

Clasificador y Detector de Personas A través de Visión Artificial

Tapia Vazquez Alan Eduardo
Facultad De Ingeniería Eléctrica y Electrónica
Universidad Veracruzana
Veracruz, México
zs19002886@estudiantes.uv.mx

Resumen—Este proyecto de comprende de la posibilidad de detectar personas dentro de un área fijada por una camara, procesando dicha información y determinando la cantidad de personas que han estado en el mismo lugar durante un lapso de tiempo.

Index Terms—vision artificial, clasificación de imagenes, machine learning.

I. INTRODUCCIÓN

EN las instituciones escolares la seguridad de los estudiantes es una de las partes más importantes a tener en cuenta, sin embargo, el uso de personal especializado no siempre cubre la necesidad en amplio panorama, es por eso, que en diversas instituciones el uso de cámaras de seguridad es esencial, pero esto también recae sobre la tentación de no cumplir con el mayor nivel de la necesidad. Es por eso, que la visión por computadora ha tomado un valor importante en los últimos años, pues ha permitido incrementar la cantidad de herramientas que hacen uso de esta, brindando así la oportunidad de mejorar e innovar la forma en que la seguridad en una institución es aplicada.

II. OBJETIVOS

- Comprender el uso de la visión por computadora y sus ventajas.
- Implementar la visión por computadora en un entorno controlado.
- Obtener resultados que sirvan para el análisis administrativo de la institución.

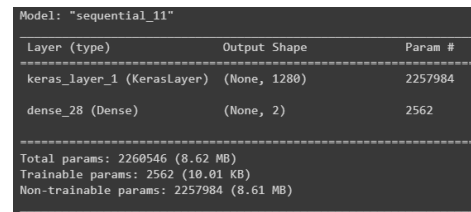
III. ESTADO DEL ARTE

El problema mencionado ha tenido un panorama de pasa un tanto desapercibido, sin embargo, cuenta con la difusión de diversos autores que han puesto en el plano los beneficios que conlleva una seguridad inteligente. En esencia uno de estos autores han sido Mark A. y Neil S. en [1] refiriéndose a la tecnología de reconocimiento facial como una herramienta que se ha ido integrando de manera obligatoria en el ámbito escolar

abordando así cuestiones de seguridad y registros. Ciertamente los sistemas que permiten el conteo de personas tienen un gran papel, pues así como menciona H. Hakan en [] estos sistemas tienen como objetivo calcular automaticamente estimados del número de personas en lugares interiores y exteriores y pueden ser utilizados en sistemas de videovigilancia.

IV. DISEÑO DEL PROYECTO

Para la creación de este mini proyecto inicialmente se considero hacer uso de TensorFlow pero de una manera distinta, es decir, haciendo uso de tfrecords para realizar un entrenamiento controlado a través del etiquetado de dataset. Consecuentemente se obtuvieron resultados no esperados, por lo que la alternativa fue realizarlo de una manera un tanto más sencilla. A través de un modelo ya entrenado llamado MobilenetV2 hacemos uso de su experiencia para facilitar el entrenamiento de nuestra red. En la figura 1, observamos la estructura base de nuestra red neuronal densa.



Layer (type)	Output Shape	Param #
keras_layer_1 (KerasLayer)	(None, 1280)	2257984
dense_28 (Dense)	(None, 2)	2562

Total params: 2260546 (8.62 MB)
Trainable params: 2562 (10.01 KB)
Non-trainable params: 2257984 (8.61 MB)

Figura 1. Estructura de la red neuronal

El uso de DataSets tiene una importancia grande, puesto que al querer entrenar una red para la clasificación y detección de personas, hacerlo con 1 solo dataset de una sola clase nos impediría tener resultados confiables; es por esto, que se tomó la decisión de hacer uso de 2 datasets uno para la clasificación de Humanos-Personas y otro para la clasificación de Animales, eso podría incrementarse, sin embargo, sería preferible mantener un equilibrio entre cada set de datos, de esta forma evitaremos posibles incongruencias al momento del aprendizaje de nuestra red neuronal.

IV-A. DataSet Person y Dataset Animal

En conjunto de ambos dataset podemos obtener un total de 2892 imágenes:

- 1863 imágenes para el conjunto de datos Person
- 1029 imágenes para el conjunto de datos Animal

Estos datasets fueron descargados e importados de la plataforma Roboflow[], los cuales vienen seccionados en carpetas de train/valid/test esto con el fin de facilitar su movilidad entre carpetas, sin embargo, esto puede llegar a causar problemas al momento de determinar las clases de cada imagen, para eso ocupamos la libreria splitfolder que nos permite dividir en partes especificadas un conjunto de datos de todas las clases, abriendonos una ventana de posibilidad a tener Train/Val/Test con imágenes necesarias pero no faltantes para el entrenamiento posterior,

```
for dirname, __, filenames in os.walk('data_split3'):
    print(dirname)

data_split3
data_split3/val
data_split3/val/Animal
data_split3/val/Person
data_split3/test
data_split3/test/Animal
data_split3/test/Person
data_split3/train
data_split3/train/Animal
data_split3/train/Person
```

