



UNIVERSIDAD TÉCNICA DE AMBATO

**FACULTAD DE INGENIERIA EN SISTEMAS,
ELECTRÓNICA E INDUSTRIAL**

Inteligencia Artificial

Alumno: Christian Núñez

Nivel: Séptimo “A”

Docente: Ing. Rubén Nogales

Octubre 2022 – Marzo 2023

Clasificador de colores.

Objetivos.

El objetivo de este informe es evaluar la eficacia de una red neuronal para resolver un problema de clasificación de imágenes, comparándola con otros métodos de aprendizaje automático y analizando los resultados obtenidos para identificar las fortalezas y debilidades del modelo y recomendar posibles mejoras para futuras investigaciones.

Introducción.

El reconocimiento de colores es un problema común en la visión por computadora y tiene aplicaciones en diferentes industrias, como la automoción, la robótica, entre otros. En este informe, se presenta una solución utilizando redes neuronales para el reconocimiento de colores en imágenes.

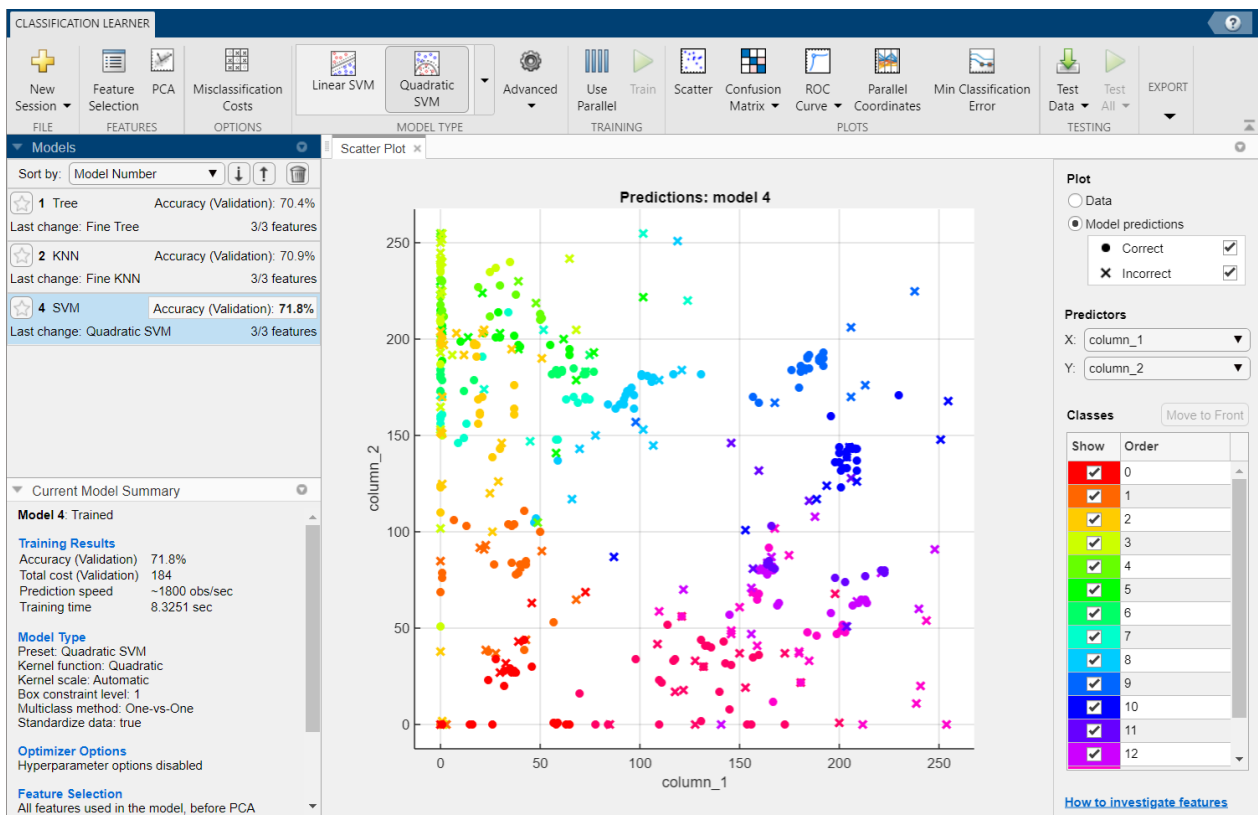
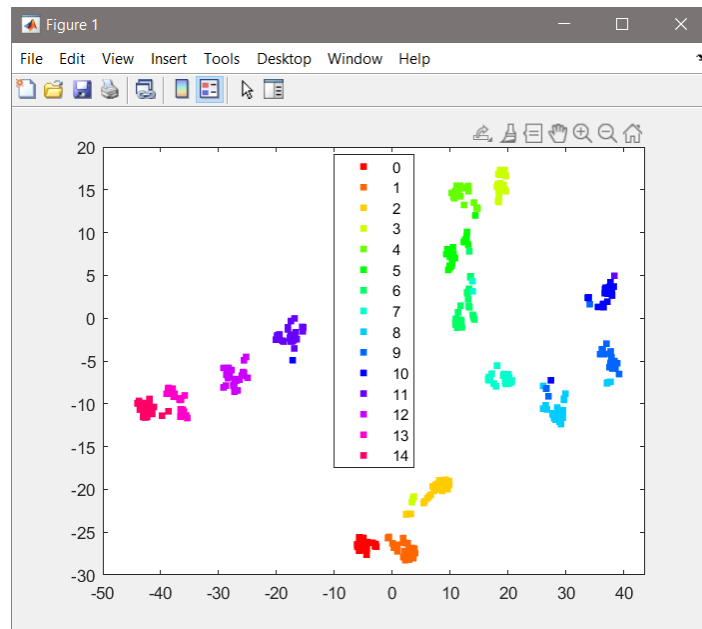
El objetivo de este informe es evaluar la eficacia de una red neuronal para resolver el problema de reconocimiento de colores y comparar su desempeño con otros métodos de aprendizaje automático. Se describirá en detalle la metodología utilizada para construir y entrenar la red neuronal, incluyendo la arquitectura, los algoritmos de optimización y la evaluación de los resultados obtenidos.

