sometimiento a estrés térmico y tres funcionales o dinámicas obtenidas después de ser aplicado un estrés térmico (frontal, oblicua 45 grados izquierda y oblicua 45 grados derecha para basales y funcionales). Para el diseño y construcción del siguiente trabajo se emplearon únicamente imágenes basales frontales.

La cámara utilizada es el modelo FLIR T650sc, la cual ha sido empleada en diversos estudios presentados anteriormente en este trabajo. Además de ser específica para desarrollos de investigación en ciencias biológicas. Las especificaciones técnicas de esta cámara se presentan en la tabla 5.

Tabla 5. Especificaciones de las cámaras T659sc y T630sc del fabricante FLIR.

	T650sc	T630sc
Modelo		
Resumen del Sistema		
Tipo de Detector	Microbolómetro no enfriado	
Intervalo Espectral	7.5 – 13.0 μm	
Resolución	640 × 480	
Detector Pitch 25 μm	17 μm	
NETD	<20 mK	<40 mK

Electrónica / Imagen		
Constante de Tiempo	<8 ms	
Tasa de Actualización	30 Hz	
Margen Dinámico	14-bit	
Medición		
Intervalo de Temperatura Estándar	-40°C a +150°C (-40°F a +302°F) +100°C a +650°C (+212°F a +1202°F)	
Exactitud	±1°C (±1.8°F) o ±1% de lectura para un intervalo de temperatura limitado por la medición de un objeto dentro de +5°C a +120°C (+41°F a +248 °F) y temperatura ambiente de and +10°C a +35°C (+49°F a +95°F). Esto es únicamente valido para el intervalo de temperatura -40°C a +120°C (-40°F a +248°F)	+/- 2 °C +/- 2% (intervalo limitado)

6.1 Construcción de Conjuntos de Entrenamiento y validación

Para el desarrollo de este prototipo se utilizaron imágenes de 212 pacientes de las cuales 172 eran sanas y 40 positivas a algún tipo de carcinoma mamario. Debido a que cada paciente tiene más de una imagen y no es deseable tener imágenes en el conjunto de entrenamiento y validación de un mismo paciente con el fin de evitar sobreestimar las métricas de validación, se decidió conformar aleatoriamente los conjuntos de entrenamiento y validación utilizando el expediente completo.

Los conjuntos resultantes se reportan en la tabla 6.

Tabla 6. Conjuntos de entrenamiento y validación.

	Imágenes sanas	Imágenes enfermas	Imágenes Totales	Pacientes sanas	Pacientes enfermas	Pacientes totales
Conjunto de Entrenamiento	2735 (81.95%)	602 (18.04%)	3337 (100%)	140 (82.84%)	29 (17.15%)	169 (100%)
Conjunto de Validación	1433 (86.61%)	231 (13.88%)	1664 (100%)	32 (74.41%)	11 (25.58%)	43 (100%)

Para mayor detalle del proceso de selección de los elementos de cada conjunto refiérase al anexo A e l.

6.2 Transformación lineal

Debido a que los datos proporcionados para la elaboración de este trabajo consistían en archivos .txt con los valores netos de temperatura, se propuso una transformación lineal con el objetivo de convertir estos valores a imágenes de 8 bits en escala de grises. Cabe mencionar que este tipo de imágenes también se pueden ser obtenidas directamente de la cámara.

Se halló la temperatura máxima y mínima de las imágenes en el conjunto de entrenamiento:

Temperatura máxima 37°C

Temperatura mínima 18.65°C

Con el objetivo de abarcar la mayor cantidad de valores posibles en el conjunto de validación se establece un umbral arbitrario superior e inferior de más 2 grados la temperatura máxima y menos 2 grados la temperatura mínima-

Valor superior de temperatura 39°C

Valor inferior de temperatura 16.65°C

Posteriormente se realiza una regresión lineal utilizando los puntos de entrada (16.65, 0), (39, 255). Esto con el objetivo de hallar los parámetros de regresión de la línea recta m (pendiente) y b (cruce en el eje y).

Resultando:

$$m = 11.409395973154359$$
$$b = -189.96644295302002$$

En la figura 23 se muestra el modelo obtenido de la regresión anterior.

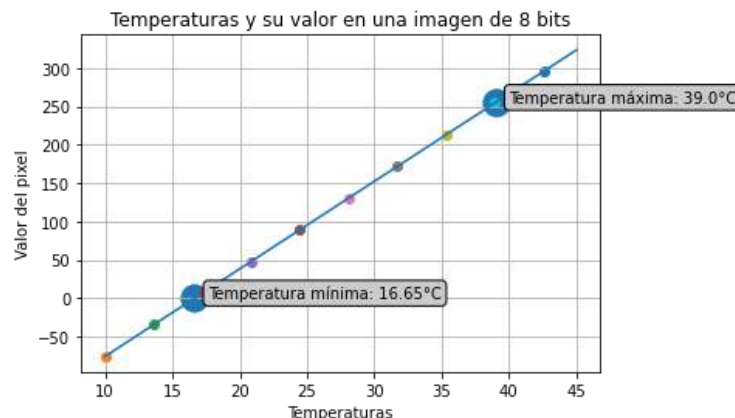


Figura 23. Línea recta aproximada mediante regresión lineal.

El detalle de la obtención de este modelo se describe en el anexo B.

6.3 Modelo por caracterización bilateral

El primer modelo de reconocimiento requiere segmentar, caracterizar, aplicar un PCA y entrenar un reconocedor artificial proceso simplificado en la figura 24.

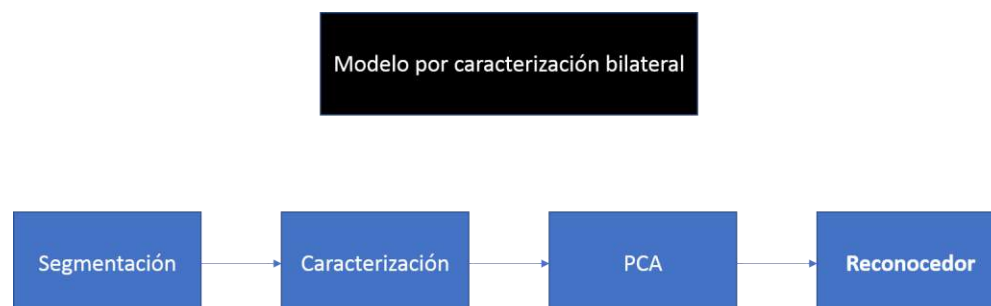


Figura 24. Modelo de caracterización bilateral.

6.3.1 Segmentación automática

Este trabajo propone dos métodos de segmentación. El primero es un método basado en la forma elíptica de los bordes inferiores de las mamas y el segundo es la utilización del plano sagital del cuerpo para realizar la segmentación.

6.3.1.1 Segmentación con base en parábolas

La primera propuesta para este prototipo realiza los pasos descritos en la figura 25.

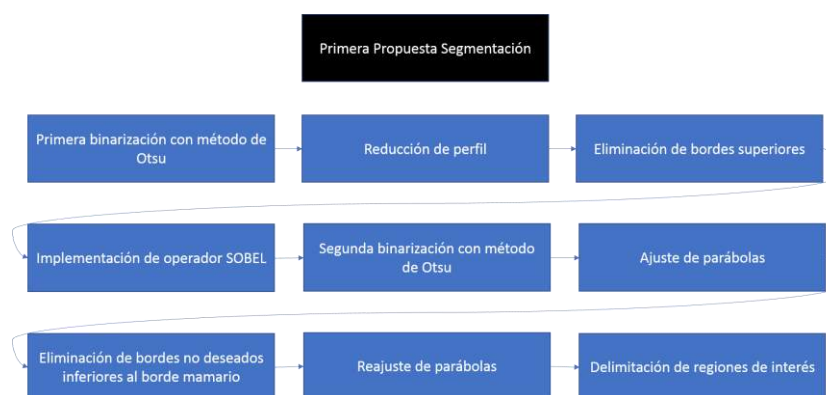


Figura 25. Segmentación por parábolas.

6.3.1.1.1 Primera Binarización con el método de Otsu

Mediante el método de Otsu se halló un umbral en la imagen con el objetivo de separar al objeto de interés con el fondo. Posteriormente se binariza la imagen con ayuda de este umbral.

En la figura 26 se muestran algunos histogramas de termogramas del conjunto de entrenamiento con el fin de ejemplificar la idea anterior. El primer conjunto de montañas representa al fondo de la imagen, mientras que el segundo representa al cuerpo u objeto de interés.

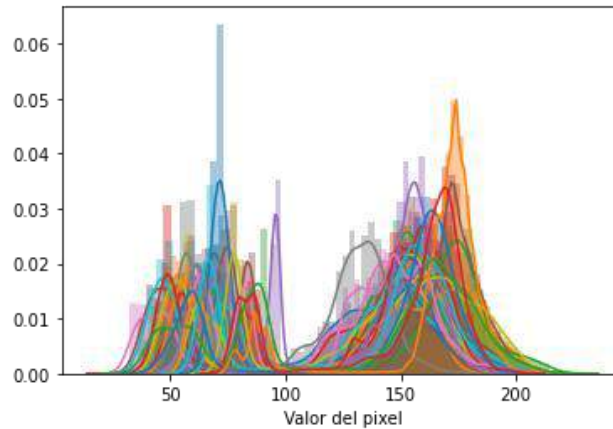


Figura 26. Histogramas de diferentes termogramas aleatorios.

En la imagen 27 se muestra un histograma de ejemplo de un termograma.

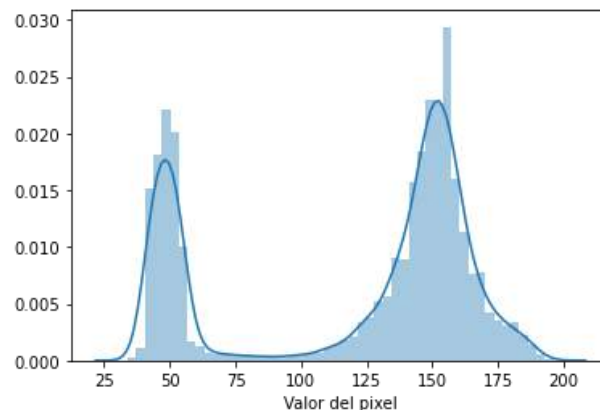


Figura 27. Histograma de un termograma.

En la siguiente imagen se observa el histograma coloreado en azul para representar al fondo y en naranja al objeto. La línea azul, que separa a estas secciones, es el umbral hallado mediante el método de Otsu.

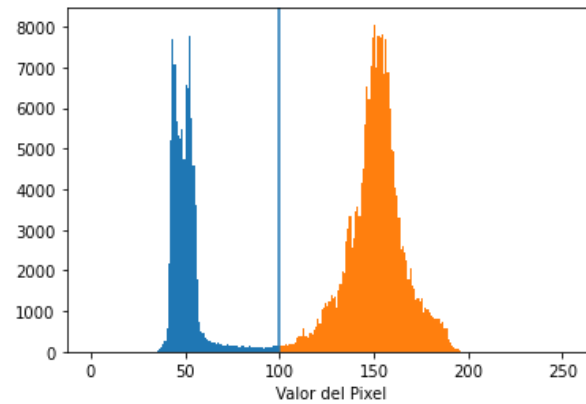


Figura 28. Histograma umbralizado por método de Otsu.

Hallado el umbral se binariza la imagen y posteriormente se envían a cero los valores del fondo como se muestra en la figura 29.

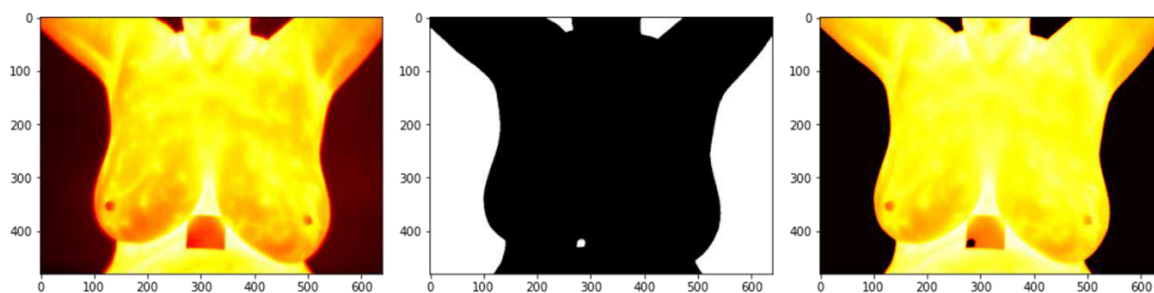


Figura 29. Proceso de binarización para separación de objeto con el fondo.

6.3.1.1.2 Reducción de Perfil

Con el objetivo de eliminar el ruido generado por los bordes exteriores del cuerpo se propone el desgaste de la imagen en un 20% del lado derecho e izquierdo.

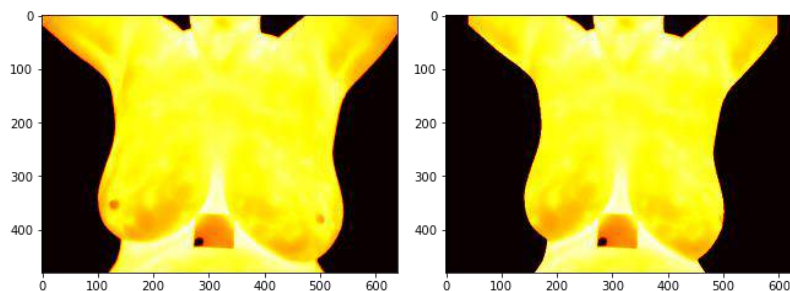


Figura 30. Reducción de perfil.

6.3.1.1.3 Eliminación de Bordes Superiores

Se envían a cero el 30% de la imagen en dirección arriba-abajo para eliminar el potencial ruido de los bordes en esta sección.

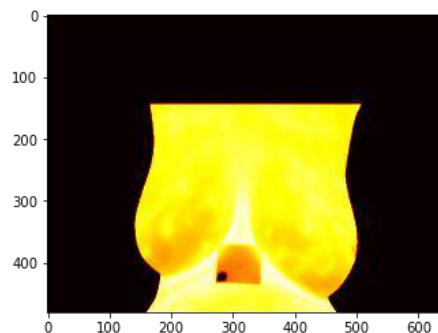


Figura 31. Eliminación de bordes superiores.

6.3.1.1.4 Implementación de operador SOBEL

Sobre la imagen del paso anterior se realiza una operación SOBEL con la intención de asignar un valor de probabilidad de pertenencia a un borde a cada punto en la imagen, operación de la que se obtiene una nueva imagen como la del ejemplo de la figura 32.

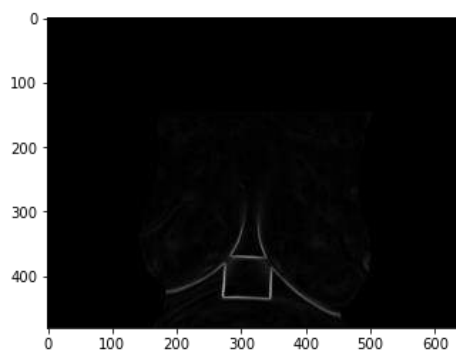


Figura 32. Resultado de operación SOBEL.

6.3.1.1.5 Segunda binarización con método de Otsu

Ahora se realiza una nueva umbralización por el método de Otsu para hallar el umbral que permita separar a los objetos que se espera (con suerte) sean los bordes de interés de la imagen.

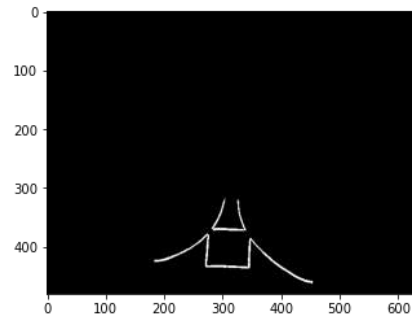


Figura 33. Bordes de segunda binarización por Otsu

6.3.1.1.6 Ajuste de parábolas

Se ajustan dos parábolas utilizando los bordes obtenidos del paso anterior.

6.3.1.1.7 Eliminación de bordes no deseados inferiores al borde mamario

Se eliminan los puntos inferiores a las parábolas ajustadas con el objetivo de suprimir el ruido de los bordes del artefacto cuadrado (ruido) que fue puesto durante la adquisición

6.3.1.1.8 Reajuste de parábolas

Se reajustan nuevamente dos parábolas.

6.3.1.1.9 Delimitación de Regiones de interés.

Se delimitan las regiones de interés con ayuda de la intersección de estas parábolas, los bordes externos y una línea superior que fue definida al momento de enviar a cero el 30% de la imagen en dirección arriba-abajo.

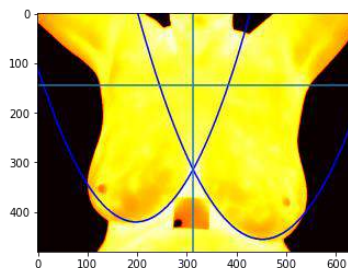


Figura 34. Delimitación de regiones de interés.

La implementación de este método en python se documenta en los anexos C y D.

6.3.1.2 Segmentación con base en plano sagital

La segunda propuesta consiste en binarizar con auxilio del método de umbralización de Otsu, definir el plano sagital que divide al cuerpo en dos secciones con ayuda de la circunferencia de la cintura y, finalmente, delimitación de las regiones de interés como se muestra en la figura 35.

