

# Práctica Imagen

Francisco Rodríguez Cuenca

El objetivo de esta práctica es el desarrollo de un modelo que pueda diferenciar entre películas de animación y vida real a partir de sus posters.

Para ello hemos realizado cuatro modelos, todos utilizando una red base VGG16: uno con la red congelada y otros tres con diferentes niveles de descongelación.

Al final nuestro mejor modelo ha sido el último que hemos probado: descongelando únicamente la última red convolucional del VGG16.

Debido al poco tamaño de nuestra muestra, los modelos se sobreentrenaban con facilidad así que hemos limitado las epochs a 30.

## Configuración

```
In [1]: import keras  
keras.__version__
```

```
Out[1]: '2.4.3'
```

```
In [2]: from keras.applications import VGG16  
  
conv_base = VGG16(weights='imagenet',  
                    include_top=False,  
                    input_shape=(150, 150, 3))
```

```
In [3]: base_dir = base_dir = 'data'  
  
train_dir = os.path.join(base_dir, 'training')  
validation_dir = os.path.join(base_dir, 'validation')  
test_dir = os.path.join(base_dir, 'test')
```

```
In [4]: from keras.preprocessing.image import ImageDataGenerator

train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=40,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest')

# Note that the validation data should not be augmented!
test_datagen = ImageDataGenerator(rescale=1./255)

train_generator = train_datagen.flow_from_directory(
    # This is the target directory
    train_dir,
    # All images will be resized to 150x150
    target_size=(150, 150),
    batch_size=20,
    # Since we use binary_crossentropy loss, we need binary labels
    class_mode='binary')

validation_generator = test_datagen.flow_from_directory(
    validation_dir,
    target_size=(150, 150),
    batch_size=5,
    class_mode='binary')

test_generator = test_datagen.flow_from_directory(
    test_dir,
    target_size=(150, 150),
    batch_size=5,
    class_mode='binary')
```

Found 2115 images belonging to 2 classes.  
Found 500 images belonging to 2 classes.  
Found 500 images belonging to 2 classes.

## Primer modelo

```
In [5]: from keras import models
        from keras import layers
        from keras import optimizers

model = models.Sequential()
model.add(conv_base)
model.add(layers.Flatten())
model.add(layers.Dense(256, activation='relu'))
model.add(layers.Dense(1, activation='sigmoid'))
```

```
In [6]: conv_base.trainable = False
```

```
In [7]: model.compile(loss='binary_crossentropy',
                    optimizer=optimizers.RMSprop(lr=2e-5),
                    metrics=['acc'])

history = model.fit_generator(
    train_generator,
    steps_per_epoch=100,
    epochs=30,
    validation_data=validation_generator,
    validation_steps=50,
    verbose=2)
```

WARNING:tensorflow:From <ipython-input-7-76fc7293b408>:5: Model.fit\_generator (from tensorflow.python.keras.engine.training) is deprecated and will be removed in a future version.

Instructions for updating:

Please use Model.fit, which supports generators.

Epoch 1/30

100/100 - 59s - loss: 0.6371 - acc: 0.6371 - val\_loss: 0.5991 - val\_acc: 0.6920

Epoch 2/30

100/100 - 62s - loss: 0.5581 - acc: 0.7083 - val\_loss: 0.5146 - val\_acc: 0.7520

Epoch 3/30

100/100 - 62s - loss: 0.5174 - acc: 0.7499 - val\_loss: 0.5624 - val\_acc: 0.7320

Epoch 4/30

100/100 - 62s - loss: 0.4960 - acc: 0.7589 - val\_loss: 0.5314 - val\_acc: 0.7760

Epoch 5/30

100/100 - 62s - loss: 0.4856 - acc: 0.7629 - val\_loss: 0.5195 - val\_acc: 0.7640

Epoch 6/30

100/100 - 61s - loss: 0.4852 - acc: 0.7694 - val\_loss: 0.4968 - val\_acc: 0.7680

Epoch 7/30

100/100 - 61s - loss: 0.4700 - acc: 0.7719 - val\_loss: 0.5197 - val\_acc: 0.7480

Epoch 8/30

100/100 - 61s - loss: 0.4594 - acc: 0.7865 - val\_loss: 0.5249 - val\_acc: 0.7360

Epoch 9/30

100/100 - 61s - loss: 0.4419 - acc: 0.7910 - val\_loss: 0.5663 - val\_acc: 0.7760

Epoch 10/30

100/100 - 61s - loss: 0.4639 - acc: 0.7865 - val\_loss: 0.5273 - val\_acc: 0.7480

Epoch 11/30

100/100 - 61s - loss: 0.4459 - acc: 0.7900 - val\_loss: 0.4818 - val\_acc: 0.7960

Epoch 12/30

100/100 - 61s - loss: 0.4496 - acc: 0.7865 - val\_loss: 0.4874 - val\_acc: 0.7680

Epoch 13/30

100/100 - 61s - loss: 0.4314 - acc: 0.7980 - val\_loss: 0.4608 - val\_acc: 0.8040

Epoch 14/30

100/100 - 61s - loss: 0.4177 - acc: 0.8145 - val\_loss: 0.4832 - val\_acc: 0.8160

Epoch 15/30

100/100 - 61s - loss: 0.4245 - acc: 0.8010 - val\_loss: 0.4957 - val\_acc: 0.7920

