

Comparación de modelos de inteligencia artificial para el reconocimiento de placas vehiculares.

Comparison of artificial intelligence models for vehicle license plate recognition

Paul Pasaca^{1,*}

¹ Carrera de Ingeniería en Sistemas, Universidad Nacional de Loja, Ecuador

* Autor para correspondencia: paul.pasaca@unl.edu.ec

Fecha de recepción del manuscrito: 12/09/2022

Fecha de aceptación del manuscrito: DD/MM/YYYY

Fecha de publicación: DD/MM/YYYY

Resumen—El reconocimiento de placas vehiculares es una tecnología en la cual se detecta la placa de un vehículo para luego hacer uso del reconocimiento óptico de caracteres para crear datos a partir de esta detección. El objetivo de este documento es poder realizar el entrenamiento de 3 modelos pre entrenados haciendo uso de el dataset elegido, con la finalidad de verificar cual es el modelo que devuelve mejores resultados. Para esto se ha utilizado Google Colab que es un entorno adecuado para tareas de aprendizaje automático y que brinda GPUs gratuitas. Dentro de los resultados obtenidos se llegó a determinar que el modelo con mayor nivel de precisión es el VGG16 con un 90 %, en conclusión, al utilizar modelos pre entrenados se reduce en gran medida el tiempo de entrenamiento, de igual forma el nivel de precisión es bastante elevado, ya que estos modelos han sido validados con una gran cantidad de datos.

Palabras clave—alpr, ocr,crisp-dm,placa, vehicular

Abstract—License plate recognition is a technology in which the license plate of a vehicle is detected to then make use of optical character recognition to create data from this detection. The objective of this paper is to train 3 pre-trained models using the chosen dataset, in order to verify which model gives the best results. For this purpose, Google Colab has been used, which is a suitable environment for machine learning tasks and provides free GPUs. Among the results obtained, it was determined that the model with the highest level of accuracy is VGG16 with 90 %. In conclusion, by using pre-trained models, the training time is greatly reduced, and the level of accuracy is also quite high, since these models have been validated with a large amount of data.

Keywords—alpr, ocr,crisp-dm,plate, vehicular

INTRODUCCIÓN

La visión Artificial es una de las diversas rama que abarca la Inteligencia Artificial, haciendo uso de esta se puede obtener, procesar y analizar diferente tipo de información capturada en imágenes, mediante las características propias de los objetos que se encuentran dentro de la imagen. Arrighi *et al.* (2012)

El reconocimiento óptico de caracteres es una tecnología permite reconocer automáticamente los caracteres mediante un mecanismo óptico. OCR es una tecnología que funciona como la capacidad humana de leer, sin embargo, no puede competir con la capacidad de lectura humana. Mithe *et al.* (2013)

MATERIALES Y MÉTODOS

Método Inductivo

Se lo ha utilizado para identificar los inconvenientes que tiene el actual control de ingreso hacia la universidad, al no contar con un sistema el cual controle y gestione el ingreso

de los vehículos hacia la facultad.

Método Deductivo

Se lo ha utilizado para dar una solución adecuada a las necesidades requeridas en el control del ingreso de los vehículos hacia la facultad y a los requerimientos obtenidos de las partes interesadas.

Metodología CRISP-DM

Cuenta con 5 fases: Comprensión del negocio, comprensión de los datos, preparación de los datos modelado, evaluación e implantación.

En la primera fase, se comprende el negocio, aquí se determinan los objetivos del negocio, se evalúa la situación actual y el plan para poder obtener los datos requeridos para el entrenamiento.

En la segunda fase, se recolecta los datos necesarios, se procede a describirlos, para luego explorar como están compuestos y finalmente se verifica la calidad de los datos obtenidos. En la tercera fase, se procede a verificar que los datos se encuentren correctos y equilibrados, pasando por una revisión rigurosa, para evitar errores durante el entrenamiento.

En la cuarta fase, se elige el modelo a implementar, en este caso se eligen 3, luego se construye el modelo dependiendo de las capas a utilizar, se procede a entrenar el modelo con los diferentes parámetros construidos anteriormente, para finalmente verificar los resultados obtenidos entre los 3 modelos elegidos y con diferentes épocas.

En la última fase, se grafican los resultados obtenidos para obtener un mejor panorama del nivel de acierto y pérdida de los modelos. Huber *et al.* (2019)

RESULTADOS

Elección del modelo

La recopilación de los métodos y técnicas, sirvió para determinar la técnica más óptimas, ya que no todas las técnicas de Deep Learning se pueden usar par aun mismo propósito; nuestra finalidad es la detección de objetos, por lo tanto, se recopiló diversos métodos los cuales has sido evaluados con un conjunto de datos llamado MSCOCO el cual contiene 80 categorías, y más de 160 000 imágenes, las cuales están divididas en entrenamiento, validación y prueba *Keras Applications* (s.f.).