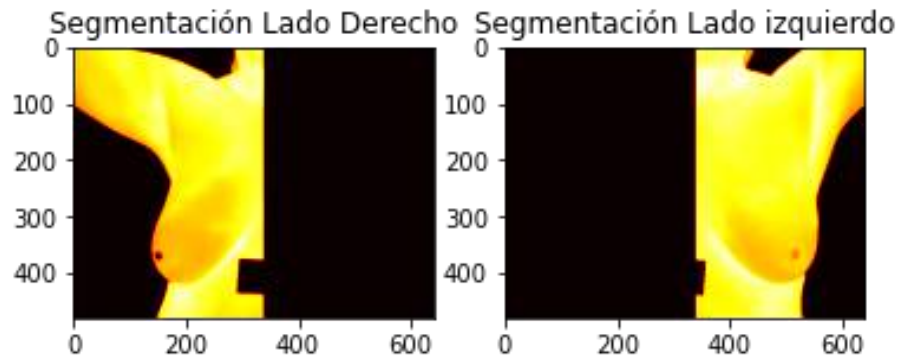


Figura 35. Segmentación con plano sagital.

La implementación al detalle de este método se puede consultar en el anexo F.

6.3.2 Caracterización

Se caracterizó cada imagen obtenida del paso anterior y se realizó la diferencia absoluta por cada descriptor. Diez de estos descriptores obtenidos directamente del histograma y siete de la matriz de co-ocurrencia.

Los descriptores estadísticos se describen en el capítulo 5 de este trabajo. Para mayor información de la forma práctica en que se implementó este proceso refiérase a los anexos G y H.

6.3.3 PCA

Posterior a la caracterización se realizó una disminución dimensional de las características con ayuda de la técnica Análisis de Componente Principales (PCA). Esta reducción se propuso a tres dimensiones por dos razones. La primera es por la facilidad para visualizar estos nuevos puntos en una imagen y, la segunda, fue para disminuir lo mayor posible el costo computacional necesario para entrenar al modelo reconocedor.

Para mayor detalle de la aplicación de este punto refiérase al anexo K.

Imágenes de Entrenamiento

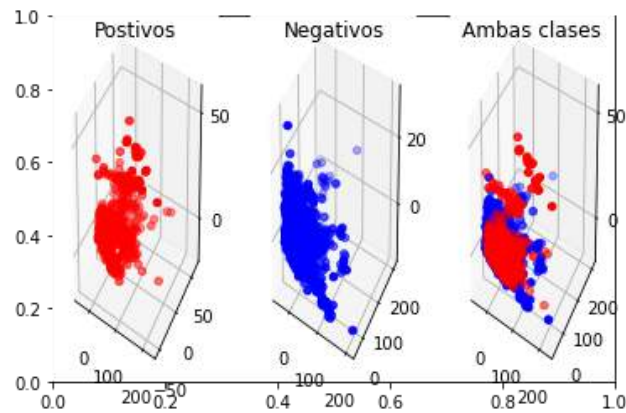


Figura 36. Diagrama de dispersión de imágenes de entrenamiento.

Imágenes de Validación

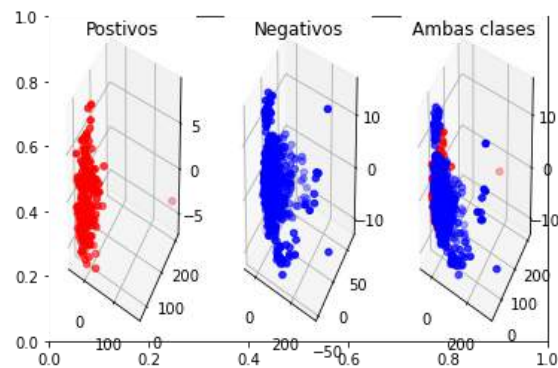


Figura 37. Diagrama de dispersión de imágenes de validación.

6.3.4 Reconocedores

Los reconocedores propuestos son:

Regresión Logística

Red Neuronal Artificial

Estos reconocedores se detallaron en el capítulo 5. Para consultar en detalle la implementación práctica de este punto refiérase al anexo K.

6.4 Modelo a partir de datos clínicos

Se realizó un análisis de los datos clínicos de los individuos involucrados en este estudio para saber cuántos y cuales contenían información suficiente. Para los casos donde no existía información suficiente se propuso una técnica simple de minería de datos para “rellenar” estos con el valor promedio de los demás- Esta técnica es una solución relativamente simple al problema de falta de datos. [98]

Se propusieron los siguientes modelos:

1. Decision Tree
2. Random Forest
3. Máquina de Soporte Vectorial
4. Regresión Logística
5. Red Neuronal

Se evaluó su desempeño utilizando diferentes métricas con el objeto de elegir al mejor, el cual posteriormente se evaluó pero, se utilizó una reducción dimensional con auxilio de PCA antes de ingresar las características al modelo. Los resultados de este punto se detallan en el capítulo 5 y en los anexos L y M.

6.5 Modelo con base en Red Neuronal Convolutacional

El segundo modelo propuesto en este trabajo que requiere como entrada la información proveniente del termograma es una Red Neuronal Convolutacional, modelo el cual se detalló su funcionamiento en el capítulo 5.

Una consideración importante es que las imágenes que los modelos que se describen a continuación recibieron como entrada fueron reducidas intencionalmente a 128x128 pixeles. La razón de esta disminución de la resolución fue técnica relacionada a la infraestructura computacional disponible, ya que estos modelos requieren una inversión importante en tiempo y recursos computacionales para ser entrenados y, mientras la imagen sea menor en resolución, menor será el tiempo para realizar un entrenamiento.

El mejor reconocedor con base en su rendimiento en AUC (área bajo la curva ROC) fue seleccionado como reconocedor para el prototipo final.

6.5.1 Red Neuronal Artificial Convolutacional EfficientNet

En este punto se propone la implementación de la arquitectura EfficientNet.

La arquitectura EfficientNet fue propuesta en 2019 como una alternativa para manipular tres parámetros que empíricamente se han observado influyen en modelos reconocedores de imágenes:

la resolución de la imagen de entrada, la profundidad de la red y el número de canales. Los experimentos con diferentes arquitecturas mencionan que una resolución “alta” junto con un número de canales “grande” mejoran la capacidad del modelo para hallar detalles en la imagen. Sin embargo, esto no es necesariamente independiente de la profundidad de la red neuronal y, hallar esta relación, no es una tarea trivial ni rápida. Ya que es posible alterar la resolución no sólo de la imagen de entrada sino las salidas de cada convolución, así como el número de canales resultante.

La arquitectura EfficientNet es un modelo específico que se utilizó para aplicar el método que resuelve el problema de encontrar “la mejor” correlación de estos tres parámetros con el número de Flops (operaciones de punto flotante). Para esto los autores proponen un parámetro ϕ que indica la potencia a la que se elevan tres constantes (que minimicen el número de FLOPS necesarios) llamadas α (relacionado a la resolución), β (relacionado a los canales) y γ (relacionado a la profundidad). Esto simplifica el proceso de “sintonización” de los tres parámetros y demostró mejores resultados que sus propuestas contemporáneas (estado del arte). Este método no es exclusivo de la arquitectura EfficientNet, es posible aplicar este principio a un sin número de arquitecturas, algunas de ellas fueron documentadas por los autores en la publicación original de 2019. A pesar de ello, también demostraron que la arquitectura EfficientNet superaba a la mayoría de los modelos existentes, con excepción de un “super” modelo que a pesar de contar con una cantidad ocho (8) veces mayor de parámetros tuvo un rendimiento “ligeramente” mayor a la EfficientNet pero con un costo computacional seis (6) veces mayor. [99]

La implementación de esta arquitectura se especifica en el anexo N.

6.5.2 Red Neuronal Artificial Convolutacional ResNet50

Esta arquitectura fue propuesta en 2016. La idea principal que se implementó en este modelo es la modificación de la función de activación de los bloques convolucionales, función que se propuso considerar segmentos de bloques en lugar de sólo uno, de ahí el prefijo que el “res” haga alusión a la palabra-residual o, en otras palabras, se considera a las salidas anteriores o “residuos”. Esto mejora empíricamente el rendimiento en diferentes problemas de reconocimiento de imágenes donde ha sido comparado con diferentes arquitecturas “clásicas”. [100]

El detalle técnico de la implementación de este modelo se presenta en el anexo O.

6.5.3 Red Neuronal Artificial Convolutacional GoogleNet

Esta arquitectura fue propuesta en 2015 y se implementa la idea de “inception”. Con el objetivo de construir redes convolucionales más profundas a las que en aquellos años se habían propuesto. Por tanto, los creadores diseñaron bloques que disminuían y aumentaban la dimensión de la salida de los bloques convolucionales con el propósito de crear un modelo que fuese cada vez más profundo, pero sin inconvenientes por las dimensiones de una imagen y las convoluciones internas. El propósito era mejorar el entrenamiento disminuyendo deliberadamente el tamaño de la salida, ya

que al conseguir lo anterior, el modelo podría entrenarse más “fácilmente” y se podrían realizar modelos más profundos a los propuestos en aquel año, lo cual permitiría y permitió definir modelos más “complejos”. Para la realización de este modelo los autores proponen dos bloques Inception (InceptionV1 e InceptionV2) los cuales realizan convoluciones paralelas para su posterior concatenación. Modificaciones de este modelo fueron propuestos en los últimos años como la InceptionResNet y la InceptionV3. [101][102][103]

Capítulo 7 Resultados

7.1 Resultados de Segmentación

A continuación, se presentan los resultados de las dos propuestas de segmentación.

7.1.1 Segmentación con base en parábolas

Se segmentaron evaluó el algoritmo en todas las imágenes de los conjuntos de entrenamiento y validación y se evaluó su rendimiento de forma visual imagen por imagen obteniendo los resultados de las figuras 38 Y 39.

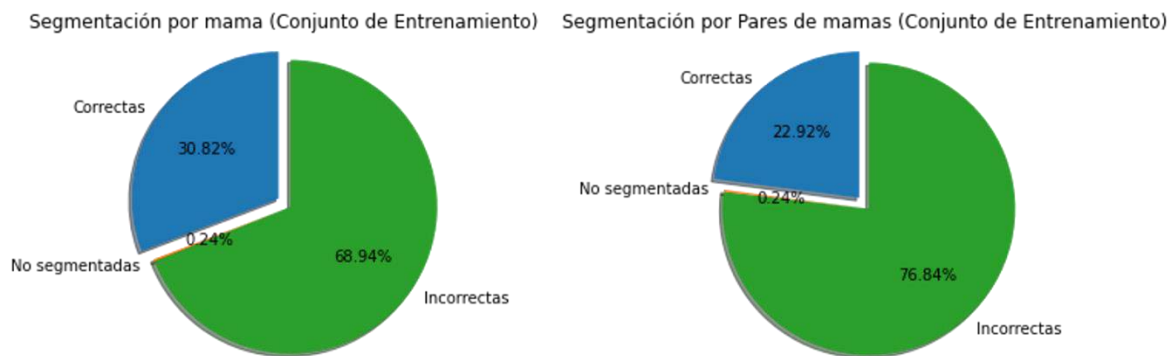


Figura 38. Resultados de segmentación de pares de mama (individuos).

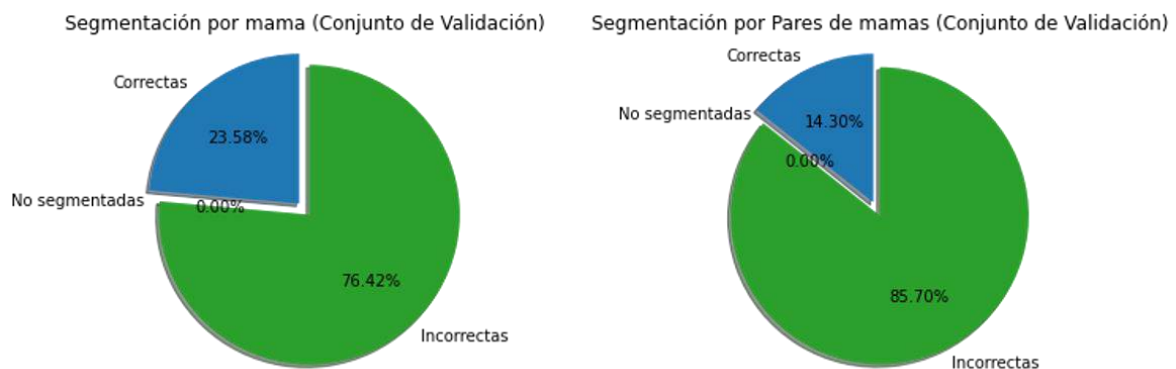


Figura 39. Resultados de segmentación por mama (individuos).

Como se aprecia en las figuras anteriores el método propuesto fue incapaz de segmentar correctamente más del 50% de los pares de mamas y de las mamas individuales. Por estos resultados y para utilizar la mayor cantidad de datos para la experimentación con los modelos reconocedores, no se utilizó este método para el prototipo final.

7.1.2 Segmentación con base en plano sagital

Al igual que el primer método, el segmentador se evaluó visualmente imagen por imagen, obteniendo segmentaciones favorables en el cien por ciento de los casos.

7.2 Resultado de reconocedores con base en caracterización bilateral

Tabla 7. Resultados de reconocedores por asimetría estadística Entrenamiento.

	R. Logística	Neural Net
Accuracy	81.87%	72.70%
Precision	48.45%	33.00%
Recall	7.81%	49.83%
F1-Score	13.45%	39.71%
AUC	64.97%	69.64%

Tabla 8. Resultados de reconocedores por asimetría estadística Validación.

	R. Logística	Neural Net
Accuracy	68.27%	59.50%
Precision	2.86%	25.00%
Recall	0.43%	22.94%
F1-Score	0.75%	23.93%
AUC	34.40%	51.63%

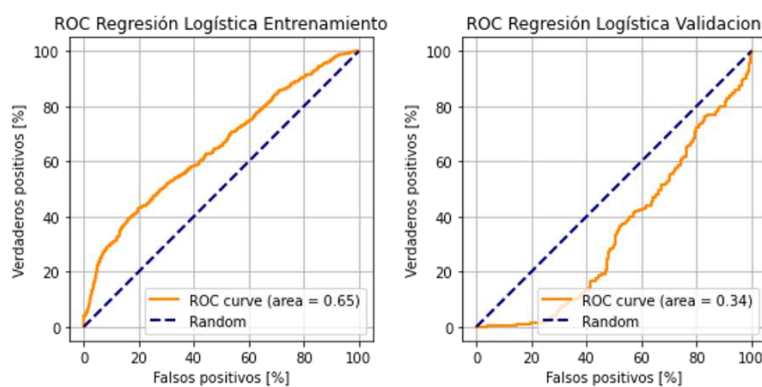


Figura 40. Curvas ROC Regresión Logística.

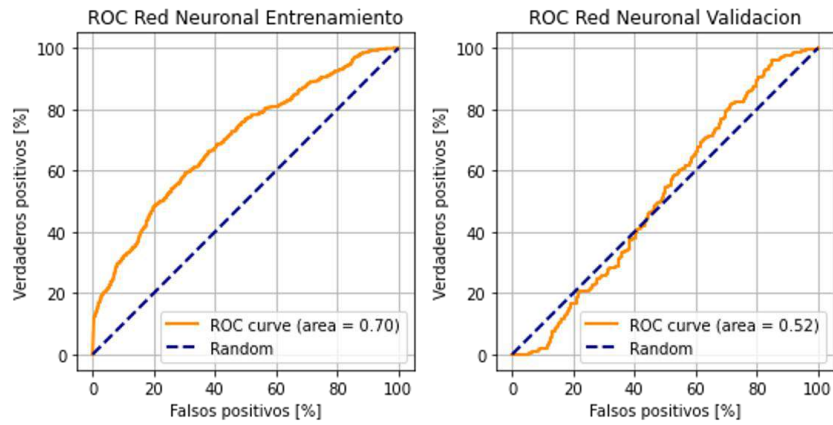


Figura 41. Curvas ROC Red Neuronal.

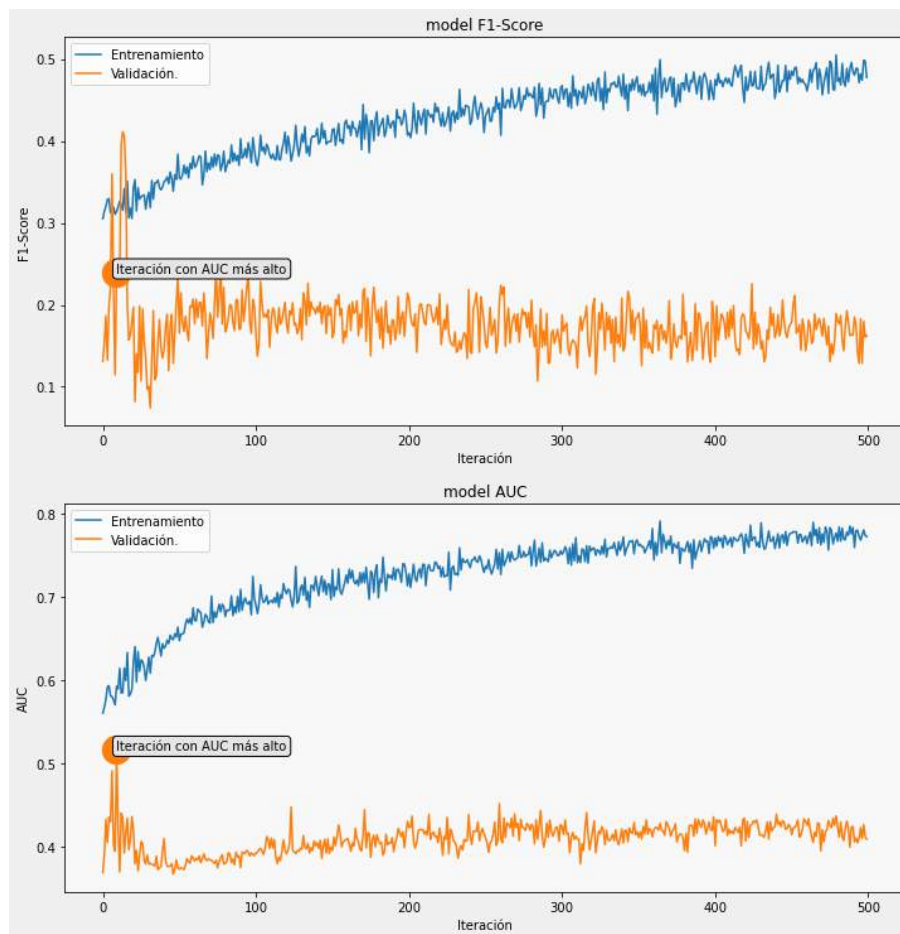


Figura 42. Curvas de entrenamiento de la Red Neuronal.