Epoch 16/30

```
100/100 - 61s - loss: 0.4256 - acc: 0.7995 - val_loss: 0.5509 - val_acc: 0.7400
Epoch 17/30
100/100 - 61s - loss: 0.4148 - acc: 0.8090 - val_loss: 0.4915 - val_acc: 0.7840
Epoch 18/30
100/100 - 61s - loss: 0.4280 - acc: 0.8020 - val_loss: 0.5478 - val_acc: 0.7680
Epoch 19/30
100/100 - 61s - loss: 0.4039 - acc: 0.8140 - val_loss: 0.4898 - val_acc: 0.7880
Epoch 20/30
100/100 - 61s - loss: 0.4061 - acc: 0.8110 - val_loss: 0.5361 - val_acc: 0.7680
Epoch 21/30
100/100 - 61s - loss: 0.4079 - acc: 0.8190 - val_loss: 0.4678 - val_acc: 0.7880
Epoch 22/30
100/100 - 61s - loss: 0.4136 - acc: 0.8155 - val_loss: 0.4878 - val_acc: 0.8120
Epoch 23/30
100/100 - 61s - loss: 0.3997 - acc: 0.8216 - val_loss: 0.5026 - val_acc: 0.7720
Epoch 24/30
100/100 - 61s - loss: 0.3974 - acc: 0.8140 - val_loss: 0.4809 - val_acc: 0.7840
Epoch 25/30
100/100 - 61s - loss: 0.4106 - acc: 0.8100 - val_loss: 0.4931 - val_acc: 0.7680
Epoch 26/30
100/100 - 61s - loss: 0.3956 - acc: 0.8155 - val_loss: 0.4947 - val_acc: 0.7920
Epoch 27/30
100/100 - 61s - loss: 0.4052 - acc: 0.8140 - val_loss: 0.4730 - val_acc: 0.8040
Epoch 28/30
100/100 - 61s - loss: 0.3973 - acc: 0.8271 - val_loss: 0.4820 - val_acc: 0.7920
Epoch 29/30
100/100 - 61s - loss: 0.3895 - acc: 0.8216 - val_loss: 0.4941 - val_acc: 0.7560
Epoch 30/30
100/100 - 61s - loss: 0.3828 - acc: 0.8366 - val_loss: 0.5004 - val_acc: 0.7760
```

```
In [8]: model.save('posters.h5')
```

```
In [9]: import matplotlib.pyplot as plt

acc = history.history['acc']
val_acc = history.history['val_acc']
loss = history.history['loss']
val_loss = history.history['val_loss']

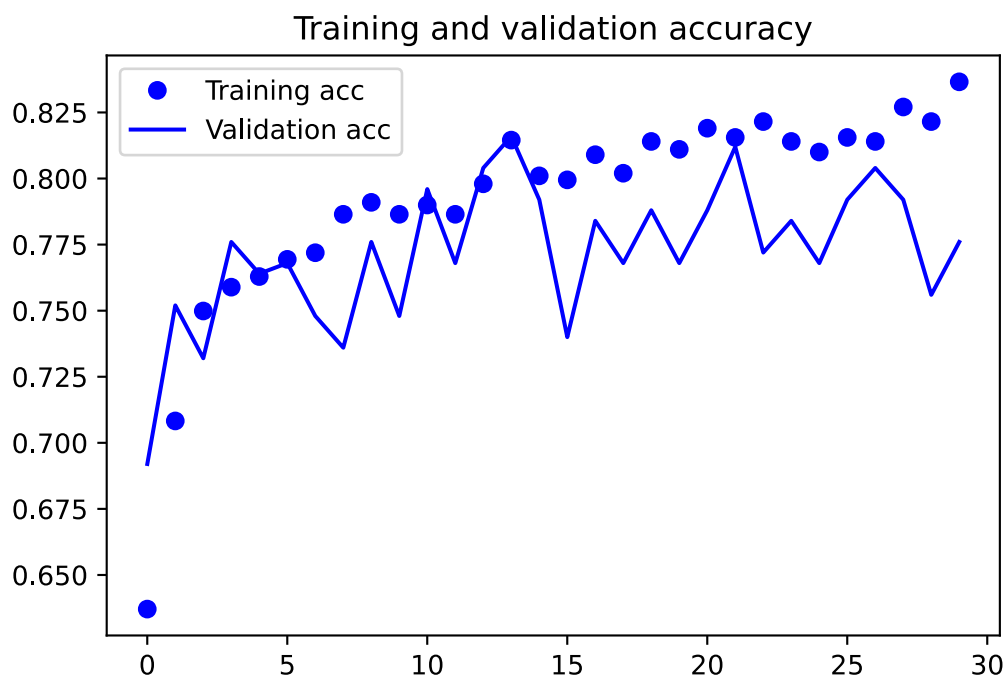
epochs = range(len(acc))

plt.plot(epochs, acc, 'bo', label='Training acc')
plt.plot(epochs, val_acc, 'b', label='Validation acc')
plt.title('Training and validation accuracy')
plt.legend()

plt.figure()

plt.plot(epochs, loss, 'bo', label='Training loss')
plt.plot(epochs, val_loss, 'b', label='Validation loss')
plt.title('Training and validation loss')
plt.legend()

plt.show()
```



