



**EDUCACIÓN**  
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO  
NACIONAL DE MÉXICO



TECNOLÓGICO NACIONAL DE  
MÉXICO  
INSTITUTO TECNOLÓGICO DE  
SAN LUIS POTOSÍ

---

# Ingeniería en sistemas computacionales

Evaluación

Prof:

CORDERO MARTINEZ STEPHANIE

Alumnos:

CONTRERAS MARTINEZ ROBERTO CARLOS

INTELIGENCIA ARTIFICIAL

5:00 – 6:00 pm

10/Diciembre/24

Link del Repositorio de GitHub donde se encuentra el programa de la Evaluación e Imágenes que se utilizan.

[https://github.com/RobertoContrerasM/IA\\_PYTHON\\_EVALUACION\\_RCCM](https://github.com/RobertoContrerasM/IA_PYTHON_EVALUACION_RCCM)

## REPORTE DE LA EVALUACION

CONTRERAS MARTINEZ	ROBERTO CARLOS	Identificación de tipos de vehículos (coches, camiones, motocicletas)
2. Adicional el alcance 1 el programa ahora realizara la predicción tomando una imagen desde una cámara web.		60%

### Código:

```
import os
import cv2
import numpy as np
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense
from tensorflow.keras.preprocessing.image import ImageDataGenerator

# Ruta del dataset
data_dir = "C:/Users/VRNK1/Documents/python/dataset"

# Parámetros del modelo
img_size = (64, 64)
batch_size = 32

# Preparación del dataset con ImageDataGenerator
train_datagen = ImageDataGenerator(rescale=1.0/255, validation_split=0.2)

# Generador de entrenamiento
train_generator = train_datagen.flow_from_directory(
    os.path.join(data_dir, 'train'), # Ruta a la carpeta de entrenamiento
    target_size=img_size,
    batch_size=batch_size,
    class_mode='categorical',
    subset='training'
)
```

```

# Generador de validación
validation_generator = train_datagen.flow_from_directory(
    os.path.join(data_dir, 'validation'), # Ruta a la carpeta de validación
    target_size=img_size,
    batch_size=batch_size,
    class_mode='categorical',
    subset='validation'
)

# Crear el modelo
model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(64, 64, 3)),
    MaxPooling2D((2, 2)),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Flatten(),
    Dense(128, activation='relu'),
    Dense(len(train_generator.class_indices), activation='softmax')
])

# Compilar el modelo
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

# Entrenar el modelo
model.fit(
    train_generator,
    validation_data=validation_generator,
    epochs=10
)

# Guardar el modelo
model.save('vehicle_classifier_model.h5')

# Función para predecir imágenes
def predict_vehicle(image_path, model):
    image = cv2.imread(image_path)
    if image is None:
        raise ValueError("La imagen no pudo ser cargada. Verifica la ruta.")

    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB) # Convierte de BGR a RGB
    image = cv2.resize(image, img_size) # Cambia el tamaño de la imagen
    image = np.expand_dims(image, axis=0) # Añade la dimensión del batch
    image = image / 255.0 # Normaliza la imagen

    # Realiza la predicción
    predictions = model.predict(image)
    class_idx = np.argmax(predictions) # Obtén el índice de la clase con mayor
probabilidad
    return class_idx, predictions[0]

