Data augmentation para imágenes

Visión por Computador, curso 2024-2025

Silvia Martín Suazo, silvia.martin@u-tad.com

6 de noviembre de 2024

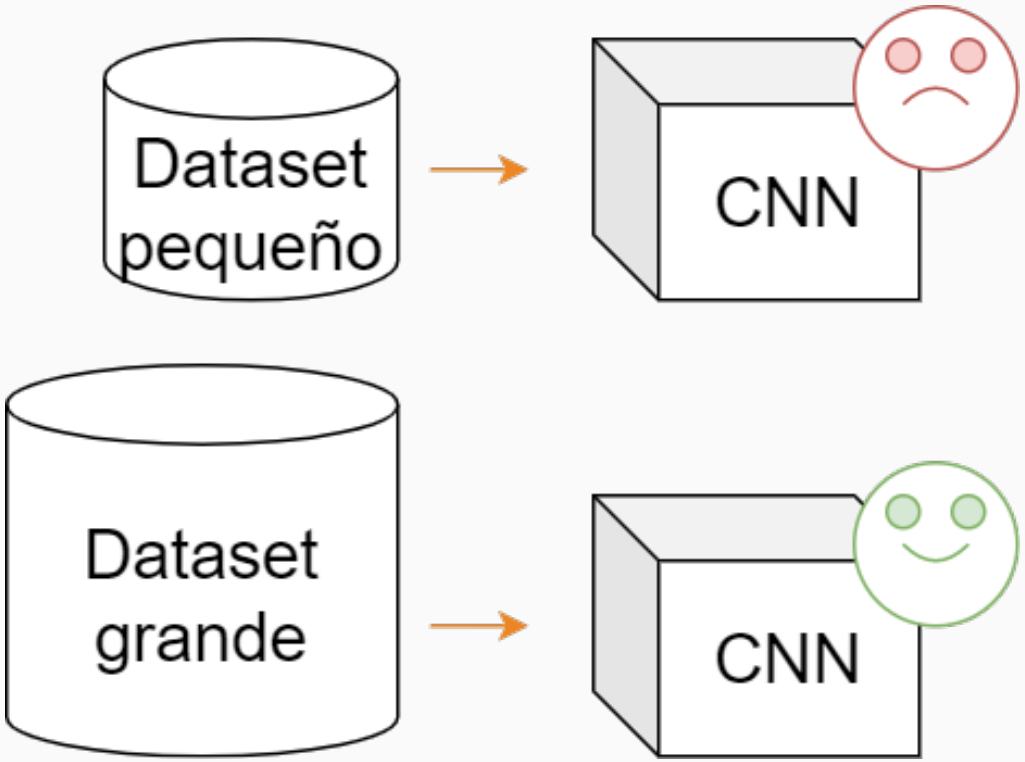
U-tad | Centro Universitario de Tecnología y Arte Digital

Introducción al Data Augmentation

Motivación

Cuando se realiza un entrenamiento con redes neuronales uno de los aspectos que más limitan su rendimiento es la falta de datos.

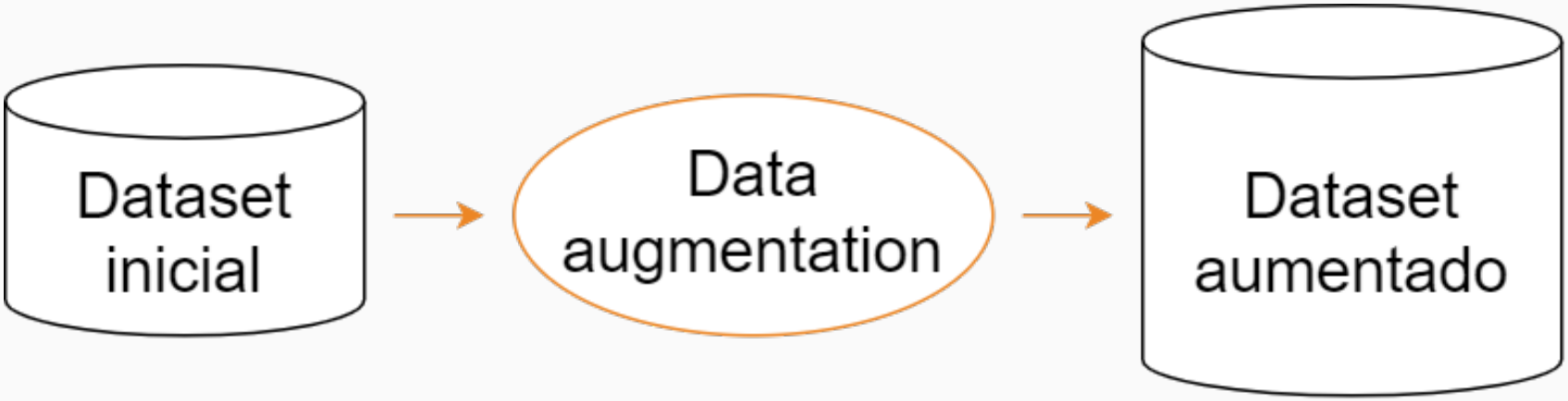
Cabe recordar que las redes neuronales son modelos altamente exigentes con la disponibilidad de datos. La convergencia, rendimiento, resistencia al overfitting o estabilidad son algunos de los aspectos sensibles a la variabilidad en los datos.



¿Qué es el data augmentation?