El modelo con mejor rendimiento fue la red Neuronal, sin embargo, la curva ROC de la validación indica resultados cercanos al azar.

Su implementación en el prototipo final será para fines meramente demostrativos ya que es el modelo implementado en la Parte 1 de este trabajo.

7.3 Análisis, minería de datos y reconocedor a partir de datos clínicos

Se analizaron los datos clínicos del conjunto de entrenamiento y se obtuvo el promedio de cada uno de estos. Para las variables categóricas binarias también se realizó este promedio. Esto con el propósito de utilizar este valor promedio como el valor a utilizar en los pacientes que no tuviesen registrado ningún valor. Considerando al promedio de una variable categórica como la probabilidad de respuesta afirmativa a, por ejemplo, la pregunta ¿fumó el paciente?

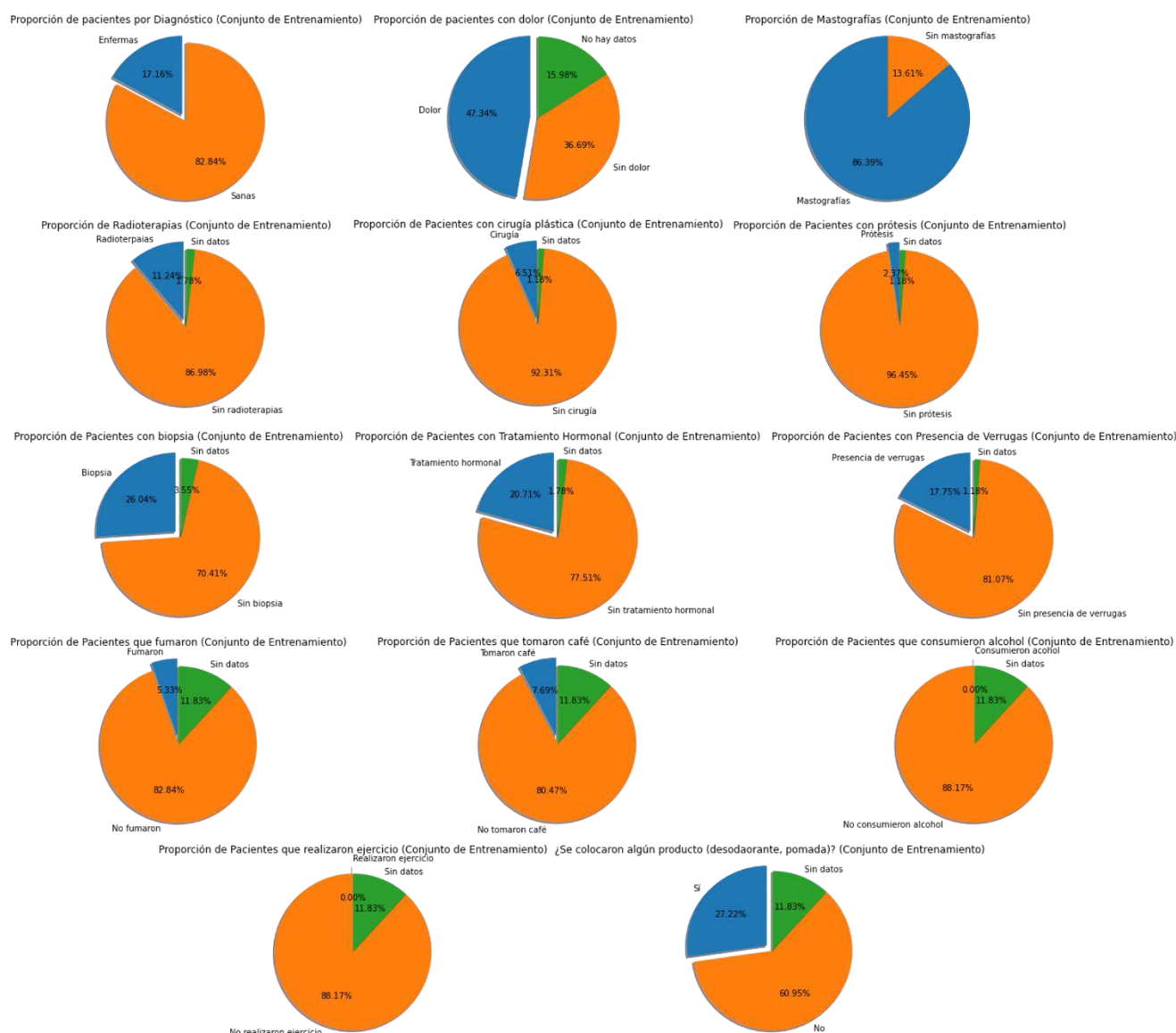


Figura 43. Descripción de datos clínicos.

Para la temperatura corporal medida con termómetro clínico se utilizó el promedio, pero en la mayoría de los casos no fue necesario su empleo. Esta es la púnica variable de la cual no se realizó una gráfica de pastel.

Conjunto de Entrenamiento

Porcentaje de pacientes con datos de temperatura corporal 97.04%

Porcentaje de pacientes sin datos de temperatura corporal 2.96%

7.3.1 Resultados de reconocedores para datos clínicos

Tabla 9. Reconocedores para datos clínicos Entrenamiento

	Decision Tree	Random Forest	SVM	R. Logística	Neural Net
Accuracy	92.90%	95.27%	75.74%	82.84%	68.05%
Precision	79.31%	95.65%	0.00%	0.00%	34.18%
Recall	79.31%	75.86%	0.00%	0.00%	93.10%
F1-Score	79.31%	84.62%	0.00%	0.00%	50.00%
AUC	N/A	N/A	N/A	68.40%	85.91%

Tabla 10. Reconocedores para datos clínicos Validación.

	Decision Tree	Random Forest	SVM	R. Logística	Neural Net
Accuracy	72.09%	76.74%	55.81%	74.42%	67.44%
Precision	45.45%	100.00%	34.62%	0.00%	42.11%
Recall	45.45%	9.09%	81.82%	0.00%	72.73%
F1-Score	45.45%	16.67%	48.65%	0.00%	53.33%
AUC	N/A	N/A	N/A	67.90%	65.34%

N/A: No Aplica

El árbol de decisión, el Random Forest y la SVM colocados en las tablas anteriores fueron elegidos por su mejor desempeño en función de la métrica F1-Score durante la validación.

Para mayor información sobre otros modelos y su rendimiento refiérase al anexo L.

La estructura del mejor árbol de decisión en esta tarea se muestra en la figura 44.

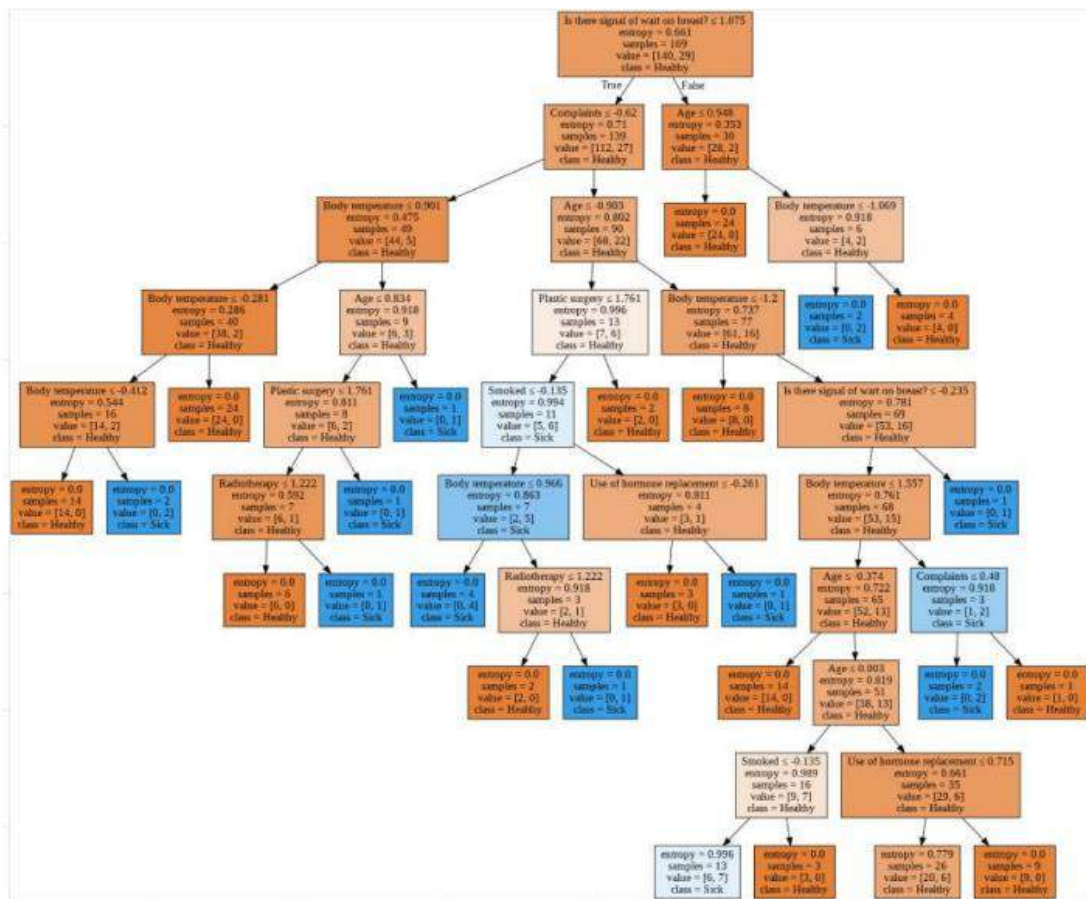


Figura 44. Árbol de decisión

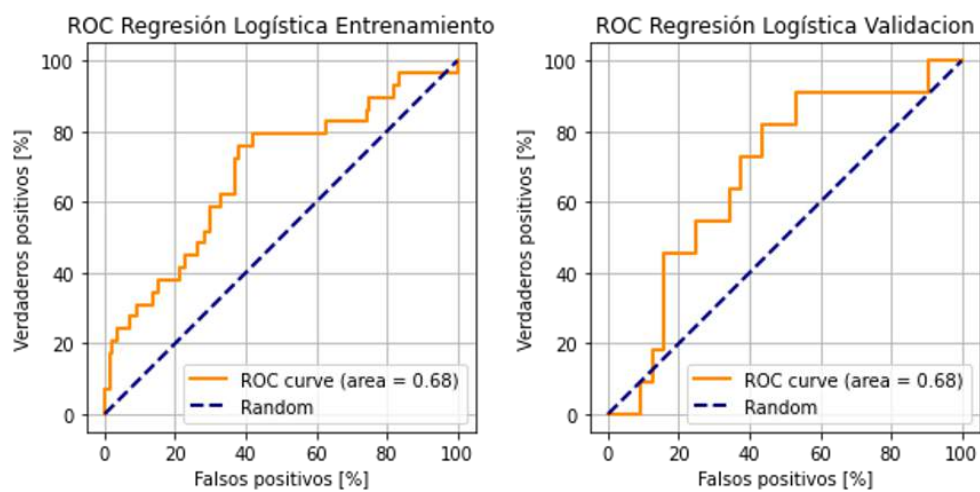


Figura 45. ROC Regresión Logística.

La regresión Logística tuvo un rendimiento ineficiente en función de la AUC registrada, aunque “poco” sobreajustada. Sin embargo, con precisión y sensibilidad cero se puede concluir que fue incapaz de reconocer correctamente positivos. Por estas razones no se utilizó como reconocedor final.

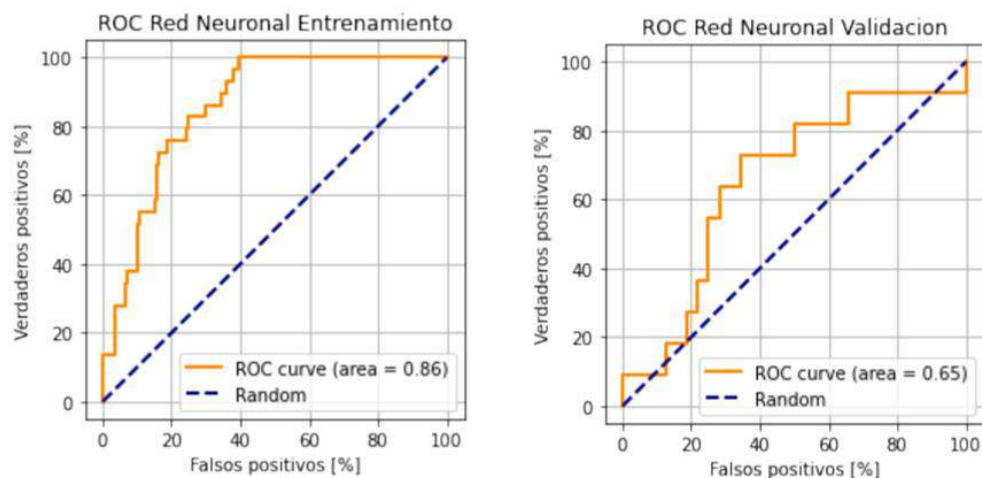


Figura 46. ROC Red Neuronal.

La red neuronal fue el modelo que presentó el mejor rendimiento en las métricas de validación, sin embargo, el valor del AUC en la curva ROC fue muy cercano al azar. Por consiguiente, no se consideró como reconocedor en el prototipo final.

7.4 Redes Neuronales Convolucionales

Tabla 11. Resultados Convolucionales Entrenamiento.

	EfficientNet	ResNet50	GoogleNet
Accuracy	87.71%	76.39%	90.17%
Precision	73.19%	42.88%	65.02%
Recall	50.33%	93.02%	98.50%
F1-Score	59.65%	58.70%	78.34%
AUC	90.83%	83.33%	98.71%

Tabla 12. Resultados Convolucionales Validación

	EfficientNet	ResNet50	GoogleNet
Accuracy	84.62%	67.67%	80.89%
Precision	89.92%	45.68%	66.98%
Recall	50.22%	87.01%	61.47%
F1-Score	64.44%	59.91%	64.11%
AUC	90.24%	73.49%	84.38%

EfficientNet

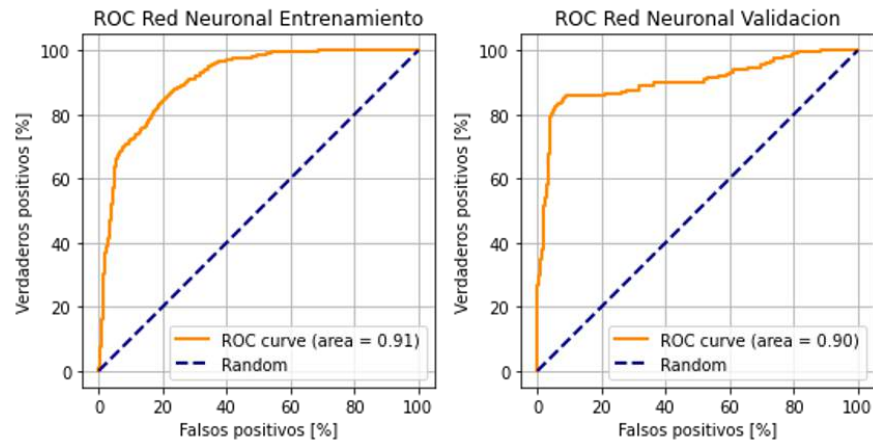


Figura 47. ROC EfficientNet.

Fe de Erratas: la ilustración dice Red Neuronal pero debería ser EfficientNet.