## Training and validation loss



```
In [10]: test_loss, test_acc = model.evaluate_generator(test_generator, steps=50)
print('test acc:', test_acc)
```

WARNING:tensorflow:From <ipython-input-10-a056cf674a87>:1: Model.evaluate\_generator (from tensorflow.python.keras.engine.training) is deprecated and will be removed in a future version.  
Instructions for updating:  
Please use Model.evaluate, which supports generators.  
test acc: 0.7639999985694885

## Fine tuning - Segundo modelo

```
In [11]: conv_base.trainable = True

set_trainable = False
for layer in conv_base.layers:
    if layer.name == 'block5_conv1':
        set_trainable = True
    if set_trainable:
        layer.trainable = True
    else:
        layer.trainable = False
```

```
In [12]: model.compile(loss='binary_crossentropy',
                        optimizer=optimizers.RMSprop(lr=1e-5),
                        metrics=['acc'])

history = model.fit_generator(
    train_generator,
    steps_per_epoch=100,
    epochs=30,
    validation_data=validation_generator,
    validation_steps=50)
```

```
Epoch 1/30
100/100 [=====] - 74s 737ms/step - loss: 0.4175 -
acc: 0.7925 - val_loss: 0.5564 - val_acc: 0.7240
Epoch 2/30
100/100 [=====] - 74s 737ms/step - loss: 0.3857 -
acc: 0.8266 - val_loss: 0.4807 - val_acc: 0.7560
Epoch 3/30
100/100 [=====] - 74s 736ms/step - loss: 0.3497 -
acc: 0.8536 - val_loss: 0.4851 - val_acc: 0.7800
Epoch 4/30
100/100 [=====] - 74s 737ms/step - loss: 0.3335 -
acc: 0.8506 - val_loss: 0.4802 - val_acc: 0.7880
Epoch 5/30
100/100 [=====] - 74s 739ms/step - loss: 0.3231 -
acc: 0.8665 - val_loss: 0.4946 - val_acc: 0.7840
Epoch 6/30
100/100 [=====] - 74s 736ms/step - loss: 0.2956 -
acc: 0.8767 - val_loss: 0.6034 - val_acc: 0.7360
Epoch 7/30
100/100 [=====] - 74s 737ms/step - loss: 0.3170 -
acc: 0.8707 - val_loss: 0.4257 - val_acc: 0.8040
Epoch 8/30
```

```
100/100 [=====] - 74s 736ms/step - loss: 0.2775 -  
acc: 0.8792 - val_loss: 0.4824 - val_acc: 0.8200  
Epoch 9/30  
100/100 [=====] - 74s 737ms/step - loss: 0.2651 -  
acc: 0.8952 - val_loss: 0.4887 - val_acc: 0.8000  
Epoch 10/30  
100/100 [=====] - 74s 739ms/step - loss: 0.2699 -  
acc: 0.8820 - val_loss: 0.4877 - val_acc: 0.8120  
Epoch 11/30  
100/100 [=====] - 74s 737ms/step - loss: 0.2741 -  
acc: 0.8807 - val_loss: 0.4539 - val_acc: 0.8080  
Epoch 12/30  
100/100 [=====] - 74s 737ms/step - loss: 0.2511 -  
acc: 0.9018 - val_loss: 0.4063 - val_acc: 0.8680  
Epoch 13/30  
100/100 [=====] - 74s 736ms/step - loss: 0.2306 -  
acc: 0.9058 - val_loss: 0.5284 - val_acc: 0.7840  
Epoch 14/30  
100/100 [=====] - 74s 737ms/step - loss: 0.2244 -  
acc: 0.9033 - val_loss: 0.4403 - val_acc: 0.8440  
