

# Enfrentando el Melanoma: Un Estudio Comparativo entre Redes Neuronales Convolucionales y k-nearest neighbors



## Colaboración

Jaime Jalomo Cuevas; José Alexander Sánchez Vargas; Citlali Navarrete Cova; Daniel Fajardo Delgado; Favio Rey Lúa Madrigal, Tecnológico Nacional de México / Instituto Tecnológico de Ciudad Guzmán

Fecha de recepción: 12 de marzo 2024

Fecha de aceptación: 21 de junio de 2024

**RESUMEN:** En el presente trabajo se diseñó e implementó un sistema de red neuronal para poder prevenir y detectar el melanoma maligno a tiempo, con el objetivo de generar una herramienta para el sector salud, se enfoca en la aplicación y comparación de información de datos presentes en imágenes. Además de comparar metodologías como redes neuronales convolucionales y k-vecinos más cercanos, con el fin de entender que método se adapta mejor a este tipo de problema. Se optó por el desarrollo de este proyecto debido a los costos elevados que tiene los equipos de análisis médico en las marcas comerciales para la detección del melanoma, así como la escasez de desarrollos tecnológicos de código abierto para estudios de investigación de Inteligencia Artificial aplicado a este tipo de cáncer, además del alto índice de incidencia que tiene esta enfermedad de carácter cutáneo.

En este estudio, se empleó tecnología de bajo costo como Raspberry Pi y cámaras de alta definición para la extracción de conjuntos de imágenes de prueba para la clasificación en tiempo real, mediante el uso del lenguaje de programación Python. Con la generación de esta herramienta se pretende impactar en la independencia tecnológica de nuestro país con respecto de la tecnología del extranjero.

**PALABRAS CLAVE:** Clasificación, Diagnóstico, k-NN, Melanoma, Redes Neuronales.

**ABSTRACT:** In this work, a neural network system was designed and implemented to prevent and detect malignant melanoma in time, with the aim of generating a tool for the health sector, focused on the application and comparison of data information present in images. In addition to comparing methodologies such as convolutional neural networks and k-Nearest Neighbors (kNN) in order to understand which method best adapts to this type of problem.

The development of this project was chosen due to the high costs of medical analysis equipment in commercial brands for the detection of melanoma, as well as the cases of open source technological developments for Artificial Intelligence research studies. applied to this type of cancer, in addition to the high incidence rate of this skin disease.

For this study, it was decided to use low-cost technology such as Raspberry Pi and high-definition cameras to extract sets of test images for real-time classification, using the Python programming language. The generation of this tool is intended to impact the technological independence of our country with respect to foreign technology.

**KEYWORDS:** Classification, Diagnosis, k-NN, Melanoma, Neural Networks.

## INTRODUCCIÓN

Como antecedente, es importante saber que el melanoma es la transformación maligna del melanocito y es responsable del 80% de muertes por cáncer de piel a nivel mundial

[1,12]. La mayor parte de la población que va del 65% al 97%, presenta esta anomalía en la piel, la causa es por la exposición al sol [18,16]. La exposición prolongada a los rayos UV es la causa de este fenómeno en la piel, ya que penetra la superficie afectando la epidermis en donde daña directamente el ADN celular [23,1, 13].

Cabe destacar que el factor que tiene más importancia es el grosor tumoral, es por eso que el diagnóstico precoz es muy importante para la supervivencia de los pacientes que se encuentren infectados [25,19].

Podemos decir que un programa de inteligencia artificial (IA), manipula informaciones simbólicas bajo conceptos de objetos o reglas [10]. La medicina es otro de los campos que podrían beneficiarse con la interacción de la IA para optimizar procesos complejos e imperfectos [15, 6, 24]. Los algoritmos que se utilizan para la IA son especialmente útiles para tratar gran cantidad de datos en particular de naturaleza compleja [4, 22].

En este trabajo se buscó la comparación de dos métodos que van de lo general a lo particular el primero k-NN y el otro las redes neuronales convolucionales (RNC).

El algoritmo k-NN es una técnica muy utilizada en el reconocimiento de información, está dada por un conjunto de clases en donde una muestra se clasificará en una clase dependiendo de si la distancia a la muestra es la mínima [17, 20]. k-NN es uno de los algoritmos más reconocidos en investigación de clasificación de datos [7], sin embargo, es ineficiente para la resolución de los conjuntos de datos de gran volumen pues tiene que escanear todas las categorías de los conjuntos para lograr la clasificación.