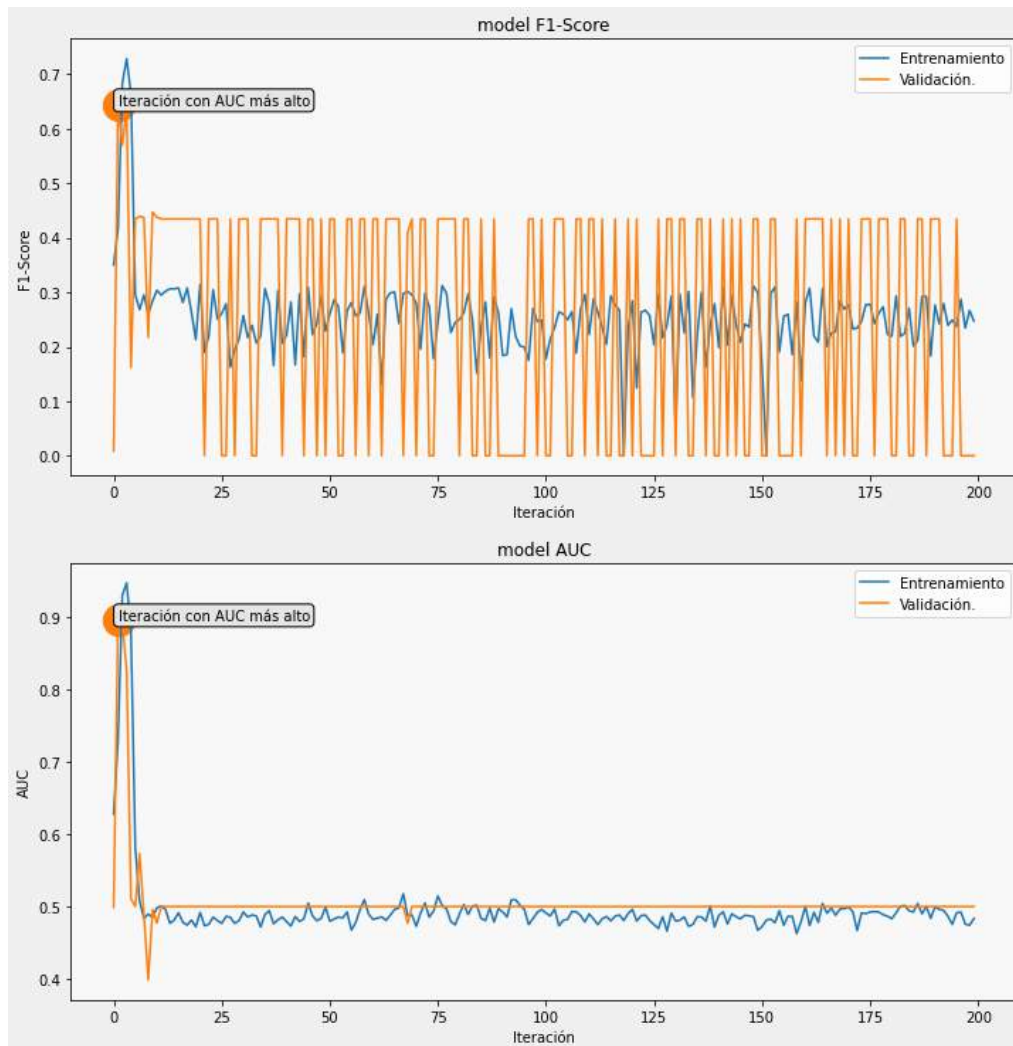


Figura 48.-Curvas de entrenamiento EfficientNet.

ResNet50

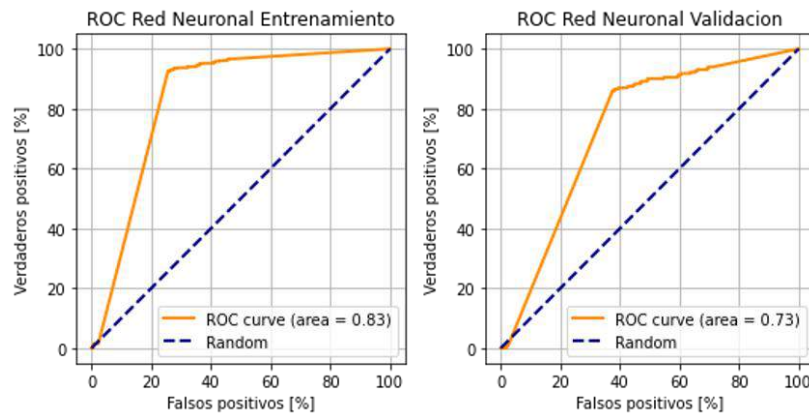


Figura 49. ROC ResNet50.

Fe de Erratas: la ilustración dice Red Neuronal pero debería decir ResNet50.

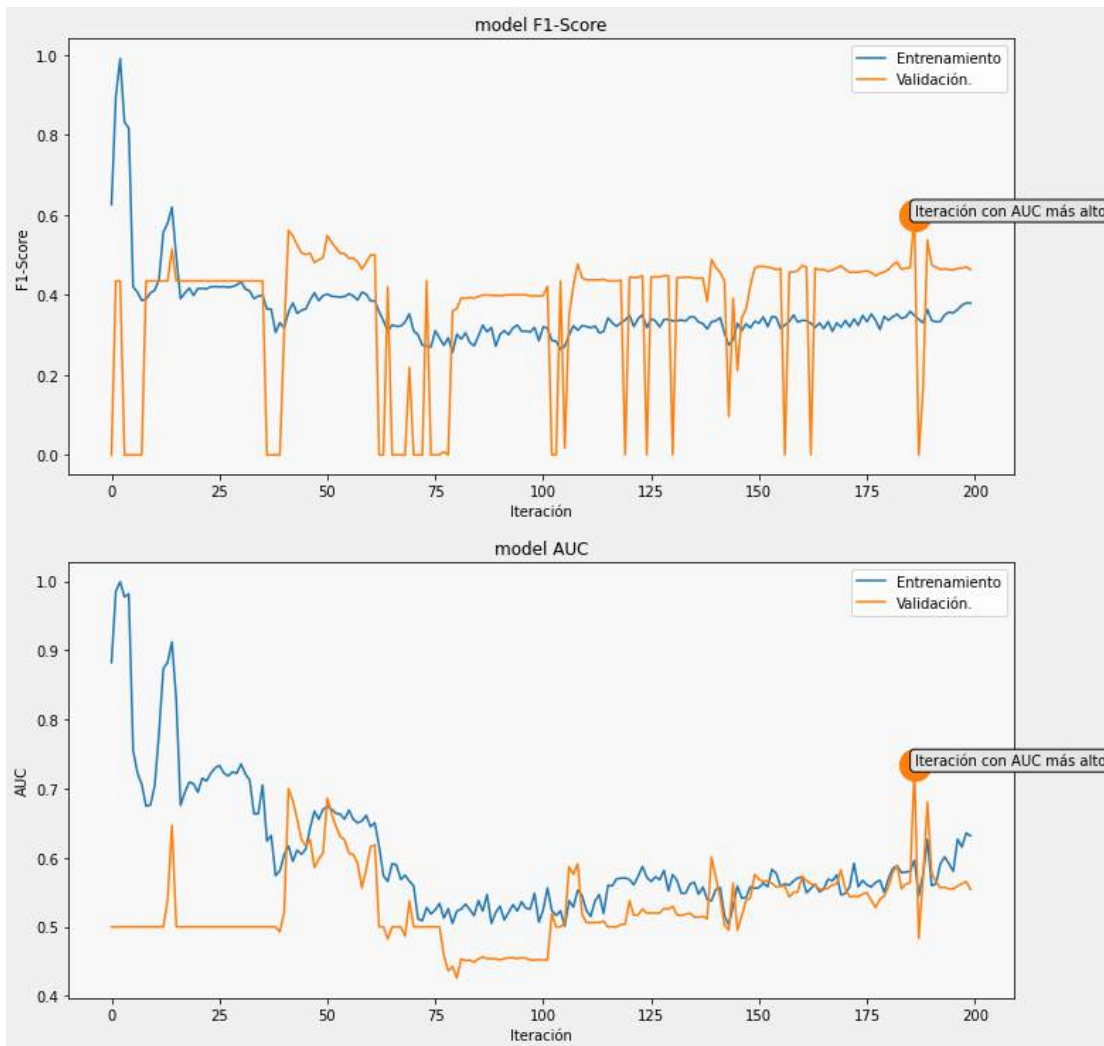


Figura 50. Curvas de entrenamiento ResNet50.

GoogleNet

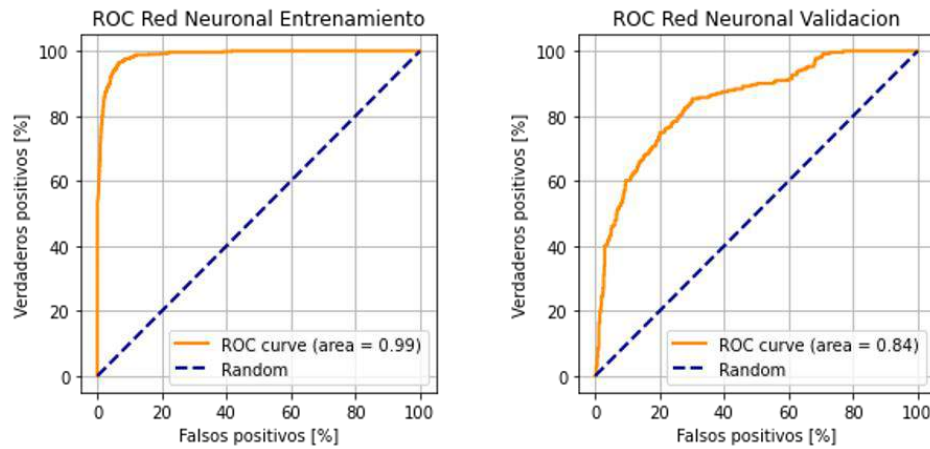


Figura 51. ROC GoogleNet.

Fe de Erratas: la ilustración dice Red Neuronal pero debería decir GoogleNet.

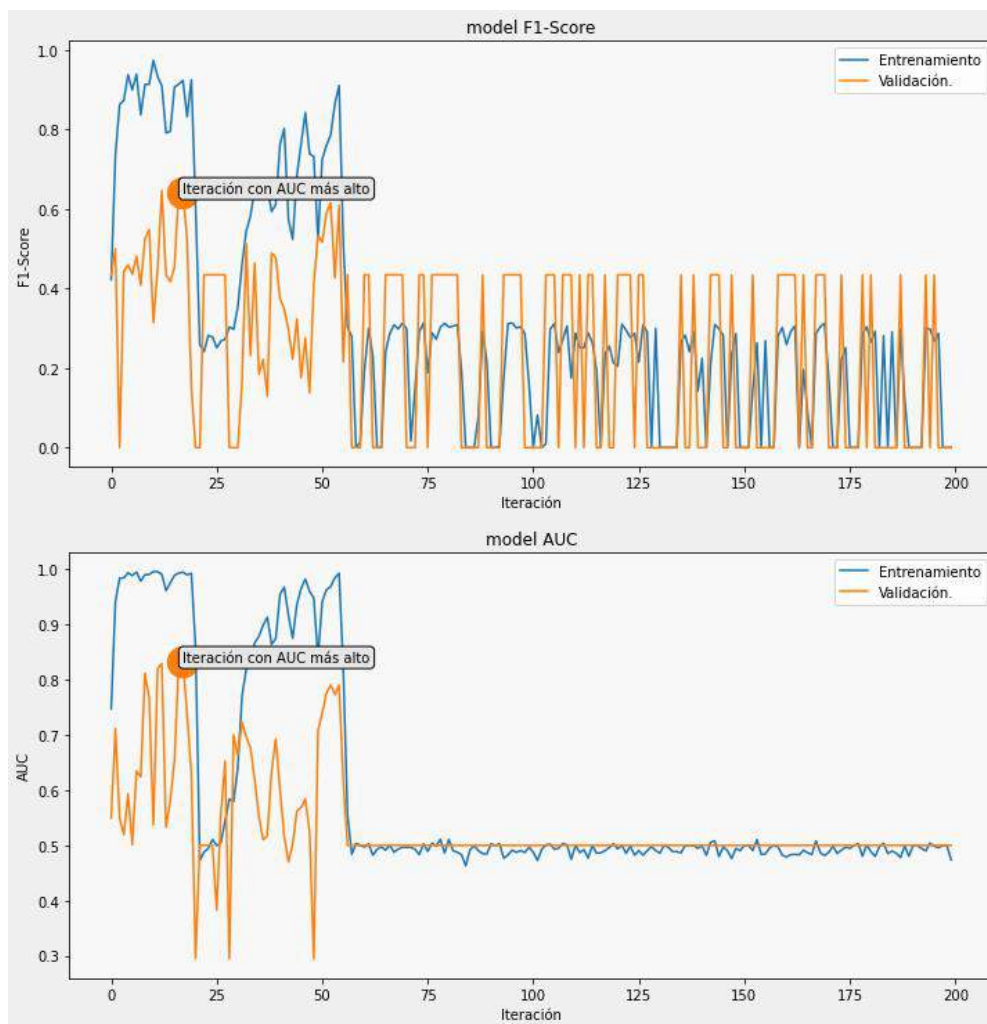


Figura 52. Curvas de entrenamiento GoogleNet.

Con base en los resultados anteriores es posible concluir que la Red Neuronal Convolucional EfficientNet presenta mejor rendimiento con respecto a las demás propuestas utilizando como referencia la mayoría de las métricas de validación.

Sin embargo, el modelo EfficientNet presenta una debilidad con respecto a los demás si se toma en cuenta la sensibilidad. Al ser ésta cercana al 50% implicaría que la probabilidad de que la prueba resulte positiva si el paciente es en verdad positivo es prácticamente el azar. Pero la precisión, la cual es la probabilidad de ser positivo si la prueba da positivo, es cercana al 90%. En otras palabras, el modelo clasifica correctamente a casi el 90% de sus resultados positivos (altamente preciso) pero es incapaz de encontrar a casi la mitad de los enfermos (precariamente sensible).

7.5 Implementación del modelo en IBM Watson

Al momento del despliegue del modelo EfficientNet en el servicio gratuito de IBM Watson (proceso que se documenta en los anexos S, T y U) se presentaron algunas problemáticas. El principal problema fue la falta de compatibilidad entre las bibliotecas utilizadas en el desarrollo de este trabajo y las disponibles en IBM Watson para el despliegue de modelos en la nube. Aunque el entorno de desarrollo de IBM Watson es un servicio que permite modificar modelos, capas o funciones de activación para experimentación, en la práctica (para servicios gratuitos), limita el uso de bibliotecas a las predefinidas sin opción de actualización. Esto representó un inconveniente porque la propuesta de Tan et al [99], propone dos innovaciones en su modelo que, aunque menores en la matemática, conllevan importantes implicaciones en la práctica. La primera es la función de activación “*swish*” y, la segunda, es una capa llamada “*DropOut modificada*” que resuelve el potencial problema de hallar valores NaN (valores no determinados, como divisiones entre cero etc.) durante el entrenamiento. La capa *DropOut* no representaría mayor inconveniente si se ignora y se implementa una “normal” o “clásica”, esto debido a que esta capa solamente actúa durante el entrenamiento “apagando” cierta cantidad de unidades (neuronas) en función de su hiperparámetro p . Pero, la función de activación, sí representa un cambio que puede modificar las operaciones del modelo predictivo (*Forward Propagation*), ya que esta función es la encargada de transformar la respuesta entregada por una unidad a un valor que indique una probabilidad (valor entre 0 y 1) o a un valor continuo positivo, según sea el caso. Un ejemplo de la primera es una función sigmoidea, la cual se aborda al detalle en el Capítulo 5 Marco Teórico, y un ejemplo de la segunda es la función ReLU, la cual se puede simplificar como $\max(x, 0)$, donde x es el valor de entrada de la neurona, como se observa, la función ReLU excluye a todos los valores menores a cero.

Otro problema al que se afrontó la propuesta de este trabajo es la limitada capacidad de procesamiento que el servicio gratuito de IBM Watson proporciona para realizar peticiones a modelos desplegados. En particular, el modelo EfficientNet consume una cantidad de recursos “considerable” que están en función del tamaño en Bytes y la cantidad de neuronas; más de 200 millones para este modelo. Lo anterior causa un consumo de los recursos gratuitos muy rápidamente, lo que limita el número de peticiones (ejemplos a evaluar) a menos de 20 por mes. Para mayor información refiérase a los anexos N, V, U y W.

Por los argumentos anteriores se realizaron dos prototipos. El primero es un prototipo que funciona de forma local y que puede ser implementado en un servicio no gratuito de alojamiento o en un servidor propio, la particularidad de este prototipo es que el modelo EfficientNet propuesto por Tan et al [99] y al cual se referirá en las siguientes tablas y gráficos como *EfficientNet Local*. El segundo prototipo consiste en un despliegue en la nube utilizando el servicio de IBM Watson y su realización se utilizó para demostrar la capacidad del prototipo para entrar rápidamente en etapa de producción, la diferencia con el anterior es que la red neuronal se modificó en los aspectos mencionados anteriormente (De función *swish* a *ReLU* y de capa *DropOut modificada* a *DropOut "clásica"*) y, a partir de este punto, se hará referencia a este modelo como *EfficientNet IBM*.

7.5.1 Sustitución de la función de activación “swish” por “ReLU”

La función *swish* propuesta por Tan et al [99] es una modificación de la función sigmoidea. La idea de la implementación de esta función de activación es conservar las propiedades de la función sigmoidea y, simultáneamente, evitar el fenómeno conocido como “vanishing”.

La función *swish* está definida como $x * \text{sigmoidea}(x)$, donde x es la entrada a la unidad o neurona. Como se observa, se utiliza el valor numérico del resultado de la función sigmoidea, pero se escala con el valor de la entrada. El propósito de este escalamiento es evitar que las multiplicaciones sucesivas de valores en el entrenamiento (*Backpropagation*) tiendan a crear valores cada vez más pequeños (que tienden a cero) en redes neuronales “profundas”, como es el caso del modelo EfficientNet. Este fenómeno se denomina comúnmente como “vanishing” y ocurre durante la ejecución del algoritmo de retro propagación (*Backpropagation*) ya que es necesario multiplicar las salidas de la función sigmoidea (números entre 0 y 1) con otras salidas de la misma función. Al realizar esta operación, los números resultantes son cada vez más pequeños y calcular los gradientes con estos valores se vuelve cada vez más costoso e ineficiente, tanto, que un punto hipotético puede llegar a ser imposible debido a la capacidad limitada de memoria para números de punto flotante de una computadora. Por lo que la solución de la función *swish* es una forma relativamente sencilla de superar este problema.

