

CT5133 / CT5145 Deep Learning (/Online) 2022-2023

Assignment 2

James McDermott

- Student ID(s): 21249929
- Student name(s): Caolan McDonagh

Due date: midnight Sunday 19 March (end Week 10).

Weighting: 20% of the module.

In this assignment the goal is to take advantage of pre-trained NN models to create an embedding with a dataset of movie posters, and demonstrate how to use that embedding.

The dataset is provided, along with some skeleton code for loading it.

The individual steps to be carried out are specified below, with `### YOUR CODE HERE` markers, together with the number of marks available for each part.

- **Topics:** in Part 5 below, students are asked to add some improvement to their models. In general, these improvements will differ between students (or student groups). **The proposed improvement must be notified to the lecturer at least 1 week before submission, and approved by the lecturer.** If working in a group, the two members of the group should not work on different topics in Part 5: they must work on the same topic and submit identical submissions.
- Students are not required to work incrementally on the parts. It is ok to do all the work in one day, so long as you abide by the rules on notifying groups and notifying topics.
- **Groups:** students may work solo or in a group of two. A student may not work together in a group with any student they have previously worked on a group project with, in this module or any other in the MSc programme. **Groups must be notified to the lecturer in writing**

before beginning work and at least 1 week before submission. If working in a group, both students must submit and both submissions must be identical. If working in a group, both students may be asked to explain any aspect of the code in interview (see below), therefore working independently on separate components is not recommended. Any emails concerning the project should be cc-ed to the other group member.

- **Libraries:** code can be written in Keras/Tensorflow, or in PyTorch.
- **Plagiarism:** students may discuss the assignment together, but you may not look at another student or group's work or allow other students to view yours (other than within a group). You may use snippets of code (eg 1-2 lines) from the internet, **if you provide a citation with URL.** You may also use a longer snippet of code if it is a utility function, again only with citation. You may not use code from the internet to carry out the core of the assignment. You may not use a large language model to generate code.
- **Submission:** after completing your work in this Jupyter notebook, submit the notebook both in `.ipynb` and `.pdf` formats. The content should be identical.
- **Interviews:** a number of students may be selected for interview, post-submission. The selection will depend on submissions, and random chance may be used also. Interviews will be held in-person (CT5133) or online (CT5145). Interviews will last approximately 10 minutes. The purpose of interviews will be to assess students' understanding of their own submission.

Dataset Credits

The original csv file is from:

<https://www.kaggle.com/datasets/neha1703/movie-genre-from-its-poster>

I have added the *year* column for convenience.

I believe most of the information is originally from the famous MovieLens dataset:

- <https://grouplens.org/datasets/movielens/>
- <https://movielens.org/>

However, I'm not clear whether the poster download URLs (Amazon AWS URLs) which are in the csv obtained from the Kaggle URL above are from a MovieLens source, or elsewhere.

To create the dataset we are using, I have randomly sampled a small proportion of the URLs in the csv, and downloaded the images. I have removed those which fail to download. Code below also filters out those which are in black and white, ie 1 channel only.

Imports

You can add more imports if needed.

```
In [ ]: import numpy as np
import pandas as pd
import os
from PIL import Image
import matplotlib.pyplot as plt
from scipy.spatial.distance import cdist, pdist, squareform # useful for distances in the embedding
```

```
In [ ]: import tensorflow as tf
from tensorflow import keras
from keras import layers, models
```

Utility functions

These functions are provided to save you time. You might not need to understand any of the details here.

```
In [ ]: # walk the directory containing posters and read them in. all are the same shape: (268, 182).
# all have 3 channels, with a few exceptions (see below).
# each is named <imdbId>.jpg, which will later allow us to get the metadata from the csv.
IDs = []
images = []
for dirname, _, filenames in os.walk('DL_Sample'):
    for filename in filenames:
        if filename.endswith('.jpg'):
            ID = int(filename[:-4])
            pathname = os.path.join(dirname, filename)
            im = Image.open(pathname)
            imnp = np.array(im, dtype=float)
            if len(imnp.shape) != 3: # we'll ignore a few black-and-white (1 channel) images
                print("This is 1 channel, so we omit it", imnp.shape, filename)
                continue # do not add to our list
```

```
    IDs.append(ID)
    images.append(imnp)
```

