



FORMATO DE TRABAJO FINAL

I. PORTADA

UNIVERSIDAD TÉCNICA DE AMBATO

Facultad de Ingeniería en Sistemas, Electrónica e Industrial

“Proyecto Académico de Fin de Semestre: abril -septiembre 2022”

Título: Reconocimiento de placas vehiculares

Carrera: Software

Unidad de Organización Curricular: Profesional

Línea de Investigación:

Nivel y Paralelo: Séptimo

Alumnos participantes: Barriga Sánchez Ludwing José

Núñez Arroba Noelia Abigail

Módulo y Docente: Inteligencia Artificial – Ing. Rubén Nogales

II. INFORME DEL PROYECTO

1.1 Título

Reconocimiento de placas vehiculares

1.2 Objetivos

Objetivo general

Desarrollar un sistema de reconocimiento de caracteres mediante la aplicación de algoritmos de inteligencia artificial para el registro de placas de vehículos.

Objetivos Específicos

- Calcular el error del modelo de cada algoritmo aplicado.
- Establecer el algoritmo a ocupar mediante el menor error de testeo y tiempo de ejecución.

1.3 Introducción



Hoy en día existen varios aspectos de la vida donde se utilizan algoritmos de Inteligencia artificial. Uno de estos es en el reconocimiento de imágenes por computadora, la cual trata de reconocer elementos de interés dentro de una imagen [1]; esto tiene muchos usos ya sea en la medicina, en construcción o en la agricultura. En el presente trabajo, usaremos el reconocimiento de imágenes para detectar las letras de una placa de vehículo haciendo uso de algoritmos de aprendizaje supervisado como son: Redes neuronales artificiales, K-Nearest-Neighbor, Support Vector Machine y árboles de decisión.

1.4 Materiales y Metodología

Marco teórico

Recolección y elaboración de datos

Para la recolección de datos, se hizo uso de imágenes de placas vehiculares, las cuales sirvieron para la generación del Data Set, el cual está conformado por imágenes binarizadas, cada una guardada en un vector de 784 características. El procesamiento que se llevó a cabo consta de un tratamiento a la imagen (limpieza de ruido, aumento de contraste), para luego extraer cada letra y número perteneciente a la misma. Además, se realizó un aumento de datos, girando la imagen en 45, 90, 135, 180, 225 y 270 grados, además que una disminución del 25, 50 y 75 por ciento.

Las etiquetas serán colocadas del 1 al 10 para los números del 0 al 9, y del 11 al 36 para las letras mayúsculas.

Materiales y métodos

Los materiales utilizados para esta práctica son: computadora, herramienta de software MATLAB para el tratamiento de imágenes, creación del Data Set, aplicación de algoritmos, y creación del software.

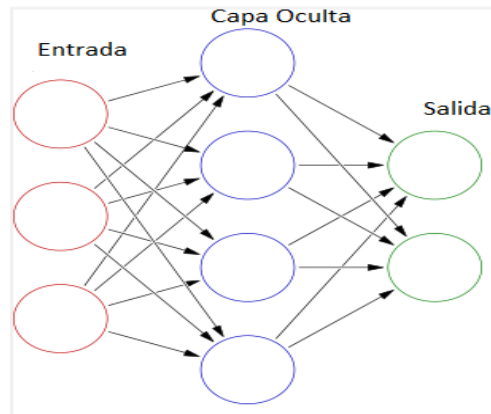
Para realizar el siguiente trabajo se tomó en cuenta el uso de Redes Neuronales artificiales, K-Nearest-Neighbor (KNN), Support Vector Machine (SVM) y árboles de decisión (DT).

Data Set

El Data Set está conformado por 630000 datos, los cuales se dividen en 75 por ciento para entrenamiento y 25 por ciento para testeo. Además, está balanceado teniendo 13125 datos de cada clase.

Red Neuronal artificial

La red neuronal artificial que se ocupó para este trabajo es del tipo Feed-Forward Backpropagation.



Para la capa oculta se utilizaron 25 neuronas, y su función de activación fue una RELU. Mientras que, para la capa de salida, la función de activación utilizada fue una softmax. Además, se decidió el uso de la función de regularización, de esta manera disminuir el coste original; el valor del factor de regularización ocupado fue de 1.0e-6.