Sin embargo, durante la implementación del modelo EfficientNet y su despliegue en IBM Watson, esta función no es posible definirla por problemas de compatibilidad entre las versiones de las bibliotecas que no permiten implementar funciones modificadas (esto sólo para la versión gratuita del servicio). Por lo que, se decidió modificar esta función por una función de activación *ReLU*.

Intuitivamente se pensaría que la solución natural sería una función sigmoidea, pero, por el fenómeno *vanishing* descrito y, si se entrena nuevamente al modelo, siendo éste relativamente mayor a otros modelos utilizados en el estado del arte [99], se decidió implementar la función *ReLU*.

Existe otro inconveniente contrario al *vanishing* durante el entrenamiento. Este fenómeno es conocido como “*exploding*” o “*explosión de gradientes*” y se refiere al problema de multiplicar números que cada vez incrementan el valor de la salida lo cual puede {para redes profundas como el modelo EfficientNet} llegar a representar un problema debido a que el incremento de los valores por multiplicaciones sucesivas podría alcanzar un comportamiento exponencial. Este problema

tiene una solución sencilla el cual se describe en los anexos N, O y P, el cual consiste en recortar intencionalmente el valor de los gradientes a un valor umbral arbitrario y así evitar que crezcan “demasiado”. Afortunadamente, para el problema a resolver en este trabajo, la no implementación de esta solución no alteró los resultados ni interfirió en el entrenamiento de los modelos.

7.5.2 Resultados EfficientNet con modificación de función *swish*

En primera instancia se implementó el modelo EfficientNet con las modificaciones descritas en el punto previo. Con los pesos o valor de los parámetros previamente entrenados para el modelo *EfficientNet local* o clásico, obteniéndose los siguientes resultados:

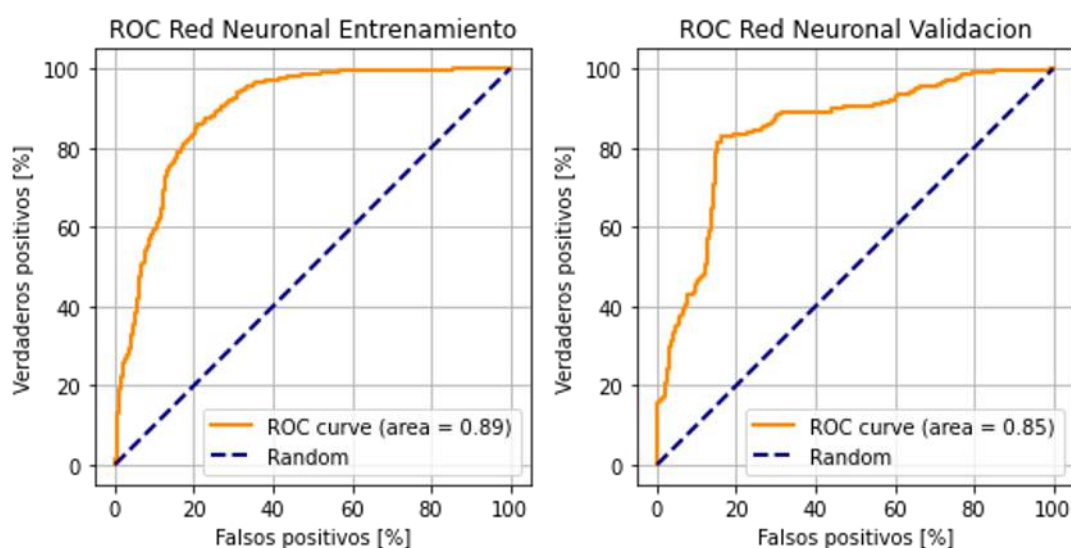


Figura 53. ROC EfficientNet Modificada con pesos de EfficientNet “Original”.

Fe de Erratas: la ilustración dice Red Neuronal pero debería decir EfficientNet.

Tabla 13. Resultados de EfficientNet modificada con pesos entrenados en EfficientNet original Conjunto de Entrenamiento.

EfficientNet modificada (pesos entrenados en original)	
Accuracy	81.96 %
Precision	0.00%
Recall	0.00%
F1-Score	0.00%
AUC	89.22%

Tabla 14. Resultados de EfficientNet modificada con pesos entrenados en EfficientNet original Conjunto de Validación.

EfficientNet modificada (pesos entrenados en original)	
Accuracy	72.24%
Precision	0.00%
Recall	0.00%
F1-Score	0.00%
AUC	84.68%

Como es posible observar en la figura 53 y en las tablas 13 y 14. El modelo fue incapaz de discriminar casos positivos. Aunque la métrica AUC no disminuyó por debajo del 80%, su capacidad de discriminación de positivos medida por las métricas sensibilidad, precisión y F1-Score fueron del 0%.

El proceso para la obtención de los resultados previos se documenta en el anexo V.

7.5.3 Reentrenando EfficientNet y Resultados obtenidos (EfficientNet IBM)

Con base en los resultados anteriores, se entrenó nuevamente al modelo, pero utilizando las modificaciones descritas anteriormente (EfficientNet IBM). El primer entrenamiento se realizó con 150 iteraciones alcanzando los siguientes resultados:

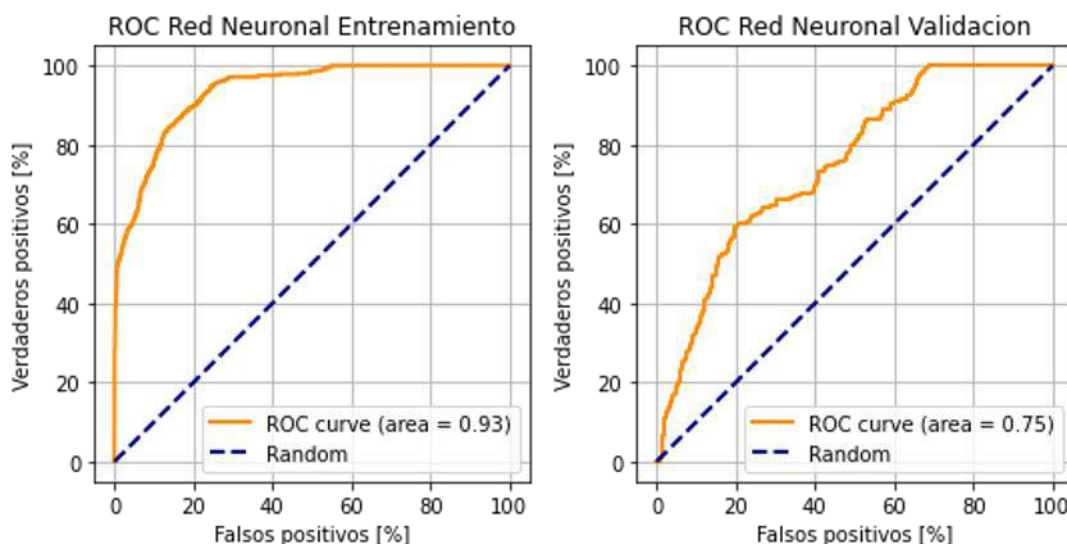


Figura 54. ROC EfficientNet modificada con 150 iteraciones.

Fe de Erratas: la ilustración dice Red Neuronal pero debería decir EfficientNet.

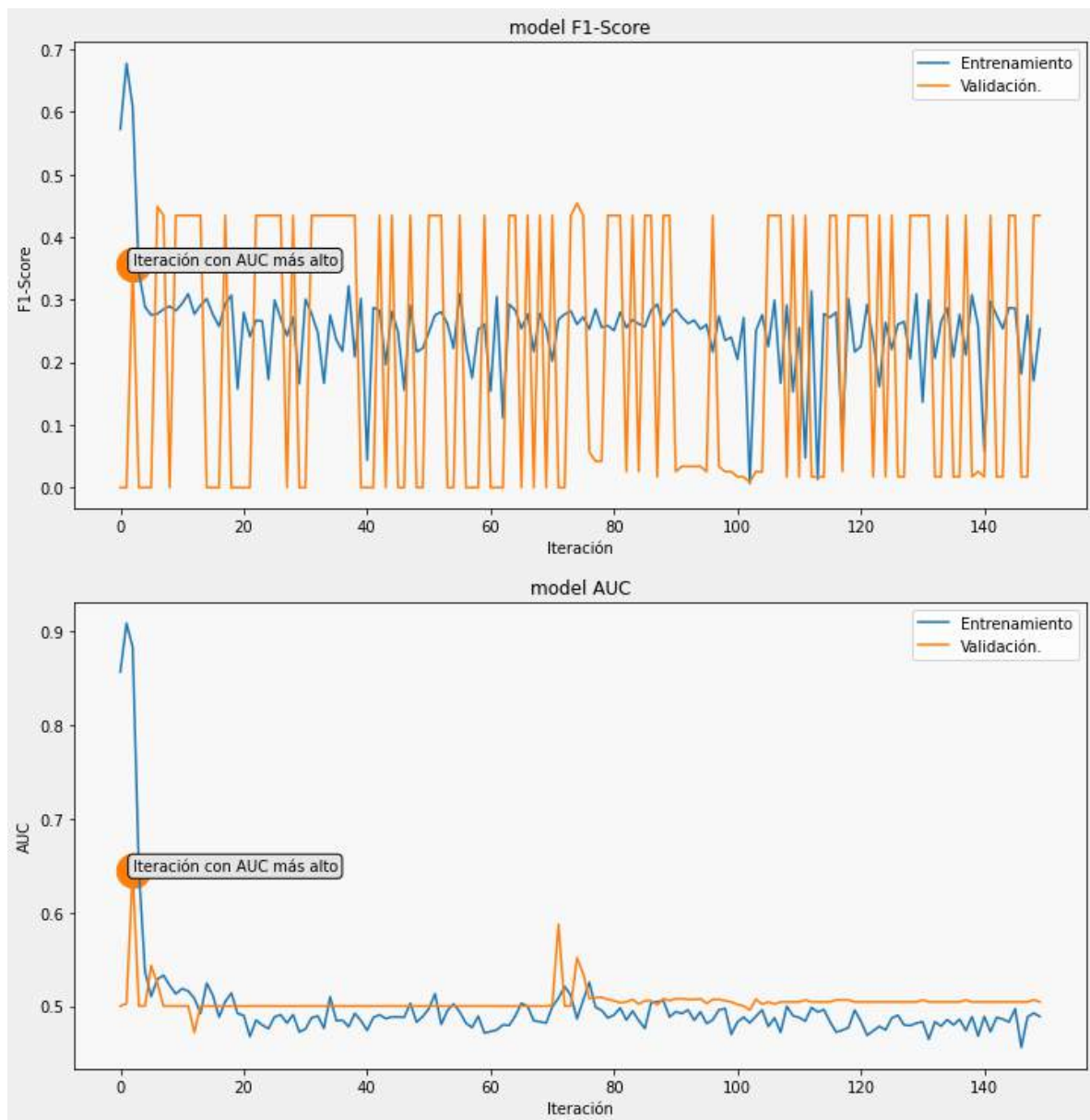


Figura 55. Curvas de entrenamiento EfficientNet modificada 150 iteraciones.

Tabla 15. Resultados de EfficientNet modificada 150 iteraciones Conjunto de Entrenamiento.

EfficientNet modificada 150 iteraciones	
Accuracy	87.71%
Precision	63.72%
Recall	74.09%
F1-Score	68.51%
AUC	93.49%

Tabla 16. Resultados de EfficientNet modificada 150 iteraciones Conjunto de Validación.

EfficientNet modificada 150 iteraciones	
Accuracy	74.28%
Precision	58.42%
Recall	25.54%
F1-Score	35.54%
AUC	75.10%

La figura 53 y las tablas 15 y 16 muestran que el modelo ahora es capaz de discriminar casos positivos, pero con detrimento de la métrica AUC. El proceso para la obtención de estos resultados se describe en el anexo W.

Con base en la tendencia observada en las métricas de la figura 55, realizar más iteraciones sobre el modelo con los pesos aprendidos podría no resultar en un mejor modelo. Lo anterior se explica porque la tendencia apunta a que el modelo alcanza su punto óptimo rápidamente, pero, inmediatamente después, cae y oscila en valores bajos de AUC (el AUC fue la métrica de oro para elegir el modelo durante el entrenamiento). En consecuencia, y después de haber realizado varias pruebas donde se observaba este mismo comportamiento, se decidió utilizar el experimento con la mejor métrica AUC en validación de los cuales se presentan los siguientes resultados:

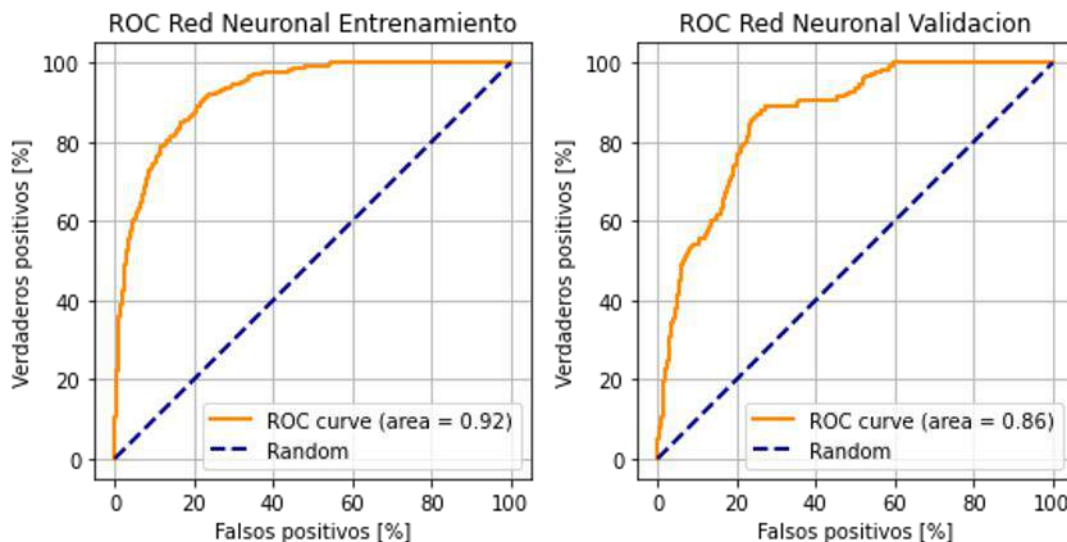


Figura 56. ROC EfficientNet modificada IBM final.

Fe de Erratas: la ilustración dice Red Neuronal pero debería decir EfficientNet.

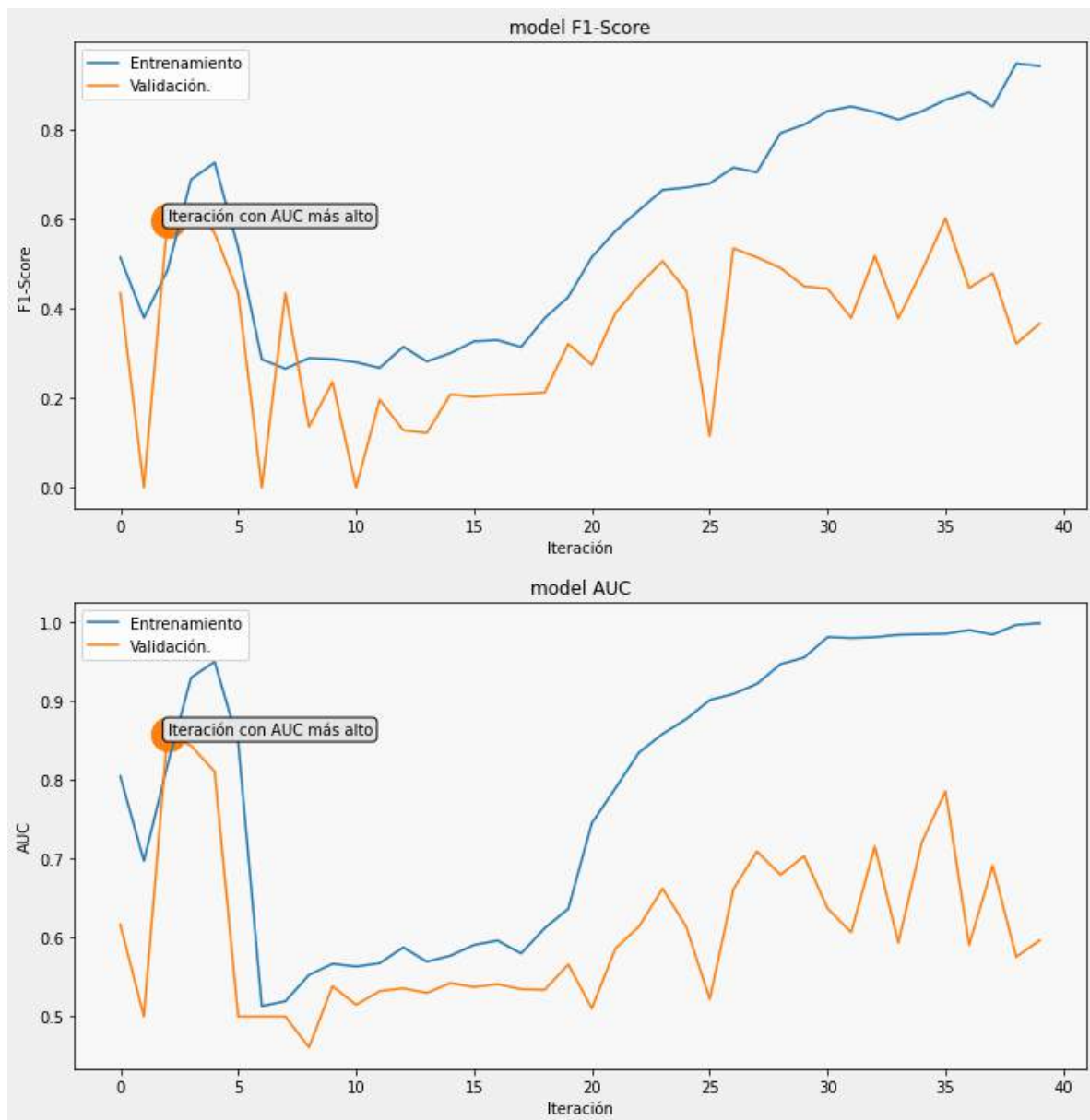


Figura 57. Curvas de entrenamiento EfficientNet modificada IBM final.

