



PORTADA

UNIVERSIDAD TÉCNICA DE AMBATO

Facultad de Ingeniería en Sistemas, Electrónica e Industrial

Título: Proyecto Final Reconocimiento de Colores

Carrera: Ingeniería de Software

Nivel y Paralelo: Séptimo “A”

Alumnos: Aguas Núñez Christian Xavier

Arcos Arcos John David

Módulo y Docente: Inteligencia Artificial

Ing. Rubén Nogales Mg.

Ambato-Ecuador 2023



Introducción

El rápido avance de la tecnología conjuntamente de la mano de la Inteligencia Artificial ha sido muy exponencial, es por eso que con el planteamiento del proyecto se puede ayudar a optimizar los procesos para el reconocimiento de colores mediante una cámara web. El trabajo propuesto propone una aplicación que puede reconocer colores de objetos pasado en redes neuronales y algoritmos de machine learning ya estudiamos como son: KNN, SVM, DT y ANN.

Objetivos

Objetivo General

- Desarrollar una aplicación que permita reconocer colores a través de una cámara web, mediante visión por computadora usando algoritmos de machine learning.

Objetivos Específicos

- Comparar los modelos y verificar cual es el mejor identificando colores.
- Usar algoritmos paramétricos en los modelos.
- Usar algoritmos no paramétricos en los modelos.

Desarrollo

División del data set para testing y training.

```

clc
close all
clear all
colores = imageDatastore('dataSetColores\','IncludeSubfolders',true,'LabelSource','foldernames');
contador = 0;

while hasdata(colores)
    [imgColores,datos] = read(colores);
    imshow(imgColores);
    imgColores = fprepararimgds(imgColores);
    contador = contador + 1;
    if(datos.Label == "amarillo")
        img(contador,:) = imgColores;
        etiqueta(contador,1) = 1;
    elseif(datos.Label == "azul")
        img(contador,:) = imgColores;
        etiqueta(contador,1) = 2;

    elseif(datos.Label == "blanco")
        img(contador,:) = imgColores;
        etiqueta(contador,1) = 3;

    elseif(datos.Label == "celeste")
        img(contador,:) = imgColores;
        etiqueta(contador,1) = 4;

    elseif(datos.Label == "gris")
        img(contador,:) = imgColores;
        etiqueta(contador,1) = 5;

    elseif(datos.Label == "morado")
        img(contador,:) = imgColores;
        etiqueta(contador,1) = 6;

    elseif(datos.Label == "naranja")
        img(contador,:) = imgColores;
        etiqueta(contador,1) = 7;

    elseif(datos.Label == "negro")
        img(contador,:) = imgColores;
        etiqueta(contador,1) = 8;

    elseif(datos.Label == "rojo")
        img(contador,:) = imgColores;
        etiqueta(contador,1) = 9;

    elseif(datos.Label == "verde")
        img(contador,:) = imgColores;
        etiqueta(contador,1) = 10;

    end
end

dataSetColores = horzcat(img,etiqueta);
caracteristicas = dataSetColores(:,1:end-1);
etiquetas = dataSetColores(:,end);

%%
% Manejo de las salidas para Red Neuronal de Reconocimiento de patrones
numclas=unique(etiquetas);
oneHotMat = (etiquetas ==1:length(numclas));

%%
% Entrenamiento de la Red Neuronal
train_data = load('train_data.mat');
train_inputs = train_data.train_data(:,1:end-1);
train_targets = train_data.train_data(:,end);
valorExactitud_train = 100 * (1 - mse(myNeuralNetworkFunctionColors(train_inputs), train_inputs, train_targets))
error_train = 100 - valorExactitud_train

% Testeo de la Red Neuronal entrenada
test_data = load('test_data.mat');
test_inputs = test_data.test_data(:,1:end-1);
test_targets = test_data.test_data(:,end);
valorExactitud = 100 * (1 - mse(myNeuralNetworkFunctionColors(test_inputs), test_inputs, test_targets))
error = 100 - valorExactitud

```

Resultados K-NN.

Código:

```
1 %% Limpieza de workspace
2 clc
3 close all
4 clear all
5
6 %% Inicialización de parámetros
7 % data
8 load .\dataSetColores\modeloEntrenamientoColoresknn.mat
9 X = dataset.train.imagenes(:,:);
10 L = dataset.train.labels(:,1);
11 N = length(dataset.train.imagenes(:,1));
12
13 % K value
14 K = floor(round(log10(N)));
15 if(mod(K,2) == 0)
16     K = K+1;
17 end
18
19 %% Input
20 a = 1;
21 b = length(dataset.test.imagenes);
22 red = randi([a b],1);
23 imgTest = dataset.test.imagenes(red,:);
24 labelTest = dataset.test.labels(red, 1);
25
26 %% Distance calo (euclidean)
27 distance = zeros(length(X),3);
28 for i=1: length(X)
29     distance(i,1) = sum(sqrt((X(i,:)-imgTest(1,:)).^2));
30     distance(i,2) = L(i,1);
31     distance(i,3) = i;
32 end
33
34 %% Sort distances
35 [~, s] = sort(distance(:, 1));
36 ord = distance(s, :);
```

```

%% Sort distances
[~, s] = sort(distance(:, 1));
ord = distance(s, :);
%% Get k short values
kVector = zeros(K,3);
for i = 1: K
    kVector(i,:) = ord(i,:);
end

%% Get Answer
[cnt_unique, unique_a] = histogram(kVector(:,2),unique(kVector(:,2)));
unique_a(:,2) = cnt_unique(:)*100/K;

mayor = false;
res = unique_a(1,2);
for i = 1 : size(unique_a,1)
    if(unique_a(i,2) > res)
        res = unique_a(i,1);
        mayor = true;
        break
    end
end

if(mayor)
    a = find(kVector(:,2) == res);
    res = kVector(a(1),:);
else
    res = kVector(1,:);
end

%% test
subplot(1,2,1)
imshow(reshape(imgTest,[28,28]));
title("Imagen de entrada");
xlabel("Etiqueta: "+labelTest);

subplot(1,2,2)
imshow(reshape(X(res(1,3),:),[28,28]));
title("Predicción");
xlabel("Etiqueta asignada: "+res(1,2))

```

Resultados SVM.

Código

Resultados ANN.

Código

Resultados DT.

Código

Resultados del Proyecto.

Código

Link del proyecto

[Programa final](#)

Conclusiones

Se realizó el aplicativo para el reconocimiento de colores con un mayor porcentaje de acierto.

