PROYECTO FINAL

CLASIFICACIÓN DE PATRONES EN TERMOGRAMAS PROVENIENTES DE LA REGIÒN PLANTAR

AUTORES:
Santiago Humberto Ramirez Martinez

PRESENTADO A:
Francisco Calderón, Ing, MSc, Ph.D

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE ELECTRÓNICA
BOGOTÁ D.C. 2021

1. Datos utilizados

1.1 Base de datos de IEEE Access

La base de datos de termogramas plantares para el estudio de las complicaciones en el pie diabético publicada en IEEE Access [1] cuenta con 334 termogramas de 122 sujetos diabéticos y 45 sujetos no diabéticos. Los sujetos fueron reclutados del Hospital General del Norte, el Hospital General del Sur, la clínica BIOCARE y el Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE), desde el año 2012 hasta el año 2014 en la ciudad de Puebla, México. En la tabla 1 se puede observar la información demográfica de la base de datos.

	Control Group	DM Group
Volunteers	45	122
Female	16	89
Male	29	33
Age(years)	27.76 ± 8.09	55.98 ± 10.57

Tabla 1. Información demográfica de la base de datos. Tomado de [1]

Cada termograma incluye un archivo CSV con la información de la temperatura y otros 8 con la información de los angiosomas. De igual forma para cada archivo mencionado anteriormente se tiene un archivo PNG con la información gráfica de las variaciones de temperatura.

1.2 Establecimiento de clases y características a extraer

A partir de los datos contenidos en la base de datos publicada en IEEE Access se establecen dos clases, una perteneciente a los termogramas provenientes de pacientes diabéticos y otra perteneciente a los termogramas provenientes de los sujetos de control. Para la extracción de la región de interés de los termogramas se utilizó un código basado en entropía difusa [2], el cual asigna un nivel de pertenencia a cada pixel entre determinados umbrales. Finalmente se definieron las características a extraer (valor de máxima entropía, valor medio, varianza, y número de pixeles) a partir de un artículo publicado en la revista Sensors [3], en el cual se realiza la clasificación de termogramas provenientes de la región plantar entre 5 clases.

2. Diagramas de bloques

El presente proyecto se divide en 2 partes principales, las cuales son la extracción de características y la clasificación de los termogramas entre 2 clases. Dentro de la extracción de características se encuentra la extracción de la ROI (región de interés) de los termogramas, a partir de la ROI se extraen el número de pixeles, el valor medio, la varianza y el valor de máxima entropía. Por otro lado, antes realizar la clasificación entre las 2 clases usando los algoritmos de máquina de soporte vectorial (SVM), Redes neuronales (ANN), K nearest neighbours (KNN), Random forest y Regresión logística, se realiza un pre procesamiento de las características aplicando una escalización y PCA (principal component analysis).

2.1 Diagrama de bloques de la extracción de características

La extracción de la región de interés se realiza por medio un algoritmo de entropía difusa y evolución diferencial [2], para posteriormente extraer las características que se observan en la figura 3 y almacenarlas en el vector asignado a cada característica en la posición asignada.

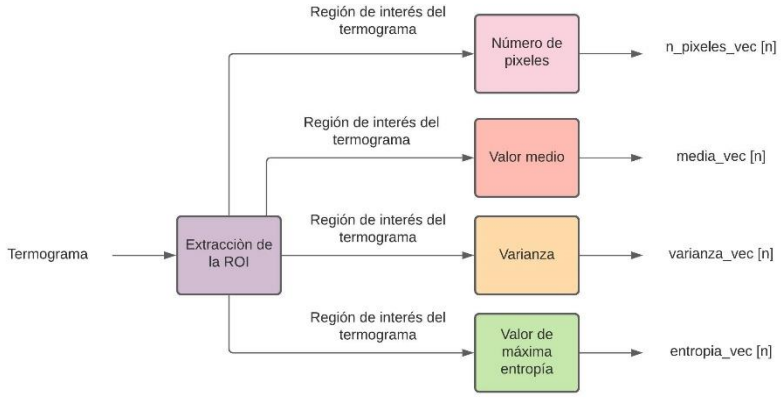

Figura 1. Diagrama de bloques de la extracción de características

1. **Extracción de la ROI (Región de interés)**
   El bloque de extracción de la ROI tiene como función la implementación de un algoritmo de entropía difusa y evolución diferencial como método de segmentación para encontrar la región de interés del termograma, con la intención de encontrar un patrón cuyas características se puedan clasificar entre las 5 clases establecidas.

2. **Número de pixeles**
   El bloque número de pixeles es el encargado de extraer la característica del número de pixeles de la imagen que contiene la ROI.

3. **Valor medio**
   El bloque valor medio es el encargado de extraer la característica del valor medio de la imagen que contiene la ROI.

4. **Varianza**
   El bloque varianza es el encargado de extraer la característica de la varianza de la imagen que contiene la ROI.

5. **Valor máximo de entropía**
   El bloque valor máximo de entropía es el encargado de extraer la característica del valor máximo de entropía de la imagen que contiene la ROI.

2.2 Diagrama de bloques de la clasificación entre 2 clases

Una vez se tiene el vector de características, la información del mismo debe ser almacenada, organizada y etiquetada. Como se observa en la figura 4, las características pasan por un procedimiento de optimización por medio de PCA (principal component analysis) y se escalizan, para posteriormente entrar a los clasificadores. Se establecieron el accuracy, la precisión, el recall y el f1-score como métricas de evaluación para todos los clasificadores.