Epoch 15/30  
100/100 [=====] - 74s 738ms/step - loss: 0.2271 -  
acc: 0.9063 - val_loss: 0.4706 - val_acc: 0.7920  
Epoch 16/30  
100/100 [=====] - 74s 736ms/step - loss: 0.1964 -  
acc: 0.9203 - val_loss: 0.5242 - val_acc: 0.8000  
Epoch 17/30  
100/100 [=====] - 74s 737ms/step - loss: 0.1971 -  
acc: 0.9248 - val_loss: 0.5315 - val_acc: 0.8200  
Epoch 18/30  
100/100 [=====] - 74s 738ms/step - loss: 0.1821 -  
acc: 0.9243 - val_loss: 0.5015 - val_acc: 0.8360  
Epoch 19/30  
100/100 [=====] - 74s 737ms/step - loss: 0.1836 -  
acc: 0.9313 - val_loss: 0.5159 - val_acc: 0.8160  
Epoch 20/30  
100/100 [=====] - 74s 737ms/step - loss: 0.1741 -  
acc: 0.9378 - val_loss: 0.4105 - val_acc: 0.8520  
Epoch 21/30  
100/100 [=====] - 74s 737ms/step - loss: 0.1639 -  
acc: 0.9333 - val_loss: 0.4575 - val_acc: 0.8120  
Epoch 22/30  
100/100 [=====] - 74s 736ms/step - loss: 0.1670 -  
acc: 0.9348 - val_loss: 0.5151 - val_acc: 0.8240  
Epoch 23/30  
100/100 [=====] - 74s 735ms/step - loss: 0.1659 -  
acc: 0.9288 - val_loss: 0.5952 - val_acc: 0.8080  
Epoch 24/30  
100/100 [=====] - 74s 736ms/step - loss: 0.1550 -  
acc: 0.9368 - val_loss: 0.6097 - val_acc: 0.8120  
Epoch 25/30  
100/100 [=====] - 73s 734ms/step - loss: 0.1383 -  
acc: 0.9484 - val_loss: 0.5103 - val_acc: 0.8040  
Epoch 26/30  
100/100 [=====] - 73s 735ms/step - loss: 0.1366 -  
acc: 0.9439 - val_loss: 0.5475 - val_acc: 0.8200  
Epoch 27/30  
100/100 [=====] - 74s 735ms/step - loss: 0.1340 -  
acc: 0.9514 - val_loss: 0.5088 - val_acc: 0.8160  
Epoch 28/30  
100/100 [=====] - 74s 737ms/step - loss: 0.1142 -  
acc: 0.9520 - val_loss: 0.5377 - val_acc: 0.8400  
Epoch 29/30  
100/100 [=====] - 73s 734ms/step - loss: 0.1365 -
```

```
acc: 0.9559 - val_loss: 0.5890 - val_acc: 0.8160
```

```
Epoch 30/30
```

```
100/100 [=====] - 73s 735ms/step - loss: 0.1157 -
```

```
In [13]: model.save('posters_conv1.h5')
```

```
In [14]: import matplotlib.pyplot as plt
```

```
acc = history.history['acc']
val_acc = history.history['val_acc']
loss = history.history['loss']
val_loss = history.history['val_loss']

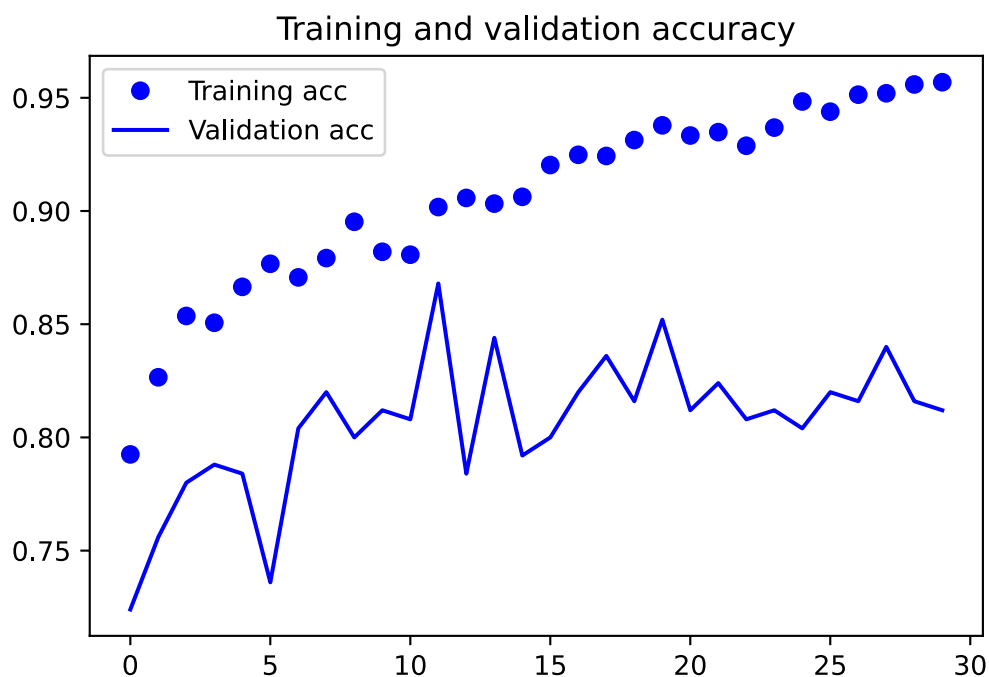
epochs = range(len(acc))

plt.plot(epochs, acc, 'bo', label='Training acc')
plt.plot(epochs, val_acc, 'b', label='Validation acc')
plt.title('Training and validation accuracy')
plt.legend()

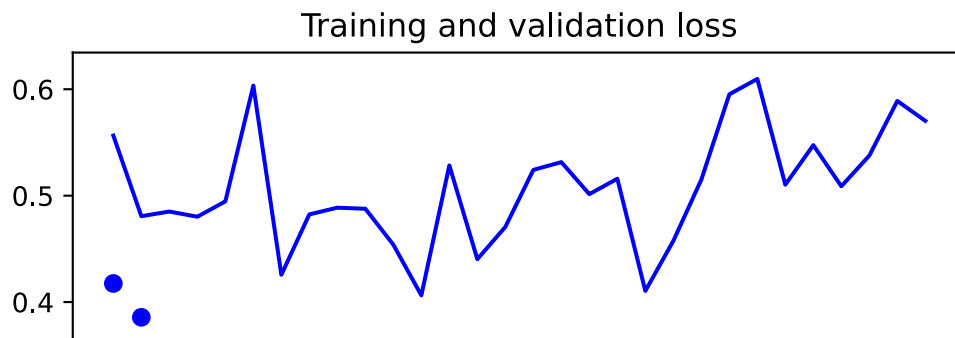
plt.figure()

plt.plot(epochs, loss, 'bo', label='Training loss')
plt.plot(epochs, val_loss, 'b', label='Validation loss')
plt.title('Training and validation loss')
plt.legend()

plt.show()
```







```
In [15]: test_loss, test_acc = model.evaluate_generator(test_generator, steps=50)
print('test acc:', test_acc)
```