In []: `img_array = np.array(images)`

In []: `img_array.shape`

Out[]: `(1238, 268, 182, 3)`

In []: `# read the csv`
`df = pd.read_csv("Movie_Genre_Year_Poster.csv", encoding="ISO-8859-1", index_col="Unnamed: 0")`
`df.head()`

Out[]:

	imdbId	Imdb Link	Title	IMDB Score	Genre	Poster	Year
0	114709	http://www.imdb.com/title/tt114709	Toy Story (1995)	8.3	Animation Adventure Comedy	https://images-na.ssl-images-amazon.com/images...	1995.0
1	113497	http://www.imdb.com/title/tt113497	Jumanji (1995)	6.9	Action Adventure Family	https://images-na.ssl-images-amazon.com/images...	1995.0
2	113228	http://www.imdb.com/title/tt113228	Grumpier Old Men (1995)	6.6	Comedy Romance	https://images-na.ssl-images-amazon.com/images...	1995.0
3	114885	http://www.imdb.com/title/tt114885	Waiting to Exhale (1995)	5.7	Comedy Drama Romance	https://images-na.ssl-images-amazon.com/images...	1995.0
4	113041	http://www.imdb.com/title/tt113041	Father of the Bride Part II (1995)	5.9	Comedy Family Romance	https://images-na.ssl-images-amazon.com/images...	1995.0

In []: `df2 = df.drop_duplicates(subset=["imdbId"]) # some imdbId values are duplicates - just drop`

In []: `df3 = df2.set_index("imdbId") # the imdbId is a more useful index, eg as in the next cell...`

In []: `df4 = df3.loc[IDs] # ... we can now use .loc to take a subset`

In []: `df4.shape # 1238 rows matches the image data shape above`

Out[]: `(1238, 6)`

```
In [ ]: years = df4["Year"].values
titles = df4["Title"].values

assert img_array.shape[0] == years.shape[0] == titles.shape[0]
assert all(np.isfinite(years)) # no missing values in years
print(years.min(), years.max())
```

1888.0 2016.0

```
In [ ]: def imread(filename):
    """Convenience function: we can supply an ID or a filename.
    We read and return the image in Image format.
    """

    if type(filename) == int:
        # assume its an ID, so create filename
        filename = f"DL_Sample/{filename}.jpg"

        # now we can assume it's a filename, so open and read
        im = Image.open(filename)

    return im

def imshow(im):
    plt.imshow(im)
    plt.axis('off')
```

Part 1. Create embedding [3 marks]

Use a pretrained model, eg as provided by Keras, to create a flat (ie 1D) embedding vector of some size `embedding_size` for each movie poster, and put all of these together into a single tensor of shape `(n_movies, embedding_size)`.

```
In [ ]: n_movies = img_array.shape[0]
embedding_size = 76032 # Acquired via output.shape of initModel.
X = tf.zeros((n_movies, embedding_size))
### YOUR CODE HERE

#Imports relevant to part 1
from keras.applications.efficientnet import EfficientNetB2
from keras.models import Model
```

```
#include_top = false to disconnect final layer.  
initModel = EfficientNetB2(include_top = False, weights = 'imagenet', input_shape=(268,182,3))  
#Create our flat layer for embeddings  
flatLayer = layers.Flatten()(initModel.output)  
  
trainedModel = Model(inputs=initModel.input,outputs=flatLayer)  
  
trainedModel.summary()  
  
#Get the embeddings of the img_array  
embeddingVectors = trainedModel.predict(img_array)  
  
X = np.vstack(embeddingVectors)  
  
assert len(X.shape) == 2 # X should be (n_movies, embedding_size)  
assert X.shape[0] == n_movies
```