Figura 2. Forma de carpetas y división

IV-B. MODELO DE LA RED NEURONAL

Para el uso de nuestra red, se ha utilizado un modelo preentrenado llamado MobileNetV2, el cual nos sirve para la extracción de características de nuestras imágenes, este modelo se hace uso a través de una red neuronal secuencial como se mira en la figura 3.

```
[116] URL = "https://tfhub.dev/google/tf2-preview/mobilenet_v2/feature_vector/4"
feature_extractor = hub.KerasLayer(URL, input_shape=(IMG_SIZE, IMG_SIZE, 3))
feature_extractor.trainable = False

[131] from tensorflow.keras.models import Model, load_model
from tensorflow.keras.layers import Input, Dense, Conv2D, Flatten
from tensorflow.keras import Sequential

red = Sequential()

red.add(feature_extractor)
red.add(Dense(2, activation='softmax'))

red.summary()
```

Figura 3. Modelo programado de la red neuronal

IV-C. ENTRENAMIENTO

Para el entrenamiento de nuestra red neuronal, fue importante considerar cuantas epocas se utilizarían y que tipo de perdida queríamos que existiera, para disminuir la cantidad de la perdida y evitar inconsistencias se opto por hacer uso de $\text{sparse_categorical_crossentropy}$.

Para un mejor resultado de entrenamiento, las imágenes recuperadas de los datasets son previamente procesadas, generando así variaciones de la misma, gracias a esto, nos permite encontrar más características de la imagen y posteriormente realizar una mejor comparación

```
[132] red.compile(
    optimizer='adam',
    loss="sparse_categorical_crossentropy",
    metrics=['accuracy']
)
```

Figura 4. Categorical utilizado

```
r = red.fit(
    train_data_gen,
    validation_data = val_data_gen,
    epochs = 15,
)
```

Figura 5. Epocas utilizadas

IV-D. RESULTADOS DE ENTRENAMIENTO

Grafica obtenida a partir de Training Validation Accuracy

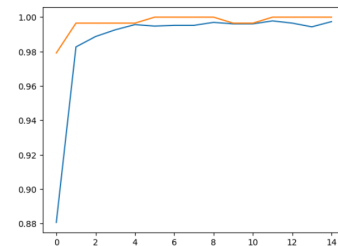


Figura 6. Resultados Accuracy

Grafica obtenida a partir de Training Validation Loss

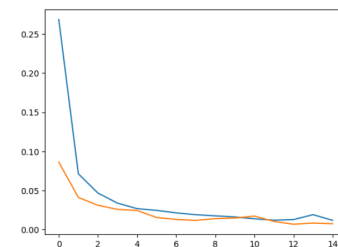
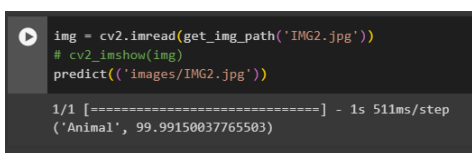


Figura 7. Resultados loss

V. PREDICCIONES E IMPLEMENTACIÓN

Con el modelo entrenado es posible ocuparlo ahora, para poder generar predicciones a partir de imágenes, si bien esta tarea puede empezar pareciendo simple, el tiempo de trabajo detrás es laborioso con lo que surge la pregunta, ¿Realmente está funcionando nuestro modelo preentrenado?, pues para eso ocuparemos funciones que nos ayuden a convertir una imagen a un tensor el cual nos servirá para pasarlo a través de nuestro modelo para crear una predicción a partir de este, obteniendo así los resultados vistos en la figura 8 y 9 mostradas a continuación.



```
img = cv2.imread(get_img_path('IMG2.jpg'))
# cv2_imshow(img)
predict(['images/IMG2.jpg'])

1/1 [=====] - 1s 511ms/step
('Animal', 99.99150037765503)
```

Figura 8. Predicción

VI. CONCLUSIONES

Es gracias a la inteligencia artificial que podemos realizar tareas que antes nos podían llegar a parecer difíciles de realizar, sin embargo, hoy en día podemos ocupar esta tecnología a nuestro favor; el uso de la visión artificial nos permite ir un paso más allá de aquellas personas que no saben utilizarla correctamente, es por eso que muchas instituciones escolares están pensando cada vez más en implementar este tipo de práctica.

El objetivo de este trabajo es poder dar una noción pequeña acerca de como podría ser posible realizar una implementación sencilla en los diferentes centros de computo y laboratorios de prácticas dentro de la universidad con el fin de mantener un registro de los propios estudiantes y profesores, brindando así una mejor seguridad y orden público.