Este algoritmo no es considerado como inteligente debido al rendimiento pues la precisión puede ser deficiente y con un alta coste computacional [14, 21].

### A continuación, el algoritmo de k-NN

k-NN se basa en la suposición de que la clase del patrón a etiquetar,  $x$  es la del prototipo más cercano  $R$  al que se denotó como  $XNN$  [3].

Para la mejora de rendimiento de estos tipos de sistemas cuyos parámetros son las imágenes, el diseño de un sistema de iluminación adecuado es importante ya que sean presentado fallas debido a la proyección de sombras [11].

Las RNC son un tipo de red neuronal artificial que imita el cortex visual humano para la identificación de características, debido a esto contiene varias capas ocultas especializadas y con jerarquía. Dependiendo de la cantidad de píxeles a utilizar y los canales a usar, es la cantidad de neuronas que utilizará nuestro modelo [5]

La teoría de la señales ofrece herramientas para analizar el comportamiento mediante estímulos de entrada. Se dice que  $S$  es un sistema que cumple con propiedades de linealidad e invarianza en el tiempo, si se verifican las siguientes propiedades:

Podemos obtener la respuesta del sistema con un impulso unitario centrado en el cero [8]. La convolución es una operación de productos y sumas entre la imagen de entrada y un filtro generando un mapa con características [9].

## MATERIAL Y MÉTODOS

En los dos casos, tanto para k-NN y RNC se usa un 'Dataset' que contiene tanto datos positivos y negativos. En la siguiente Figura se puede observar una muestra de nuestros datos.

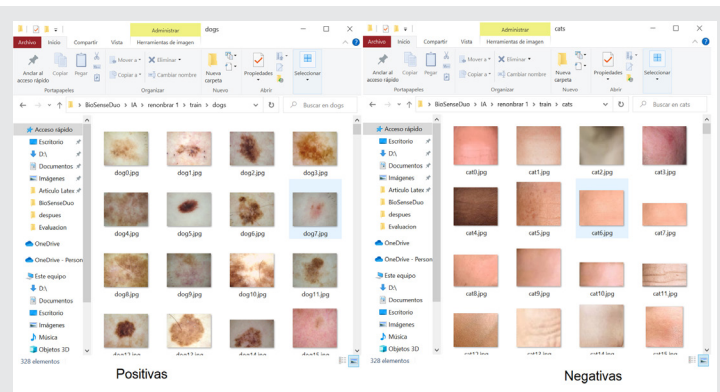


Figura 1. Datos positivos y negativos.  
Fuente: Elaboración propia.

En primer término se trabajo k-NN para el análisis de imágenes positivas y negativas. A continuación se presenta un gráfico que explica el funcionamiento del k-NN para los datos positivos y negativos, en donde  $k$  es el número de conjuntos en el espacio.

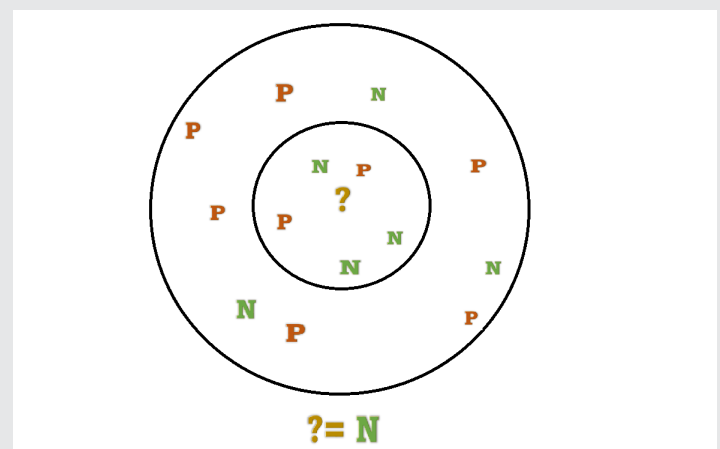


Figura 2. En la figura donde  $k=5$  tenemos que  $? = N$  mientras que para  $k = 14$  tenemos que  $? = P$ .  
Fuente: Elaboración propia.

Durante el desarrollo del proyecto, se vio la necesidad de utilizar el sensor resistivo LDR para evitar errores

con la proyección de sombras, logrando así una iluminación adecuada para el análisis. Además de utilizó una ArduCam y una Raspberry Pi 3B+ para la captura y procesamiento de la información.