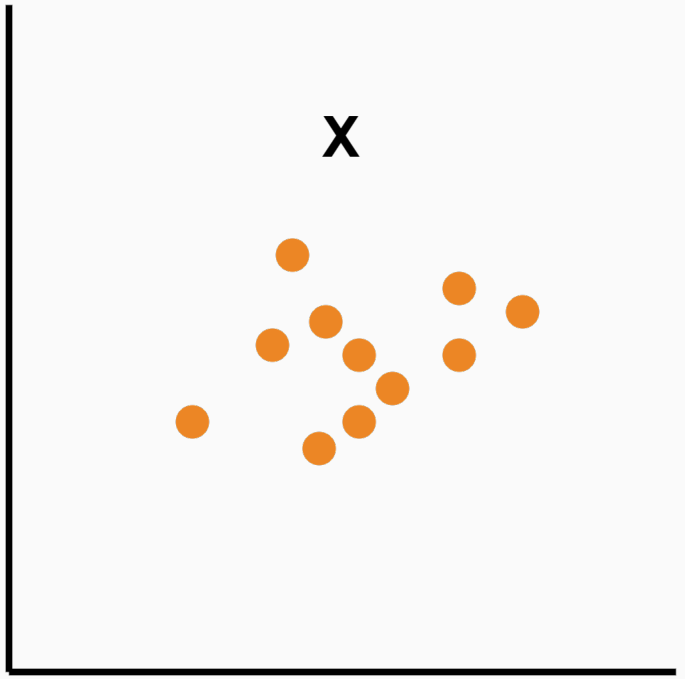
El proceso de data augmentation tiene como objetivo aumentar sintéticamente el tamaño del dataset que se usa. Para ello se aplican diferentes transformaciones al conjunto de datos que se tiene.

Con ello se obtienen más datos a partir de la información que se tiene originalmente. La calidad de estos datos dependerá de los algoritmos utilizados, el objetivo que se busca es replicar lo más fielmente la distribución de datos original.



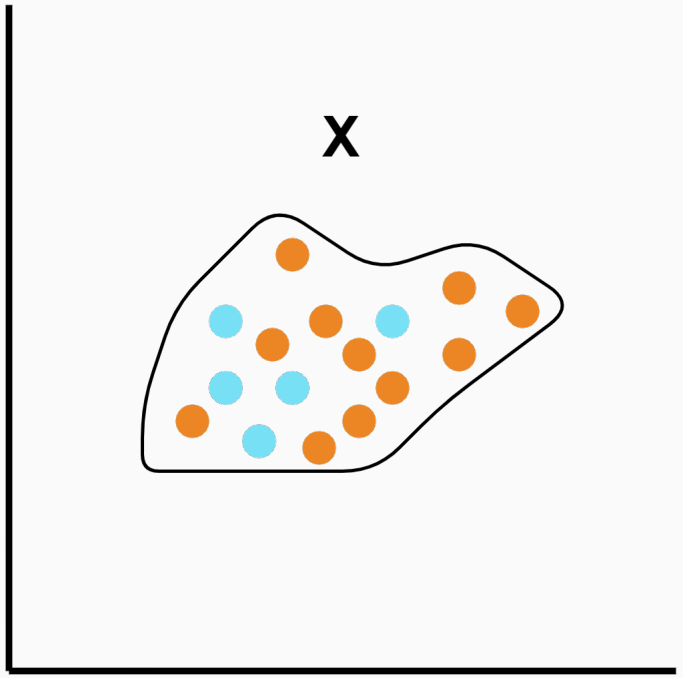
Data augmentation como replica de la distribución de datos

Desde un punto de vista matemático, se pretender replicar la distribución de datos original, con cierta variabilidad en los nuevos datos, pero manteniendo su distribución original. Se busca generar nuevas instancias indistinguibles de las originales.



Data augmentation como replica de la distribución de datos

Desde un punto de vista matemático, se pretender replicar la distribución de datos original, con cierta variabilidad en los nuevos datos, pero manteniendo su distribución original. Se busca generar nuevas instancias indistinguibles de las originales.



Ventajas de uso de data augmentation

En el campo de las redes neuronales artificiales el uso de técnicas de aumento de datos permite mejorar el rendimiento en ciertos aspectos:

- Overfitting: Una de las principales causas de este problema es que los modelos son capaces de "aprender" los casos con los que fueron entrenados.
- Mejor rendimiento: Al conseguir una distribución de datos más rica se consiguen mejores precisiones a la hora de clasificar, al permitir a la red una mejor generalización.

Implementaciones del data augmentation

Las técnicas de data augmentation, se utilizan para todo tipo de datos, independientemente de su origen. Sin embargo, no todas las técnicas pueden ser aplicadas a todos los conjuntos de datos de la misma manera.

Existen algoritmos de data augmentation específicos para distintos tipos de datos:

- Señales
- Imágenes
- Datos tabulares

Como norma general, las técnicas aplicadas deben ser cuidadosamente estudiadas para adaptarse al problema que se trate concreto.