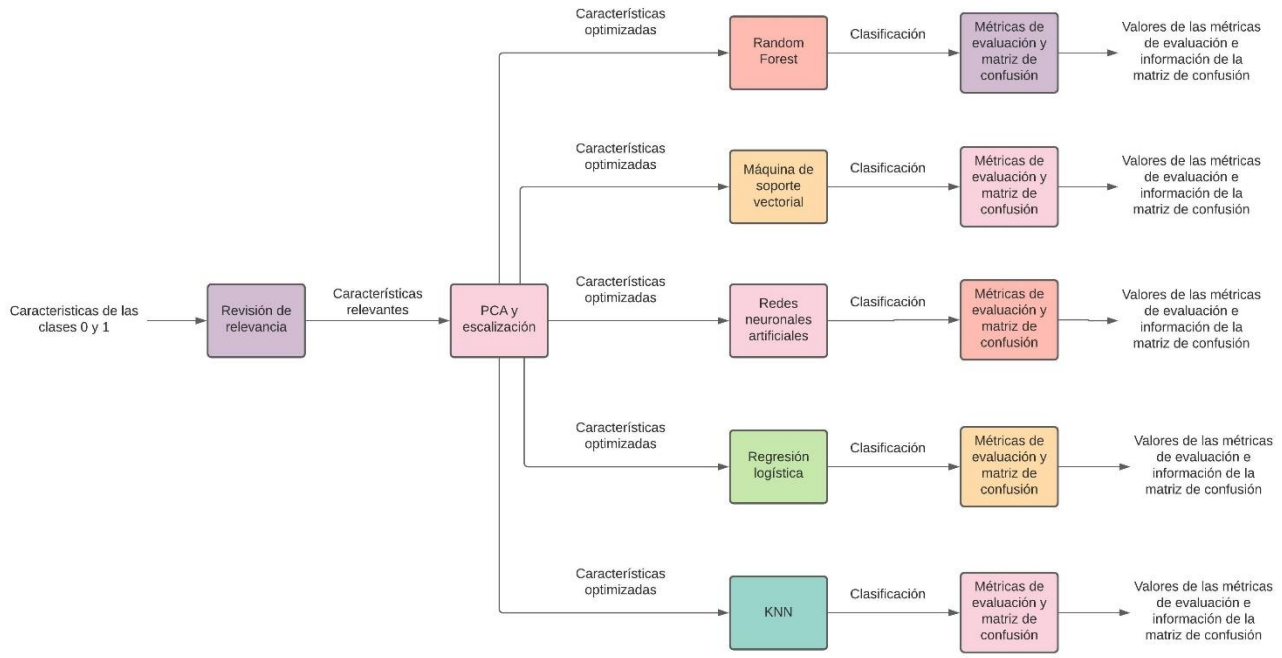

Figura 2. Diagrama de bloques de la clasificación

1. **Revisión de relevancia**
   El bloque de revisión de relevancia tiene como función establecer que características son más significativas para la clasificación a partir de valores de probabilidad.

2. **PCA y escalización**
   El bloque de PCA (principal component analysis) y escalización tiene como función hacer las características lo más significativas posibles.

3. **Random Forest**
   Bloque encargado de crear un modelo siguiendo el algoritmo Random Forest mediante el cual se clasifican las características de la ROI de los termogramas en 5 clases.

4. **Máquina de soporte vectorial**
   Bloque encargado de crear un modelo siguiendo el algoritmo de support vector machine mediante el cual se clasifican las características de la ROI de los termogramas en 5 clases.

5. **Redes neuronales artificiales**
   Bloque encargado de crear un modelo a partir de redes neuronales artificiales mediante el cual se clasifican las características de la ROI de los termogramas en 5 clases.

6. **Regresión logística**
   Bloque encargado de crear un modelo siguiendo el algoritmo de regresión logística mediante el cual se clasifican las características de la ROI de los termogramas en 5 clases.

7. **KNN**
   Bloque encargado de crear un modelo siguiendo el algoritmo KNN (k nearest neighbours) mediante el cual se clasifican las características de la ROI de los termogramas en 5 clases.

8. **Métricas de evaluación y matriz de confusión**

Bloque encargado de implementar las métricas de evaluación accuracy, precisión, recall y f1-score y de extraer las matrices de confusión para cada uno de los clasificadores.

3. Diagrama de flujo

En la figura 5 se puede observar un diagrama de flujo de alto nivel del código a implementar en el cual se utiliza el set de datos extraído de los termogramas de la base de datos de IEEE Access para el entrenamiento y prueba de los clasificadores.

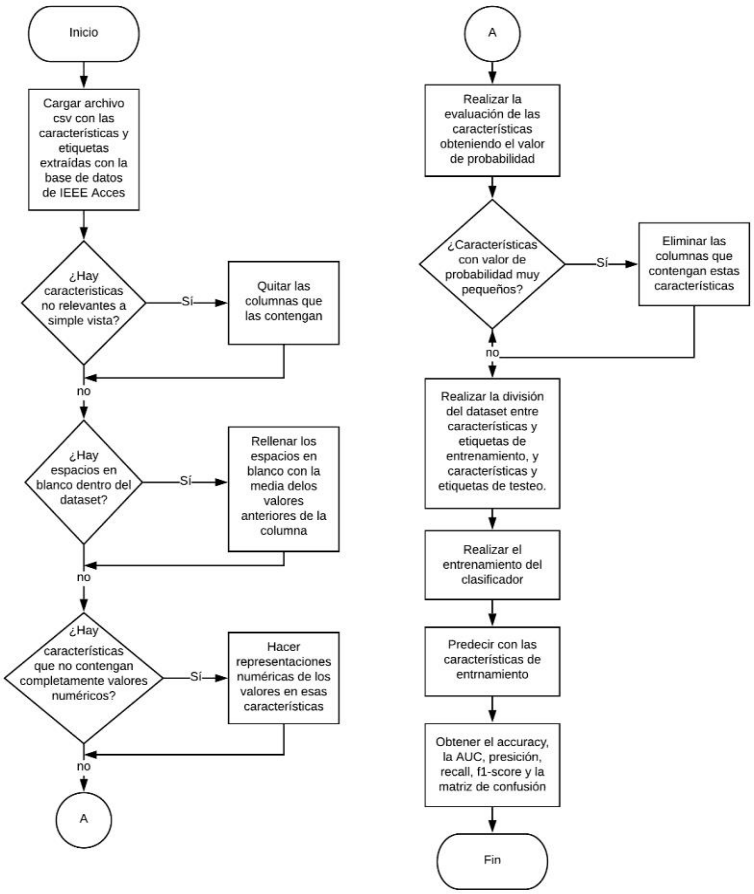

Figura 3. Diagrama de flujo de alto nivel del sistema.