Model: "model_10"

Layer (type)	Output Shape	Param #	Connected to
input_11 (InputLayer)	[(None, 268, 182, 3 0)]	0	[]
rescaling_5 (Rescaling)	(None, 268, 182, 3) 0	0	['input_11[0][0]']
normalization_5 (Normalization)	(None, 268, 182, 3) 7	7	['rescaling_5[0][0]']
tf.math.truediv_2 (TFOpLambda)	(None, 268, 182, 3) 0	0	['normalization_5[0][0]']
stem_conv_pad (ZeroPadding2D)	(None, 269, 183, 3) 0	0	['tf.math.truediv_2[0][0]']
stem_conv (Conv2D)	(None, 134, 91, 32) 864	864	['stem_conv_pad[0][0]']
stem_bn (BatchNormalization)	(None, 134, 91, 32) 128	128	['stem_conv[0][0]']
stem_activation (Activation)	(None, 134, 91, 32) 0	0	['stem_bn[0][0]']
block1a_dwconv (DepthwiseConv2D)	(None, 134, 91, 32) 288	288	['stem_activation[0][0]']
block1a_bn (BatchNormalization)	(None, 134, 91, 32) 128	128	['block1a_dwconv[0][0]']
block1a_activation (Activation)	(None, 134, 91, 32) 0	0	['block1a_bn[0][0]']
block1a_se_squeeze (GlobalAveragePooling2D)	(None, 32) 0	0	['block1a_activation[0][0]']
block1a_se_reshape (Reshape)	(None, 1, 1, 32) 0	0	['block1a_se_squeeze[0][0]']
block1a_se_reduce (Conv2D)	(None, 1, 1, 8) 264	264	['block1a_se_reshape[0][0]']
block1a_se_expand (Conv2D)	(None, 1, 1, 32) 288	288	['block1a_se_reduce[0][0]']
block1a_se_excite (Multiply)	(None, 134, 91, 32) 0	0	['block1a_activation[0][0]', 'block1a_se_expand[0][0]']
block1a_project_conv (Conv2D)	(None, 134, 91, 16) 512	512	['block1a_se_excite[0][0]']

block1a_project_bn (BatchNormaliza lization)	(None, 134, 91, 16) 64	['block1a_project_conv[0][0]']
block1b_dwconv (DepthwiseConv2 D)	(None, 134, 91, 16) 144	['block1a_project_bn[0][0]']
block1b_bn (BatchNormalizatio n)	(None, 134, 91, 16) 64	['block1b_dwconv[0][0]']
block1b_activation (Activatio n)	(None, 134, 91, 16) 0	['block1b_bn[0][0]']
block1b_se_squeeze (GlobalAver agePooling2D)	(None, 16) 0	['block1b_activation[0][0]']
block1b_se_reshape (Reshape)	(None, 1, 1, 16) 0	['block1b_se_squeeze[0][0]']
block1b_se_reduce (Conv2D)	(None, 1, 1, 4) 68	['block1b_se_reshape[0][0]']
block1b_se_expand (Conv2D)	(None, 1, 1, 16) 80	['block1b_se_reduce[0][0]']
block1b_se_excite (Multipli cation)	(None, 134, 91, 16) 0	['block1b_activation[0][0]', 'block1b_se_expand[0][0]']
block1b_project_conv (Conv2D)	(None, 134, 91, 16) 256	['block1b_se_excite[0][0]']
block1b_project_bn (BatchNorma lization)	(None, 134, 91, 16) 64	['block1b_project_conv[0][0]']
block1b_drop (Dropout)	(None, 134, 91, 16) 0	['block1b_project_bn[0][0]']
block1b_add (Add)	(None, 134, 91, 16) 0	['block1b_drop[0][0]', 'block1a_project_bn[0][0]']
block2a_expand_conv (Conv2D)	(None, 134, 91, 96) 1536	['block1b_add[0][0]']
block2a_expand_bn (BatchNormal ization)	(None, 134, 91, 96) 384	['block2a_expand_conv[0][0]']
block2a_expand_activation (Act ivation)	(None, 134, 91, 96) 0	['block2a_expand_bn[0][0]']
block2a_dwconv_pad (ZeroPaddin g)	(None, 135, 93, 96) 0	['block2a_expand_activation[0][0]