Además, se analizarán los resultados y se discutirán las fortalezas y debilidades del modelo, proporcionando recomendaciones para futuras investigaciones y mejoras en la solución. En resumen, este informe pretende demostrar la capacidad de las redes neuronales para resolver problemas de reconocimiento de colores de manera efectiva.

Desarrollo.

1. *Evaluación de los algoritmos:* Se evaluaron cuatro algoritmos de aprendizaje automático para resolver el problema de reconocimiento de colores, incluyendo Decision Trees (DT), k-Nearest Neighbors (KNN), Support Vector Machines (SVM) y una red neuronal. Se utilizaron métricas como la precisión, recall y medidas de error para comparar el desempeño de cada algoritmo.
2. *Implementación del mejor algoritmo en la red neuronal:* Luego de evaluar los algoritmos, se identificó que el SVM fue el que obtuvo los mejores resultados. Por lo tanto, se decidió utilizar el SVM como preprocesamiento para la construcción de la red neuronal. La red neuronal utilizó una arquitectura de feedforward con varias capas ocultas y se entrenó utilizando un algoritmo de optimización como el stochastic gradient descent (SGD).
3. *Resultados de la red neuronal:* Se presentaron los resultados obtenidos al aplicar la red neuronal al conjunto de datos de prueba. Se compararon estos resultados con los obtenidos por los otros algoritmos evaluados previamente y se discutió la eficacia del modelo en términos de precisión y medidas de error.
4. *Análisis de los resultados:* Se analizaron los resultados obtenidos por la red neuronal y se discutió la influencia del preprocesamiento con el SVM en la mejora del desempeño del modelo. Se identificaron las fortalezas y debilidades del modelo y se proporcionaron recomendaciones para futuras investigaciones y mejoras en la solución.