K-Nearest-Neighbor(KNN)

KNN es un algoritmo no parametrizado el cual mide la distancia existente entre observaciones y selecciona al vecino más cercano. Para esto se utilizó la fórmula de la distancia euclidiana.

$$d(x, y) = \sqrt{\sum_{i=1}^n (y_i - x_i)^2}$$

Para este caso se utilizó el Data Set de entrenamiento debido a que posee más datos.

```
clc
close all;
clear all;
%cargar dataset
m=load('datasetBinarizado.mat');
DSX=m.dataset.train.imagenes;
DSY=m.dataset.train.labels;

%valor de k
k=round(log10(length(DSX)));
if rem(k,2)==0
    k=k+1;
end
```



```
% distancia euclidiana (vector)
V=[];
for i=1: length(DSX)
    r=sqrt(sum((DSXN(12,:)-DSX(i,:)).^2));
    V=[V;r DSX(i)];
end
%ordenar distancias
[~, s] = sort(V(:, 1));
V=V(s, :);
%Tomar las K primeras
P=[V(1:k, :)];
%contar número de cada etiqueta
o=unique(P(:,2));
pr=[];
for i=1:length(o)
    n=numel(find(P(:,2)==o(i, :)));
    pr=[pr;o(i),n];
end
%Tomar etiqueta con más probabilidad
[res, index]= max(pr(:,2));
respuesta=pr(index,1);
```

Support Vector Machine (SVM)

Para la aplicación de este algoritmo se utilizó la toolbox de MATLAB, siendo la APP escogida “Classification Learner”. Tomando como datos de entrada el Data Set de entrenamiento.



UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL
CARRERA DE SOFTWARE

Cdla. Universitaria (Predios Huachi) / Casilla 334 / Telefax: 03-2851894 – 2411537, Correo Electrónico: carrera.sistemas@uta.edu.ec
AMBATO-ECUADOR



New Session from Workspace

Data set

Data Set Variable

DSX 472507x784 double

☒ Use columns as variables
☐ Use rows as variables

Response

☐ From data set variable
☒ From workspace

DSY 472507x1 double ...

Predictors

	Name	Type	Range
<input checked="" type="checkbox"/>	column_1	double	0 .. 1
<input checked="" type="checkbox"/>	column_2	double	0 .. 1
<input checked="" type="checkbox"/>	column_3	double	0 .. 1
<input checked="" type="checkbox"/>	column_4	double	0 .. 1

[How to prepare data](#)

Validation

☒ **Cross-Validation**
Protects against overfitting by partitioning the data set into folds and estimating accuracy on each fold.

Cross-validation folds: 5

☐ **Holdout Validation**
Recommended for large data sets.

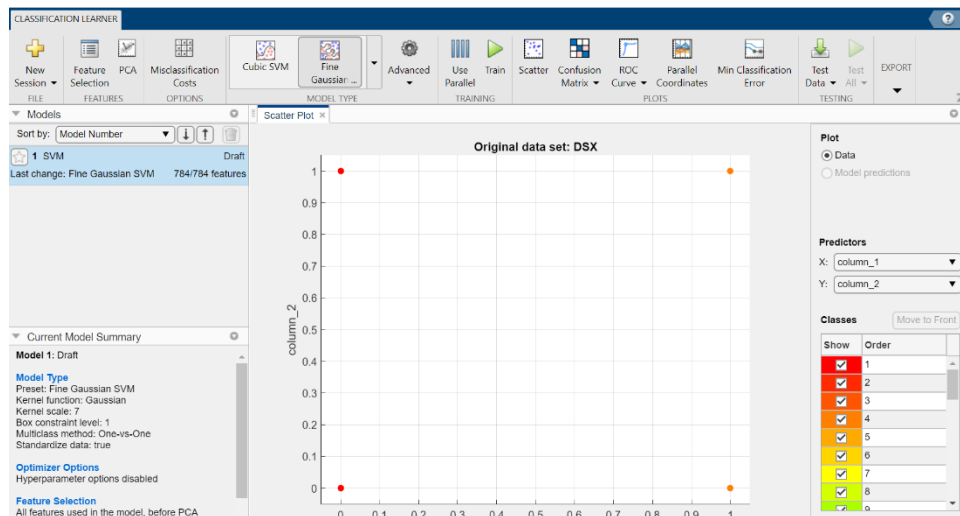
Percent held out: 25

☐ **Resubstitution Validation**
No protection against overfitting. The app uses all the data for both training and validation.