Data augmentation para imágenes

Introducción

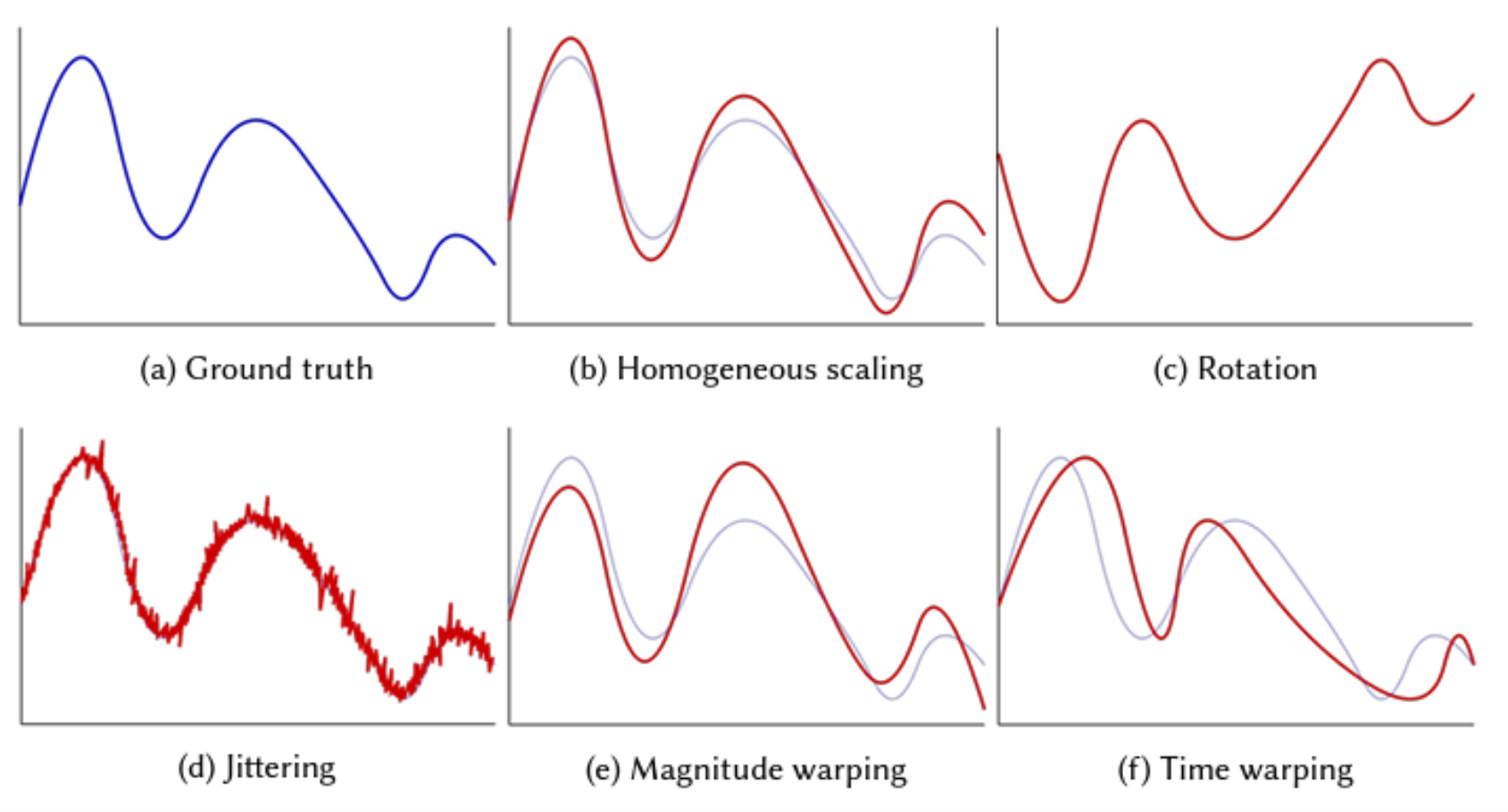
Existen un conjunto de técnicas de data augmentation que se pueden aplicar de manera general a las imágenes. Una de las ventajas de las imágenes es su fácil comprensión debido a que los resultados pueden ser validados de manera visual.

Sin embargo, cada problema es distinto y no siempre será posible aplicar todas las técnicas a cualquier problema. Siempre debe ser analizado cómo modifican los algoritmos el conjunto de datos.

Mantener un equilibrio entre variabilidad y similitud al conjunto inicial es un proceso complicado.

Relaciones entre distintas técnicas de data augmentation

Por otra parte, ciertas técnicas de data augmentation pueden ser utilizadas para distintos tipos de datos. Por ejemplo existen grandes similitudes entre datos de imágenes y señales.



[1]

Resumen

Existen una gran cantidad de algoritmos distintos para aumentar el número de imágenes de un dataset. Por ejemplo:

- Recorte
- Volteo horizontal
- Volteo vertical
- Rotación
- Translación
- Cambios en brillo
- Cambios en contraste
- Cambios en matiz
- Cambios en la calidad

Implementación en Keras

Existen dos posibilidades a la hora de realizar la implementación de los métodos de aumento de datos a la hora de realizar entrenamientos con redes neuronales.

1. Capas de aumento de datos dentro del modelo.
2. Aplicando las capas al dataset directamente.
3. Mediante generadores.
4. Con la librería Tensorflow.
5. Aumento de datos con otras librerías.