Tabla 1: Modelos encontrados, con sus respectivos parámetros.

| Modelo | Tamaño (MB) | Precisión | Parámetros |
|-------------------|-------------|-----------|----------------|
| Xception | 88 | 94,5% | 22,9 millones |
| VGG16 | 528 | 90,1% | 138,4 millones |
| VGG19 | 549 | 90,0% | 143,7 millones |
| ResNet50V2 | 98 | 93,0% | 25,6 millones |
| ResNet101V2 | 171 | 93,8% | 44,7 millones |
| ResNet152V2 | 232 | 94,2% | 60,4 millones |
| InceptionV3 | 92 | 93,7% | 23,9 millones |
| InceptionResNetV2 | 215 | 95,3% | 55,9 millones |
| MobileNetV2 | 14 | 90,1% | 3,5 millones |
| DenseNet201 | 80 | 93,6% | 20,2 millones |
| EfficientNetB7 | 256 | 97,0% | 66,7 millones |
| EfficientNetV2B3 | 59 | 95,8% | 14,5 millones |
| EfficientNetV2L | 479 | 97,5% | 119.0 millones |

Elección del dataset

Se emplea la plataforma Kaggle, la cual es la brinda las herramientas y recursos más importantes para el aprendizaje automático *Kaggle: Your Home for Data Science* (s.f.).

Tabla 2: Datasets encontrados y sus números de datos.

Finalmente se hizo uso del dataset llamado “Car License Plate Detection”, este conjunto de datos contiene 433 imágenes con etiquetas de cuadros delimitadores de las matrículas de automóviles dentro de la imagen. Las etiquetas se proporcionan en formato PASCAL VOC.

Construcción del modelo

Para los 3 modelos utilizados, primeramente, se exportó desde keras, luego se utilizó la función Flatten para aplanar los datos, se agregaron dos capas densas: la primera de 500 neuronas con una activación relu y la segunda con 250 neuronas igualmente con una activación relu. Finalmente, se aplicó una capa densa de salida con activación sigmoid.

| Nombre del dataset | Número de datos |
|--|--------------------------|
| Car License Plate Detection | 433 imágenes |
| | 433 etiquetas |
| license-plate-detection | No definido |
| License Plate Characters - | 209 imágenes |
| Detection OCR | 236 etiquetas |
| Two Wheeler Number / License Plate Detection | 750 imágenes |
| License Plate Digits | 35 directorios, cada uno |
| Classification Dataset | con más de 1000 imágenes |
| Indian License Plates with | 2083 imágenes |
| Labels | 2021 etiquetas |

En la Figura 1 se muestra la importación del modelo VGG16 y la agregación de las capas densas con su respectiva activación.

```
inception_resnet = VGG16(weights="imagenet", include_top=False, input_tensor=Input(shape=(224, 224, 3)))
# .....
headmodel = InceptionResNetV2.output
headmodel = Flatten()(headmodel)
headmodel = Dense(500, activation="relu")(headmodel)
headmodel = Dense(250, activation="relu")(headmodel)
headmodel = Dense(4, activation="sigmoid")(headmodel)
```

Fig. 1: Construcción del modelo

En la Figura 2 se muestra la compilación del modelo VGG16, con un optimizado Adam y una ponderación de 0.0001.

```
model.compile(loss='binary_crossentropy',
              optimizer=tf.keras.optimizers.Adam(learning_rate=1e-4),
              metrics=['accuracy'])
model.summary()
```

Fig. 2: Optimizado y ponderación

Evaluación del modelo

Para realizar una evaluación de cuál es el modelo más óptimo, se procedió a entrenar los modelos con más épocas, los resultados obtenidos se muestran en la Tabla 3.

Tabla 3: Resultados obtenidos de los 3 modelos entrenados.

| Modelo pre entrenado | Épocas | Accuracy | Loss | Val accuracy | Val loss |
|----------------------|--------|----------|--------|--------------|----------|
| InceptionResNetV2 | 50 | 0.9653 | 0.5969 | 0.8506 | 0.6279 |
| | 90 | 0.9740 | 0.5976 | 0.8506 | 0.6632 |
| | 180 | 0.9682 | 0.5972 | 0.8851 | 0.6965 |
| VGG16 | 50 | 0.9566 | 0.5995 | 0.8966 | 0.6085 |
| | 90 | 0.9653 | 0.5984 | 0.8966 | 0.6075 |
| | 180 | 0.9624 | 0.5987 | 0.9425 | 0.6081 |
| InceptionV3 | 50 | 0.9335 | 0.5999 | 0.8736 | 0.6119 |
| | 90 | 0.9653 | 0.5986 | 0.9310 | 0.6076 |
| | 180 | 0.9595 | 0.5989 | 0.8621 | 0.6110 |