[Read about validation](#)

Response variable is numeric. Distinct values will be interpreted as class labels.

[Start Session](#) [Cancel](#)



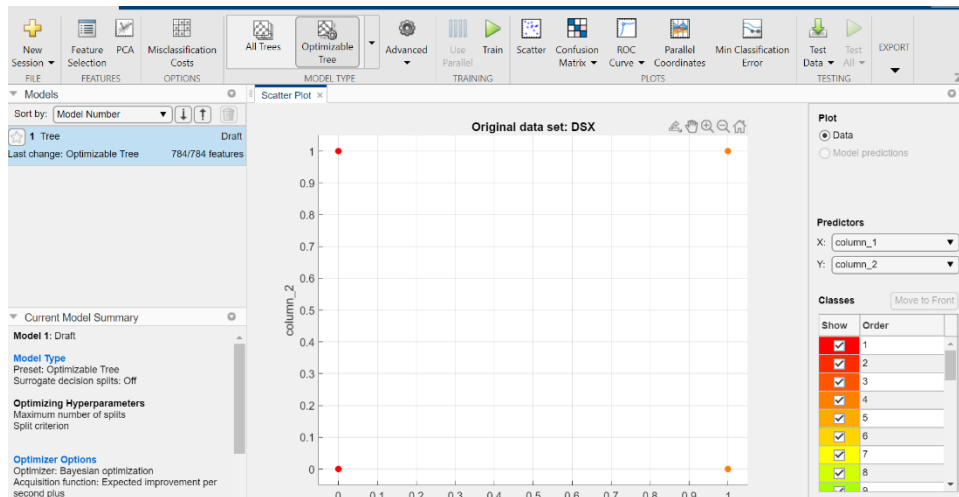
Árboles de decisión (DT)

Para los árboles de decisión se utilizó la toolBox de MATLAB. Siguiendo los mismos pasos que en SVM, pero en esta ocasión se utilizó el tipo de modelo “Optimizable Tree”.



UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL
CARRERA DE SOFTWARE

Cdla. Universitaria (Predios Huachi) / Casilla 334 / Telefax: 03-2851894 – 2411537, Correo Electrónico: carrera.sistemas@uta.edu.ec
AMBATO-ECUADOR



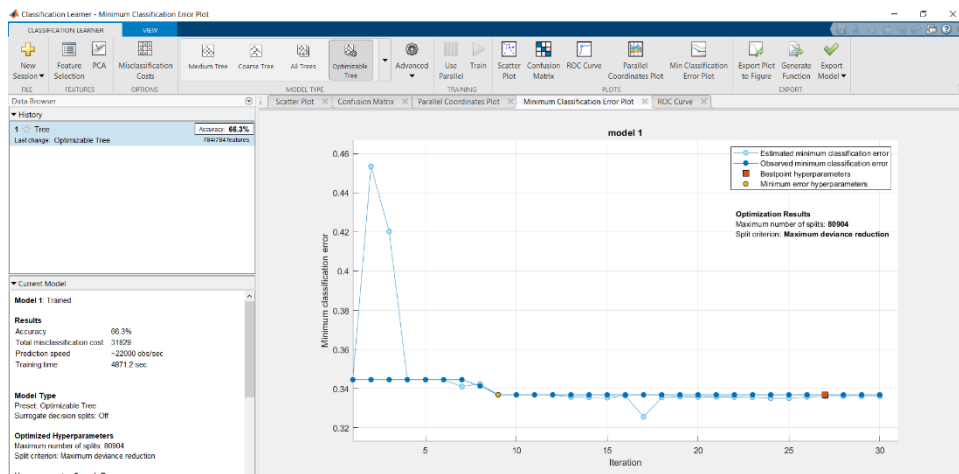
1.5 Resultados y Discusión

Una vez utilizado los diferentes algoritmos se obtuvieron los siguientes resultados:

Árboles de decisión

Con el uso de árboles de decisión se obtuvo los siguientes resultados:

El modelo obtuvo un 66.3% de exactitud, por ende, existe un 33.7% de error en el modelo, por lo cual varias predicciones son fallidas.