Se decidió por el uso de la tarjeta Raspberry Pi debido a su bajo costo en comparación de un sistema de cómputo de escritorio o portátiles convencionales.

Para el análisis de las imágenes es necesario tener conjuntos de datos para el entrenamiento y para la validación y en base a esa relación llevar a cabo el cálculo de la eficiencia de los dos algoritmo estudiados.

El análisis de una imagen se realiza al extraer datos de la misma que se conocen como descriptores, para posteriormente realizar la transformación de la imagen a una señal, para que mediante procedimientos matemáticos podamos obtener las características para analizarla. Para esto se utilizó el lenguaje de programación Python debido al su fácil uso mediante la tarjeta Raspberry Pi.

Para realizar la clasificación, la característica que se debe de extraer es desviación estándar para el conjunto de datos positivos, datos negativos y validación, después se calcula la media del histograma generado para que nos produzca un único valor. Ese valor nos representa las diferencias de tonalidades que tiene cada imagen. El procesamiento de la información se tarda con relación a el número de conjuntos para cada descriptor.

se puede observar la dispersión de los resultados de nuestros descriptores.

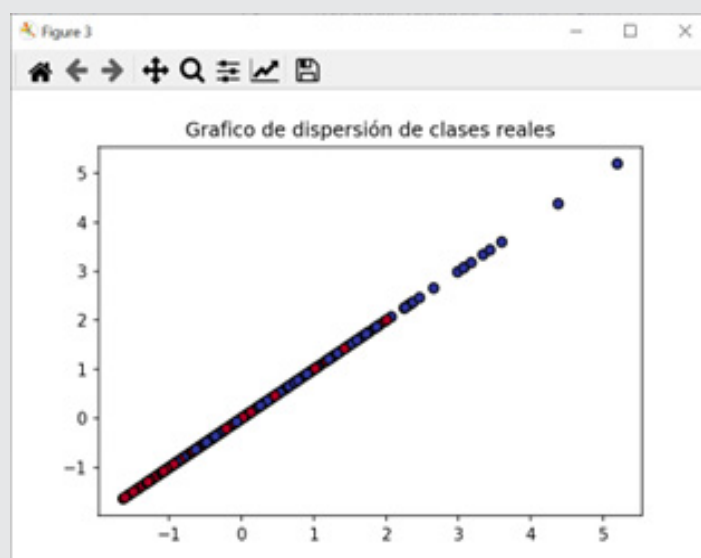


Figura 4. Gráfico de clases.  
Fuente: Elaboración propia.

Para el desarrollo del modelo es necesario descargar la librería: `from sklearn.neighbors import KNeighborsClassifier`. Después con los datos almacenados en los archivos de texto se les asignan etiquetas del tipo de descriptor que son positivo o negativo, estandarizando los datos para tener una mejor visualización en el diagrama. A continuación se ejecuta el programa y después de un periodo de tiempo obtenemos los resultados mostrados en la Figura 5.

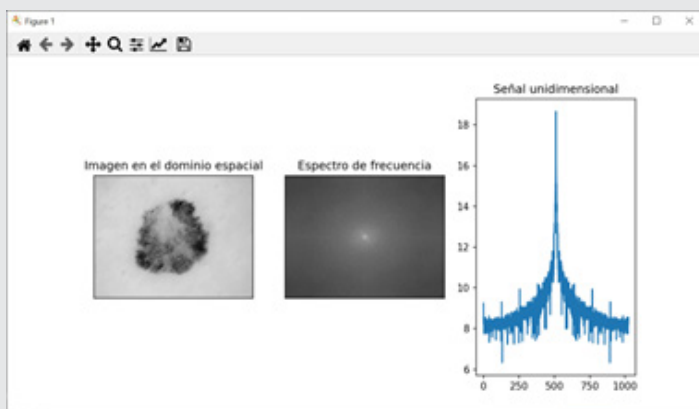


Figura 3. Espectro en frecuencia de una imagen.  
Fuente: Elaboración propia.