4. Marco teórico

4.1 Inteligencia artificial

La inteligencia artificial se define como un conjunto de técnicas que permiten a un computador imitar funciones humanas [4]. El término inteligencia artificial fue acuñado formalmente en 1956 durante la conferencia de Dartmouth. Existen dos grandes grupos de inteligencia artificial, los cuales son inteligencia artificial general e inteligencia artificial débil. Dentro de la inteligencia artificial débil se encuentran las aplicaciones que se restringen a una sola tarea en específico, mientras que en la inteligencia artificial general se encuentran las máquinas que experimentan conciencia, las cuales por ahora solo están en la ciencia ficción [5].

Las técnicas de inteligencia artificial débil se dividen en varias categorías, tales como aprendizaje de máquina, visión por computador, inteligencia computacional y sistemas expertos. Estas categorías a su vez se dividen en

subcategorías, tales como sistemas supervisados, no supervisados y por refuerzo en el caso del aprendizaje de máquina. La rama de inteligencia computacional puede dividirse en dos, teniendo por un lado las redes neuronales, y por el otro los algoritmos evolutivos. Finalmente, los sistemas expertos pueden ser basados en reglas, redes bayesianas o en casos [6].

4.2 Aprendizaje de máquina y algunos métodos

En el aprendizaje de máquina los sistemas supervisados son aquellos que crean un modelo matemático que busca explicar unas etiquetas de entrada o salida a partir de un conjunto de características de entrada. Estos sistemas se pueden dividir principalmente en sistemas supervisados de clasificación y sistemas supervisados de regresión. De igual forma existen otros sub-métodos tales como aprendizaje activo, "similarity learing" y "recomended systems". Por otro lado, los sistemas no supervisados son aquellos que crean un modelo que busca explicar las características de entrada sin contar con etiquetas. Se pueden dividir en agrupamiento, estimación de densidad y reducción dimensional [7].

4.3.1 Support Vector Machine

Support vector machine (SVM), en español "máquina de soporte vectorial", es uno de los métodos de machine learning supervisados más utilizados en áreas tales como el procesamiento de señales, aplicaciones médicas, reconocimiento de imágenes entre otros. Las SVM funcionan encontrando un hiperplano optimo que separe las clases. Para encontrar este hiperplano es necesario que el problema sea linealmente separable. En la figura 6 se puede observar el hiperplano separando las clases más (+) y menos (-). Además de lo anterior se puede observar el margen, el cual indica el ancho máximo paralelo al hiperplano que no contiene datos y es el parámetro que debe ser optimizado. Finalmente, los vectores de soporte son los puntos que conforman las dos líneas paralelas al hiperplano, siendo la margen la mayor posible. [8]

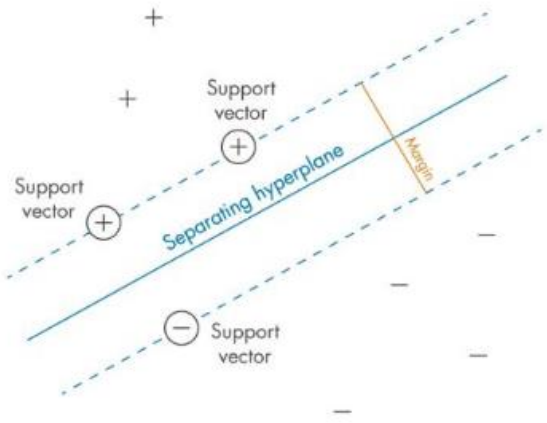

Figura 4. Componentes del método SVM. Tomado de [8]

4.3.2 K Nearest Neighbors

Otro de los algoritmos ampliamente usados perteneciente al grupo de aprendizaje supervisado es el K Nearest Neighbors (KNN), en español "k vecinos más cercanos", el cual clasifica cada nuevo dato calculando la distancia del mismo con todos los del conjunto de entrenamiento. La predicción se hace en base a las clases más cercanas al nuevo dato, y el rango de comparación está dado por K, es decir, este valor determina cuantos vecinos se deben tener en cuenta. Para encontrar el valor K la técnica más usada es llamada cross validation, la cual consiste en partir el conjunto de entrenamiento y comparar varios valores de k pudiendo así establecer cuál es el valor que más se acomoda al conjunto de entrenamiento [9].

En la Figura 7 se puede observar un ejemplo gráfico donde se toman 2 valores con el objetivo de clasificar el circulo verde, si se toma el K de la línea continua el circulo será clasificado como un triángulo. Por el contrario,

si se toma el K de la línea discontinua, el circulo se clasificará como cuadrado, debido a que hay 2 triángulos y tres cuadrados.

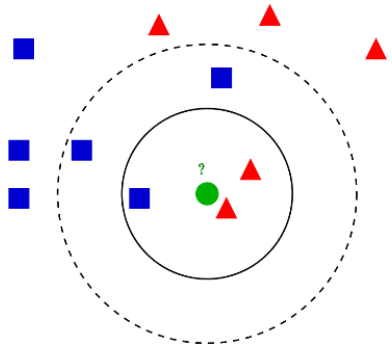

Figura 5. Ejemplo gráfico de clasificación usando KNN adoptando dos valores de K. Tomado de [10]

4.3.3 Regresión logística

La regresión logística es un método supervisado de los más utilizados en el área de machine learning y es idóneo para clasificación binaria. Las características en la regresión logística pueden están en R^n, mientras que las etiquetas son 1 y 0. Se utiliza la función logística, la cual se ilustra en la figura 8, para determina la probabilidad de la variable dependiente, y se establece un valor umbral de discriminación entre las dos clases. En términos simples, se busca es establecer una probabilidad para un conjunto de características, dónde los resultados son probabilidades de pertenecer a la clase 1 o la clase 0, las características se clasifican a la clase con mayor probabilidad. La descripción matemática de la función logística se puede observar en la ecuación 1, dónde x es una combinación lineal entre las características y los pesos de entrada [11] .