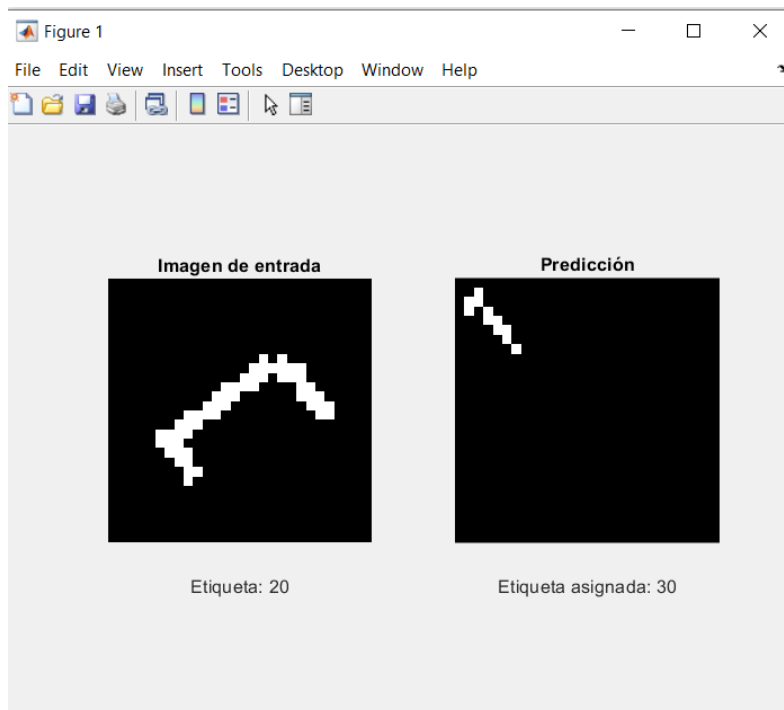


Para realizar el testeado de este modelo se desarrolló el siguiente código:

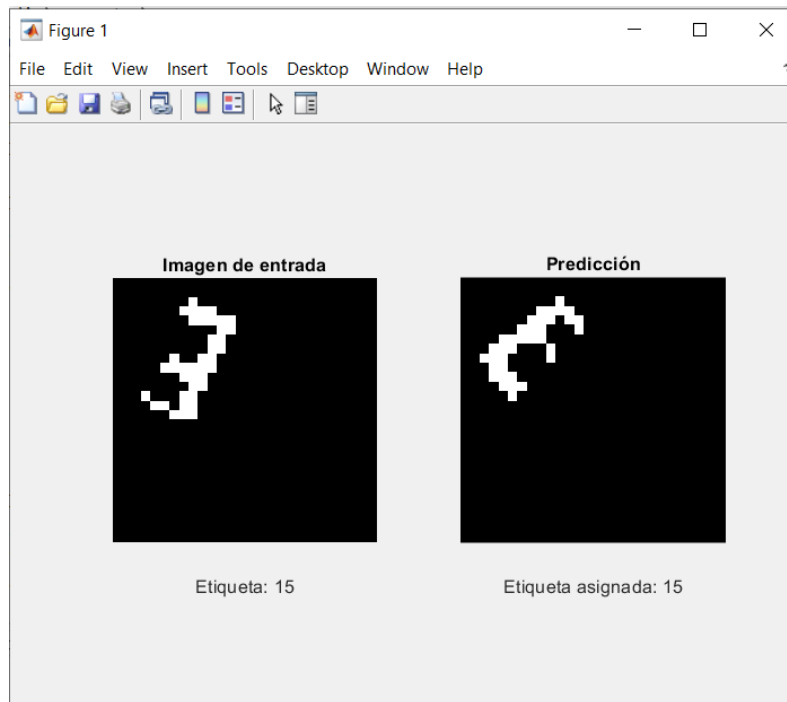


```
dt.m x redNeuronal.m x dtTest.m x prueba.m x +
1 - clc
2
3 - load datasetBinarizado.mat
4 - dsX=dataset.test.imagenes;
5 - dsY=dataset.test.labels;
6
7 - a = 1;
8 - b = length(dataset.test.imagenes);
9 - red = randi([a b],1);
10 - imgTest = dataset.test.imagenes(red,:);
11 - labelTest = dataset.test.labels(red, 1);
12
13 - subplot(1,2,1)
14 - imshow(reshape(imgTest, [28,28]));
15 - title("Imagen de entrada");
16 - xlabel("Etiqueta: " + labelTest);
17
18 - etiquetaTest = trainedModel.predictFcn((imgTest));
19
20 - subplot(1,2,2)
21 - imshow(reshape(dataset.train.imagenes(etiquetaTest,:), [28 28]))
22 - title("Predicción");
23 - xlabel("Etiqueta asignada: " + etiquetaTest)
24
```

Predicción errónea



Predicción acertada



ANN

Con el uso del algoritmo de ANN se pudo obtener un porcentaje de error del 60% aproximadamente y un porcentaje de exactitud del 40%, para realizar los respectivos testeos se desarrolló el siguiente código.