g2D)		']
block2a_dwconv (DepthwiseConv2D) (None, 67, 46, 96) 864		['block2a_dwconv_pad[0][0]']
block2a_bn (BatchNormalization) (None, 67, 46, 96) 384		['block2a_dwconv[0][0]']
block2a_activation (Activation) (None, 67, 46, 96) 0		['block2a_bn[0][0]']
block2a_se_squeeze (GlobalAveragePooling2D)	0	['block2a_activation[0][0]']
block2a_se_reshape (Reshape) (None, 1, 1, 96)	0	['block2a_se_squeeze[0][0]']
block2a_se_reduce (Conv2D) (None, 1, 1, 4)	388	['block2a_se_reshape[0][0]']
block2a_se_expand (Conv2D) (None, 1, 1, 96)	480	['block2a_se_reduce[0][0]']
block2a_se_excite (Multiply) (None, 67, 46, 96) 0		['block2a_activation[0][0]', 'block2a_se_expand[0][0]']
block2a_project_conv (Conv2D) (None, 67, 46, 24)	2304	['block2a_se_excite[0][0]']
block2a_project_bn (BatchNormalization) (None, 67, 46, 24)	96	['block2a_project_conv[0][0]']
block2b_expand_conv (Conv2D) (None, 67, 46, 144)	3456	['block2a_project_bn[0][0]']
block2b_expand_bn (BatchNormalization) (None, 67, 46, 144)	576	['block2b_expand_conv[0][0]']
block2b_expand_activation (Activation) (None, 67, 46, 144) 0		['block2b_expand_bn[0][0]']
block2b_dwconv (DepthwiseConv2D) (None, 67, 46, 144) 1296		['block2b_expand_activation[0][0]']
block2b_bn (BatchNormalization) (None, 67, 46, 144) 576		['block2b_dwconv[0][0]']
block2b_activation (Activation) (None, 67, 46, 144) 0		['block2b_bn[0][0]']

block2b_se_squeeze (GlobalAveragePooling2D)	(None, 144)	0	['block2b_activation[0][0]']
block2b_se_reshape (Reshape)	(None, 1, 1, 144)	0	['block2b_se_squeeze[0][0]']
block2b_se_reduce (Conv2D)	(None, 1, 1, 6)	870	['block2b_se_reshape[0][0]']
block2b_se_expand (Conv2D)	(None, 1, 1, 144)	1008	['block2b_se_reduce[0][0]']
block2b_se_excite (Multiply)	(None, 67, 46, 144)	0	['block2b_activation[0][0]', 'block2b_se_expand[0][0]']
block2b_project_conv (Conv2D)	(None, 67, 46, 24)	3456	['block2b_se_excite[0][0]']
block2b_project_bn (BatchNormalization)	(None, 67, 46, 24)	96	['block2b_project_conv[0][0]']
block2b_drop (Dropout)	(None, 67, 46, 24)	0	['block2b_project_bn[0][0]']
block2b_add (Add)	(None, 67, 46, 24)	0	['block2b_drop[0][0]', 'block2a_project_bn[0][0]']
block2c_expand_conv (Conv2D)	(None, 67, 46, 144)	3456	['block2b_add[0][0]']
block2c_expand_bn (BatchNormalization)	(None, 67, 46, 144)	576	['block2c_expand_conv[0][0]']
block2c_expand_activation (Activation)	(None, 67, 46, 144)	0	['block2c_expand_bn[0][0]']
block2c_dwconv (DepthwiseConv2D)	(None, 67, 46, 144)	1296	['block2c_expand_activation[0][0]']
block2c_bn (BatchNormalization)	(None, 67, 46, 144)	576	['block2c_dwconv[0][0]']
block2c_activation (Activation)	(None, 67, 46, 144)	0	['block2c_bn[0][0]']
block2c_se_squeeze (GlobalAveragePooling2D)	(None, 144)	0	['block2c_activation[0][0]']
block2c_se_reshape (Reshape)	(None, 1, 1, 144)	0	['block2c_se_squeeze[0][0]']