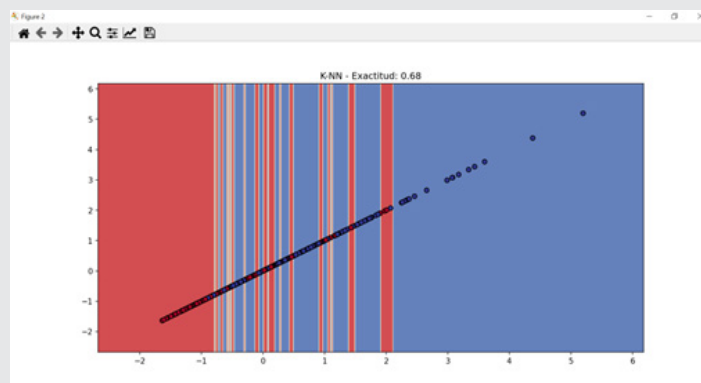


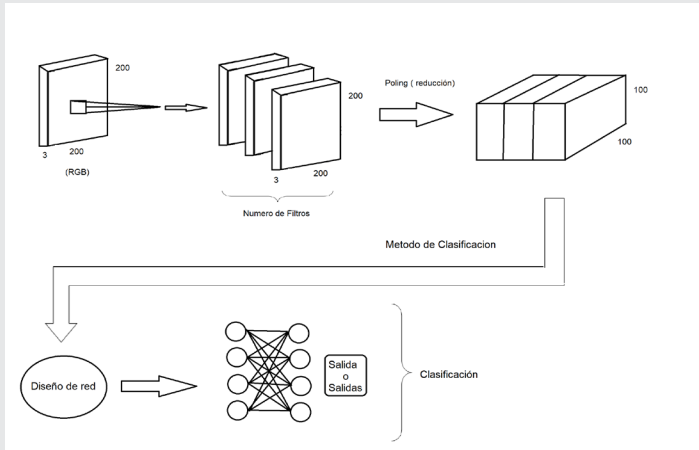
Figura 5. k-NN.  
Fuente: Elaboración propia.

De esta manera, obtenemos los puntos de la desviación estándar de las tonalidades de las imágenes y mediante el uso de la librería `matplotlib.pyplot` podemos graficar los puntos generados para cada tipo de clase o descriptor. Es necesario que para la extracción de esta información generemos un archivo que almacene nuestros datos dado que el procesamiento resulta tardado debido a la gran cantidad de información. En el gráfico que se muestra a continuación

De acuerdo a el valor que le asignemos a k, es el tiempo que el algoritmo se va a demorar en analizar los datos, pues tiene que evaluar todos los valores que estén dentro del rango como se puede ver también en la Figura 2.

Como resultado se obtuvo una exactitud de 68%, este índice se puede calcular con `'\texttt{accuracy\_score}'` haciendo una comparación con las clases reales y las predicciones en el conjunto de prueba.

El esquema de las redes convolucionales estaría dado como se muestra en la figura 6.



**Figura 6. RNC.**  
Fuente: Elaboración propia.

La Figura 6 nos dice que al tomar una imagen, analizamos una pequeña parte de la misma, generando un tipo de ventana con la operación matemática de convolución, permitiendo que podamos generar varias versiones de ese pequeño fragmento de las imágenes. Este proceso se realiza de acuerdo con el número de filtros que se halla elegido (como rotar, desenfocar la imagen etc.), lo cual nos genera un extenso mapa de características. Esto resulta muy bueno ya que podemos extraer un buen número de datos de un solo pixel. En la etapa de reducción se extrae la información más importante, permitiendo disminuir la imagen a un valor menor para que se pueda procesar mediante una red neuronal multicapa y poder tomar la decisión de la categoría a la que pertenece una imagen.

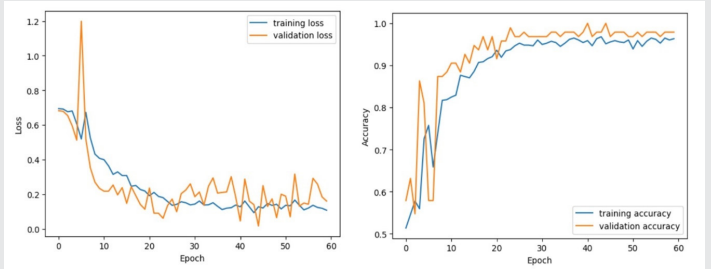
Para las redes convolucionales se utilizó la biblioteca de Tensorflow, debido a los recursos computacionales que requiere una red neuronal como esta. Para la implementación de este trabajo se utilizó la plataforma de desarrollo Colab que nos permite escribir y ejecutar código en el lenguaje de Python, todo esto a través de un navegador web, esto nos facilita el acceso gratuito a recursos de hardware como el GPU, haciéndolo útil para trabajos de aprendizaje profundo como este, ya que como se mencionó anteriormente se requieren bastantes recursos computacionales. Para el procesamiento de las imágenes, se subió a la nube los descriptores separados por carpetas para poder hacer posible el procesamiento y mandar a llamar cada carpeta mediante código.