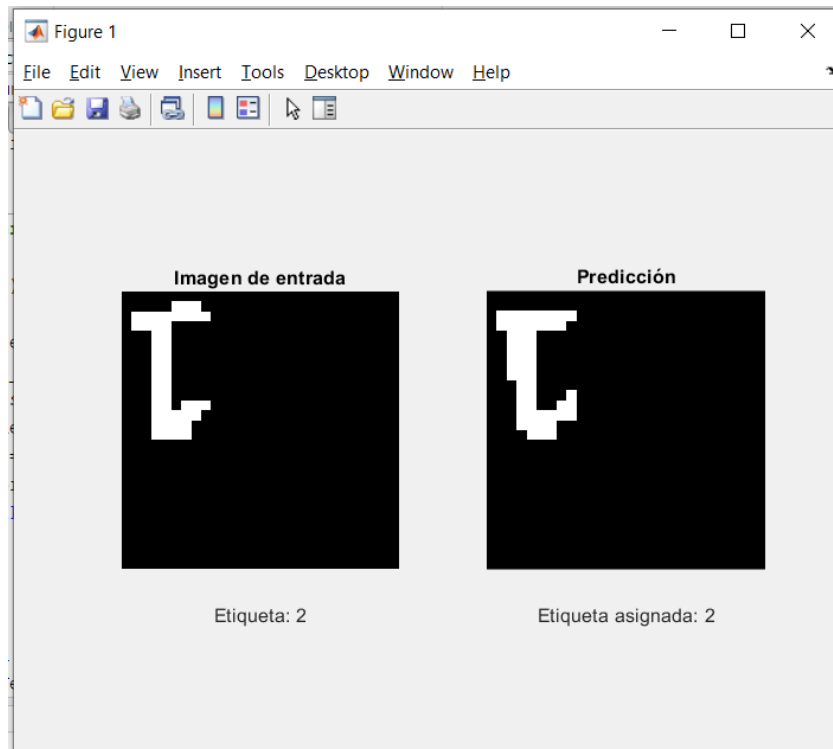


```
classProject.m x test.m x +
1 -   clc,clear,close all;
2 -   load('trainDS.mat')
3
4 -   img = imread('k.png');
5 -   img = mat2gray(imbinarize(rgb2gray(img)));
6 -   img= ~img;
7 -   entrada=reshape(img,[1 784]);
8
9 -   metaParameters.numIterations = 100;
0 -   metaParameters.lambda = 0.1e-6;
1 -   metaParameters.reluThresh = 1e-4;
2
3 -   transferFunctions{1} = 'none';
4 -   transferFunctions{2} = 'relu';
5 -   transferFunctions{3} = 'softmax';
6
7 -   [Z, A] = ffnnmTest(entrada, W, transferFunctions, metaParameters);
8
9 -   P = A(end);
0 -   [dummyVar, predictedLabels] = max(P, [], 2);
1
2 -   figure();
3 -   subplot(2,2,1);
4 -   imshow(img);
5 -   title("Imagen de entrada con clase 21");
6 -   xlabel("Clase predicha: "+ predictedLabels);
```

