



PRÁCTICA 2 - VPC

Gender Recognition

Cristian Villarroya Sánchez

01-06-2021

Índice

1.	Descripción de la tarea.....	2
2.	Modelo propuesto	2
3.	Conclusiones.....	5
4.	Bibliografía	6

1. Descripción de la tarea

Se dispone de un dataset de imágenes de caras en diferentes posturas, gestos y escenarios de tamaño 100x100 en RGB, separadas en dos conjuntos de entrenamiento y test.

El código de referencia está disponible en el GitHub del profesor de la asignatura[1]

Los objetivos de esta tarea son:

- Construir un modelo con >97% de precisión en el conjunto de test
- Construir un modelo con >92% de precisión en el conjunto de test con menos de 100K parámetros

En este documento, se describe un modelo que cumple con los dos objetivos propuestos, precisión mayor de un 97% y menos de 100K parámetros.

2. Modelo propuesto

La implementación de este modelo se ha llevado a cabo siguiendo la idea del artículo propuesto [2].

Lo primero que se ha visto es que, las imágenes contienen “demasiado fondo”, que puede dar problemas a la hora de identificar las caras. Por eso, se ha optado por recortar las imágenes para que contengan la menor cantidad posible de fondo y la mayor cantidad de cara. Se ha optado por cortar las imágenes a un tamaño de 64x64. Por ejemplo, la primera imagen sufriría la siguiente transformación:

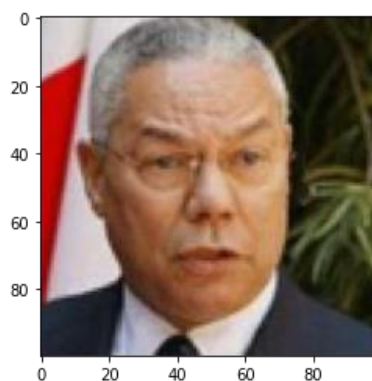


Ilustración 1. Imagen Original 100x100

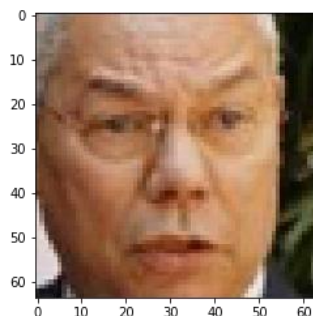


Ilustración 2. Imagen Recortada 64x64

Reduciendo el tamaño de la imagen, además de mejorar la precisión, se consigue reducir el número de parámetros del modelo. Sin esta transformación, el modelo no llegaba al 97% de

precisión, se quedaba en torno al 94% de precisión y el número de parámetros era de 164.066, sin embargo, con esta transformación, el modelo superó el 97% de precisión y además consiguió un número de parámetros igual a 93.282.

```
Total params: 164,066
Trainable params: 163,298
Non-trainable params: 768
```

Ilustración 3. Parámetros sin recorte

```
Total params: 93,282
Trainable params: 92,770
Non-trainable params: 512
```

Ilustración 4. Parámetros con Recorte

Otro aspecto importante, que ayudo a mejorar notablemente la precisión, es el data augmentation. Hay que tener cuidado de no hacer demasiado zoom, puesto que la imagen ya ha sido recortada previamente. Lo mismo con la rotación y los cambios en altura y anchura, si se aplica un valor demasiado alto, puede que la cara deje de ser una cara. Lo mismo con dar la vuelta, en este caso, se ha optado por hacer únicamente un horizontal flip, no vertical flip ya que, con este último, una cara no sería una cara.

Otro de los aspectos importantes es el optimizador del gradiente. En este ejercicio se ha probado SGD y Adam. Además del optimizador, se han probado también diferentes learning rates. Con el optimizador SGD y learning rates de 0.1 y 0.01, la precisión rondaba el 97% pero no llego a sobrepasarlo, se quedó cerca. Sin embargo, con Adam, ambos valores fueron capaces de sobrepasar el 97%, finalmente se decidió utilizar el optimizador Adam con un learning rate de 0.1.

Al modelo se le ha añadido un checkpoint para que guarde los pesos del mejor resultado obtenido. Con todo esto, la configuración final del modelo es la siguiente:

- Tamaño imagen entrada → 64x64
- Data Augmentation:
 - Width_shift_range: 0.18
 - Height_shift_range: 0.18
 - zoom_range: 0.18
 - rotation_range: 10,
 - horizontal_flip = True
 - vertical_flip = False
- Batch size: 128
- Optimizador: Adam
- Learning Rate: 0.01

El modelo posee la siguiente estructura:

- Input 64x64x3
- Conv2D 3x3 32 padding=same
- BN 0.1
- Relu
- Conv2D 3x3 32 padding=same
- BN 0.1
- Relu
- Max Pooling 2x2
- Conv2D 3x3 32 padding=same
- BN 0.1
- Relu
- Conv2D 3x3 32 padding=same
- BN 0.1
- Relu
- Max Pooling 2x2
- Conv2D 3x3 64 padding=same
- BN 0.1
- Relu
- Conv2D 3x3 64 padding=same
- BN 0.1
- Relu
- Max Pooling 2x2
- Flatten()
- Dense 2
- Softmax

Se ha adjuntado una imagen del modelo en formato png (model_plot.png), ya que es muy grande y no se ve bien al insertarse en este documento