$$y = \frac{1}{1+e^{-x}} \quad \text{Ecuación 1}$$

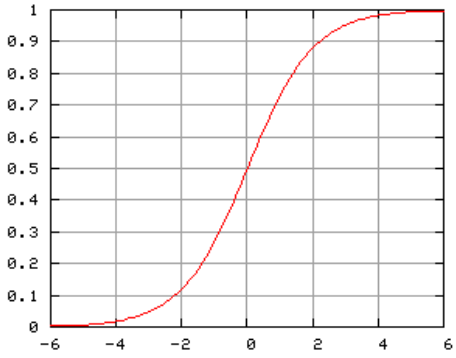

Figura 6. Función logística. Tomado de [11]

4.3.4 Random forest

Es un método de clasificación que parte del concepto de los árboles de decisión, dónde se establecen una serie de relaciones entre las características y las etiquetas con el fin de determinar la impureza de las características, medida por medio del coeficiente de impuridad de Gini. La característica con menor coeficiente Gini se convierte en el nodo raíz del árbol, y posteriormente se recalcula Gini para el resto de características, construyendo así un árbol. En Random Forest este mismo procedimiento se repite seleccionando datos aleatoriamente, creando una cantidad de set de datos aleatorios, de los cuales se crean arboles de decisión[12]. La clasificación de un dato nuevo se escoge de acuerdo a la votación de cada uno de los arboles creados, en la figura 9 se puede observar la arquitectura del algoritmo Random Forest.

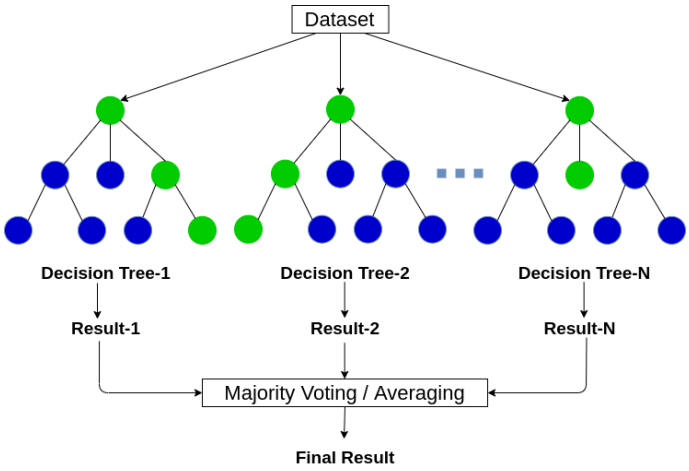

Figura 7. Arquitectura del método de clasificación Random Forest. Tomado de [13]

4.3 Inteligencia computacional y algunos métodos

4.4.1 Artificial Neural Networks

En la inteligencia computacional se utilizan elementos de aprendizaje, adaptación evolución y lógica difusa con el objetivo de crear programas inteligentes, es decir, capaces de tomar buenas decisiones. Las redes neuronales artificiales, en inglés artificial neural networks (ANN), son una de las herramientas prácticas de la inteligencia computacional, las cuales constan de una capa de entrada de neuronas, de una a tres capas escondidas de neuronas y una capa final de salida de neuronas. Las neuronas de entrada se conectan con las capas escondidas, y estas a su vez se conectan con la capa de salida como se muestra en la figura 10 [14].

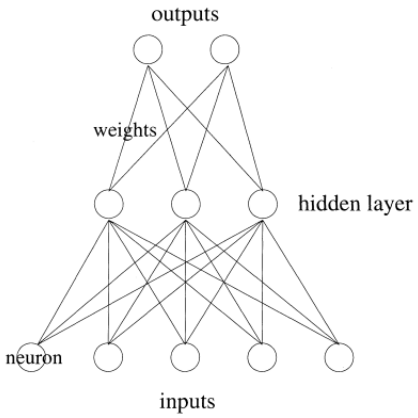

Figura 8. Estructura general de una red neuronal artificial. Tomado de [14]

Cada conexión existente se asocia con un valor numérico que indica un peso, las entradas se multiplican por el peso asociado a ellas y se suman, para luego usar una función de activación que tiene como tarea acotar los valores de salida del sistema para mantenerlos en ciertos rangos, regulando las neuronas divergentes. La ecuación 2 muestra la salida de una neurona en la capa escondida, dónde h_i es la salida, $\sigma()$ es la función de activación, N el número de neuronas de entrada, V_{ij} el peso, x_j la entrada de las neuronas de entrada y T_i^{hid} el umbral de las neuronas ocultas.

$$h_i = \sigma(\sum_{J=1}^{N} V_{ij} x_j + T_i^{hid}) \qquad \text{Ecuación 2}$$

Existen varios tipos de redes neuronales, tales como el perceptrón multicapa, las redes neuronales convolucionales, y las redes neuronales probabilísticas. Estas redes neuronales se clasifican dependiendo de su tipo de conexión y número de capas [14]. El perceptrón multicapa es una generalización de las redes neuronales mono capa

observadas en la Figura 6. En esta clasificación se tienen 2 o más capas ocultas entre la capa de entrada y la capa de salida, de igual manera, dependiendo del número de conexiones, el perceptrón multicapa puede clasificarse entre parcial o totalmente conectado.

4.5 Algoritmo de extracción de la región de interés

El algoritmo de extracción de la región de interés de los termograma se basa en definir umbrales multinivel basados en entropía difusa utilizando evolución diferencial. Para un correcto entendimiento de este algoritmo se debe realizar una descripción matemática de la entropía difusa multinivel [15].