Forma 1: Capas de aumento de datos dentro del modelo

Capas de data augmentation dentro del modelo

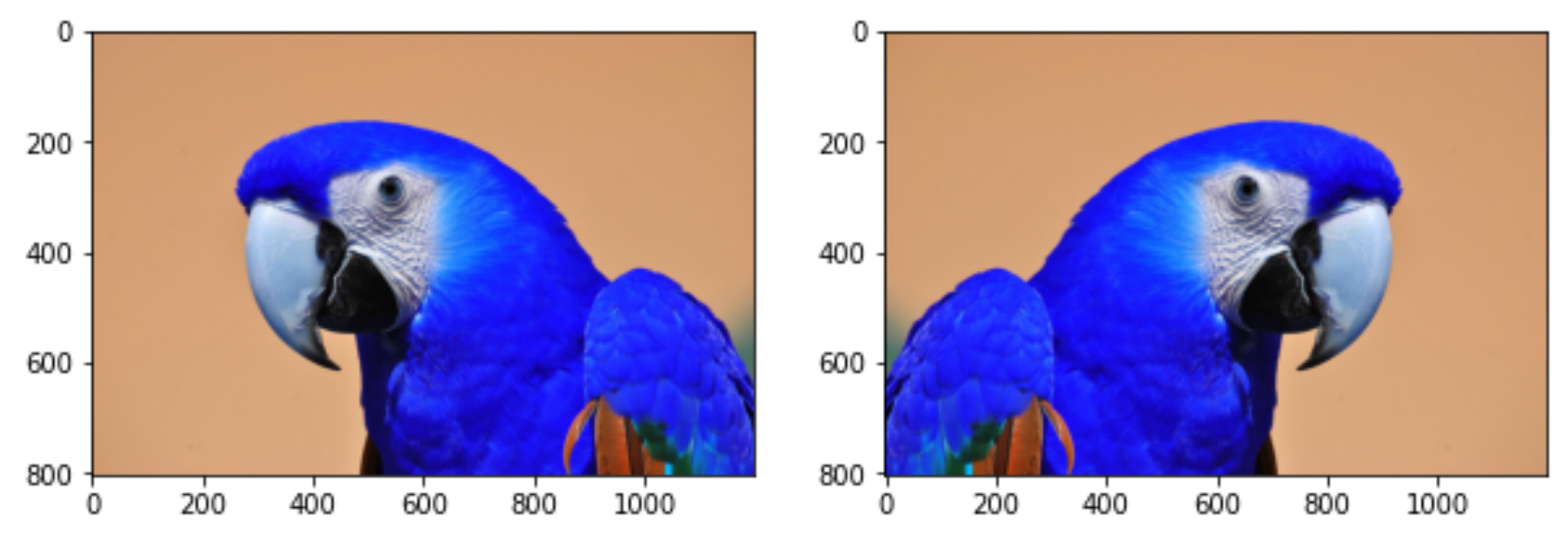
A la hora de crear un modelo con las librerías Tensorflow/Keras se pueden aplicar nuevas capas de aumento de datos en la entrada del modelo.

Esto hará que cada vez que el modelo reciba las imágenes como entrada, les aplique aleatoriamente las transformaciones deseadas.

```
model = tf.keras.Sequential([
  # Add the preprocessing layers you created earlier.
  resize_and_rescale,
  data_augmentation,
  layers.Conv2D(16, 3, padding='same', activation='relu'),
  layers.MaxPooling2D(),
  # Rest of your model.
])
```

Recorte

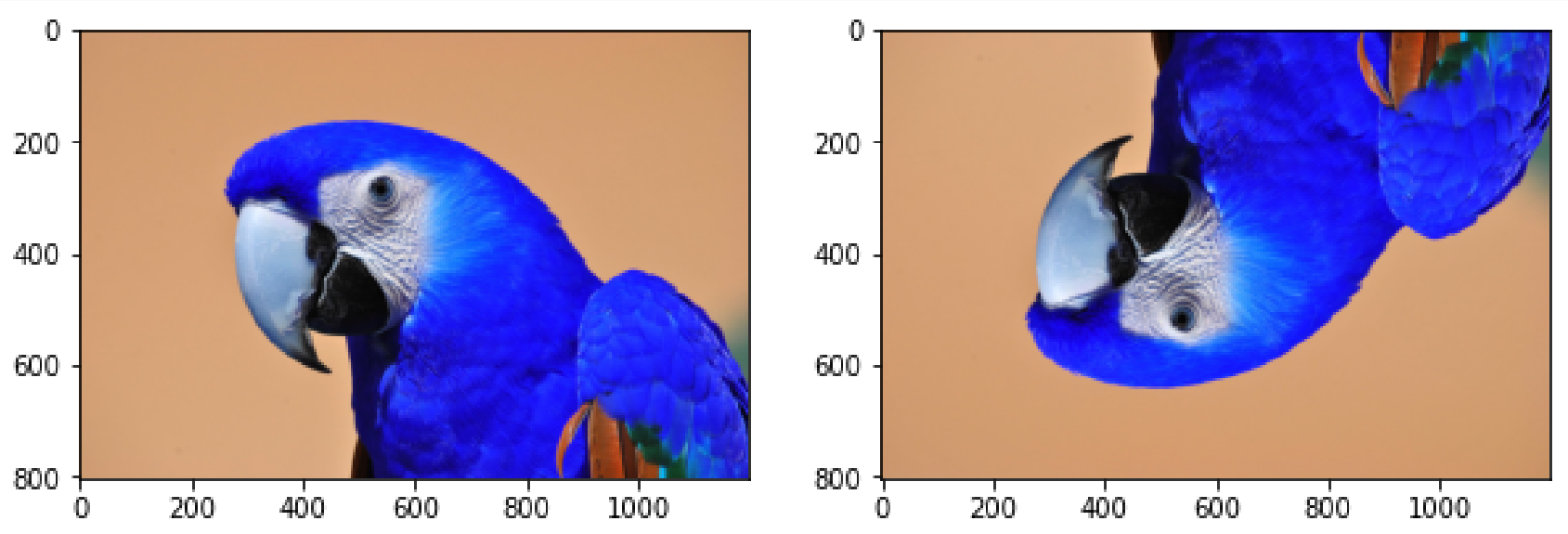
Documentación de la capa RandomFlip



* aplicar el formato “horizontal”