test acc: 0.8119999766349792

## Fine tuning - Tercer modelo

```
In [16]: conv_base.trainable = True

set_trainable = False
for layer in conv_base.layers:
    if layer.name == 'block5_conv2':
        set_trainable = True
    if set_trainable:
        layer.trainable = True
    else:
        layer.trainable = False
```

```
In [17]: model.compile(loss='binary_crossentropy',
                        optimizer=optimizers.RMSprop(lr=1e-5),
                        metrics=['acc'])

history = model.fit_generator(
    train_generator,
    steps_per_epoch=100,
    epochs=30,
    validation_data=validation_generator,
    validation_steps=50)
```

```
Epoch 1/30
100/100 [=====] - 68s 682ms/step - loss: 0.1059 -
acc: 0.9554 - val_loss: 0.4868 - val_acc: 0.8560
Epoch 2/30
100/100 [=====] - 68s 682ms/step - loss: 0.1019 -
acc: 0.9614 - val_loss: 0.5365 - val_acc: 0.8200
Epoch 3/30
100/100 [=====] - 68s 684ms/step - loss: 0.0941 -
acc: 0.9624 - val_loss: 0.5311 - val_acc: 0.8360
Epoch 4/30
100/100 [=====] - 68s 684ms/step - loss: 0.1014 -
acc: 0.9639 - val_loss: 0.6964 - val_acc: 0.7960
Epoch 5/30
100/100 [=====] - 68s 685ms/step - loss: 0.0902 -
acc: 0.9680 - val_loss: 0.6528 - val_acc: 0.8200
Epoch 6/30
100/100 [=====] - 68s 683ms/step - loss: 0.0950 -
acc: 0.9694 - val_loss: 0.5962 - val_acc: 0.8280
Epoch 7/30
100/100 [=====] - 68s 683ms/step - loss: 0.0861 -
acc: 0.9689 - val_loss: 0.7763 - val_acc: 0.7720
Epoch 8/30
```

```
100/100 [=====] - 68s 684ms/step - loss: 0.0712 -  
acc: 0.9770 - val_loss: 0.5854 - val_acc: 0.8360  
Epoch 9/30  
100/100 [=====] - 69s 685ms/step - loss: 0.0842 -  
acc: 0.9704 - val_loss: 0.7462 - val_acc: 0.7760  
Epoch 10/30  
100/100 [=====] - 68s 684ms/step - loss: 0.0879 -  
acc: 0.9654 - val_loss: 0.7542 - val_acc: 0.8040  
Epoch 11/30  
100/100 [=====] - 68s 684ms/step - loss: 0.0838 -  
acc: 0.9714 - val_loss: 0.4791 - val_acc: 0.8440  
Epoch 12/30  
100/100 [=====] - 68s 684ms/step - loss: 0.0745 -  
acc: 0.9774 - val_loss: 0.6017 - val_acc: 0.8320  
Epoch 13/30  
100/100 [=====] - 68s 684ms/step - loss: 0.0814 -  
acc: 0.9664 - val_loss: 0.7402 - val_acc: 0.7800  
Epoch 14/30  
100/100 [=====] - 68s 684ms/step - loss: 0.0705 -  
acc: 0.9764 - val_loss: 0.7658 - val_acc: 0.8280  
Epoch 15/30  
100/100 [=====] - 68s 683ms/step - loss: 0.0843 -  
acc: 0.9694 - val_loss: 0.7845 - val_acc: 0.8120  
Epoch 16/30  
100/100 [=====] - 68s 684ms/step - loss: 0.0721 -  
acc: 0.9724 - val_loss: 0.6445 - val_acc: 0.8280  
Epoch 17/30  
100/100 [=====] - 68s 684ms/step - loss: 0.0672 -  
acc: 0.9749 - val_loss: 0.6085 - val_acc: 0.8360  
Epoch 18/30  
100/100 [=====] - 68s 683ms/step - loss: 0.0714 -  
acc: 0.9749 - val_loss: 0.7700 - val_acc: 0.8120  
Epoch 19/30  
100/100 [=====] - 68s 683ms/step - loss: 0.0623 -  
acc: 0.9764 - val_loss: 0.7297 - val_acc: 0.8000  
Epoch 20/30  
100/100 [=====] - 68s 684ms/step - loss: 0.0687 -  
acc: 0.9734 - val_loss: 0.5964 - val_acc: 0.8160  
Epoch 21/30  
100/100 [=====] - 68s 684ms/step - loss: 0.0669 -  
acc: 0.9794 - val_loss: 0.7710 - val_acc: 0.8200  
Epoch 22/30  
100/100 [=====] - 68s 684ms/step - loss: 0.0680 -  
acc: 0.9759 - val_loss: 0.8429 - val_acc: 0.8080  
Epoch 23/30  
100/100 [=====] - 68s 684ms/step - loss: 0.0588 -  
acc: 0.9764 - val_loss: 0.6545 - val_acc: 0.8400  
Epoch 24/30  
100/100 [=====] - 68s 683ms/step - loss: 0.0598 -  
acc: 0.9789 - val_loss: 0.8070 - val_acc: 0.8240  
Epoch 25/30  
100/100 [=====] - 68s 684ms/step - loss: 0.0571 -  
acc: 0.9830 - val_loss: 0.6898 - val_acc: 0.8360  
Epoch 26/30  
100/100 [=====] - 68s 683ms/step - loss: 0.0485 -  
acc: 0.9815 - val_loss: 0.7863 - val_acc: 0.7960  
Epoch 27/30  
100/100 [=====] - 68s 684ms/step - loss: 0.0530 -  
acc: 0.9820 - val_loss: 0.8238 - val_acc: 0.8200  
Epoch 28/30  
100/100 [=====] - 68s 683ms/step - loss: 0.0532 -  
acc: 0.9789 - val_loss: 0.6680 - val_acc: 0.8280  
Epoch 29/30  
100/100 [=====] - 69s 686ms/step - loss: 0.0499 -
```

```
acc: 0.9810 - val_loss: 0.8569 - val_acc: 0.8160
```