En resumen, el informe demuestra cómo la combinación de diferentes técnicas de aprendizaje automático puede ser efectiva para resolver problemas de reconocimiento de colores. La implementación del SVM como preprocesamiento para la red neuronal resultó en una mejora significativa en el desempeño del modelo y sugiere la importancia de considerar diferentes técnicas en la construcción de soluciones de aprendizaje automático.



```

1 -   clc
2 -   close all
3 -   clear
4 -   data = readmatrix("colores.csv");
5 -   y = data(:,end);
6 -   dataset = normalize(data(:,1:end-1));
7 -   dataset = [dataset y];
8 -   n = size(dataset,1); % Número de filas en los datos
9 -   cv = cvpartition(n,'Holdout',0.2);
10  % Crea dos conjuntos de datos, uno para entrenamiento y otro para prueba
11 -   dataTrain = dataset(cv.training,:);
12 -   dataTest= dataset(cv.test,:);
13 -   dataXTrain =dataTrain(:,1:end-1);
14 -   dataYTrain=dataTrain(:,end);
15 -   dataXTest=dataTest(:,1:end-1);
16 -   dataYTest=dataTest(:,end);
17 -   ts_training = tsne(dataTrain);
18 -   gscatter(ts_training(:,1),ts_training(:,2),dataTrain(:,end));
19 -   classificationLearner;
20  %% testeo
21 -   yfit = trainedModelANN.predictFcn(dataXTest);
22 -   ErrorBest = sum( yfit ~= dataYTest )/n;
23 -   exactitud=1-ErrorBest;
24 -   fprintf('Error de entrenamiento = %3.2f %%\n',100*ErrorBest);
25 -   fprintf('Exactitud del modelo = %3.2f %%\n',100*exactitud);
26  %% Ejecucion
27  %bgr
28 -   imagen='rojo.jpg';
29 -   val=RGB(imagen);
30 -   predic = trainedModelANN.predictFcn(val);
31 -   fprintf('prediccion = %i\n',predic);
32
33

```

(Entrenamiento en Matlab)

Por otra parte, el reconocimiento de colores es un problema importante en diferentes aplicaciones, como la robótica, la visión por computadora y la industria automotriz. Python es un lenguaje de programación ampliamente utilizado en el desarrollo de soluciones de aprendizaje automático y ofrece una amplia gama de bibliotecas y herramientas para resolver problemas de reconocimiento de colores.