Volteo horizontal

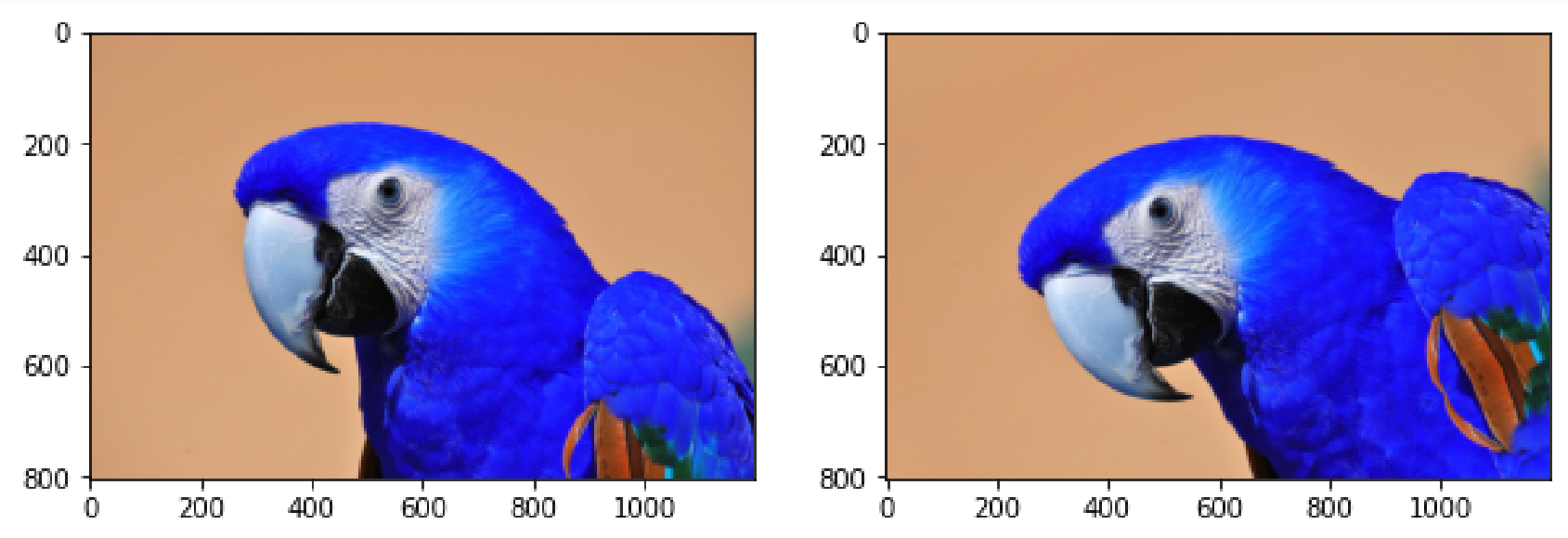
Documentación de la capa RandomFlip



* aplicar el formato “vertical”