Tabla 17. Resultados de EfficientNet modificada IBM final Conjunto de Entrenamiento.

EfficientNet modificada IBM final	
Accuracy	87.95%
Precision	65.87%
Recall	68.94%
F1-Score	67.37%
AUC	92.33%

Tabla 18. Resultados de *EfficientNet* modificada IBM final Conjunto de Validación.

	EfficientNet modificada IBM final
Accuracy	79.09%
Precision	64.18%
Recall	55.84%
F1-Score	59.72%
AUC	86.21%

Las métricas son mejores con respecto al experimento previo y alcanzan valores comparables a los modelos GoogleNet y ResNet50, los cuales fueron expuestos previamente. Los detalles de este experimento se documentan en el Anexo X.

7.6 Comparativo de Parte 1 y Parte 2 de este trabajo

Para finalizar este capítulo se muestra el rendimiento de los modelos seleccionados para el prototipo final de la Parte 2 con el modelo utilizado en la Parte 1.

Se utiliza el término *EfficientNet local* para hacer referencia al modelo empleado en la versión local (no desplegado en la nube) del prototipo de software. Por otra parte, el nombre *EfficientNet modificada IBM* es el modelo empleado en la versión web.

Tabla 19. Comparativo modelos Conjunto de Entrenamiento.

	EfficientNet local	EfficientNet modificada IBM	Neural Net
Accuracy	87.71%	87.95%	72.70%
Precision	73.19%	65.87%	33.00%
Recall	50.33%	68.94%	49.83%
F1-Score	59.65%	67.37%	39.71%
AUC	90.83%	92.33%	69.64%

Tabla 20. Comparativo modelos Conjunto de Validación.

	EfficientNet local	EfficientNet modificada IBM	Neural Net
Accuracy	84.62%	79.09%	59.50%
Precision	89.92%	64.18%	25.00%
Recall	50.22%	55.84%	22.94%
F1-Score	64.44%	59.72%	23.93%
AUC	90.24%	86.21%	51.63%

Los modelos propuestos en esta Parte 2 (EfficientNet) mostraron mejor respuesta en todas las métricas en comparación con el modelo final de la Parte 1 (Neural Net).

A continuación, se enlistan las mejoras con respecto a la Parte 1:

1. Mayor cantidad de modelos evaluados
2. Evaluación y análisis de datos clínicos (esto no se reportó en la Parte 1)
3. Mayor variabilidad en pacientes e imágenes. En la Parte 1 se utilizaron únicamente 40 imágenes (una por paciente) para entrenamiento y validación.
4. En la Parte 2 se emplearon tecnologías Open Source en sustitución de la plataforma cerrada Matlab.
5. Aplicación Web con la ventaja de ser multiplataforma y escalable a aplicación local.

NOTA FINAL: Los valores de AUC y F1-Score en las gráficas de entrenamiento de los diferentes modelos evaluados en este trabajo, no necesariamente coinciden con las tablas de resultados, debido a que se realizaron con bibliotecas diferentes, las cuales utilizan métodos diferentes para el cálculo de estas métricas. Sin embargo, para el caso de los valores en las tablas de resultados, se utilizaron los resultados de la biblioteca que se apega a las definiciones establecidas en el Capítulo 5 Marco Teórico. La biblioteca empleada se especifica en cada uno de los anexos de cada experimento. A pesar de lo anterior, se decidió mostrar estos gráficos ya que otorgan al lector una visión general del comportamiento de estas métricas durante la optimización de los modelos.

Capítulo 8 Prototipo final

Finalmente, se realizaron dos prototipos de software. En el primero se utilizó el servicio de IBM Watson para el despliegue de modelos y se construyó una aplicación Web alojada en Heroku, como se muestra en la figura 58. El segundo prototipo se implementó de forma local en un equipo de cómputo, pero, con los requerimientos necesarios para su implementación en cualquier servidor profesional.

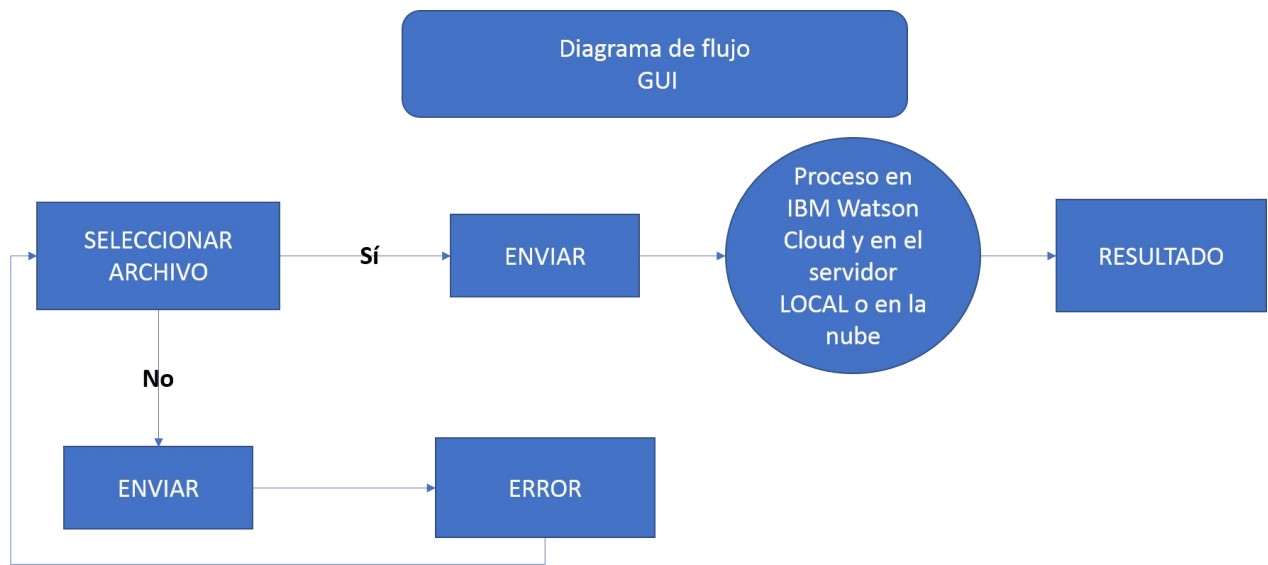


Figura 58. Diagrama GUI.

8.1 Interfaz Gráfica de Usuario (GUI)

El prototipo está diseñado de tal forma que el usuario, con mínimos conocimientos de computación y medicina, puede interactuar con la plataforma y, al final del proceso de análisis, recibir información potencialmente útil para la toma de decisiones con base en el análisis de algún termograma de la región frontal del cuerpo.

A continuación, se muestran las imágenes de la GUI final.

Termogramas de prueba <https://mega.nz/folder/UZoGBKpT#4Y6MRc-LE0EhvznY4o1TQ>

El usuario ingresa a la siguiente url: <https://aithermobreastcancer2.herokuapp.com/>

El usuario sube el archivo solicitado y lo envía. Si por alguna razón envía un archivo no permitido o no envía nada, se redirecciona a una página de error que permite al usuario regresar al inicio.

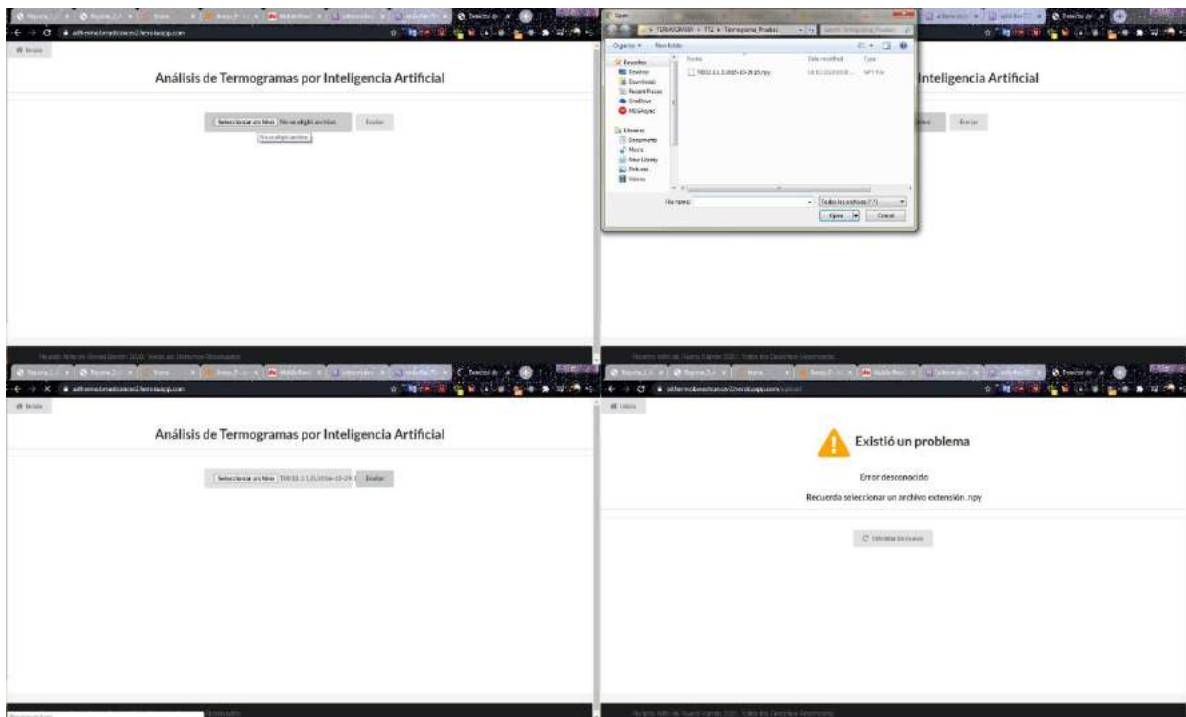


Figura 59. GUI Usuario sube archivo.

Si el usuario envía el archivo correcto se despliega una simulación de carga de trabajo. Mientras tanto el servidor realiza múltiples tareas en segundo plano. El navegador solicita al servidor de forma “silenciosa” notificaciones a la tarea en segundo plano para saber el avance y el momento en que ésta finalmente concluye.

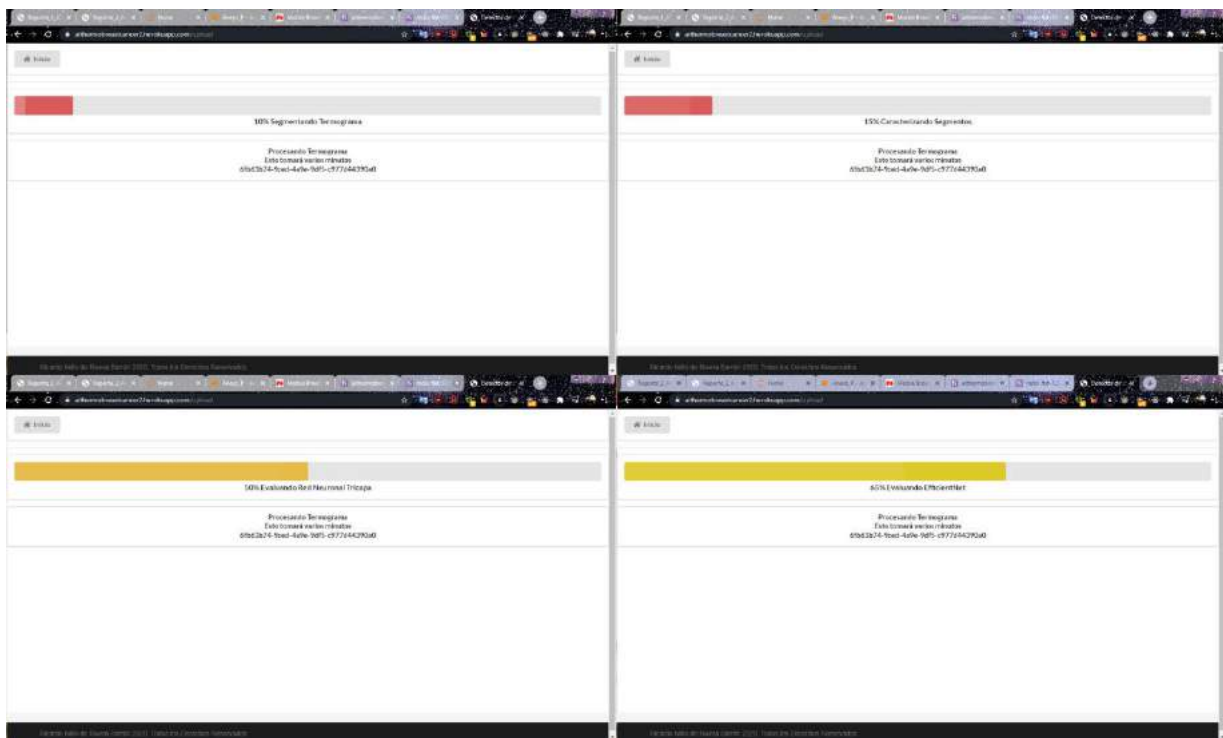


Figura 60. GUI Avance de procesamiento de termograma.

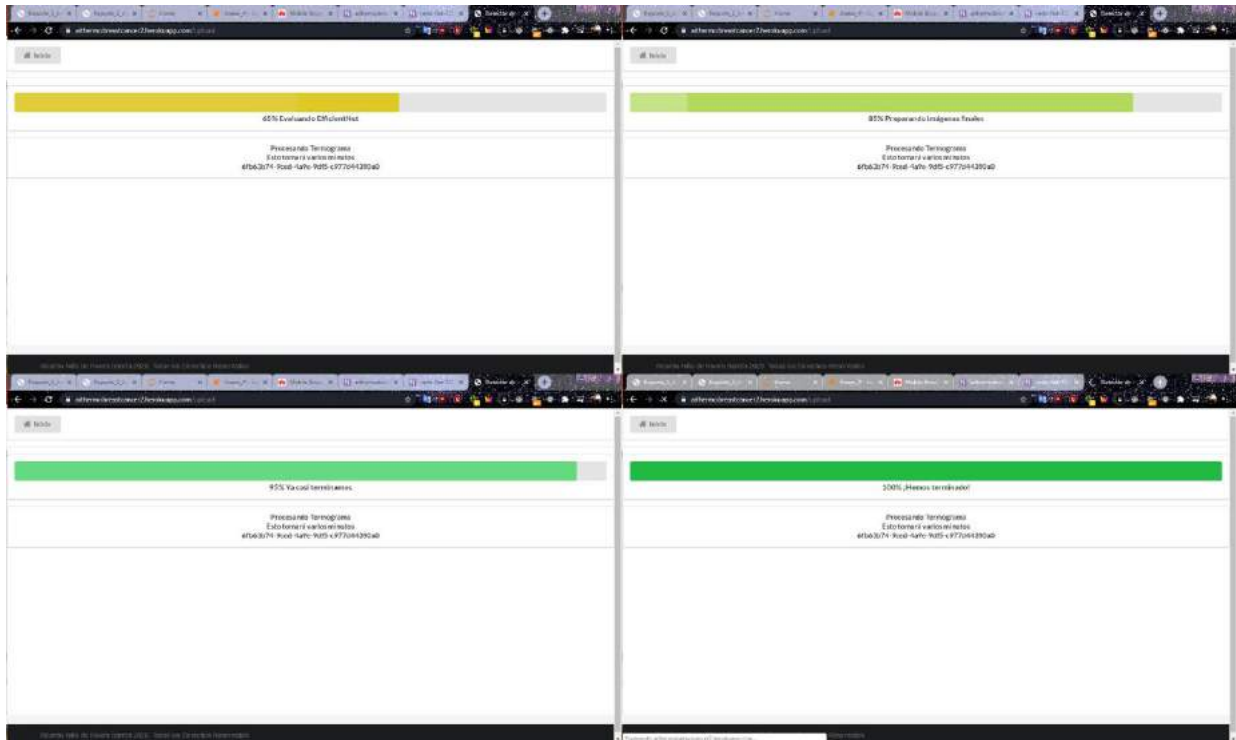


Figura 61. GUI 2 Avance de procesamiento de termograma.

Para realizar las peticiones del navegador al servidor, de tal forma que el primero sepa el momento en que la tarea en segundo plano concluye, se utilizó JQuery y Ajax. JQuery es una biblioteca de JavaScript para comunicación e interacción con páginas. Por otra parte, Ajax, es una integración de diferentes tecnologías para gestionar peticiones al servidor.