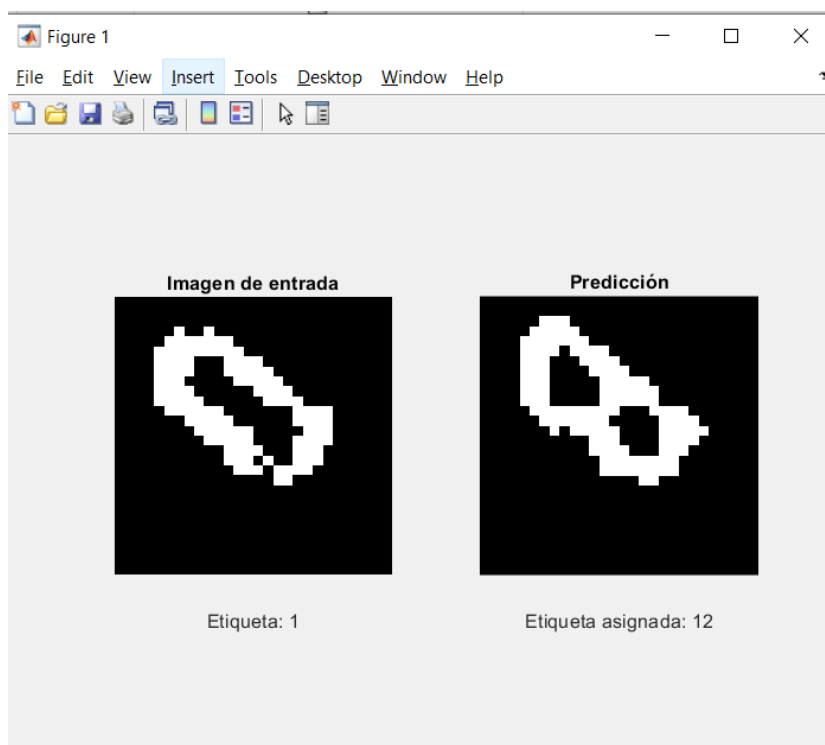
KNN

Con el uso del algoritmo de KNN desarrollado se encontró una mejor exactitud en el modelo, ya que predice correctamente 8 de 10 entradas, por lo que se puede deducir que tiene un acierto de 80% y un error del 20%, a continuación, se observa algunos de los ejemplos capturados.