A. Entropía difusa multinivel

La entropía de Shannon se define como la cantidad de información que es dada por una variable aleatoria, y se define matemáticamente como se observa en la ecuación 3, dónde p_i es la probabilidad de la variable. La entropía puede ser dada en bits, nat o hartleys, dependiendo de la base del logaritmo.

$$H(P) = -\sum_{i=1}^{n} p_i \log p_i \quad \text{Ecuación 3}$$

Para una imagen, i debe ser tomado con un tamaño de 8 bits, en escala de grises y con unas dimensiones determinadas. Para encontrar la entropía para cada uno de los t (n-1) umbrales se debe seguir la descripción matemática que se observa en las ecuaciones 4, 5 y 6, dónde P es el histograma normalizado, n es el número de clases y $L = 256$ niveles de gris.

$$H_1(t) = -\sum_{i=0}^{t_1} \frac{p_i}{P_1} \log \frac{p_i}{P_1} \quad \text{Ecuación 4}$$

$$H_2(t) = -\sum_{i=t_1+1}^{t_2} \frac{p_i}{P_2} \log \frac{p_i}{P_2} \quad \text{Ecuación 5}$$

$$H_n(t) = -\sum_{i=t_1+1}^{L-1} \frac{p_i}{P_n} \log \frac{p_i}{P_n} \quad \text{Ecuación 6}$$

Para encontrar P_1, P_2,…, P_n se deben seguir las ecuaciones 7, 8 y 9

$$P_1(t) = -\sum_{i=0}^{t1} p_i \quad \text{Ecuación 7}$$

$$P_2(t) = -\sum_{i=t1+1}^{t2} p_i \quad \text{Ecuación 8}$$

$$P_n(t) = -\sum_{i=t_{n-1}+1}^{L-1} p_i \quad \text{Ecuación 9}$$

Una vez se tiene la entropía para cada uno de los umbrales, se debe encontrar el argumento máximo de la suma de las entropías con el objetivo de encontrar el umbral óptimo, como se muestra en la ecuación 10.

$$\varphi(t_1, t_2, \dots, t_n) = Arg \max([H_1(t) + H_2(t) + \cdots + H_n(t)]). \quad \text{Ecuación 10}$$

Cuando se tiene un conjunto de elementos estos pueden pertenecer o no pertenecer a este, en un conjunto difuso, los elementos pueden pertenecer parcialmente. La pertenencia a un conjunto está dado por la función de pertenencia, denotada como $\mu_n(k)$, la cual puede tomar valores entre 0 y 1. En la ecuación 11 se puede observar la descripción matemática de un conjunto difuso, dónde A es el conjunto y x es uno de los elementos.

$$A = \{(x, \mu_A(x)) | x \in X\} \quad \text{Ecuación 11}$$

La pertenencia de los elementos a un conjunto se representa en gráficas de trapecio como se observa en la figura 11, dónde por cada uno de los umbrales se necesitan 2 parámetros difusos en pares a y c, desde a_1 y c_1, hasta a_n y c_n, dónde $0 \leq a_1 \leq c_1 \leq \cdots \leq a_{n-1} \leq c_{n-1} \leq L-1$.

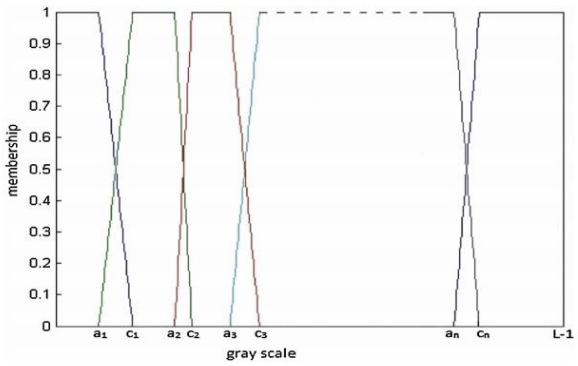

Figura 9. Función de pertenencia a un conjunto. Tomada de [15]

La función de pertenencia se describe matemáticamente como se muestra en la ecuación 12, donde se puede observar la relación de cada uno de los parámetros difusos con k.

$$\mu_n(k) = \begin{cases} 1 & k \leq a_{n-1} \\ \frac{k-a_n}{c_n-a_n} & a_{n-1} < k \leq c_{n-1} \\ 1 & k > c_{n-1} \end{cases} \quad \text{Ecuación 12}$$

Finalmente, la entropía difusa puede ser descrita como se muestra en la ecuación 13.

$$H_n(t) = -\sum_{i=0}^{L-1} \frac{p_i * \mu_n(i)}{P_n} * \ln\left(\frac{p_i * \mu_n(i)}{P_n}\right)$$

5. Resultados preliminares

5.1 Revisión de la base de datos

La base de datos publicada en IEEE Access [16] cuenta con datos etiquetados entre sujetos no diabéticos y pacientes diabéticos. Se realizó una revisión con el objetivo de verificar que todos los datos estuvieran etiquetados de forma correcta, encontrando una serie de posibles inconsistencias en 17 termogramas etiquetados como pacientes diabéticos. Los 17 termogramas encontrados fueron eliminados con el fin de tener la mejor calidad de datos posible. Los datos de etiquetaron en dos clases, siendo la clase 0 referente a los termogramas de sujetos de control y la clase 1 referente a los pacientes diabéticos.

5.2 Obtención de la región de interés

La obtención de la región de interés es un módulo fundamental dado que se busca clasificar el patrón del termograma. En la figura 12 se puede observar el termograma de un sujeto de control el cual presenta el patrón de mariposa, considerado el patrón normal, mientras que en la figura 13 se puede observar como no hay un patrón definido en el termograma, siendo este un patrón alterado. Para la obtención de la región de interés se realiza un procesamiento al termograma, primero pasándolo a escala de grises y posteriormente aplicando un algoritmo de segmentación basado entropía difusa y evolución diferencial [2], [15], en el cual los pixeles adquieren probabilidades asignadas según que tanto pertenecen a un umbral, estas probabilidades son valores entre 0 y 1. El

número de umbrales se estableció en 2 para todos los termogramas y el algoritmo usado está implementado en Matlab.

Figura 10. Extracción del patrón del termograma de un sujeto no diabético.

Figura 11. Extracción del patrón del termograma de un sujeto diabético.

5.3 Extracción de características

A partir de la región de interés, se extrajeron 4 características para todos los termogramas documentadas en el estudio publicado en la revista Sensors [3], las cuales son la media, la varianza, el número de pixeles y la entropía máxima, como se observa en la tabla 3. Se realizó una calificación de la relevancia de las características por medio del análisis de la varianza, teniendo como resultado los valores "F-statistic" (Score) y los valores de probabilidad asociados a estos (P_Value). La característica con mayor calificación fue la varianza, con valores de 5.204 y 21.291 para el Score y el P_Value respectivamente. En contraste, la característica con menor calificación fue la entropía, con valores de 0.467 y 0.907. En la tabla 2 se puede observar la calificación para todas las características.