Finalmente, con esta estructura y configuración, se ha obtenido una precisión del 97.77% tras 84 épocas de entrenamiento

```
Epoch 00083: val_accuracy did not improve from 0.97621
Epoch 84/150
82/82 [=====] - 12s 142ms/step - loss: 0.0531 - accuracy: 0.9801 - val_loss: 0.0704 - val_accuracy: 0.9777

Epoch 00084: val_accuracy improved from 0.97621 to 0.97772, saving model to drive/MyDrive/VPC/gender/weights_97.hdf5
Epoch 85/150
82/82 [=====] - 11s 139ms/step - loss: 0.0490 - accuracy: 0.9814 - val_loss: 0.1057 - val_accuracy: 0.9687

Epoch 00085: val_accuracy did not improve from 0.97772
Epoch 86/150
82/82 [=====] - 12s 139ms/step - loss: 0.0635 - accuracy: 0.9765 - val_loss: 0.1002 - val_accuracy: 0.9724

Epoch 00086: val_accuracy did not improve from 0.97772
Epoch 87/150
82/82 [=====] - 12s 146ms/step - loss: 0.0494 - accuracy: 0.9806 - val_loss: 0.1207 - val_accuracy: 0.9702
```

El número de parámetros es el especificado en la ilustración 4, 93.282.

3. Conclusiones

Se ha propuesto un modelo que cumple con los dos requisitos propuestos. Además, se han probado diferentes configuraciones hasta conseguir obtener la precisión requerida en el ejercicio.

En el artículo propuesto [2] aparecen diferentes estructuras con las cuales se superó también el 97% de precisión, pero el número de parámetros era muy elevado.

Starting CNN	Optimization: candidates at step 1				Optimization: candidate at step 2	Optimization: candidate at step 3										
	A	B	C	D	E	F	G	H	I	J	K	L	M			
Input: 128 × 128	Input: 64 × 64	Input: 32 × 32	Input: 16 × 16	Input: 32 × 32						Input: 32 × 32						
Conv: 32 @ 3 × 3	Conv: 32 @ 3 × 3	Conv: 32 @ 3 × 3	Conv: 64 @ 3 × 3	Conv: 16 @ 3 × 3						Conv: 32 @ 3 × 3						
Conv: 32 @ 3 × 3	Conv: 32 @ 3 × 3	Conv: 32 @ 3 × 3	Conv: 64 @ 3 × 3	Conv: 16 @ 3 × 3						Conv: 32 @ 3 × 3						
MaxPool: 2 × 2	MaxPool: 2 × 2	MaxPool: 2 × 2	MaxPool: 2 × 2	MaxPool: 2 × 2						MaxPool: 2 × 2						
Conv: 32 @ 3 × 3	Conv: 32 @ 3 × 3	Conv: 64 @ 3 × 3	Conv: 32 @ 3 × 3							Conv: 64 @ 3 × 3						
Conv: 32 @ 3 × 3	Conv: 32 @ 3 × 3	Conv: 64 @ 3 × 3	Conv: 32 @ 3 × 3							Conv: 64 @ 3 × 3						
MaxPool: 2 × 2	MaxPool: 2 × 2	MaxPool: 2 × 2	MaxPool: 2 × 2							MaxPool: 2 × 2						
Conv: 64 @ 3 × 3	Conv: 64 @ 3 × 3															
Conv: 64 @ 3 × 3	Conv: 64 @ 3 × 3															
MaxPool: 2 × 2	MaxPool: 2 × 2															
Conv: 64 @ 3 × 3																
Conv: 64 @ 3 × 3																
MaxPool: 2 × 2																

Ilustración 5. Modelos propuestos en el artículo [2]

Por ejemplo, con la configuración descrita en el apartado 2 de este documento y la Starting CNN y la optimización 1 de la Ilustración 5 como modelos, se superó también el 97% de precisión, pero el número de parámetros era muy elevado.

Realizando una ejecución con la Starting CNN y la configuración descrita en el anterior apartado se obtuvo el siguiente resultado:

```
Total params: 161,506
Trainable params: 160,738
Non-trainable params: 768
```

```
Epoch 00039: val_accuracy did not improve from 0.96866
Epoch 40/150
82/82 [=====] - 12s 142ms/step - loss: 0.0702 - accuracy: 0.9752 - val_loss: 0.0917 - val_accuracy: 0.9721
Epoch 00040: val_accuracy improved from 0.96866 to 0.97205, saving model to drive/MyDrive/VPC/gender/weights_97.hdf5
```

En primera instancia se pensó en hacer dos modelos diferentes, uno que supere el 97% y otro que supere el 92% con menos de 100k de parámetros, pero finalmente, como se ha visto que se pueden conseguir ambos objetivos con un único modelo, ha sido este el modelo propuesto para conseguir ambos objetivos.

En definitiva, se han probado dos modelos:

- Modelo Propuesto:
 - Parámetros: 93.282
 - Precisión: 97.77%
- Starting CNN:
 - Parámetros: 161.506
 - Precisión: 97.21%

4. Bibliografía

[1]. R. Paredes Palacios, "Computer vision lab."

<https://github.com/RParedesPalacios/ComputerVisionLab,2020>.

[2]. G. Antipov, S.-A. Berrani, and J. Dugelay, "Minimalistic cnn-based ensemble model for gender prediction from face images," Pattern Recognit. Lett., vol. 70, pp. 59–65, 2016