Predicción acertada de KNN



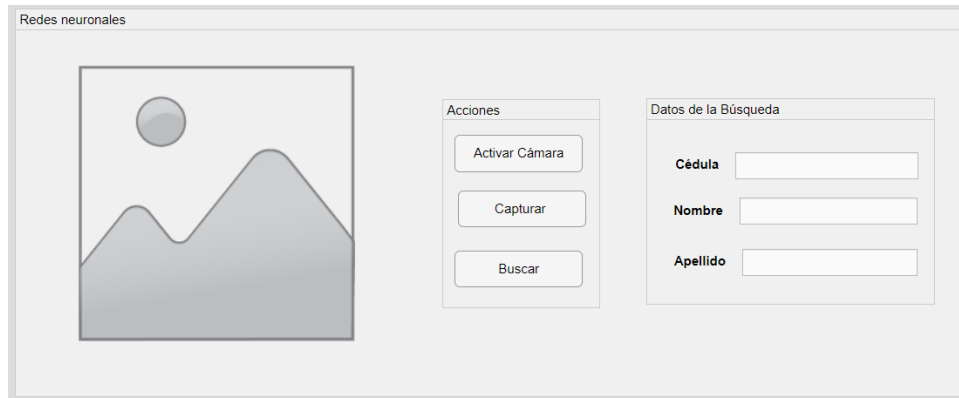
Predicción errónea con KNN



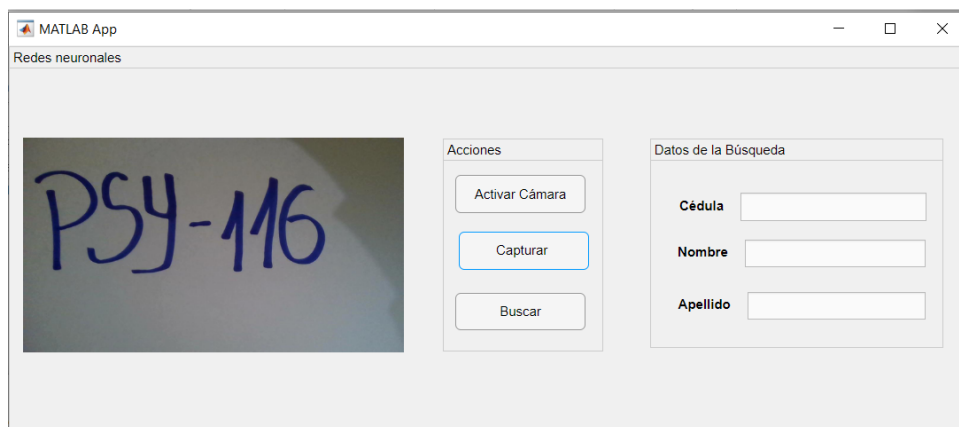
Aplicación



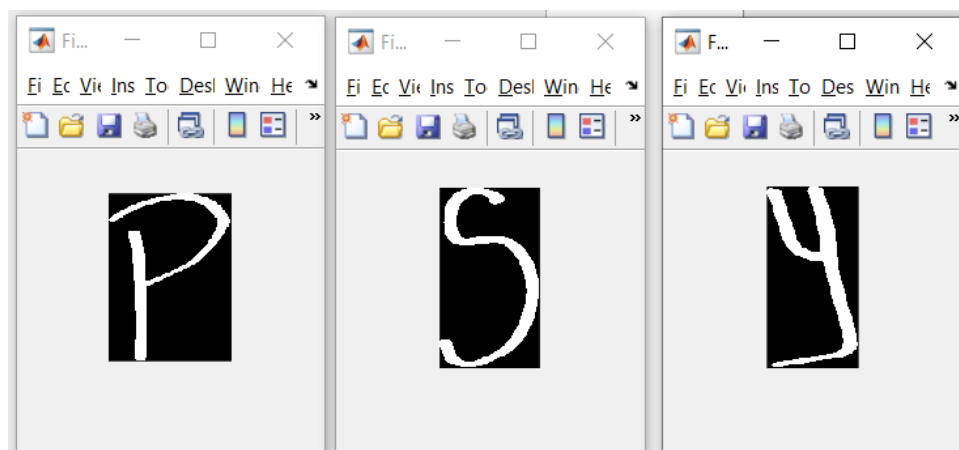
La presenta interfaz cuenta con tres botones para realizar las acciones posibles, cabe recalcar que para el uso de la cámara se usó plugins pertenecientes a MatLab.

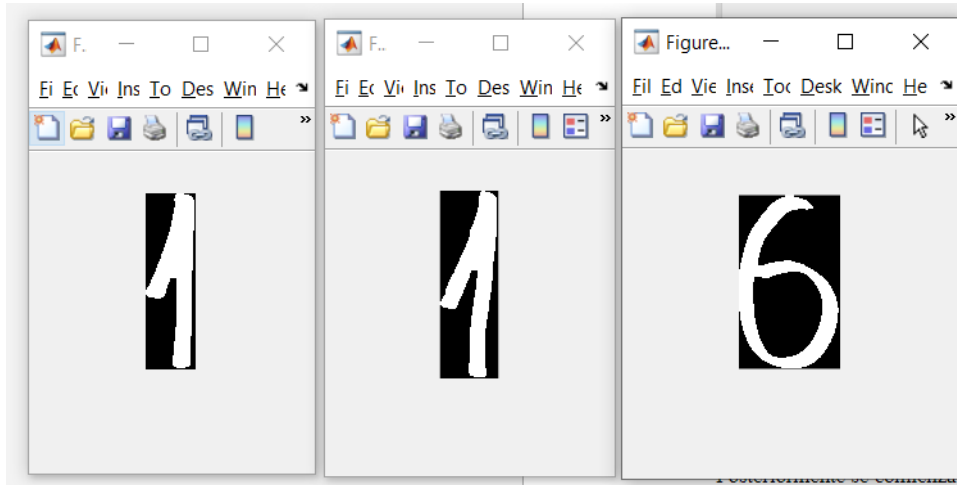


La acción de Capturar Imagen, toma una instantánea de la placa deseada.



Posteriormente se comienza a leer carácter por carácter con el uso del BoundigBox y se obtiene el siguiente resultado.





1.6 Conclusiones

Mediante el uso de Matlab y las respectivas Toolbox se pudo contemplar el entrenamiento y testeo de los modelos de aprendizaje de una manera sencilla. Aunque los resultados obtenidos no fueron los esperados.

El procesamiento de estos modelos puede tardar horas dependiendo la cantidad de datos que se obtenga y la herramienta utilizada.

Aunque no exista un método para conocer cuál de los modelos es mejor para el uso en la aplicación, se puede tener una idea al probar todos, utilizando así aquel que de mayor exactitud en los resultados tanto en test como On-line.

1.7 Referencias bibliográficas

- [1] C. Sandoval, «Reconocimiento de imágenes: Conceptos básicos,» 9 Noviembre 2021. [En línea]. Available: <https://www.lisainsurtech.com/es/2021/11/09/reconocimiento-de-imagenes-conceptos-basicos/>. [Último acceso: 25 Julio 2022].