Por último, el usuario recibe una página con el resultado y algunas imágenes del termograma evaluado.

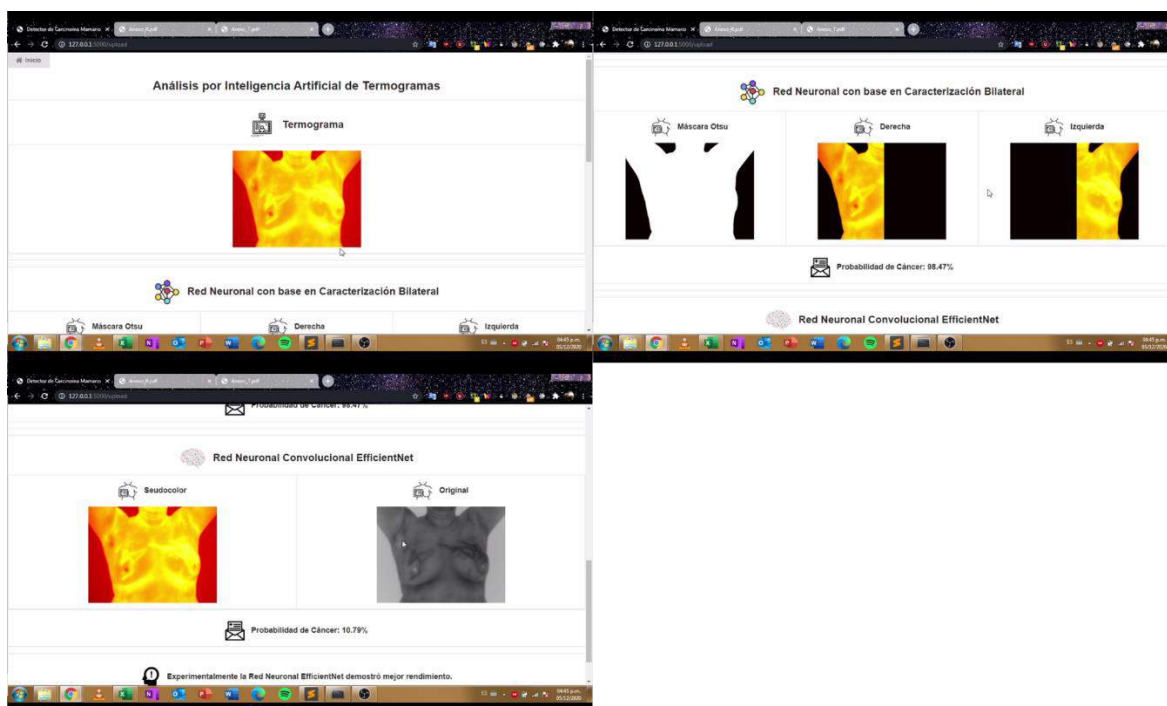


Figura 62. Resultado.

Generalmente la aplicación Web está en mantenimiento debido al servicio limitado que ofrece IBM Cloud. Los modelos desplegados en este servicio realizan procesos que requieren una cantidad importante de memoria y capacidad de procesamiento en cada petición de algún usuario, por lo que las consultas exitosas a estos modelos están condicionadas a entre diez y quince por mes.

Sin embargo, con el objetivo de no mostrar una página de error, la aplicación despliega la página de resultado con las leyendas de mantenimiento debido a problemas con IBM Watson en la sección donde se encontraría (en una situación “normal”) la probabilidad de cáncer. Lo anterior se puede observar en la figura 63.



Figura 63. Resultado cuando la aplicación se encuentra en mantenimiento.

La interacción con la aplicación, gestión y operaciones matemáticas se realizaron en el lenguaje de programación *Python* en su versión 3.7, además se utilizaron las bibliotecas *flask* para desarrollo web y *celery* para gestionar tareas en segundo plano.

Otros lenguajes utilizados fueron *JavaScript*, *JQuery* (biblioteca de *JavaScript*), *HTML* y *CSS*. Estos dos últimos utilizados para realizar la interfaz gráfica de usuario (GUI).

8.2 Diagrama de la aplicación

La aplicación se encuentra alojada en Heroku (servicio en la nube para desplegar aplicaciones web comerciales o prototipos) y utiliza IBM Watson Cloud (similar a Heroku pero especializado en Inteligencia Artificial), que es utilizado como servidor alojador de los modelos inteligentes que previamente fueron entrenados en Kaggle (comunidad en la nube de desarrolladores y científicos en Inteligencia Artificial que ofrece el servicio gratuito de alojamiento de datos y recursos para entrenamiento de modelos escritos en lenguaje *Python* y/o *R*).

Cuando el usuario ingresa a la aplicación y envía un documento, realmente está interactuando con la aplicación alojada en Heroku. A su vez, la aplicación se encarga de gestionar las peticiones a los modelos alojados en IBM Watson que evalúan los termogramas. En caso de no existir ningún inconveniente, IBM devuelve una respuesta en, aproximadamente, un minuto por modelo. Cuando este proceso no se puede llevar a cabo, por algún problema con el servidor de IBM, la aplicación no

devuelve una página de error, en su lugar, el usuario continúa observando una página de avance del proceso y, al final, se le muestra una página con imágenes del termograma con una leyenda en donde deberían ir las cifras de probabilidad que hacen referencia a que ocurrió un error con la petición a los modelos alojados en IBM Watson.

La figura 64 muestra los aspectos fundamentales del funcionamiento de la aplicación web. La diferencia con la versión local es que los modelos y la aplicación se ejecutan en el mismo servidor (computadora).

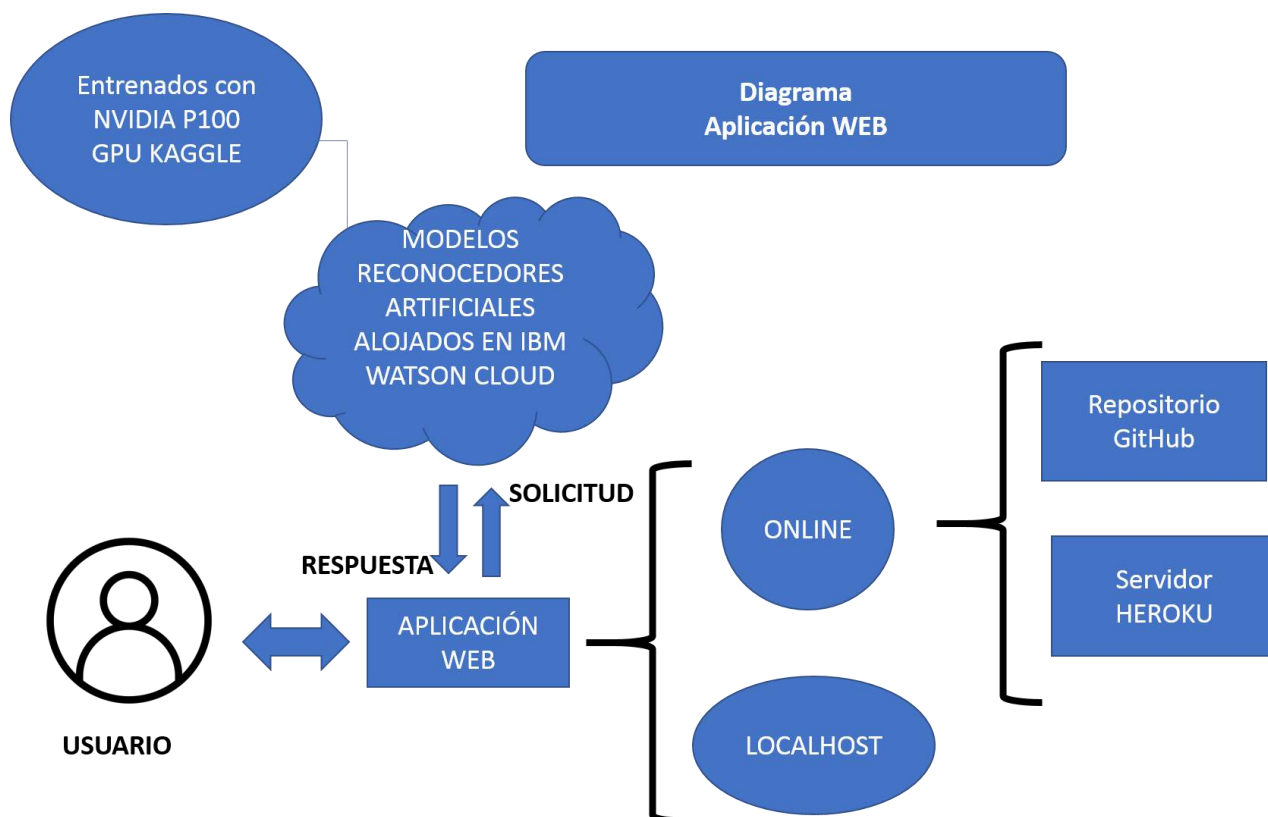


Figura 64. Diagrama de la Aplicación WEB.

Capítulo 9 Conclusiones

La termografía, junto con clasificadores artificiales inteligentes, tiene el potencial de convertirse en una herramienta complementaria para la detección de lesiones cancerígenas de la región mamaria en mujeres mayores de 40 años.

Aunque en la primera parte de este trabajo la Red Neuronal de tres capas ocultas demostró ser el modelo con mejor rendimiento (al no haber tenido ningún error de clasificación en el conjunto de validación), no mantuvo este buen rendimiento con un conjunto mayor de datos, demostrando que el modelo presenta deficiencias de generalización, a pesar de haber sido entrenado con nuevos casos. Modelo que, aparentemente, es posible sustituir con una arquitectura convolucional, la cual, funciona sin ningún proceso previo de caracterización, aunque a un mayor costo computacional.

La segmentación automática logró describir las regiones de interés sin pérdida significativa de información, sin embargo, la segmentación automática con base en parábolas no mostró resultados alentadores debido a que la morfología de la mama no es, necesariamente, parabólica, ni define, necesariamente tampoco, una figura geométrica.

Este trabajo también establece las bases teóricas para suponer que la utilización de imágenes con menor sensibilidad a la entregada por la cámara FLIR T650sc (menos a 20 mK), pueden contener suficiente información para encontrar lesiones cancerígenas. Lo anterior se concluye con base en la propuesta de transformación lineal, que se desarrolló en este trabajo, de los valores de temperatura a una imagen de 8 bits (imagen a escala de gris con 256 posibles valores). Cada cambio de valor de gris de un píxel es equivalente a, aproximadamente, 0.8°C. Es decir, una sensibilidad (en términos del sensor, no en términos estadísticos) 40 veces menor a la entregada por la cámara infrarroja utilizada en este trabajo. Además, se obtuvo evidencia que sugiere que la resolución del sensor de adquisición podría ser menor a la resolución del dispositivo que se utilizó en este estudio, lo que implicaría una disminución en los costos de la cámara infrarroja en futuras propuestas.

Capítulo 10 Sugerencias para investigaciones futuras

Realizar un estudio incluyendo mujeres menores de 40 años. La norma mexicana (NOM-041-SSA2-2011) no sugiere la realización de mastografías en mujeres jóvenes (menores de 40 años) pero, un estudio futuro puede solucionar este problema realizando un seguimiento de por lo menos un año con el fin de descartar o confirmar la presencia de enfermedad. De hecho, para algunas mujeres de este trabajo, no se adquirió un mastograma radiológico, pero se realizó un seguimiento clínico de por lo menos un año posterior a la adquisición del termograma.

Obtener una mayor cantidad de imágenes a partir de un grupo de estudio mayor al utilizado en este trabajo. Con base en una toma muestral significativa de una población específica.

Validar los modelos utilizando validación cruzada o, en otras palabras, utilizar dos conjuntos; un conjunto de validación cruzada y un conjunto de validación final. Esto con el objetivo de medir el rendimiento comparando dos conjuntos y no un sólo conjunto de validación. Con esto se podría conseguir encontrar un mejor modelo que solucione el problema planteado en este trabajo.

Utilizar la técnica *cross fold validation*. Es decir, validar el modelo en diferentes subconjuntos del conjunto original, para lo cual se generarían tantos modelos como subconjuntos de entrenamiento y, al final, se obtendría la probabilidad de cáncer con el promedio de las salidas entregadas de cada uno de los modelos.

Realizar un prototipo que ubique las regiones sospechosas en el termograma a partir del peso que la Red Neuronal de a ciertos píxeles. Existen propuestas en el estado del arte para Redes Neuronales Convolucionales que no requieren etiquetado previo de zonas sospechosas. [104][105]

Realizar un reconocedor de detección de regiones sospechosas utilizando como base un estudio clínico que previamente ubique las regiones sospechosas mediante ultrasonido o mastografía.

Realizar un reconocedor multitarea, es decir, un modelo que sea capaz de encontrar diferentes lesiones, no sólo cancerígenas.

Aumentar la resolución de entrada de las imágenes.

Aumentar la cantidad de iteraciones para el entrenamiento.

Utilizar diferentes técnicas de regularización para evitar sobreajuste.

Utilizar diferentes métodos de optimización.

Utilizar diferentes métodos para disminución de la tasa de aprendizaje durante el entrenamiento.

Utilizar DropOut y realizar un barrido de posibles valores.

Realizar modelos híbridos, como por ejemplo EfficientNet (Entrada) + GoogleNet (Salida)

Utilizar modelos preentrenados con imágenes médicas.

Realizar modelos “paralelos”, por ejemplo, 5 redes convolucionales en paralelo, entrenarlas independientemente, pero integrarlas utilizando el promedio de la probabilidad de cáncer que cada

una entrega y mostrar ese dato al usuario, con las evaluaciones de rendimiento pertinentes. La diferencia con la técnica *cross fold validation*, es que no se evaluarían los modelos en subconjuntos, sino en todo el conjunto de entrenamiento.

Integrar un modelo de datos clínicos mediante promedio (como en el párrafo anterior) o mediante una nueva arquitectura que considere su utilización durante el entrenamiento en paralelo con la imagen.

Bibliografía

- [1] SHORTLIFFE, E. H.; BUCHANAN, B. G. *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*, (eBook version). 2013
- [2] DAYAKSHINI, Sathish, et al. *Medical imaging techniques and computer aided diagnostic approaches for the detection of breast cancer with an emphasis on thermography—a review*. International Journal of Medical Engineering and Informatics, 2016, vol. 8, no 3, p. 275-299.
- [3] TORRE, Lindsey A., et al. *Global cancer statistics, 2012*. CA: a cancer journal for clinicians, 2015, vol. 65, no 2, p. 87-108.
- [4] MOSKOWITZ, Myron, et al. Lack of efficacy of thermography as a screening tool for minimal and stage I breast cancer. New England Journal of Medicine, 1976, vol. 295, no 5, p. 249-252.
- [5] HEAD, Jonathan F.; WANG, Fen; ELLIOTT, Robert L. *Breast thermography is a noninvasive prognostic procedure that predicts tumor growth rate in breast cancer patients*. Annals of the New York Academy of Sciences, 1993, vol. 698, no 1, p. 153-158.
- [6] GONZÁLEZ, Francisco J. *Theoretical and clinical aspects of the use of thermography in non-invasive medical diagnosis*. Biomedical Spectroscopy and Imaging, 2016, vol. 5, no 4, p. 347-358.
- [7] GONZÁLEZ, Francisco Javier. *Non-invasive estimation of the metabolic heat production of breast tumors using digital infrared imaging*. Quantitative InfraRed Thermography Journal, 2011, vol. 8, no 2, p. 139-148.
- [8] GONZÁLEZ, F. J. *Thermal simulation of breast tumors*. Revista mexicana de física, 2007, vol. 53, no 4, p. 323-326.
- [9] SCHAEFER, Gerald; ZÁVIŠEK, Michal; NAKASHIMA, Tomoharu. *Thermography based breast cancer analysis using statistical features and fuzzy classification*. Pattern Recognition, 2009, vol. 42, no 6, p. 1133-1137.
- [10] QI, Hairong; HEAD, Jonathan F. *Asymmetry analysis using automatic segmentation and classification for breast cancer detection in thermograms*. En Engineering in Medicine and Biology Society, 2001. Proceedings of the 23rd Annual International Conference of the IEEE. IEEE, 2001. p. 2866-2869.
- [11] KURUGANTI, Phani Teja; QI, Hairong. *Asymmetry analysis in breast cancer detection using thermal infrared images*. En Engineering in Medicine and Biology, 2002. 24th Annual Conference and the Annual Fall Meeting of the Biomedical Engineering Society EMBS/BMES Conference, 2002. Proceedings of the Second Joint. IEEE, 2002. p. 1155-1156.
- [12] NICANDRO, Cruz-Ramírez, et al. *Evaluation of the diagnostic power of thermography in breast cancer using bayesian network classifiers*. Computational and mathematical methods in medicine, 2013, vol. 2013.
- [13] ACHARYA, U. Rajendra, et al. *Thermography based breast cancer detection using texture features and support vector machine*. Journal of medical systems, 2012, vol. 36, no 3, p. 1503-1510.