block2c_se_reduce (Conv2D)	(None, 1, 1, 6)	870	['block2c_se_reshape[0][0]']
block2c_se_expand (Conv2D)	(None, 1, 1, 144)	1008	['block2c_se_reduce[0][0]']
block2c_se_excite (Multiply)	(None, 67, 46, 144)	0	['block2c_activation[0][0]', 'block2c_se_expand[0][0]']
block2c_project_conv (Conv2D)	(None, 67, 46, 24)	3456	['block2c_se_excite[0][0]']
block2c_project_bn (BatchNormala lization)	(None, 67, 46, 24)	96	['block2c_project_conv[0][0]']
block2c_drop (Dropout)	(None, 67, 46, 24)	0	['block2c_project_bn[0][0]']
block2c_add (Add)	(None, 67, 46, 24)	0	['block2c_drop[0][0]', 'block2b_add[0][0]']
block3a_expand_conv (Conv2D)	(None, 67, 46, 144)	3456	['block2c_add[0][0]']
block3a_expand_bn (BatchNormala lization)	(None, 67, 46, 144)	576	['block3a_expand_conv[0][0]']
block3a_expand_activation (Act ivation)	(None, 67, 46, 144)	0	['block3a_expand_bn[0][0]']
block3a_dwconv_pad (ZeroPaddin g2D)	(None, 71, 49, 144)	0	['block3a_expand_activation[0][0]']
block3a_dwconv (DepthwiseConv2 D)	(None, 34, 23, 144)	3600	['block3a_dwconv_pad[0][0]']
block3a_bn (BatchNormalizatio n)	(None, 34, 23, 144)	576	['block3a_dwconv[0][0]']
block3a_activation (Activation)	(None, 34, 23, 144)	0	['block3a_bn[0][0]']
block3a_se_squeeze (GlobalAver agePooling2D)	(None, 144)	0	['block3a_activation[0][0]']
block3a_se_reshape (Reshape)	(None, 1, 1, 144)	0	['block3a_se_squeeze[0][0]']
block3a_se_reduce (Conv2D)	(None, 1, 1, 6)	870	['block3a_se_reshape[0][0]']

