

```
clc; clear; close all;
```

```
%% 1. Descargar las fotos del dataset desde Kaggle y colocarlas en las carpetas correspondientes
```

```
%% 2. Procesar las fotos para obtener las matrices Xtrain y Xtest
```

```
% Directorios de las imágenes de entrenamiento
```

```
dir_hombres = 'C:/Users/EstJustinXavierRosil/OneDrive - Universidad Politecnica Salesiana/Documentos/Universidad/Periodo 64/Inteligencia artificial/Parcial2/proyecto/archive/Training/male';
```

```
dir_mujeres = 'C:/Users/EstJustinXavierRosil/OneDrive - Universidad Politecnica Salesiana/Documentos/Universidad/Periodo 64/Inteligencia artificial/Parcial2/proyecto/archive/Training/female';
```

```
%% Listar archivos de imágenes
```

```
archivos_hombres = dir(fullfile(dir_hombres, '*.jpg'));
```

```
archivos_mujeres = dir(fullfile(dir_mujeres, '*.jpg'));
```

```
%% Seleccionar las primeras 100 imágenes (si están disponibles)
```

```
num_imagenes = min(100, length(archivos_hombres)); % Asegura que no se acceda a más archivos de los disponibles
```

```
archivos_hombres = archivos_hombres(1:num_imagenes);
```

```
archivos_mujeres = archivos_mujeres(1:num_imagenes);
```

```
%% Conversión a escala de grises y redimensionamiento
```

```
alto_img = 90;
```

```
ancho_img = 60;
```

```
tam_img = [alto_img, ancho_img];
```

```
Xtrain = zeros(num_imagenes * 2, alto_img * ancho_img);
```

```
ytrain = [ones(num_imagenes, 1); -ones(num_imagenes, 1)];
```

```
for i = 1:num_imagenes
```

```
    % Procesar imágenes de hombres
```

```
    img = imread(fullfile(dir_hombres, archivos_hombres(i).name));
```

```
    img_gris = rgb2gray(img);
```

```
    img_redimensionada = imresize(img_gris, tam_img);
```

```
    Xtrain(i, :) = img_redimensionada(:)';
```

```
    % Procesar imágenes de mujeres
```

```
    img = imread(fullfile(dir_mujeres, archivos_mujeres(i).name));
```

```
    img_gris = rgb2gray(img);
```

```
    img_redimensionada = imresize(img_gris, tam_img);
```

```
    Xtrain(num_imagenes + i, :) = img_redimensionada(:)';
```

```
end
```

```
%% Directorios de las imágenes de validación
```

```
dir_val_hombres = 'C:/Users/EstJustinXavierRosil/OneDrive - Universidad  
Politecnica Salesiana/Documentos/Universidad/Periodo 64/Inteligencia  
artificial/Parcial2/proyecto/archive/Validation/male';
```

```
dir_val_mujeres = 'C:/Users/EstJustinXavierRosil/OneDrive - Universidad  
Politecnica Salesiana/Documentos/Universidad/Periodo 64/Inteligencia  
artificial/Parcial2/proyecto/archive/Validation/female';
```

```
%% Listar archivos de imágenes
```

```
archivos_val_hombres = dir(fullfile(dir_val_hombres, '*.jpg'));
```

```
archivos_val_mujeres = dir(fullfile(dir_val_mujeres, '*.jpg'));
```

```
%% Verificar cuántas imágenes hay disponibles
```

```

num_val_hombres = min(20, length(archivos_val_hombres));
num_val_mujeres = min(24, length(archivos_val_mujeres));

%% Seleccionar las primeras imágenes disponibles
archivos_val_hombres = archivos_val_hombres(1:num_val_hombres);
archivos_val_mujeres = archivos_val_mujeres(1:num_val_mujeres);

%% Procesar imágenes de validación
num_val_imagenes = num_val_hombres + num_val_mujeres;
xtest = zeros(num_val_imagenes, alto_img * ancho_img);
yetiquetas_val = [ones(num_val_hombres, 1); -ones(num_val_mujeres, 1)];

for i = 1:num_val_hombres
    % Procesar imágenes de hombres
    img = imread(fullfile(dir_val_hombres, archivos_val_hombres(i).name));
    img_gris = rgb2gray(img);
    img_redimensionada = imresize(img_gris, tam_img);
    xtest(i, :) = img_redimensionada(:)';
end

for i = 1:num_val_mujeres
    % Procesar imágenes de mujeres
    img = imread(fullfile(dir_val_mujeres, archivos_val_mujeres(i).name));
    img_gris = rgb2gray(img);
    img_redimensionada = imresize(img_gris, tam_img);
    xtest(num_val_hombres + i, :) = img_redimensionada(:)';
end