```

```

# Cargar el modelo
model = tf.keras.models.load_model('vehicle_classifier_model.h5')

# Mapeo de clases
class_labels = list(train_generator.class_indices.keys()) # Asegúrate que las clases
estén bien mapeadas

# Predicción desde una imagen
image_path = "C:/Users/VRNK1/Documents/python/dataset/test_image.jpg" # Cambia por la
ruta de tu imagen
class_idx, probabilities = predict_vehicle(image_path, model)
print(f"La imagen pertenece a la clase: {class_labels[class_idx]} con probabilidad
{probabilities[class_idx]:.2f}")

# Predicción desde cámara web
cap = cv2.VideoCapture(0)
while True:
    ret, frame = cap.read()
    if not ret:
        print("No se pudo acceder a la cámara")
        break

    # Procesar la imagen capturada
    resized_frame = cv2.resize(frame, img_size)
    normalized_frame = resized_frame / 255.0
    input_frame = np.expand_dims(normalized_frame, axis=0)

    # Predicción
    predictions = model.predict(input_frame)
    class_idx = np.argmax(predictions)
    label = class_labels[class_idx]

    # Mostrar el resultado en la imagen
    cv2.putText(frame, f"{label}", (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0),
2)
    cv2.imshow('Vehicle Classifier', frame)

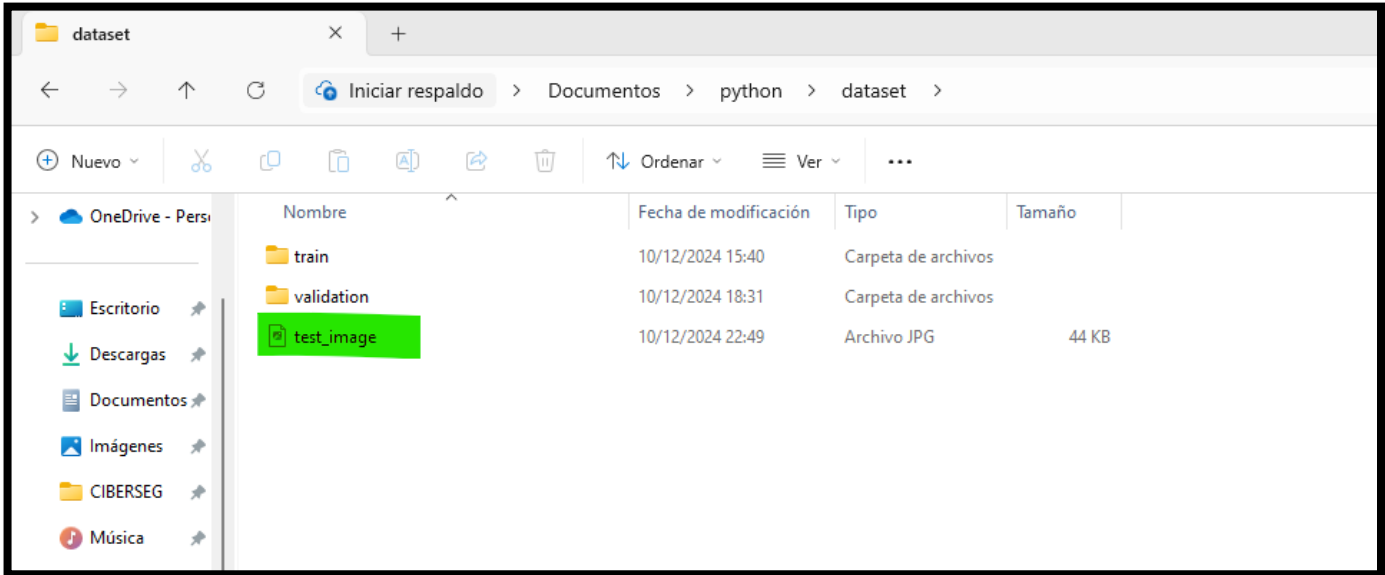
    # Salir con 'q'
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()

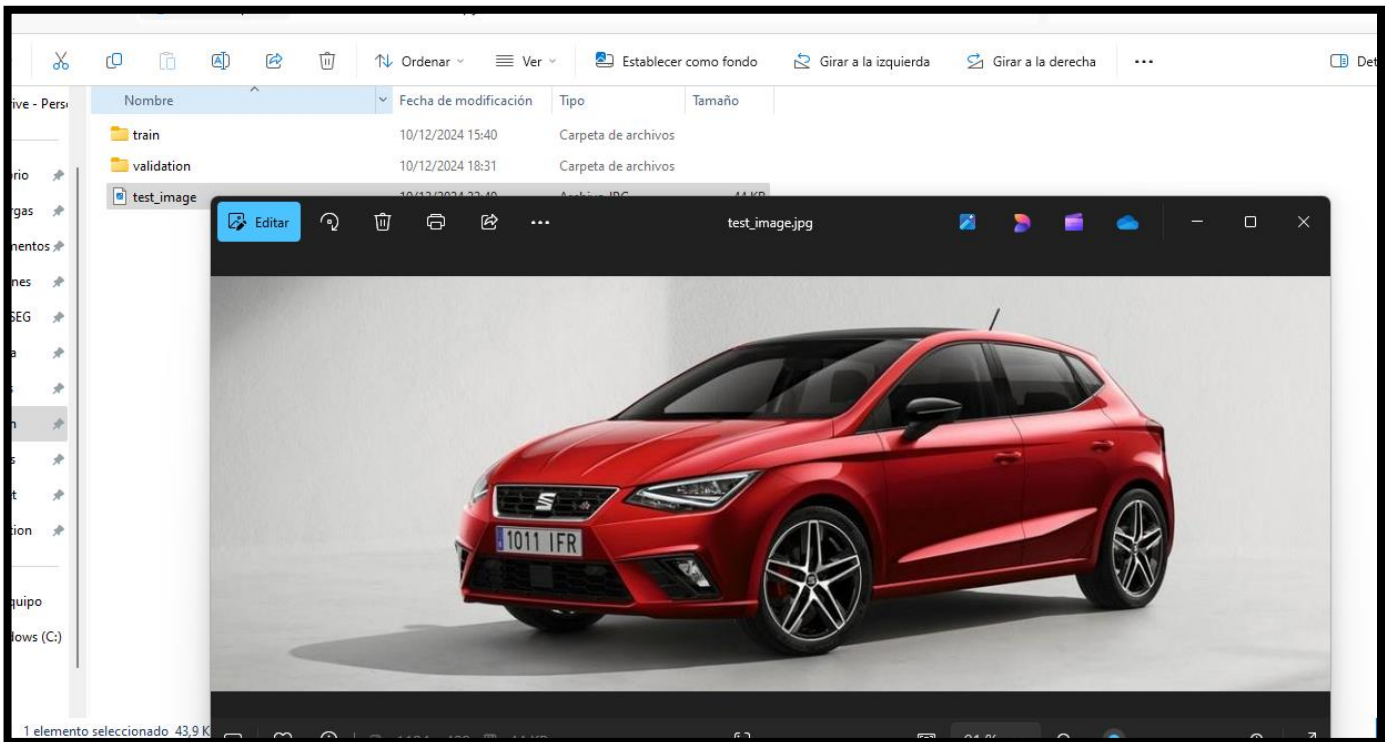
```