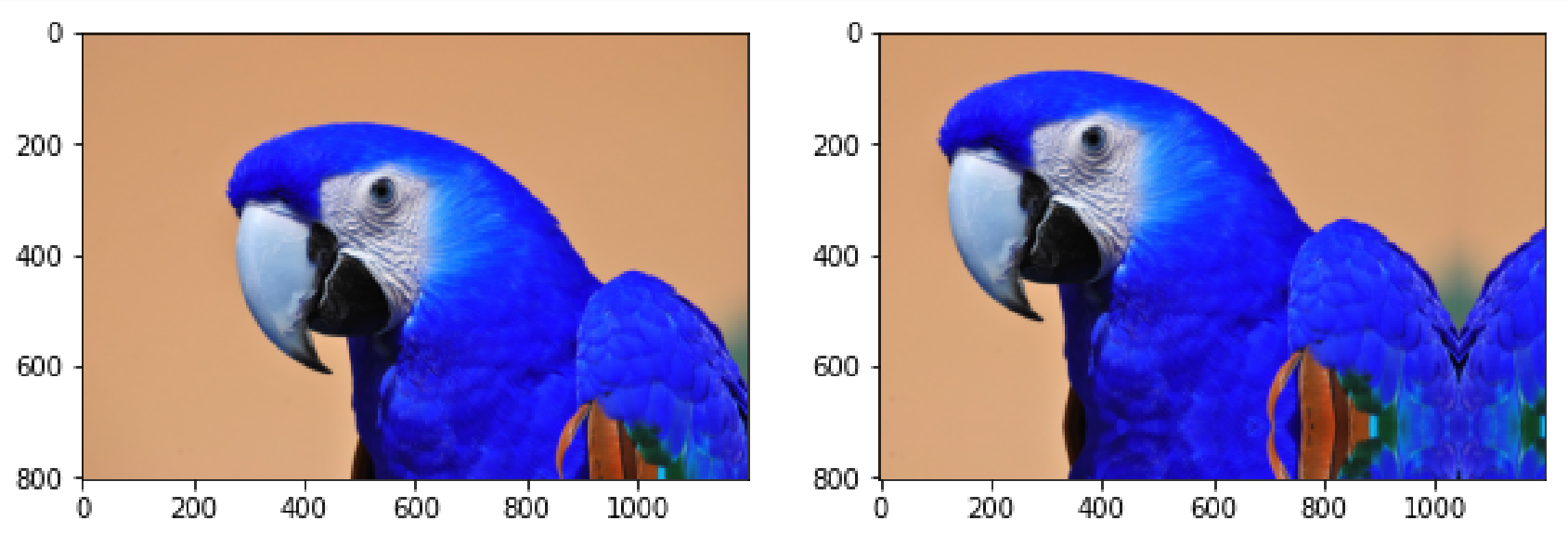
Rotación

Documentación de la capa RandomRotation



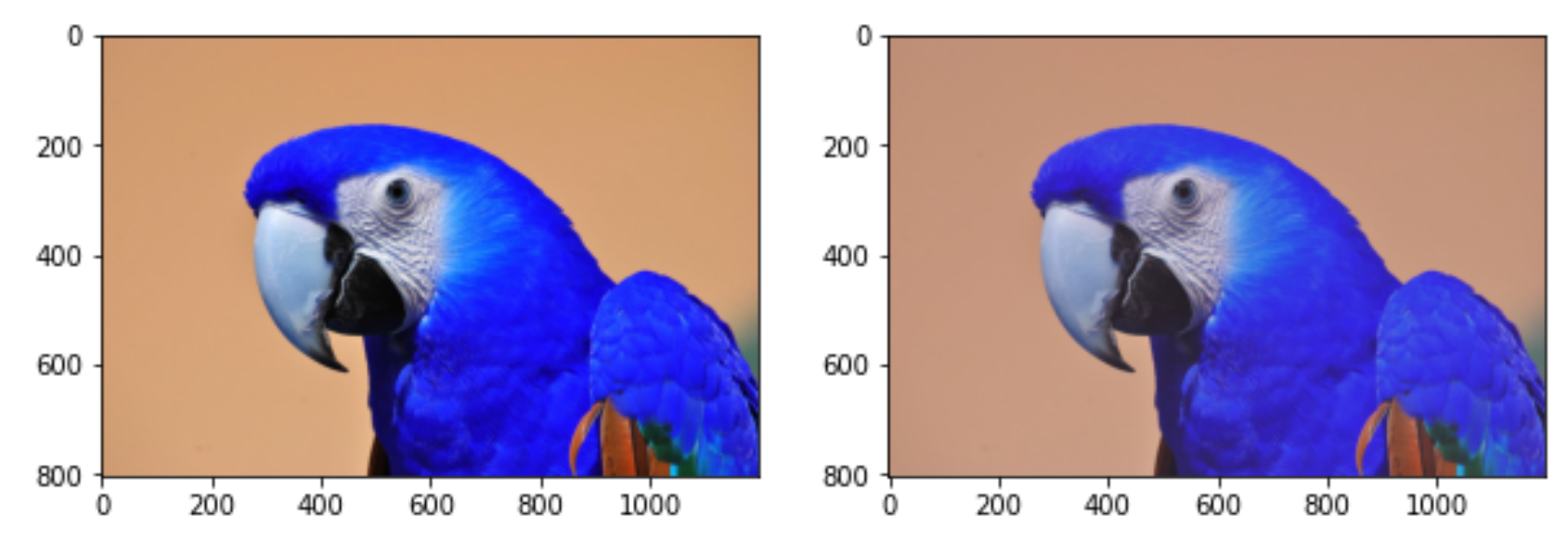
Translación

Documentación de la capa RandomTranslation



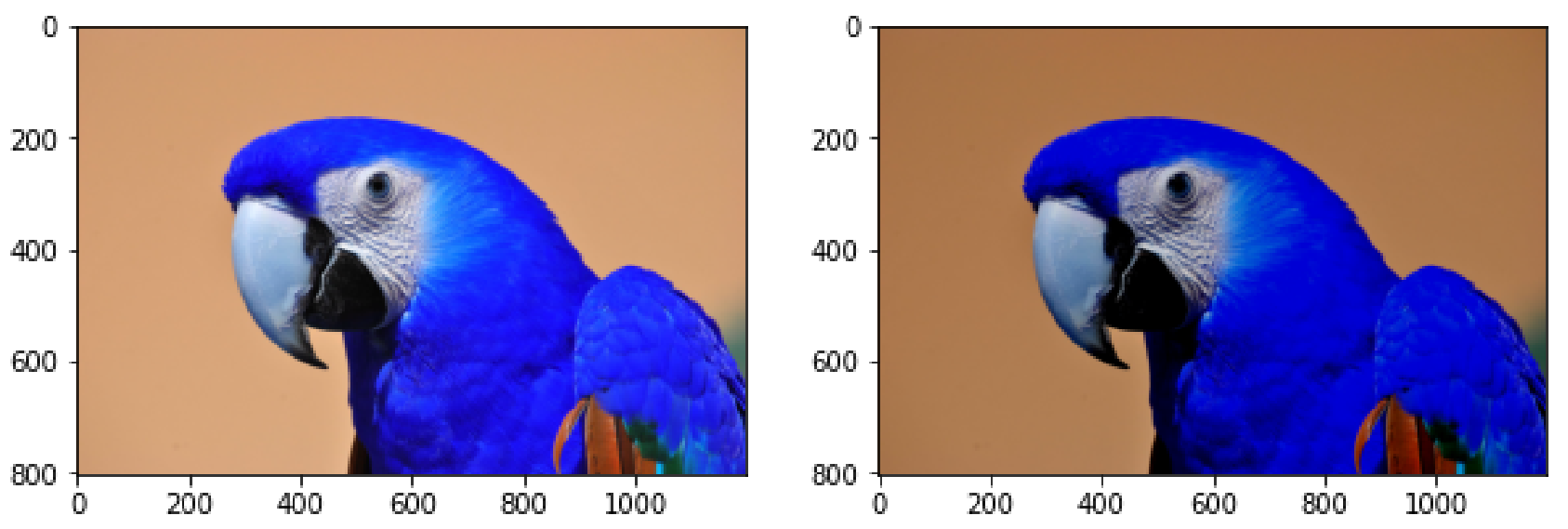
Cambios en contraste

Documentación de la capa RandomContrast



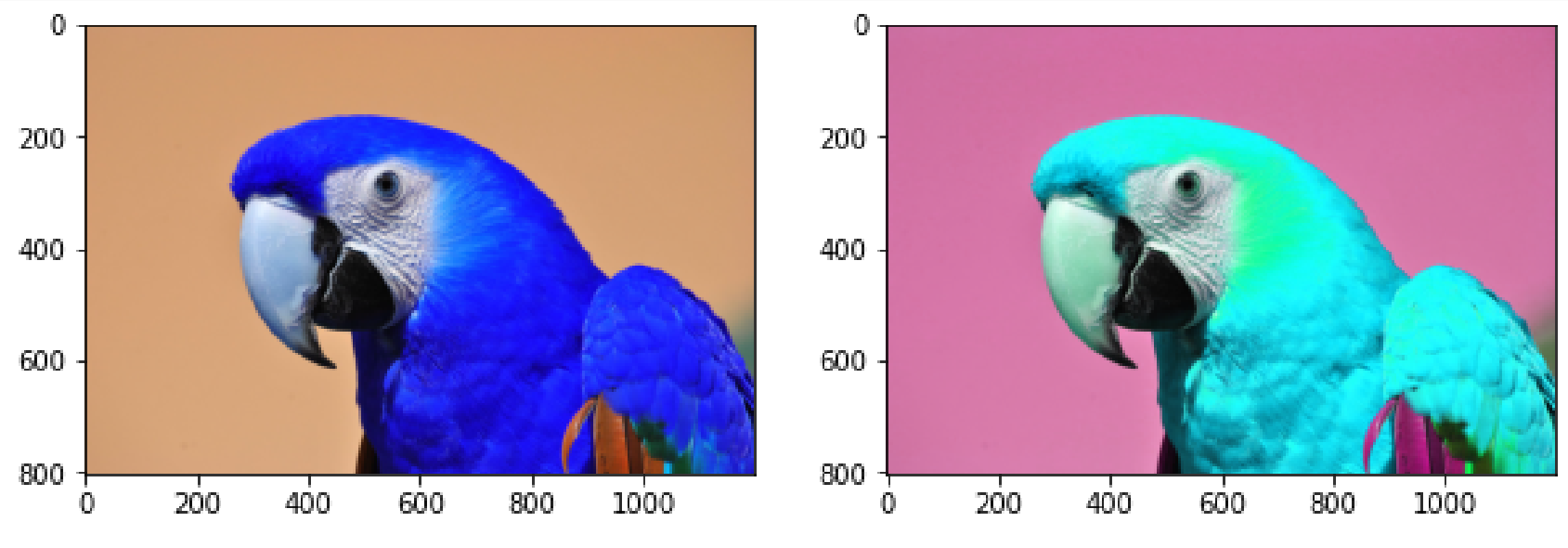
Cambios en brillo

Documentación de la capa stateless_random_brightness



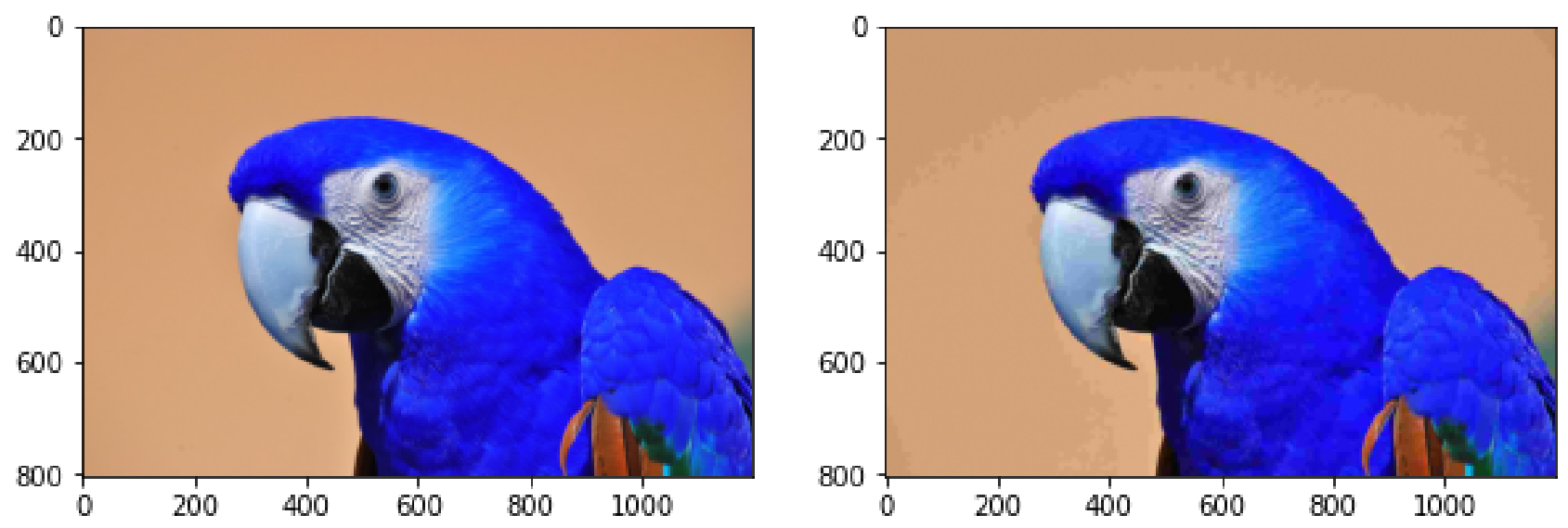
Cambios en matiz

Documentación de la capa stateless_random_hue



Cambios en calidad

Documentación de la capa stateless_random_jpeg_quality



Forma 2: Aplicando las capas al dataset directamente

Aplicando las capas al dataset directamente

Otra alternativa es la de aplicar el aumento de datos después de cargar el dataset

```
aug_ds = train_ds.map(