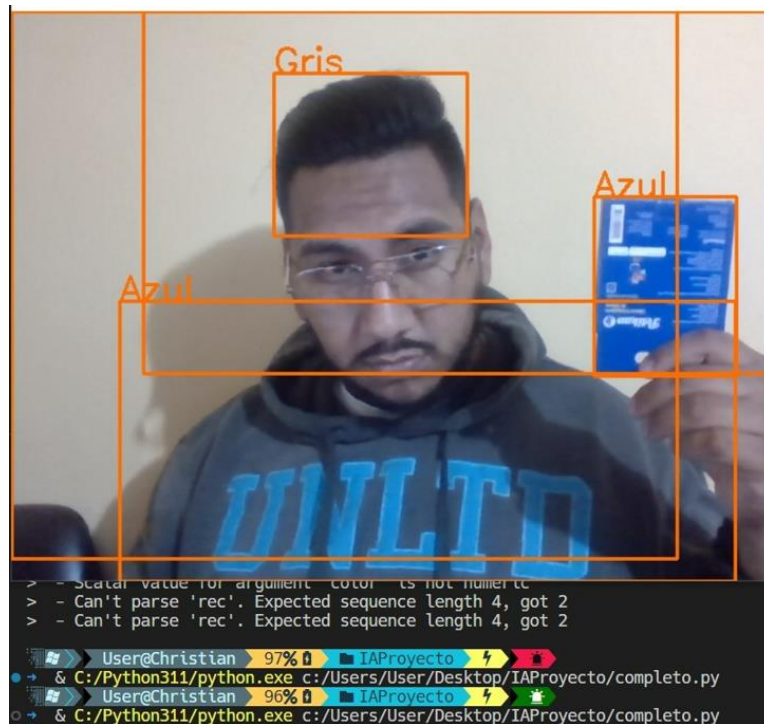
```

IAProyecto - completo.py

1  import cv2
2  import numpy as np
3
4  # Diccionario con los colores y sus rangos en HSV
5  colors = {
6      "Negro": (np.array([0, 0, 0]), np.array([180, 255, 30])),
7      "Azul": (np.array([100, 50, 50]), np.array([140, 255, 255])),
8      "Marrón": (np.array([10, 100, 20]), np.array([20, 255, 255])),
9      "Verde": (np.array([35, 50, 50]), np.array([90, 255, 255])),
10     "Rojo": (np.array([0, 50, 50]), np.array([10, 255, 255])),
11     "Amarillo": (np.array([20, 100, 100]), np.array([30, 255, 255])),
12     "Blanco": (np.array([0, 0, 200]), np.array([180, 30, 255])),
13     "Gris": (np.array([0, 0, 70]), np.array([180, 30, 220]))
14 }
15
16 # Inicializa la cámara
17 cap = cv2.VideoCapture(0)
18
19 while True:
20     # Captura un frame de la cámara
21     ret, frame = cap.read()
22
23     # Convierte la imagen a HSV
24     hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
25
26     # Itera sobre los colores
27     for color_name, (lower, upper) in colors.items():
28
29         # Crea una máscara para el color actual
30         mask = cv2.inRange(hsv, lower, upper)
31
32         # Encuentra los contornos de los objetos en la máscara
33         contours, hierarchy = cv2.findContours(
34             mask, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
35
36         # Itera sobre los contornos encontrados
37         for contour in contours:
38
39             # Calcula el área del contorno
40             area = cv2.contourArea(contour)
41
42             # Si el área es mayor a 10000, dibuja el contorno y muestra el nombre del color
43             if area > 10000:
44                 x, y, w, h = cv2.boundingRect(contour)
45                 cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 108, 255), 2)
46                 cv2.putText(frame, color_name, (x, y),
47                             cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 108, 255), 2)
48
49     # Muestra la imagen con los contornos y los nombres de los colores
50     cv2.imshow("Detector de Colores", frame)
51
52     # Si se presiona la tecla "q", sale del bucle
53     if cv2.waitKey(1) == 27:
54         break
55
56 # Libera la cámara y destruye las ventanas
57 cap.release
58

```

(Código de Python para la detección de colores).



(Ejecución del clasificador de colores)

Sin embargo, el uso de una red neuronal entrenada con SVM puede mejorar significativamente el desempeño de la solución. El SVM es un algoritmo de aprendizaje automático que puede utilizarse como preprocesamiento para la construcción de la red neuronal, lo que resulta en una mejora en la precisión y el rendimiento del modelo. Además, las redes neuronales son una herramienta poderosa y versátil para el reconocimiento de colores y ofrecen una base sólida para futuras investigaciones y mejoras en la solución.

Conclusiones.

La combinación de técnicas de aprendizaje automático es efectiva para resolver problemas de reconocimiento de colores. La implementación del SVM como preprocesamiento para la red neuronal resultó en una mejora significativa en el desempeño del modelo, sugiriendo la importancia de considerar diferentes técnicas en la construcción de soluciones de aprendizaje automático.

Las redes neuronales son una herramienta poderosa para el reconocimiento de colores. Los resultados obtenidos en este informe demuestran la capacidad de las redes neuronales para resolver el problema de reconocimiento de colores de manera efectiva y ofrecen una base sólida para futuras investigaciones y mejoras en la solución. La combinación de preprocesamiento con algoritmos como el SVM y la utilización de una arquitectura adecuada y algoritmos de optimización efectivos son claves para lograr un buen desempeño en la solución.

Bibliografía.

<https://aws.amazon.com/es/what-is/neural-network/#:~:text=Una%20red%20neuronal%20es%20un,lo%20hace%20el%20cerebro%20humano.>

https://es.wikipedia.org/wiki/Red_neuronal_artificial

<https://www.atriainnovation.com/que-son-las-redes-neuronales-y-sus-funciones/>

<https://omes-va.com/deteccion-de-colores/>

<https://medium.com/@gastonace1/detecci%C3%B3n-de-objetos-por-colores-en-im%C3%A1genes-con-python-y-opencv-c8d9b6768ff>

<https://www.delftstack.com/es/howto/python/opencv-inrange/>