La información se procesó con un total de 60 épocas y un conjunto de filtros para las imágenes que van desde la rotación hasta el rescalamiento de la imagen con el comando zoom.

Con una cantidad de imágenes de prueba se pudo obtener una eficiencia de 97%.

```
Epoch 25/60
7/7 [=====] - 11s 2s/step - loss: 0.1576 - accuracy: 0.9466 - val_loss: 0.1347 - val_accuracy: 0.9684
Epoch 26/60
7/7 [=====] - 9s 1s/step - loss: 0.1355 - accuracy: 0.9527 - val_loss: 0.1716 - val_accuracy: 0.9684
Epoch 27/60
7/7 [=====] - 12s 2s/step - loss: 0.1411 - accuracy: 0.9482 - val_loss: 0.0985 - val_accuracy: 0.9789
```

**Figura 7. Término del entrenamiento y visualización de eficiencia.**  
Fuente: Elaboración propia.



**Figura 8. Diagrama de eficiencia con relación al número de épocas.**  
Fuente: Elaboración propia.

Al igual que en el algoritmo de k-NN se generó un archivo con los datos de prueba, para no tener que esperar todo el procesamiento de los datos, así también se generó un archivo con extensión .h5 con el entrenamiento neuronal finalizado ya que el proceso es muy tardado.



**Figura 9. Instalación de equipo electrónico.**  
Fuente: Elaboración propia.

Mediante el uso de código se manda a llamar el modelo para que comience a clasificar las imágenes que obtiene a través de la cámara como se puede ver en la Figura 10.

Para el control de luminosidad se optó por un regulador PID con Arduino para obtener un diseño adaptativo al ambiente. La sintonización del control PID se llevó a cabo mediante la captura de los datos a través del puerto serial con la ayuda del mismo Arduino, además con el programa CoolTerm se creó un block de notas

con los valores del LDR. Posteriormente, en Matlab generamos una tabla con todos los valores obtenidos del sensor. Matlab tiene una comando que nos ayuda en la sintonización del PID, simplemente tecleamos ident.

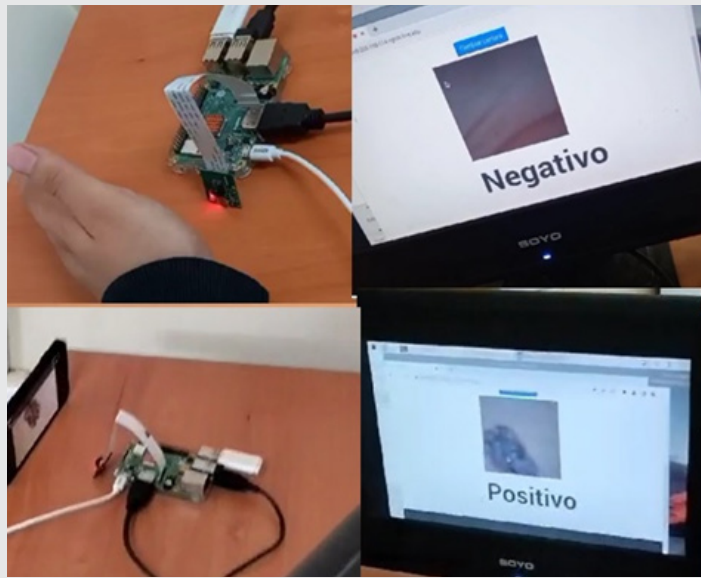


Figura 10. Pruebas.  
Fuente: Elaboración propia.

Finalmente, importamos nuestros valores de la tabla generada, para obtener la identificación de una función de transferencia y el cálculo de una estimación para este tipo de control mostrándonos la eficiencia de la estimación.