Feature	P_Value	Score
Varianza	5.240	21.291
Media	4.786	19.159
n_pixeles	3.288	12.312
Entropia	0.467	0.907

Tabla 2. Calificación de las características extraídas de la región de interés.

	Media	Varianza	Entropia	n_pixeles	Clase
0	84.206044	14381.883370	0.915151	10818	0
1	106.718367	15824.373744	0.980750	13842	0
2	55.587645	6786.256082	1.213999	12957	0
3	56.010202	6197.980464	1.204194	13755	0
4	51.286727	4627.608295	1.076427	15900	0
...	...	...	...	...	...
312	103.654971	15687.664581	0.974621	10635	1
313	73.519833	6811.780821	1.328999	13002	1
314	95.200000	15212.960000	0.953197	6552	1
315	101.012903	15554.683704	0.968641	7368	1
316	60.057566	6301.142930	1.236328	11373	1

Tabla 3. Set de datos a usar en la clasificación

5.4 Prueba de algoritmos en Python

Se implementaron los algoritmos de máquina de soporte vectorial, ANN (redes neuronales artificiales), regresión logística, KNN (K Nearest Neighbours) y Random Forest, utilizando el 30% del set de datos como set de prueba y el 70% como set de entrenamiento, con los datos no escalizados, en las figuras 14 y 15 se puede observar el conjunto de entrenamiento y de prueba con respecto a las características media, varianza y número de pixeles. Al implementar la máquina de soporte vectorial se tuvo en cuenta un kernel cuadrático, mientras que para las redes neuronales artificiales se utilizaron 4 neuronas de entrada y una capa oculta con 10 neuronas. Por otro lado, para el algoritmo KNN se utilizaron como parámetros 8 vecinos, pesos uniformes, y distancia euclidiana. Finalmente, para el algoritmo de Random Forest se utilizaron 100 árboles y una máxima profundidad en cada árbol de 5. Los algoritmos fueron probados utilizando la escalización de las características y aplicando PCA.

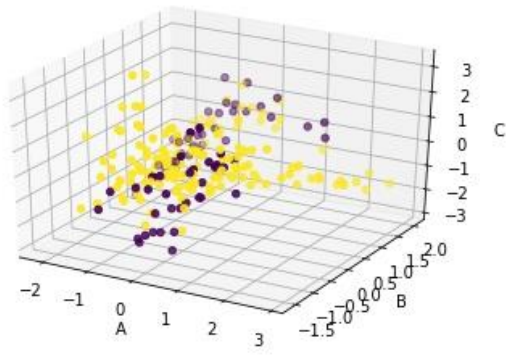

Figura 12. Conjunto de datos de entrenamiento.

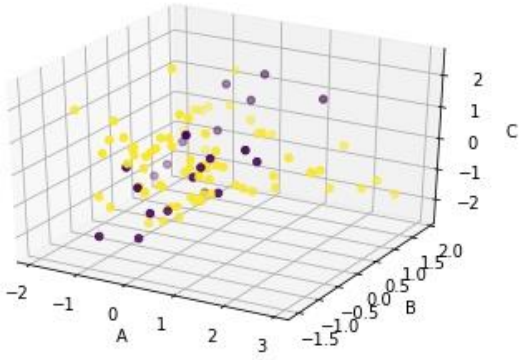

Figura 13. Conjunto de datos de prueba.

De cada uno de los algoritmos se extrajeron las métricas de accuracy, AUC (área bajo la curva) basada en la ROC (curva de característica operativa del recepto) y las matrices de confusión. En la tabla 4 se listan los valores para cada uno de los algoritmos implementados. Los resultados más adecuados se obtuvieron implementando el algoritmo Random forest, dónde se obtuvo un accuracy de 0.80 y una AUC de 0.75.

Algoritmo	**Accuracy**	**AUC**
Máquina de soporte vectorial	0.75	0.66
ANN	0.75	0.71
Regresión logística	0.75	0.65
KNN	0.63	0.73
Random forest	0.80	0.75

Tabla 4. Valores de las métricas de valuación para cada algoritmo implementado.

Las matrices de confusión para cada uno de los algoritmos se pueden observar de la figura 16 a la 20, dónde se pueden apreciar las etiquetas predichas y las etiquetas verdaderas en cada clasificador. La clase 0 se representa con 0, mientras que la clase 1 se representa con 1.