## EVIDENCIA:

### Evidencia con imagen cargada y con el dataset



**Imagen de coche, el cual al dar el entrenamiento y la clasificación debería ponerlo en la clase de coches.**

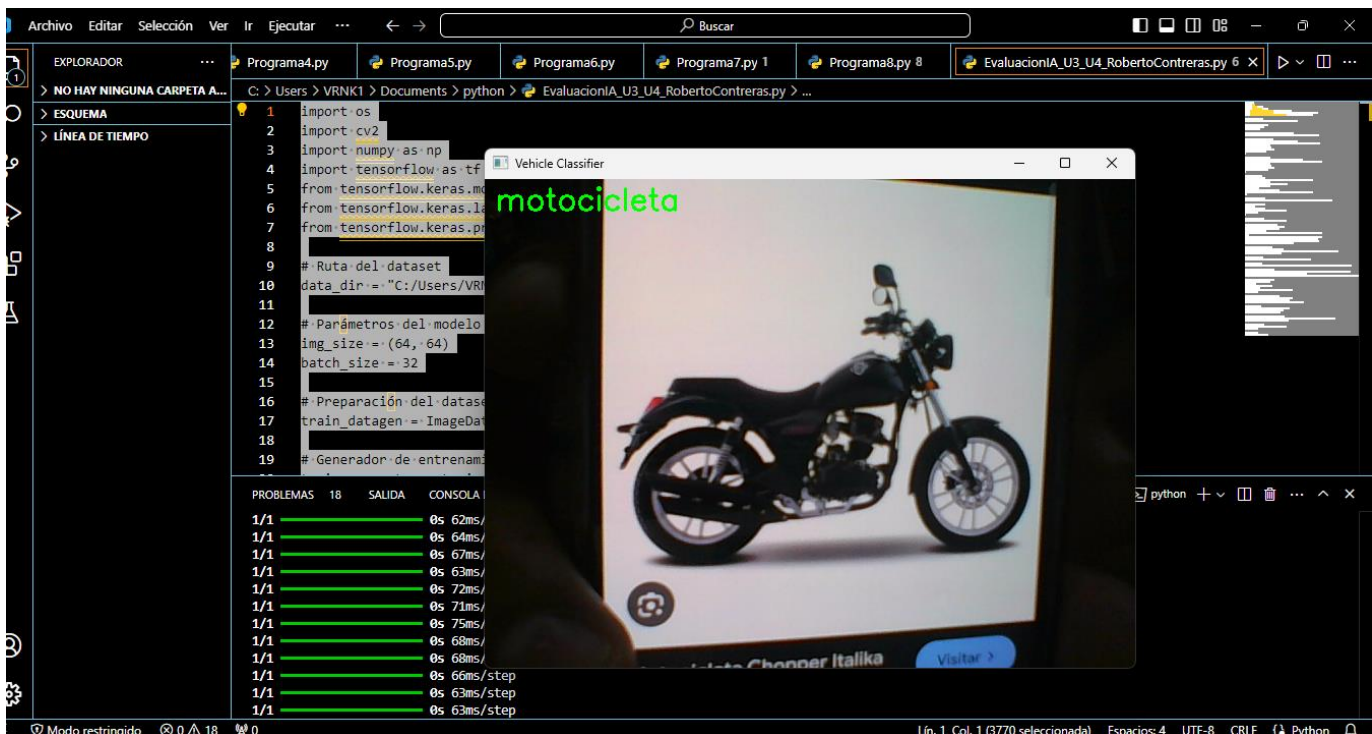


```
1 import os
2 import cv2
3 import numpy as np
4 import tensorflow as tf
5 from tensorflow.keras.models import Sequential
6 from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense
7 from tensorflow.keras.preprocessing.image import ImageDataGenerator
8
9 # Ruta del dataset
10 data_dir = "C:/Users/VRNK1/Documents/python/dataset"
11
12 # Parámetros del modelo
13 img_size = (64, 64)
14 batch_size = 32
15
16 # Preparación del dataset con ImageDataGenerator
17 train_datagen = ImageDataGenerator(rescale=1.0/255, validation_split=0.2)
18
19 # Generador de entrenamiento
20 train_generator = train_datagen.flow_from_directory(
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

11/11 14s 1s/step - accuracy: 0.9678 - loss: 0.2316 - val\_accuracy: 0.7949 - val\_loss: 0.5067  
WARNING:absl:You are saving your model as an HDF5 file via 'model.save()' or 'keras.saving.save\_model(model)'. This file format is considered legacy. We recommend using instead the native Keras format, e.g. 'model.save('my\_model.keras')' or 'keras.saving.save\_model(model, 'my\_model.keras')'.  
WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be built. 'model.compile\_metrics' will be empty until you train or evaluate the model.  
1/1 0s 118ms/step  
La imagen pertenece a la clase: coches con probabilidad 0.56  
1/1 0s 58ms/step  
1/1 0s 66ms/step  
1/1 0s 75ms/step  
1/1 0s 62ms/step  
1/1 0s 72ms/step

Resultado de pertenencia a la clase coches.

## EVIDENCIA CON LA IDENTIFICACION DE VEHICULOS MEDIANTE LA WEBCAM





Archivo Editar Selección Ver Ir Ejecutar ... Buscar

EXPLORADOR ... Programa4.py Programa5.py Programa6.py Programa7.py 1 Programa8.py 8 EvaluacionIA\_U3\_U4\_RobertoContreras.py 6 X

> NO HAY NINGUNA CARPETA A...> ESQUEMA> LÍNEA DE TIEMPO

C: > Users > VRNK1 > Documents > python > EvaluacionIA\_U3\_U4\_RobertoContreras.py > ...