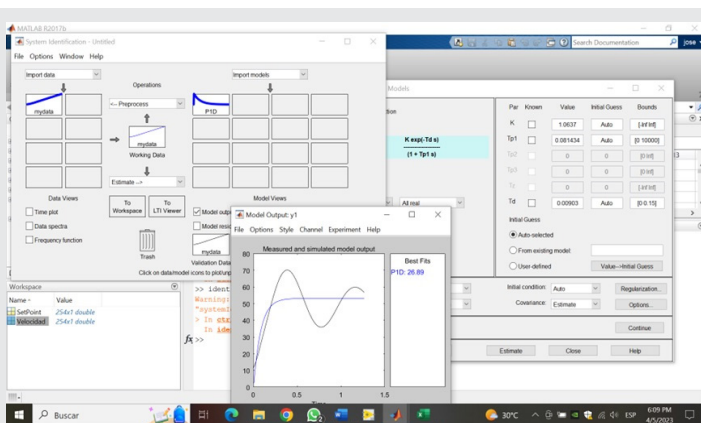


Figura 11. Sintonización de PID utilizando Matlab con el comando ident.  
Fuente: Elaboración propia.

Se explota el valor generado de la estimación y se pasa a la herramienta "Tune", ahí se importa la estimación y se selecciona el tipo de control deseado, generando los valores de la parte proporcional, integral y derivativo.

Los valores obtenidos previamente se pasan a el código de Arduino para que controle la iluminación.

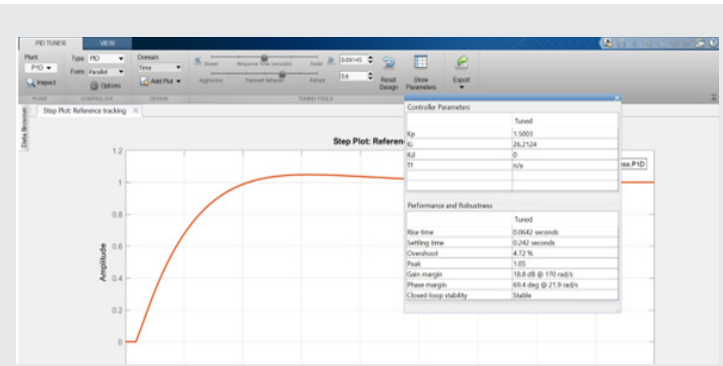


Figura 12. Visualización de PID.  
Fuente: Elaboración propia.

En la siguientes imagenes se puede observar el prototipo.

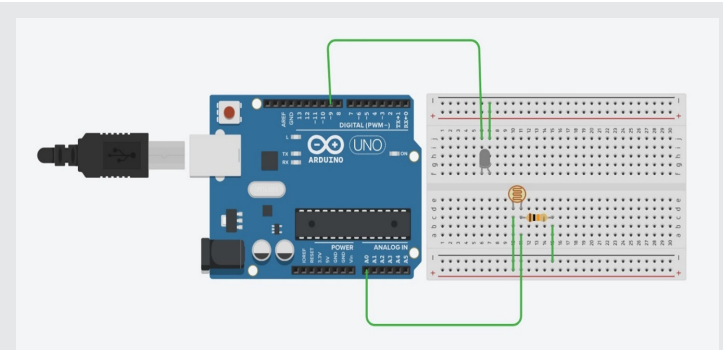


Figura 13. Circuito Arduino.  
Fuente: Elaboración propia.

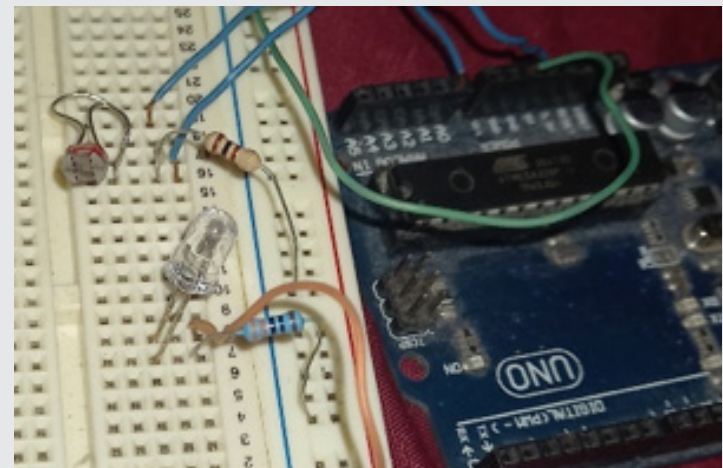


Figura 14. Montaje de circuito de pruebas.  
Fuente: Elaboración propia.