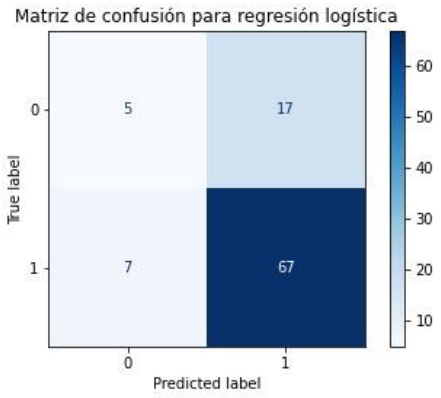

Figura 14. Matriz de confusión para regresión logística

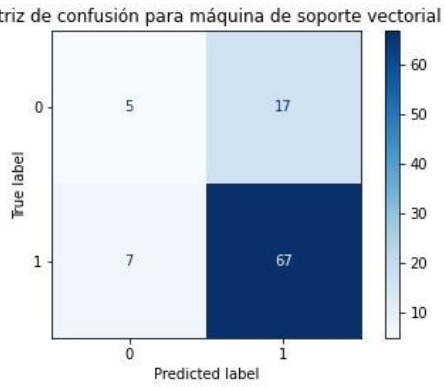

Figura 15. Matriz de confusión para la máquina de soporte vectorial

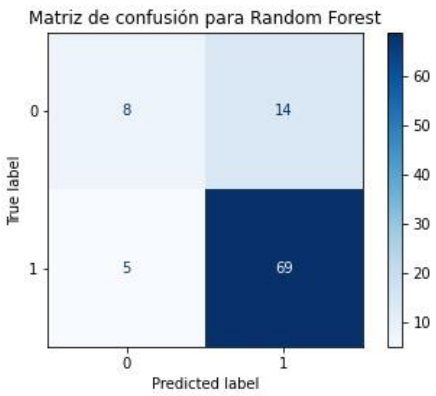

Figura 16. Matriz de confusión para Random forest

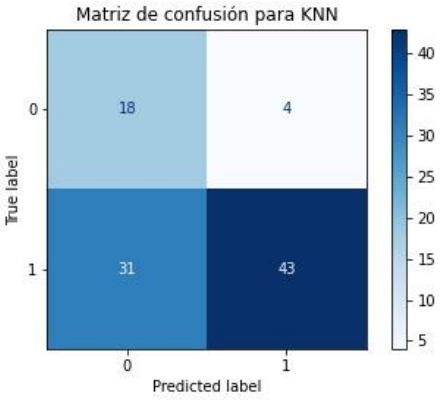

Figura 17. Matriz de confusión para KNN

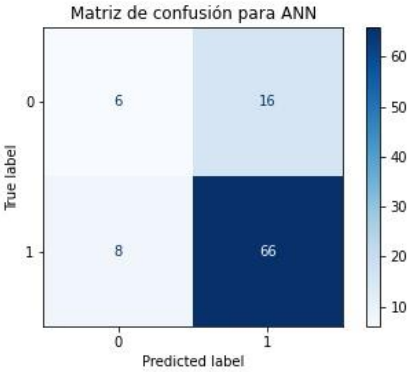

Figura 18. Matriz de confusión para ANN

Se establecieron otras métricas de evaluación para los clasificadores las cuales tienen que ver con los verdaderos positivos, falsos positivos y falsos negativos observados en la matriz de confusión. En la tabla 5 se listan los resultados para cada uno de los clasificadores, dónde se evidencia que todos obtienen mejor resultado clasificando la clase 0 (región de interés de sujetos no diabéticos), con valores mayores a 0.58 en las métricas establecidas.

Clases	**Clasificador**	**Precisión**	**Recall**	**f1-score**
	Máquina de soporte vectorial			
Clase 0		0.42	0.23	0.29
Clase 1		0.80	0.91	0.85
	Regresión logística			
Clase 0		0.42	0.23	0.29
Clase 1		0.80	0.91	0.85
	KNN			
Clase 0		0.37	0.82	0.51
Clase 1		0.91	0.58	0.71
	ANN			
Clase 0		0.43	0.27	0.33

Clase 1	0.80	0.89	0.85
Random forest			
Clase 0	0.67	0.36	0.47
Clase 1	0.83	0.95	0.89

Tabla 5. Métricas de evaluación de la clasificación para todos los algoritmos

5.5 Prueba de algoritmos en Matlab

Se realizó una prueba en Matlab utilizando la herramienta Classification Learner, en la cual se probaron todos los algoritmos disponibles con el objetivo de extraer los 5 clasificadores con el accuracy y AUC más adecuados. Para esta prueba se utilizó validación cruzada (cross validation) para obtener un mejor resultado mejorando el sesgo de selección entre clases, de igual forma se implementó PCA en las características. En la tabla 6 se listan los resultados obtenidos, teniendo como el algoritmo más adecuado KNN con un accuracy de 0.78 y una AUC de 0.81.

Algoritmo	**Accuracy**	**AUC**
KNN (Medium KNN)	0.78	0.81
Máquina de soporte vectorial (Cubic SVM)	0.77	0.80
Boosted trees	0.77	0.76
ANN (Wilde neural network)	0.76	0.71
Regresión logística	0.74	0.70

Tabla 6. Valores de accuracy y AUC de los mejores clasificadores arrojados por la herramienta Classification Learner.

Las matrices de confusión para cada uno de los algoritmos se pueden observar de la figura 21 a la 25, dónde se pueden apreciar las etiquetas predichas y las etiquetas verdaderas en cada clasificador. La clase 0 se representa con 0, mientras que la clase 1 se representa con 1.

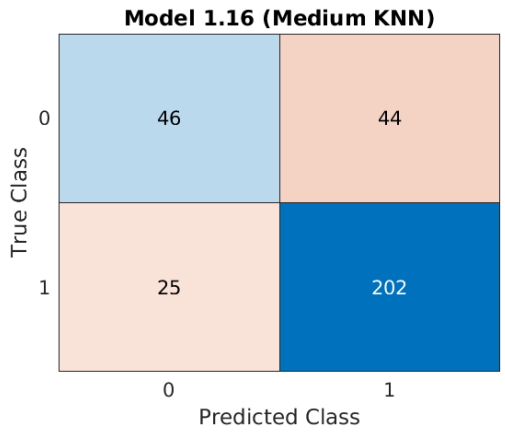

Figura 19. Matriz de confusión para el clasificador KNN

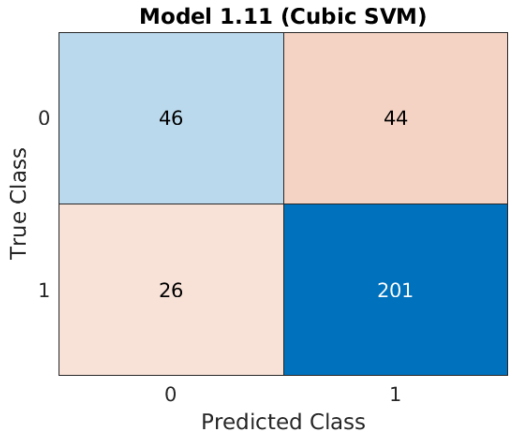

Figura 22. Matriz de confusión para SVM

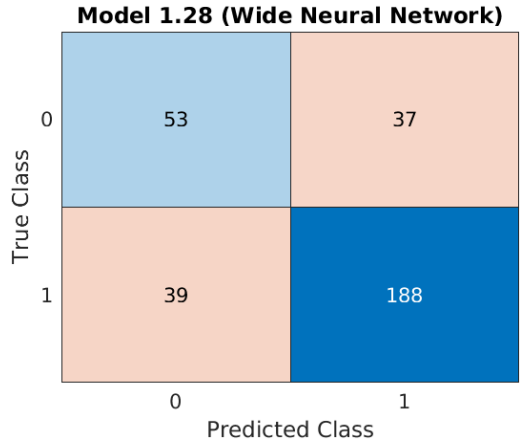

Figura 23. Matriz de confusión para ANN.