```

```

%% 3. generar el vector w_iniciales y y_train

w_inicial = zeros(1, alto_img*ancho_img);

ytrain=[ones(num_imagenes,1);-ones(num_imagenes,1)];

%%

%% Función perceptronLearning

function [w_final, W_values] = perceptronLearning(Xtrain, ytrain, w_inicial)

    % Inicialización

    [num_samples, num_features] = size(Xtrain);

    w = w_inicial;

    W_values = w;

    % Parámetros

    tasa_aprendizaje = 0.1; % Tasa de aprendizaje para el ajuste de pesos

    max_iter = 1000; % Número máximo de iteraciones permitidas

    iter = 0;

    error = Inf;

    % Ciclo de aprendizaje

    while error > 0.1 && iter < max_iter

        error = 0;

        for i = 1:num_samples

            if ytrain(i) * (Xtrain(i, :) * w') <= 0

                w = w + tasa_aprendizaje * ytrain(i) * Xtrain(i, :);

                error = error + 1;

            end

        end

        error = error / num_samples;

        W_values = [W_values; w];

    end

```

```

        iter = iter + 1;
    end
    w_final = w;
end

%% 4. Entrenar el perceptron
[w_final, ~]=perceptronLearning(Xtrain,ytrain,w_inicial);

%% Función perceptronOutput
function y_hat = perceptronOutput(X, w)
    % X: Matriz de datos de entrada (n_samples x n_features)
    % w: Vector de pesos del perceptrón (1 x n_features)
    % y_hat: Vector de predicciones (n_samples x 1)

    % Calcular el número de muestras
    num_samples = size(X, 1);

    % Inicializar el vector de predicciones
    y_hat = zeros(num_samples, 1);

    % Calcular las predicciones
    for i = 1:num_samples
        y_hat(i) = sign(X(i, :) * w');
    end
end

%% 5. Probar el perceptron en el conjunto Xtes
y_hat=perceptronOutput(xtest,w_final);

```

```
%% imagen con la segunda y penultima de las fotos de cada clase, en escala de  
grises y redimensionada a 90x60
```

```
%% segunda foto de hombres
```

```
img_hombre2 = imread(fullfile(dir_hombres,archivos_hombres(2).name));
```

```
img_hombre2_gris=rgb2gray(img_hombre2);
```

```
img_hombre2_redimensionada=imresize(img_hombre2_gris,tam_img);
```

```
%% penultima foto de hombres
```

```
img_hombre_penultima =imread(fullfile(dir_hombres,archivos_hombres(end-  
1).name));
```

```
img_hombre_penultima_gris=rgb2gray(img_hombre_penultima);
```

```
img_hombre2_pen_redimensionada=imresize(img_hombre_penultima_gris,tam_i  
mg);
```

```
%% segunda foto de mujeres
```

```
img_mujer2=imread(fullfile(dir_mujeres,archivos_mujeres(2).name));
```

```
img_mujer2_gris=rgb2gray(img_mujer2);
```

```
img_mujer2_redimensionada=imresize(img_mujer2_gris,tam_img);
```

```
%% penultima foto de mujeres
```

```
img_mujer2_penultima=imread(fullfile(dir_mujeres,archivos_mujeres(end-  
1).name));
```

```
img_mujer2_penultima_gris=rgb2gray(img_mujer2_penultima);
```

```
img_mujer2_penultima_redimensionada=imresize(img_mujer2_penultima_gris,ta  
m_img);
```

```
%% mostrar imagenes
```

```
figure;
```

```
subplot(2,2,1);
```

```
imshow(img_hombre2_redimensionada,[]);
```

```
title('Hombre - Segunda foto');
```

```
subplot(2,2,2);  
imshow(img_hombre2_pen_redimensionada,[]);  
title('Hombre - Penultima foto');
```

```
subplot(2,2,3);  
imshow(img_mujer2_redimensionada,[]);  
title('Mujer - Segunda foto');
```

```
subplot(2,2,4);  
imshow(img_mujer2_penultima_redimensionada,[]);  
title('Mujer - Penultima foto');
```

```
%% imagen con muestra aleatorio de 20 fotos con su etiqueta predicha
```

```
etiquetas_pred=cell(num_val_imagenes,1);  
etiquetas_pred(y_hat>0)={'Hombre'};  
etiquetas_pred(y_hat<=0)={'Mujer'};
```

```
%% seleccion aleatoria de 20 imagenes
```

```
idx=randperm(num_val_imagenes,20);
```

```
%% crear una figura para mostrar las imagenes con etiquetas predichas
```

```
figure
```

```
for i = 1:20
```

```
    subplot(4,5,i);  
    img=reshape(xtest(idx(i,:),:),tam_img);  
    imshow(img,[]);  
    title(['Pred:',etiquetas_pred{idx(i)}]);
```

```
end
```