## RESULTADOS

Concluyendo, los resultados obtenidos en nuestro estudio revelan un desempeño notablemente superior de las redes neuronales convolucionales, alcanzando una impresionante tasa de exactitud del 97%. Este rendimiento contrasta significativamente con el algoritmo k-NN, que exhibió una precisión limitada al 68%. Además de la diferencia en precisión, es fundamental considerar la disparidad en los tiempos de reacción entre ambos enfoques.

Las redes neuronales convolucionales demostraron una eficiencia excepcional al clasificar imágenes en tiempo real, con tiempos de reacción medidos en segundos o incluso milisegundos. Este atributo las posiciona como una elección idónea para aplicaciones que requieren la clasificación instantánea de imágenes de video en vivo. En contraste, el algoritmo k-NN presentó una desventaja notable en términos de velocidad, ya que la clasificación de una única imagen estática demandó un tiempo considerable de 35 minutos.

Tabla 1: Comparación de K-NN y RNC.

método	Eficiencia	Tiempo en minutos
K-NN	97	35
RNC	68	0.166667

Fuente: Elaboración propia.

## CONCLUSIONES

Las redes neuronales resultaron mejores debido a que pueden aprender patrones así como representaciones complejas de los datos mientras que para k-NN es necesario el uso de más formas de extracción de datos debido a que el resultado obtenido con nada más la desviación estándar no generó un buen rendimiento.

Además, las redes neuronales pueden trabajar una gran cantidad de datos eficientemente mientras que k-NN puede volverse más costoso los recursos computacionales a medida que los datos aumenten.

De esta forma, k-NN puede ser efectivo para conjuntos de datos simples y pequeños ya que si el conjunto de datos es muy extenso puede generar lentitud en el sistema, pues el algoritmo no aprende de un modelo, sino que este utiliza los mismos datos para la generación de una respuesta.

Respecto a otros trabajos y observaciones realizadas, se identificó que la eficiencia sí varía con relación a la iluminación del ambiente donde se están llevando a cabo las pruebas, por lo que la integración del led resultó beneficioso para el programa evitando en cierta media los falsos negativos y positivos.

Para próximos trabajos se recomienda el uso del control del PID con la tarjeta Raspberry Pi sin agregar otra tarjeta como lo es Arduino, para la reducción del espacio del dispositivo y el ahorro de los costos materiales.

## AGRADECIMIENTOS

Agradecemos al Tecnológico Nacional de México y al Instituto Tecnológico de Ciudad Guzmán por dar las facilidades para llevar a cabo este proyecto.

## BIBLIOGRAFÍA

[1] Aceituno-Madera, P. et al (2011). Melanoma, altitud y radiación uvb. *Actas dermo-sifiliograficas*, 102(3): 199-205.

[2] Acosta, E., Fierro, et al. (2009). Melanoma: patogénesis, clínica e histopatología. Consultada el 13 del 09 del 2023

[3] Ali, M., Tang, et al. (2020). Algoritmo semántico-k-nn: una versión mejorada del algoritmo k-nn tradicional. ISSN: 0957-4174, 151.



Figura 15. Análisis de imágenes positivas.  
Fuente: Elaboración propia.

En resumen, mientras las redes neuronales convolucionales exhibieron una precisión destacada del 97%, su capacidad para realizar clasificaciones en tiempo real las distingue significativamente de k-NN, que, a pesar de su precisión del 68%, se ve limitado por tiempos de reacción significativamente más prolongados, especialmente en la clasificación de imágenes estáticas. Estos resultados refuerzan la posición de las redes neuronales convolucionales como una opción superior en términos de rendimiento y eficacia en el contexto específico de la clasificación de imágenes.

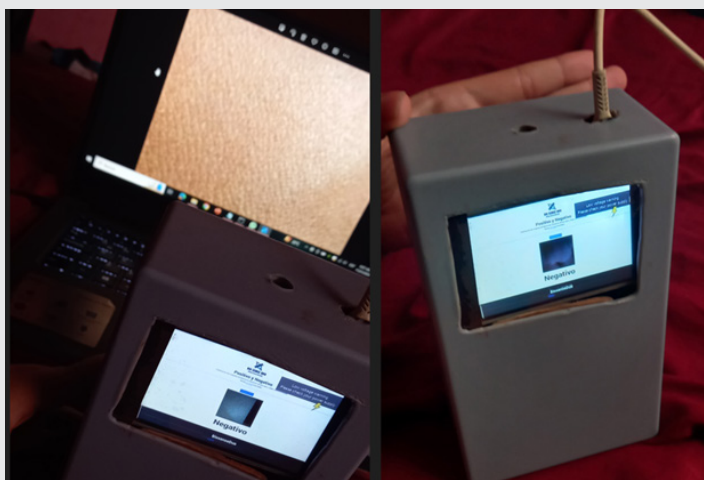


Figura 16. Análisis de imágenes negativas.  
Fuente: Elaboración propia.