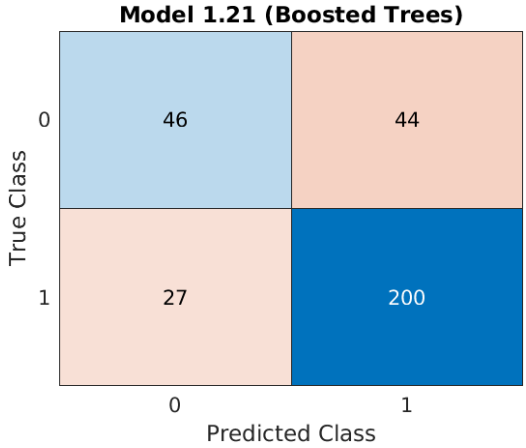

Figura 24. Matriz de confusión para Boosted Trees.

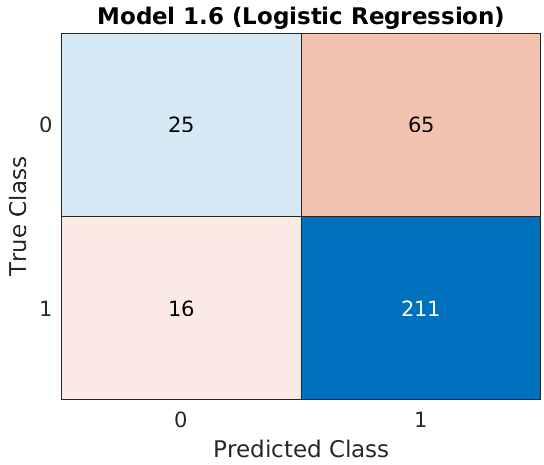

Figura 25. Matriz de confusión para regresión logística

6. Conclusiones

1. En los resultados de las métricas de evaluación precisión, recall y f1-score aplicadas a los clasificadores implementados en Python se muestra un sesgo de la clasificación de los datos hacia la clase 1, lo cual se debe a que se cuenta con un set de datos desbalanceado con 90 termogramas etiquetados con la clase 0 en contra de 227 etiquetados con la clase 1. Esto puede ser mejorado aplicando validación cruzada o aumentando la cantidad de datos la base de datos de termogramas para la clase 0 por medio de técnicas de procesamiento de imágenes.

2. En trabajos futuros se extraerán de los termogramas características relevantes que tengan que ver no solo con la forma del patrón de distribución de temperatura, sino también con los valores en grados centígrados de dicha distribución. Además, se realizará la división de la base de datos en 5 o más clases para lograr una mejor clasificación.

7. Referencias

[1] D. A. Hernandez-Contreras, H. Peregrina-Barreto, J. D. J. Rangel-Magdaleno, and F. J. Renero-Carrillo, “Plantar Thermogram Database for the Study of Diabetic Foot Complications,” *IEEE Access*, vol. 7, pp. 161296–161307, 2019, doi: 10.1109/ACCESS.2019.2951356.

[2] Sujoy Paul, "A Fuzzy Entropy based Multi-level Image Thresholding using Differential Evolution," *https://la.mathworks.com/matlabcentral/fileexchange/48055-a-fuzzy-entropy-based-multi-level-image-thresholding-using-differential-evolution*. .

[3] I. Cruz-Vega, D. Hernandez-Contreras, H. Peregrina-Barreto, J. J. Rangel-Magdaleno, and J. M. Ramirez-Cortes, "Deep learning classification for diabetic foot thermograms," *Sensors (Switzerland)*, vol. 20, no. 6, 2020, doi: 10.3390/s20061762.

[4] B. Pérez Orozco, "Inteligencia artificial," *Foro Consult. Científico y Tecnológico*, vol. 012, Mar. 2018.

[5] R. López De Mántaras, "¿Hacia una nueva Ilustración? Una década trascendente El futuro de la IA: hacia inteligencias artificiales realmente inteligentes."

[6] H. A. Banda Gamboa, "INTELIGENCIA ARTIFICIAL PRINCIPIOS Y APLICACIONES," 2014.

[7] A. Ramiro and V. Alvarado, "Machine Learning para Todos," doi: 10.13140/RG.2.2.13786.70086.

[8] "Support Vector Machine (SVM) - MATLAB & Simulink." https://la.mathworks.com/discovery/support-vector-machine.html (accessed Feb. 25, 2021).

[9] G. B. Arbeloa, "Implementación del algoritmo de los kvecinos más cercanos (k-NN) yestimación del mejor valor local de kpara su cálculo." [Online]. Available: http://ieeexplore.org/document/7368188/.

[10] Antti Ajanki, "Example of k-nearest neighbour classification." May 28, 2007.

[11] D. W. Hosmer and S. Lemeshow, *Applied Logistic Regression*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2000.

[12] "¿Qué es Decision Tree y Random Forest?" https://www.youtube.com/watch?v=tYPi6qcCQbg (accessed May 10, 2021).

[13] "Decision Tree vs. Random Forest - Which Algorithm Should you Use?" https://www.analyticsvidhya.com/blog/2020/05/decision-tree-vs-random-forest-algorithm/ (accessed May 21, 2021).

[14] S.-C. Wang, "Artificial Neural Network," in *Interdisciplinary Computing in Java Programming*, Boston, MA: Springer US, 2003.

[15] S. Sarkar, S. Paul, R. Burman, S. Das, and S. S. Chaudhuri, "A fuzzy entropy based multi-level image thresholding using differential evolution," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2015, vol. 8947, pp. 386–395, doi: 10.1007/978-3-319-20294-5_34.

[16] D. A. Hernandez-Contreras, H. Peregrina-Barreto, J. de J. Rangel-Magdaleno, and F. J. Renero-Carrillo, "Plantar Thermogram Database for the Study of Diabetic Foot Complications," *IEEE Access*, vol. 7, p., 2019, doi: 10.1109/ACCESS.2019.2951356.