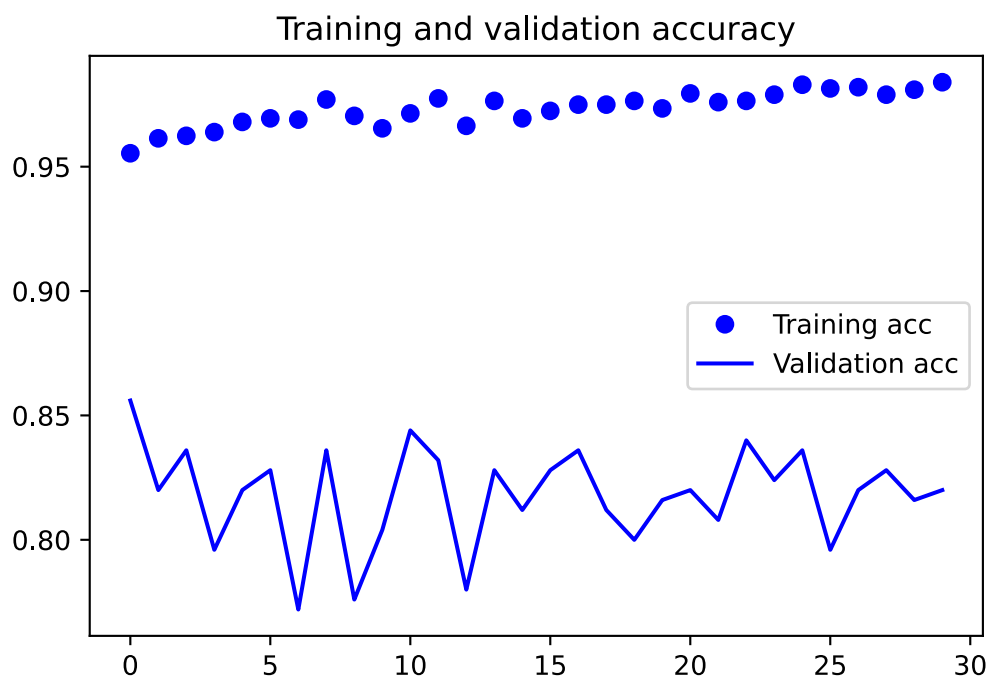
```
Epoch 30/30
```

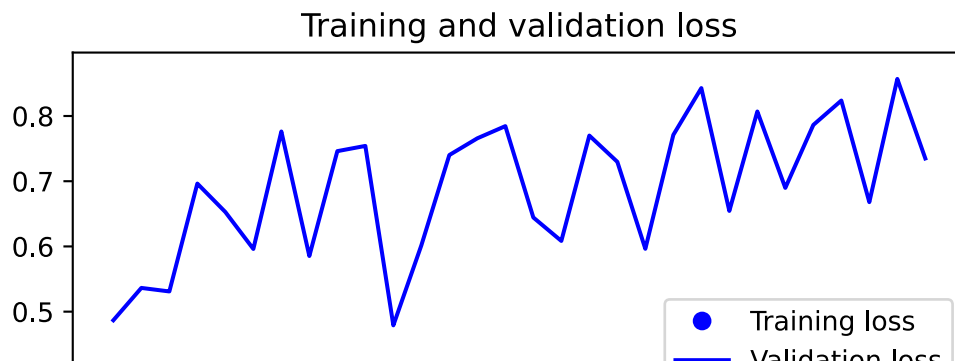
```
100/100 [=====] - 68s 684ms/step - loss: 0.0508 -  
acc: 0.9810 - val_loss: 0.8569 - val_acc: 0.8160
```

```
In [18]: model.save('posters_conv2.h5')
```

```
In [19]: import matplotlib.pyplot as plt
```

```
acc = history.history['acc']  
val_acc = history.history['val_acc']  
loss = history.history['loss']  
val_loss = history.history['val_loss']  
  
epochs = range(len(acc))  
  
plt.plot(epochs, acc, 'bo', label='Training acc')  
plt.plot(epochs, val_acc, 'b', label='Validation acc')  
plt.title('Training and validation accuracy')  
plt.legend()  
  
plt.figure()  
  
plt.plot(epochs, loss, 'bo', label='Training loss')  
plt.plot(epochs, val_loss, 'b', label='Validation loss')  
plt.title('Training and validation loss')  
plt.legend()  
  
plt.show()
```





```
In [20]: test_loss, test_acc = model.evaluate_generator(test_generator, steps=50)
print('test acc:', test_acc)
```