  lambda x, y: (resize_and_rescale(x, training=True), y))
```

** para más información se puede consultar la siguiente guía oficial de Keras*

Forma 3: Mediante generadores

Mediante generadores

Como vimos anteriormente, una forma de cargar los datos es mediante generadores. Además, mediante el uso de estos generadores podemos realizar modificaciones a los datos con el fin de realizar aumentación de datos.

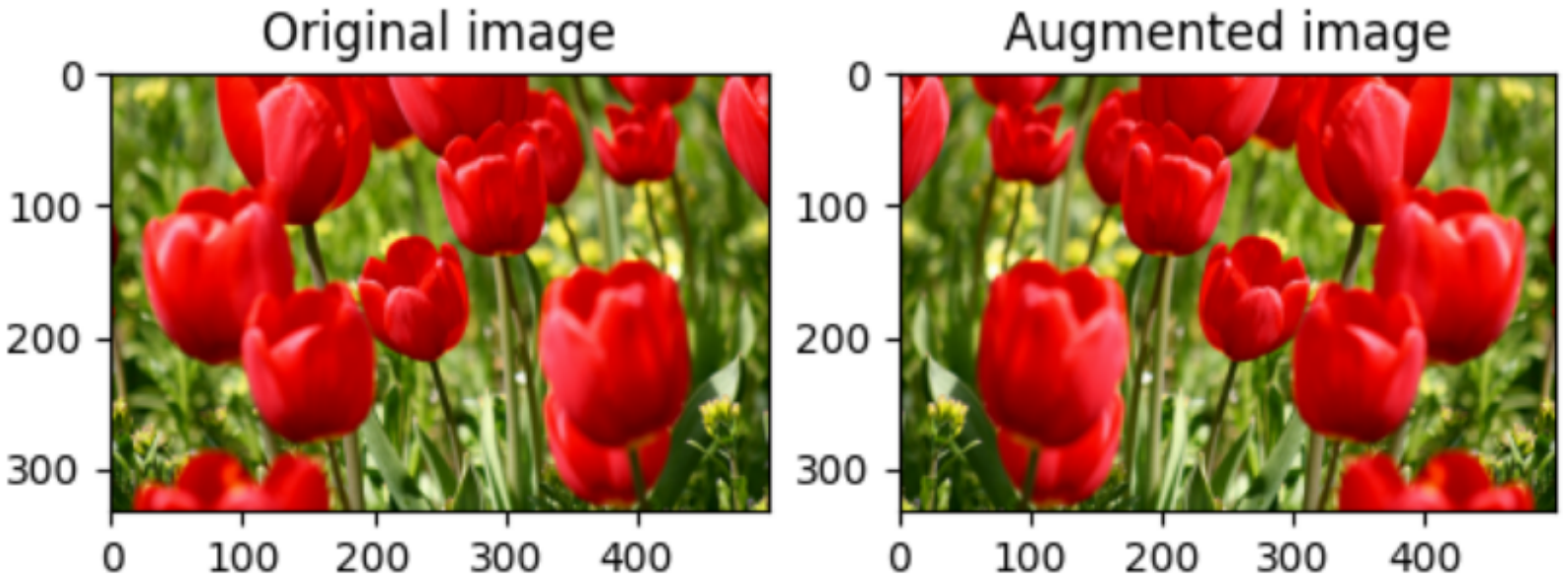
```
aug = ImageDataGenerator(
    rotation_range=20,
    zoom_range=0.15,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.15,
    horizontal_flip=True,
    fill_mode="nearest")
```

Forma 4: Con Tensorflow

Con Tensorflow

Tensorflow también ofrece funciones para realizar transformaciones a las imágenes para aumentación de datos mediante tf.image. En este caso será necesario aplicarlas independientemente a cada imagen.

```
flipped = tf.image.flip_left_right(image)
visualize(image, flipped)
```



Forma 5: Con otras librerías

Con otras librerías

Siempre se puede recurrir a la opción de realizar él data augmentation que nos interese a mano. Por ejemplo, con numpy o OpenCV.

```
img_rotate_180 = cv2.rotate(img, cv2.ROTATE_180)
```

Notebook de ejemplo, data augmentation

El siguiente notebook contiene las implementaciones de las distintas formas de hacer data augmentation.

- 03_07-DataAugmentation.ipynb

Referencias i

[1] Edgar Talavera, Guillermo Iglesias, Ángel González-Prieto, Alberto Mozo, and Sandra Gómez-Canaval.
Data augmentation techniques in time series domain: A survey and taxonomy.
arXiv preprint arXiv:2206.13508, 2022.