- [4] Arias, V. et al. (2019). Una introducción a las aplicaciones de la inteligencia artificial en medicina: Aspectos históricos. *Revista Latinoamericana de Hipertensión*, 14(5):590-600.
- [5] Artola Moreno Á. (2019). Clasificación de imágenes usando redes neuronales convolucionales en Python. pages 28-29.
- [6] Barrios Tao, H. et al. (2020). Subjetividades e inteligencia artificial: desafíos para 'lo humano'. *Veritas*, (47):81-107.
- [7] Bernal-de Lazaro, J. et al. (2011). Estudio comparativo de clasificadores empleados en el diagnóstico de fallos de sistemas industriales. *Ingeniería Mecánica*, 14(2):87-98.
- [8] Bonilla Carrión, C. (2020). Redes convolucionales. page 16.
- [9] Duran Suarez, J. (2017). Redes neuronales convolucionales en r: Reconocimiento de caracteres escritos a mano. page 25.
- [10] Hardy, T. (2001). *la (inteligencia artificial)*. *Polis: Revista Latinoamericana*, (2):18.
- [11] Herrera, J. C. a. et al. (2016). Diseño de un sistema automático de selección de frutos de café mediante técnicas de visión artificial. *Revista UIS ingenierías*, 15(1):7-14.
- [12] Infante Carbonell, M. G. et al. (2019). Melanoma cutáneo: algunas consideraciones actuales. *Medisan*, 23(1):146-164.
- [13] Leitner, R. (2006). Epidemiología del melanoma cutáneo. *Revista argentina de dermatología*, 87(2):86-97.
- [14] Lucero Carrillo, E. D. et al. (2016). Utilización de técnicas de visión artificial para la detección automática de defectos externos del mango.
- [15] Lugo-Reyes, Saúl Oswaldo, a. o. (2014). *Inteligencia artificial para asistir el diagnóstico clínico en medicina*. *Revista Alergia México*, 61(2):110-120.
- [16] Maria Zuluaga, A. I. et al. (2016). Actualización en el tratamiento quirúrgico del melanoma cutáneo primario y metastásico. *Cirugía y Cirujanos*, 84(1):77-84.
- [17] Micó, M. (1996). Algoritmos de búsqueda de vecinos más próximos en espacios métricos. PhD thesis, Tesis doctoral, Universidad Politécnica de Valencia, Valencia, España.
- [18] M.L. Villalobos León, R. M. V. et al. (2013). Melanoma. *Medicine-Programa de Formación Médica Continuada Acreditado*, 11(26):1597-1607.
- [19] Mordoh, A. (2009). clínica del melanoma. *Acta bio-química clínica latinoamericana*, 43(3):327-331.
- [20] Narváez Vilema, M. E. et al. (2021). Predicción de clientes potenciales utilizando k vecino más cercano en el área de negocios de la cooperativa Riobamba.
- [21] Nguyen Cong, R. P. et al. (2015). Aprendizaje super-visado de funciones de distancia: estado del arte. *Revista Cubana de Ciencias Informáticas*, 9(2):14-28.
- [22] Porcelli, A. M. (2020). La inteligencia artificial y la robótica: sus dilemas sociales, éticos y jurídicos. *Derecho global. Estudios sobre derecho y justicia*, 6(16):49-105.
- [23] Sordo, C. and Gutiérrez, C. (2013). Cáncer de piel y radiación solar: experiencia peruana en la prevención y detección temprana del cáncer de piel y melanoma. *Revista Peruana de Medicina Experimental y Salud Publica*, 30:113-117.
- [24] Vega, M. Á. et al. (2020). Inteligencia artificial y aprendizaje automático en medicina. *Revista médica sinergia*, 5(8):e557-e557
- Zaballos, P. et al. (2004). Criterios dermatoscópicos para el diagnóstico del melanoma. *Med Cutan Iber Lat Am*, 32(1):3-17.
- [25] Zaballos, P. et al. (2004). Criterios dermatoscópicos para el diagnóstico del melanoma. *Med Cutan Iber Lat Am*, 32(1):3-17.