DISCUSIÓN

Para el desarrollo de este trabajo se recopiló diversos métodos para la detección de objetos, se logró determinar que la mayoría de estos métodos hacen uso de las redes neuronales convolucionales, basándose en esto se hizo la búsqueda modelos pre entrenados con un alto porcentaje de precisión, cabe mencionar que la cantidad de datos utilizados influirá enormemente en la precisión del modelo al igual que en el

consumo de recursos, por estos motivos hay que ser cuidadoso en la elección del modelo.

A continuación, según los resultados obtenidos, se eligió 3 modelos, los cuales parecían ser los más óptimos, esto se realizó basándose en varios parámetros como precisión, cantidad de datos con el que ha sido probado y el tiempo de entrenamiento. De estos 3 modelos se elegirá solamente uno, el cual obtenga el mayor nivel de precisión con el dataset a utilizar.

Finalmente, se buscó diversos datasets para poder realizar el entrenamiento, cabe resaltar que si un dataset no se encuentra con una suficiente cantidad de datos y con datos incorrectos o ambiguos harán que el porcentaje en la precisión del modelo baje significativamente. Una vez elegido el dataset más adecuado se procede a entrenar el modelo, para esto lo más recomendable es utilizar GPUs gratuitas, ya que, en el caso de no contar con una computadora de altas capacidades, si se utiliza las GPUs no se ocuparán recursos de nuestra computadora. Al finalizar el entrenamiento con los 3 modelos elegidos se graficó los resultados de cada uno de ellos, eligiendo como el mejor al modelo VGG16, puesto que es el de mejor porcentaje de acierto y pérdida.

CONCLUSIONES

La utilización de un modelo pre entrenado hace que el tiempo de entrenamiento sea menor en comparación a un modelo creado desde cero, esto debido a que no tiene que empezar a calcular el peso de las entradas, sino que las equilibra según el paso de las épocas, lo mismo sucede con el nivel de precisión, ya que un modelo pre entrenado ha sido validado con enormes cantidades de datos, por lo tanto, el dataset que se ocupe

AGRADECIMIENTOS

Quiero expresar mi más sincero agradecimiento a todos los docentes que formaron parte de todo mi proceso de formación académica, ya que con sus conocimientos otorgados quedan plasmados en el desarrollo de este trabajo. Además de eso agradezco mi director de trabajo de titulación el Ing.

Pablo Ordoñez, por el apoyo y orientación para que este trabajo se culmine de manera exitosa.

CONTRIBUCIONES DE LOS AUTORES

Conceptualización: Paul Pasaca; metodología: Paul Pasaca; análisis formal: Paul Pasaca.; investigación: Paul Pasaca; recursos: Paul Pasaca; redacción — preparación del borrador original: Paul Pasaca; redacción — revisión y edición: Paul Pasaca; visualización: Paul Pasaca; supervisión: Paul Pasaca; administración de proyecto: Paul Pasaca. Todos los autores han leído y aceptado la versión publicada del manuscrito.

FINANCIAMIENTO

Los recursos utilizados durante el desarrollo del presente trabajo de titulación serán asumidos por el autor, a excepción de los gastos por tutor que serán asumidos por la Universidad.

REFERENCIAS

- Arrighi, T., Rojas, J. E., Soto, J. C., Madrigal, C. A., y Londono, J. A. (2012). Recognition and classification of numerical labels using digital image processing techniques. *STSIVA 2012 - 17th Symposium of Image, Signal Processing, and Artificial Vision*, 252–260. doi: 10.1109/STSIVA.2012.6340592
- Huber, S., Wiemer, H., Schneider, D., y Ihlenfeldt, S. (2019, jan). DMME: Data mining methodology for engineering applications – a holistic extension to the CRISP-DM model. *Procedia CIRP*, 79, 403–408. Descargado de <https://www.sciencedirect.com/science/article/pii/S2212827119302239> doi: 10.1016/J.PROCIR.2019.02.106
- Kaggle: *Your Home for Data Science*. (s.f.). Descargado 2022-06-19, de <https://www.kaggle.com/>
- Keras Applications. (s.f.). Descargado 2022-06-19, de <https://keras.io/api/applications/>
- Mitthe, R., Indalkar, S., y Divekar, N. (2013, mar). Optical Character Recognition. *International Journal of Recent Technology and Engineering*. Descargado de <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.673.8061&rep=rep1&type=pdf>