- [14] MILOSEVIC, Marina; JANKOVIC, Dragan; PEULIC, Aleksandar. *Thermography based breast cancer detection using texture features and minimum variance quantization*. EXCLI journal, 2014, vol. 13, p. 1204.
- [15] NG, E. Y. K.; CHEN, Y.; UNG, L. N. *Computerized breast thermography: study of image segmentation and temperature cyclic variations*. Journal of medical engineering & technology, 2001, vol. 25, no 1, p. 12-16.
- [16] NG, Eddie Y.-K.; CHEN, Y. *Segmentation of breast thermogram: improved boundary detection with modified snake algorithm*. Journal of Mechanics in Medicine and Biology, 2006, vol. 6, no 02, p. 123-136.
- [17] SCALES, N.; KERRY, C.; PRIZE, M. *Automated image segmentation for breast analysis using infrared images*. En Engineering in Medicine and Biology Society, 2004. IEMBS'04. 26th Annual International Conference of the IEEE. IEEE, 2004. p. 1737-1740.
- [18] GOLESTANI, N.; ETEHADTAVAKOL, M.; NG, E. Y. K. *Level set method for segmentation of infrared breast thermograms*. 2014.
- [19] PÉREZ, María G., et al. *Detección temprana del cáncer de mama mediante la termografía en Ecuador*. Maskana, 2016, p. 111-123.
- [20] GUAMÁN, Diana. *Análisis, evaluación y caracterización de imágenes termográficas utilizando redes neuronales artificiales y técnicas de extracción de características de textura de Haralick*. 2015. Tesis de Licenciatura. Universidad Técnica de Ambato. Facultad de Ingeniería en Sistemas, Electrónica e Industrial. Carrera de Ingeniería en Electrónica y Comunicaciones.
- [21] WAKANKAR, Asmita T.; SURESH, G. R. *Automatic Diagnosis of Breaste Cáncer usina Thermographic Color Analysis and SVM Classifier*. En The International Symposium on Intelligent Systems Technologies and Applications. Springer International Publishing, 2016. p. 21-32.
- [22] KOAY, J.; HERRY, C.; FRIZE, M. *Analysis of breast thermography with an artificial neural network*. En Engineering in Medicine and Biology Society, 2004. IEMBS'04. 26th Annual International Conference of the IEEE. IEEE, 2004. p. 1159-1162.
- [23] HADDADNIA, Javad, et al. *Diagnosis of breast cancer using a combination of genetic algorithm and artificial neural network in medical infrared thermal imaging*. Iranian Journal of Medical Physics, 2012, vol. 9, no 4, p. 265-274.
- [24] WAHAB, Asnida Abd; SALIM, Maheza Irna Mohamad; AZIZ, Maizatul Nadwa Che. *In Vivo Thermography-Based Image for Early Detection of Breast Cancer Using Two-Tier Segmentation Algorithm and Artificial Neural Network*. En Application of Infrared to Biomedical Sciences. Springer Singapore, 2017. p. 109-131.
- [25] ARAÚJO, Marcus C.; LIMA, Rita CF; DE SOUZA, Renata MCR. *Interval symbolic feature extraction for thermography breast cancer detection*. Expert Systems with Applications, 2014, vol. 41, no 15, p. 6728-6737.
- [26] OKUNIEWSKI, Rafał, et al. *Contour classification in thermographic images for detection of breast cancer*. En Photonics Applications in Astronomy, Communications, Industry, and High-Energy

Physics Experiments 2016. International Society for Optics and Photonics, 2016. p. 100312V-100312V-8.

[27] RAGHAVENDRA, U., et al. *An integrated index for breast cancer identification using histogram of oriented gradient and kernel locality preserving projection features extracted from thermograms*. Quantitative InfraRed Thermography Journal, 2016, vol. 13, no 2, p. 195-209.

[28] GOGOI, Usha Rani, et al. *A study and analysis of hybrid intelligent techniques for breast cancer detection using breast thermograms*. En Hybrid Soft Computing Approaches. Springer India, 2016. p. 329-359.

[29] OLESZKIEWICZ, Witold, et al. *Application of SVM classifier in thermographic image classification for early detection of breast cancer*. En Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2016. International Society for Optics and Photonics, 2016. p. 100312T-100312T-8.

[30] NEUMANN, Łukasz, et al. *Preprocessing for classification of thermograms in breast cancer detection*. En Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2016. International Society for Optics and Photonics, 2016. p. 100313A-100313A-8.

[31] LALLI, G., et al. *A development of knowledge-based inferences system for detection of breast cancer on thermogram images*. En Computer Communication and Informatics (ICCCI), 2014 International Conference on. IEEE, 2014. p. 1-6.

[32] Martínez García V. *Capítulo 12 Aparato reproductor femenino*. En; García-Porrero Juan A., Hurlé Juan M. *Anatomía Humana*. Madrid, España. McGraw-Hill Interamericana de España, 2005; 539-541.

[33] U.S. National Cancer Institute. *Breast Cancer Treatment (PDQ®)—Patient Version* [en línea]. U.S. National Institutes of Health. October 14, 2020 [ref. de 18 de diciembre de 2020]. Disponible en Web: <<https://www.cancer.gov/types/breast/patient/breast-treatment-pdq>>

[34] Organización Mundial de la Salud. *Factores de riesgo* [en línea]. 2020 [ref. de 18 de diciembre de 2020]. Disponible en Web: <http://www.who.int/topics/risk_factors/es/>

[35] Mia M. Gaudet, Susan M. Gapstur, Juzhong Sun, et al. *Active Smoking and Breast Cancer Risk: Original Cohort Data and Meta-Analysis*. **Journal of National Cancer Institute** 2013; 105 (8): 515-525.

[36] Esqueda Elizondo J. J. *Capítulo 1 Fundamentos de procesamiento de imágenes digitales*. En; Esqueda Elizondo J. J., Palafox Maestre L. E. *Fundamentos de procesamiento de imágenes*. Mexicali, México. Universidad Autónoma de Baja California, 2005; 9-13.

[37] V.S. Nalwa. *A Guided Tour to Computer Vision*. Addison-Wesley, 1993.

[38] Alférez Edwin S. *Trabajo de investigación de maestría: Detección del carcinoma de glándula mamaria fusionando variables clínicas y termográficas*. Universidad Industrial de Santander. 2010; 15.

- [39] Orel Susan G., Kay Nicole, Reynolds Carol, et al. *BI-RADS Categorization As a Predictor of Malignancy*. **RSNA Radiology**. Vol. 211 Issue 3, June 1999.
- [40] González R., Woods R. *Digital Image Processing*, Third edition, Pearson Prentice Hall, 2008.
- [41] Otsu N. *A Threshold Selection Method from Gray-Level Histograms*. **IEEE Trans. SMC**-9:1, 1979.
- [42] Sobel I. *An Isotropic 3x3 Image Gradient Operator* [en línea]. February 2014 [ref. de 19 de diciembre de 2020]. ResearchGate. Disponible en Web: <https://www.researchgate.net/publication/239398674_An_Isotropic_3x3_Image_Gradient_Operator>
- [43] Robles A., García J. *Métodos numéricos en ingeniería: prácticas con Matlab*. Segunda edición, Textos Universitarios Ediuno, 2006.
- [44] Haralick Robert M., Shanmugan K., Dinstein I. *Textural Features for Image Classification*. **IEEE Transactions on Systems, Man, and Cybernetics** (Volume: SMC-3, Issue: 6, Nov. 1973)
- [45] *Chapter 1. Computational Intelligence and Knowledge*. En; Poole D. Mackworth A. Randy G. *Computational Intelligence: A Logical Approach*. Oxford University Press, 1997; 1-22.
- [46] Turing A. *Computing Machinery and Intelligence*. **Mind**; Oxford University Press; Vol. 59 No. 253 (Oct. 1950); 433-460.
- [47] Wu Y., Giger M., Doi K., et al. *Artificial neural networks in mammography: application to decision making in the diagnosis of breast cancer*. **RSNA Radiology**. Vol. 87 Issue 1, April 1993.
- [48] Dean J. *Big Data, Data Mining and Machine Learning: Value Creation for Business Leaders and Practitioners*. John Wiley & Sons; 2014.
- [49] *Chapter 10: Giving Computers the Ability to Learn from Data*. En; Raschka S. *Python Machine Learning*. Packt Publishing, 2016; 1-16.
- [50] Rosenblatt F. *The perceptron: A probabilistic model for information storage and organization in the brain*. **Psychological Review**, Vol 65(6), Nov 1958; 386-408.
- [51] Coursera. *Week 1 Lecture notes* [en línea]. Stanford University. 2011. [ref. de 19 de diciembre de 2020]. Disponible en Web: <<https://www.coursera.org/learn/machine-learning/resources/JXWWS>>
- [52] Stanford University. *CS229 Lecture notes*. Adjunct Professor Andrew NG. 2016 [ref. de 15 de junio de 2017]. Disponible en Web: < <http://cs229.stanford.edu/notes/cs229-notes1.pdf>>
- [53] *Chapter 2. The Regression Line*. En; Freedman D. *Statistical Models: Theory and Practice*. Cambridge University Press. 2009; 18-28.
- [54] Corinna C., Vapnick V. *Support-Vector networks*. **Machine Learning**. Vol. 20, Issue 3, 1995; 273-297.
- [55] Rumelhart D., Hinton G., Williams R. *Learning Internal Representations by Error Propagation*. California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

- [56] Collins J., Sohl-dickstein J., Sussillo D. *Capacity and Trainability in Recurrent Neural Networks*. **arXiv** preprint arXiv:1611.09913, 2016.
- [57] Cox D. *The regression analysis of binary sequences*. **Journal of the Royal Statistical Society. Series B (Methodological)**, 1958; 215-242.
- [58] SAFAVIAN, S. Rasoul; LANDGREBE, David. A survey of decision tree classifier methodology. *IEEE transactions on systems, man, and cybernetics*, 1991, vol. 21, no 3, p. 660-674.
- [59] BREIMAN, Leo. Random forests. *Machine learning*, 2001, vol. 45, no 1, p. 5-32.
- [60] Rosasco L., et al. *Are loss functions all the same?*. *Neural Computation*, vol. 16, no 5, 2004; 1063-1076.
- [61] Cayre M., Jordane M., Scotto-Lomassese S., et al. *The common properties of neurogenesis in the adult brain: from invertebrates to vertebrates*. *Comparative Biochemistry and Physiology Part B: Biochemistry and Molecular Biology*. Volume 132, Issue 1, May 2002; 1-15.
- [62] Paniagua R., et al. *Citología e histología vegetal y animal*. McGraw-Hill Interamericana, 1998.
- [63] Yegnanarayana B. *Artificial neural networks*. PHI Learning Pvt. Ltd., 2009.
- [64] Minsky M., Papert S. *Perceptrons*. Oxford, England: **M.I.T. Press**, 1969.
- [65] Pearson K. *On lines and planes of closest fit to systems of points in space*. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 1901, vol. 2, no 11; 559-572.
- [66] LECUN, Yann, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998, vol. 86, no 11, p. 2278-2324.
- [67] ARDILA, Diego, et al. End-to-end lung cancer screening with three-dimensional deep learning on low-dose chest computed tomography. *Nature medicine*, 2019, p. 1.
- [68] RAJPURKAR, Pranav, et al. Deep learning for chest radiograph diagnosis: A retrospective comparison of the CheXNeXt algorithm to practicing radiologists. *PLoS medicine*, 2018, vol. 15, no 11, p. e1002686.
- [69] HUBEL, David H.; WIESEL, Torsten N. The period of susceptibility to the physiological effects of unilateral eye closure in kittens. *The Journal of physiology*, 1970, vol. 206, no 2, p. 419-436.
- [70] HUBEL, David H.; WIESEL, T. N. Shape and arrangement of columns in cat's striate cortex. *The Journal of physiology*, 1963, vol. 165, no 3, p. 559-568.
- [71] HUBEL, David H.; WIESEL, Torsten N. Integrative action in the cat's lateral geniculate body. *The Journal of Physiology*, 1961, vol. 155, no 2, p. 385-398.
- [72] HUBEL, David H.; WIESEL, Torsten N. Uniformity of monkey striate cortex: a parallel relationship between field size, scatter, and magnification factor. *Journal of Comparative Neurology*, 1974, vol. 158, no 3, p. 295-305.

- [73] HUBEL, David H.; WIESEL, Torsten N.; STRYKER, Michael P. Anatomical demonstration of orientation columns in macaque monkey. *Journal of Comparative Neurology*, 1978, vol. 177, no 3, p. 361-379.
- [74] WIESEL, Torsten N.; HUBEL, David H. Ordered arrangement of orientation columns in monkeys lacking visual experience. *Journal of comparative neurology*, 1974, vol. 158, no 3, p. 307-318.
- [75] HUBEL, D. H.; WIESEL, T. N. Receptive fields of optic nerve fibres in the spider monkey. *The Journal of physiology*, 1960, vol. 154, no 3, p. 572-580.
- [76] HUBEL, D. H.; WIESEL, T. N.; LEVAY, S. Functional architecture of area 17 in normal and monocularly deprived macaque monkeys. En *Cold Spring Harbor Symposia on Quantitative Biology*. Cold Spring Harbor Laboratory Press, 1976. p. 581-589.
- [77] FUKUSHIMA, Kunihiko. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 1980, vol. 36, no 4, p. 193-202.
- [78] WENG, John J.; AHUJA, Narendra; HUANG, Thomas S. Learning recognition and segmentation of 3-d objects from 2-d images. En *1993 (4th) International Conference on Computer Vision*. IEEE, 1993. p. 121-128.
- [79] LECUN, Yann, et al. LeNet-5, convolutional neural networks. URL: <http://yann.lecun.com/exdb/lenet>, 2015, vol. 20.
- [80] HANIN, Boris. Universal function approximation by deep neural nets with bounded width and relu activations. *arXiv preprint arXiv:1708.02691*, 2017.
- [81] STIGLER, Stephen M. *Francis Galton's account of the invention of correlation*. Statistical Science, 1989, p. 73-79.
- [82] LEHMAN, Ann, et al. *JMP for basic univariate and multivariate statistics*. SAS Institute Inc., Cary, NC, 2005.
- [83] STEHMAN, Stephen V. *Selecting and interpreting measures of thematic classification accuracy*. Remote sensing of Environment, 1997, vol. 62, no 1, p. 77-89.
- [84] POWERS, David Martin. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. 2011.
- [85] BALAGEAS, Daniel L. *Termografía infrarroja: una técnica multifacética para la Evaluación No Destructiva (END)*. En IV Conferencia Panamericana de END, Buenos Aires. 2007.
- [86] BYRNES, James (ed.). *Unexploded ordnance detection and mitigation*. Springer Science & Business Media, 2008.
- [87] MADHU, Himanshu, et al. *Extraction of medically interpretable features for classification of malignancy in breast thermography*. En Engineering in Medicine and Biology Society (EMBC), 2016 IEEE 38th Annual International Conference of the. IEEE, 2016. p. 1062-1065.

- [88] LAHIRI, B. B., et al. *Medical applications of infrared thermography: a review*. *Infrared Physics & Technology*, 2012, vol. 55, no 4, p. 221-235.
- [89] RING, E. F. J.; AMMER, K. *Infrared thermal imaging in medicine*. Physiological measurement, 2012, vol. 33, no 3, p. R33.
- [90] BUSHBERG, Jerrold T.; BOONE, John M. *The essential physics of medical imaging*. Lippincott Williams & Wilkins, 2011.
- [91] WEBB, Steve. *The physics of medical imaging*. CRC Press, 1988.
- [92] YAO, Xiaoli, et al. *A comparison of mammography, ultrasonography, and far-infrared thermography with pathological results in screening and early diagnosis of breast cancer*. *Asian Biomedicine*, 2014, vol. 8, no 1, p. 11-19.
- [93] PRASAD, Seetharam Shiva, et al. *Evaluation of efficacy of thermographic breast imaging in breast cancer: A pilot study*. *Breast Disease*, 2016, no Preprint, p. 1-5.
- [94] KANDLIKAR, Satish G., et al. *Infrared imaging technology for breast cancer detection—Current status, protocols and new directions*. *International Journal of Heat and Mass Transfer*, 2017, vol. 108, p. 2303-2320.
- [95] NEW, Katina M.; EINSTEIN, George P.; TULP, Orien L. *Analysis of Thermography and Breast Cancer Prevention Among Minority Women and Other Ethnic Groups*. *The FASEB Journal*, 2017, vol. 31, no 1 Supplement, p. 806.1-806.1.
- [96] GERASIMOVA, Evgeniya, et al. *Wavelet-based multifractal analysis of dynamic infrared thermograms to assist in early breast cancer diagnosis*. *Frontiers in physiology*, 2014, vol. 5, p. 176.
- [97] HANKARE, Priya, et al. *Breast Cancer Detection Using Thermography*. 2016.
- [98] PUJARI, Arun K. *Data mining techniques*. Universities press, 2001.
- [99] TAN, Mingxing; LE, Quoc V. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*, 2019.
- [100] HE, Kaiming, et al. Deep residual learning for image recognition. En *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016. p. 770-778.
- [101] SZEGEDY, Christian, et al. Going deeper with convolutions. En *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015. p. 1-9.
- [102] SZEGEDY, Christian, et al. Inception-v4, inception-resnet and the impact of residual connections on learning. *arXiv preprint arXiv:1602.07261*, 2016.
- [103] XIA, Xiaoling; XU, Cui; NAN, Bing. Inception-v3 for flower classification. En *2017 2nd International Conference on Image, Vision and Computing (ICIVC)*. IEEE, 2017. p. 783-787.
- [104] SELVARAJU, Ramprasaath R., et al. Grad-cam: Visual explanations from deep networks via gradient-based localization. En *Proceedings of the IEEE international conference on computer vision*. 2017. p. 618-626.

[105] XI, Pengcheng, et al. An integrated approach for medical abnormality detection using deep patch convolutional neural networks. *The Visual Computer*, 2019, p. 1-14.

ANEXOS