test acc: 0.8240000009536743

## Fine tuning - Cuarto modelo

```
In [21]: conv_base.trainable = True

set_trainable = False
for layer in conv_base.layers:
    if layer.name == 'block5_conv3':
        set_trainable = True
    if set_trainable:
        layer.trainable = True
    else:
        layer.trainable = False
```

```
In [22]: model.compile(loss='binary_crossentropy',
                        optimizer=optimizers.RMSprop(lr=1e-5),
                        metrics=['acc'])

history = model.fit_generator(
    train_generator,
    steps_per_epoch=100,
    epochs=30,
    validation_data=validation_generator,
    validation_steps=50)
```

Epoch 1/30  
100/100 [=====] - 63s 632ms/step - loss: 0.0431 -  
acc: 0.9860 - val\_loss: 0.6385 - val\_acc: 0.8360  
Epoch 2/30  
100/100 [=====] - 63s 632ms/step - loss: 0.0405 -  
acc: 0.9880 - val\_loss: 0.7490 - val\_acc: 0.8360  
Epoch 3/30  
100/100 [=====] - 63s 632ms/step - loss: 0.0394 -  
acc: 0.9855 - val\_loss: 0.9046 - val\_acc: 0.8080  
Epoch 4/30  
100/100 [=====] - 63s 633ms/step - loss: 0.0342 -  
acc: 0.9900 - val\_loss: 0.8062 - val\_acc: 0.8120  
Epoch 5/30  
100/100 [=====] - 63s 632ms/step - loss: 0.0512 -  
acc: 0.9835 - val\_loss: 0.8014 - val\_acc: 0.8200  
Epoch 6/30  
100/100 [=====] - 63s 634ms/step - loss: 0.0325 -  
acc: 0.9885 - val\_loss: 0.9386 - val\_acc: 0.7840  
Epoch 7/30  
100/100 [=====] - 63s 633ms/step - loss: 0.0387 -  
acc: 0.9865 - val\_loss: 0.6487 - val\_acc: 0.8520

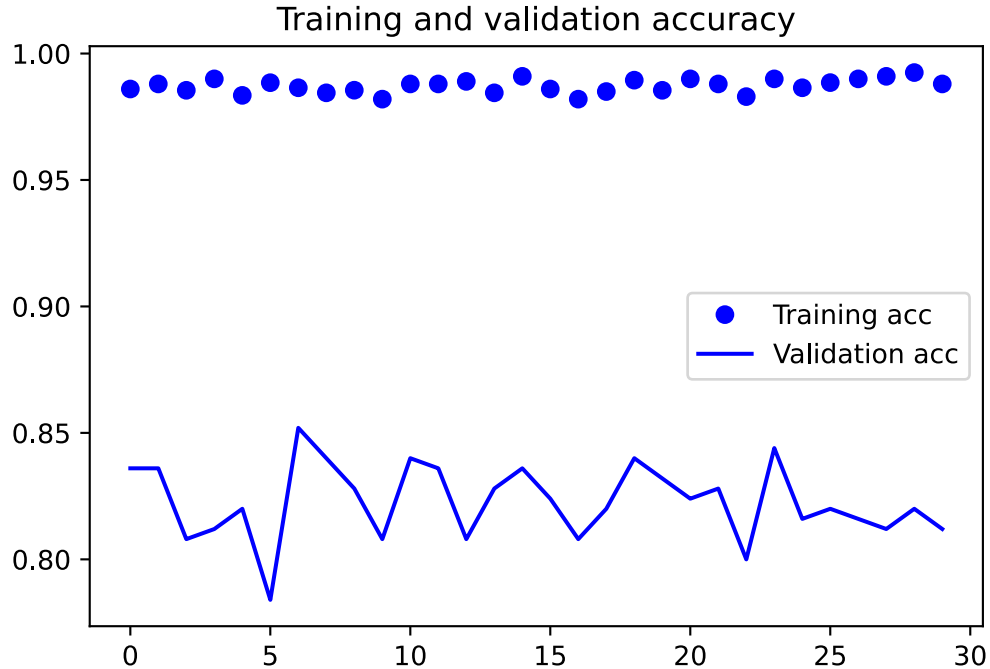
```
Epoch 8/30
100/100 [=====] - 63s 633ms/step - loss: 0.0415 -
acc: 0.9845 - val_loss: 0.6561 - val_acc: 0.8400
Epoch 9/30
100/100 [=====] - 63s 634ms/step - loss: 0.0432 -
acc: 0.9855 - val_loss: 0.7370 - val_acc: 0.8280
Epoch 10/30
100/100 [=====] - 63s 632ms/step - loss: 0.0435 -
acc: 0.9820 - val_loss: 0.9664 - val_acc: 0.8080
Epoch 11/30
100/100 [=====] - 63s 633ms/step - loss: 0.0404 -
acc: 0.9880 - val_loss: 0.7148 - val_acc: 0.8400
Epoch 12/30
100/100 [=====] - 63s 632ms/step - loss: 0.0360 -
acc: 0.9880 - val_loss: 0.7392 - val_acc: 0.8360
Epoch 13/30
100/100 [=====] - 63s 632ms/step - loss: 0.0353 -
acc: 0.9890 - val_loss: 0.9497 - val_acc: 0.8080
Epoch 14/30
100/100 [=====] - 63s 632ms/step - loss: 0.0435 -
acc: 0.9845 - val_loss: 0.7364 - val_acc: 0.8280
Epoch 15/30
100/100 [=====] - 63s 633ms/step - loss: 0.0330 -
acc: 0.9910 - val_loss: 0.8597 - val_acc: 0.8360
Epoch 16/30
100/100 [=====] - 63s 634ms/step - loss: 0.0413 -
acc: 0.9860 - val_loss: 0.7805 - val_acc: 0.8240
Epoch 17/30
100/100 [=====] - 63s 632ms/step - loss: 0.0445 -
acc: 0.9820 - val_loss: 0.9402 - val_acc: 0.8080
Epoch 18/30
100/100 [=====] - 63s 632ms/step - loss: 0.0437 -
acc: 0.9850 - val_loss: 0.9161 - val_acc: 0.8200
Epoch 19/30
100/100 [=====] - 63s 633ms/step - loss: 0.0334 -
acc: 0.9895 - val_loss: 0.9324 - val_acc: 0.8400
Epoch 20/30
100/100 [=====] - 63s 633ms/step - loss: 0.0392 -
acc: 0.9855 - val_loss: 0.7775 - val_acc: 0.8320
Epoch 21/30
100/100 [=====] - 63s 632ms/step - loss: 0.0293 -
acc: 0.9900 - val_loss: 0.7969 - val_acc: 0.8240
Epoch 22/30
100/100 [=====] - 63s 633ms/step - loss: 0.0330 -
acc: 0.9880 - val_loss: 0.7825 - val_acc: 0.8280
Epoch 23/30
100/100 [=====] - 63s 633ms/step - loss: 0.0484 -
acc: 0.9830 - val_loss: 0.9196 - val_acc: 0.8000
Epoch 24/30
100/100 [=====] - 63s 633ms/step - loss: 0.0308 -
acc: 0.9900 - val_loss: 0.7358 - val_acc: 0.8440
Epoch 25/30
100/100 [=====] - 63s 633ms/step - loss: 0.0369 -
acc: 0.9865 - val_loss: 0.8598 - val_acc: 0.8160
Epoch 26/30
100/100 [=====] - 63s 634ms/step - loss: 0.0398 -
acc: 0.9885 - val_loss: 0.9222 - val_acc: 0.8200
Epoch 27/30
100/100 [=====] - 63s 633ms/step - loss: 0.0331 -
acc: 0.9900 - val_loss: 0.8599 - val_acc: 0.8160
Epoch 28/30
100/100 [=====] - 63s 633ms/step - loss: 0.0279 -
acc: 0.9910 - val_loss: 0.9307 - val_acc: 0.8120
Epoch 29/30
```