```
1 import os
2 import cv2
3 import numpy as np
4 import tensorflow as tf
5 from tensorflow.keras.m
6 from tensorflow.keras.l
7 from tensorflow.keras.p
8
9 # Ruta del dataset
10 data_dir = "C:/Users/VRN
11
12 # Parámetros del modelo
13 img_size = (64, 64)
14 batch_size = 32
15
16 # Preparación del datase
17 train_datagen = ImageDa
18
19 # Generador de entrenam
```

PROBLEMAS 18 SALIDA CONSOLA

1/1	0s 45ms/
1/1	0s 50ms/
1/1	0s 48ms/
1/1	0s 45ms/
1/1	0s 48ms/
1/1	0s 44ms/
1/1	0s 52ms/
1/1	0s 47ms/
1/1	0s 45ms/
1/1	0s 48ms/step
1/1	0s 45ms/step
1/1	0s 46ms/step

Vehicle Classifier

coches El Español



Así es el coche eléctrico de Xiaomi: viene 'cargado' par... Visitar >

Las imágenes pueden estar sujetas a derechos de autor. Más información

python + v [ ] ... ^ x

Modo restringido 0 18 0

Lín. 1, Col. 1 (3770 seleccionada) Espacios: 4 UTF-8 CRLF Python

Archivo Editar Selección Ver Ir Ejecutar ... Buscar

EXPLORADOR ... Programa4.py Programa5.py Programa6.py Programa7.py 1 Programa8.py 8 EvaluacionIA\_U3\_U4\_RobertoContreras.py 6 X

> NO HAY NINGUNA CARPETA A...> ESQUEMA> LÍNEA DE TIEMPO

C: > Users > VRNK1 > Documents > python > EvaluacionIA\_U3\_U4\_RobertoContreras.py > ...


```
1 import os
2 import cv2
3 import numpy as np
4 import tensorflow as tf
5 from tensorflow.keras.m
6 from tensorflow.keras.l
7 from tensorflow.keras.p
8
9 # Ruta del dataset
10 data_dir = "C:/Users/VRN
11
12 # Parámetros del modelo
13 img_size = (64, 64)
14 batch_size = 32
15
16 # Preparación del datase
17 train_datagen = ImageDa
18
19 # Generador de entrenam
```

PROBLEMAS 18 SALIDA CONSOLA

1/1	0s 54ms/
1/1	0s 51ms/
1/1	0s 51ms/
1/1	0s 46ms/
1/1	0s 52ms/
1/1	0s 46ms/
1/1	0s 47ms/
1/1	0s 48ms/
1/1	0s 52ms/
1/1	0s 43ms/step
1/1	0s 50ms/step
1/1	0s 46ms/step

Vehicle Classifier

camiones



python + v [ ] ... ^ x

Modo restringido 0 18 0

Lín. 1, Col. 1 (3770 seleccionada) Espacios: 4 UTF-8 CRLF Python