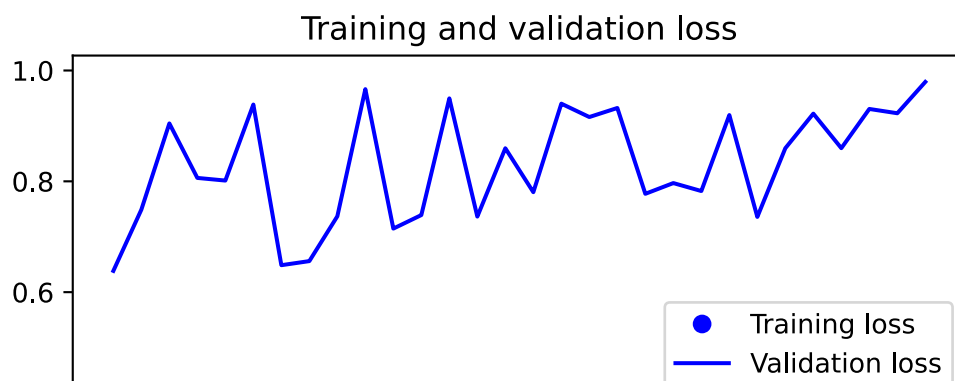
```
100/100 [=====] - 63s 632ms/step - loss: 0.0281 -  
acc: 0.9925 - val_loss: 0.9229 - val_acc: 0.8200  
Epoch 30/30  
100/100 [=====] - 63s 632ms/step - loss: 0.0363 -
```

```
In [23]: model.save('posters_conv3.h5')
```

```
In [24]: import matplotlib.pyplot as plt
```

```
acc = history.history['acc']  
val_acc = history.history['val_acc']  
loss = history.history['loss']  
val_loss = history.history['val_loss']  
  
epochs = range(len(acc))  
  
plt.plot(epochs, acc, 'bo', label='Training acc')  
plt.plot(epochs, val_acc, 'b', label='Validation acc')  
plt.title('Training and validation accuracy')  
plt.legend()  
  
plt.figure()  
  
plt.plot(epochs, loss, 'bo', label='Training loss')  
plt.plot(epochs, val_loss, 'b', label='Validation loss')  
plt.title('Training and validation loss')  
plt.legend()  
  
plt.show()
```





```
In [25]: test_loss, test_acc = model.evaluate_generator(test_generator, steps=50)
print('test acc:', test_acc)
```

test acc: 0.8320000171661377