block3a_se_expand (Conv2D)	(None, 1, 1, 144)	1008	['block3a_se_reduce[0][0]']
block3a_se_excite (Multiply)	(None, 34, 23, 144)	0	['block3a_activation[0][0]', 'block3a_se_expand[0][0]']
block3a_project_conv (Conv2D)	(None, 34, 23, 48)	6912	['block3a_se_excite[0][0]']
block3a_project_bn (BatchNorma lization)	(None, 34, 23, 48)	192	['block3a_project_conv[0][0]']
block3b_expand_conv (Conv2D)	(None, 34, 23, 288)	13824	['block3a_project_bn[0][0]']
block3b_expand_bn (BatchNormal ization)	(None, 34, 23, 288)	1152	['block3b_expand_conv[0][0]']
block3b_expand_activation (Act ivation)	(None, 34, 23, 288)	0	['block3b_expand_bn[0][0]']
block3b_dwconv (DepthwiseConv2 D)	(None, 34, 23, 288)	7200	['block3b_expand_activation[0][0] ']
block3b_bn (BatchNormalizatio n)	(None, 34, 23, 288)	1152	['block3b_dwconv[0][0]']
block3b_activation (Activation)	(None, 34, 23, 288)	0	['block3b_bn[0][0]']
block3b_se_squeeze (GlobalAver agePooling2D)	(None, 288)	0	['block3b_activation[0][0]']
block3b_se_reshape (Reshape)	(None, 1, 1, 288)	0	['block3b_se_squeeze[0][0]']
block3b_se_reduce (Conv2D)	(None, 1, 1, 12)	3468	['block3b_se_reshape[0][0]']
block3b_se_expand (Conv2D)	(None, 1, 1, 288)	3744	['block3b_se_reduce[0][0]']
block3b_se_excite (Multiply)	(None, 34, 23, 288)	0	['block3b_activation[0][0]', 'block3b_se_expand[0][0]']
block3b_project_conv (Conv2D)	(None, 34, 23, 48)	13824	['block3b_se_excite[0][0]']
block3b_project_bn (BatchNorma lization)	(None, 34, 23, 48)	192	['block3b_project_conv[0][0]']

block3b_drop (Dropout)	(None, 34, 23, 48)	0	['block3b_project_bn[0][0]']
block3b_add (Add)	(None, 34, 23, 48)	0	['block3b_drop[0][0]', 'block3a_project_bn[0][0]']
block3c_expand_conv (Conv2D)	(None, 34, 23, 288)	13824	['block3b_add[0][0]']
block3c_expand_bn (BatchNormal ization)	(None, 34, 23, 288)	1152	['block3c_expand_conv[0][0]']
block3c_expand_activation (Act ivation)	(None, 34, 23, 288)	0	['block3c_expand_bn[0][0]']
block3c_dwconv (DepthwiseConv2 D)	(None, 34, 23, 288)	7200	['block3c_expand_activation[0][0]']
block3c_bn (BatchNormalizatio n)	(None, 34, 23, 288)	1152	['block3c_dwconv[0][0]']
block3c_activation (Activation)	(None, 34, 23, 288)	0	['block3c_bn[0][0]']
block3c_se_squeeze (GlobalAver agePooling2D)	(None, 288)	0	['block3c_activation[0][0]']
block3c_se_reshape (Reshape)	(None, 1, 1, 288)	0	['block3c_se_squeeze[0][0]']
block3c_se_reduce (Conv2D)	(None, 1, 1, 12)	3468	['block3c_se_reshape[0][0]']
block3c_se_expand (Conv2D)	(None, 1, 1, 288)	3744	['block3c_se_reduce[0][0]']
block3c_se_excite (Multiply)	(None, 34, 23, 288)	0	['block3c_activation[0][0]', 'block3c_se_expand[0][0]']
block3c_project_conv (Conv2D)	(None, 34, 23, 48)	13824	['block3c_se_excite[0][0]']
block3c_project_bn (BatchNorma lization)	(None, 34, 23, 48)	192	['block3c_project_conv[0][0]']
block3c_drop (Dropout)	(None, 34, 23, 48)	0	['block3c_project_bn[0][0]']
block3c_add (Add)	(None, 34, 23, 48)	0	['block3c_drop[0][0]', 'block3b_add[0][0]']

block4a_expand_conv (Conv2D) (None, 34, 23, 288)	13824	['block3c_add[0][0]']
block4a_expand_bn (BatchNormal ization)	1152	['block4a_expand_conv[0][0]']
block4a_expand_activation (Act ivation)	0	['block4a_expand_bn[0][0]']
block4a_dwconv_pad (ZeroPaddin g2D)	0	['block4a_expand_activation[0][0]']
block4a_dwconv (DepthwiseConv2 D)	2592	['block4a_dwconv_pad[0][0]']
block4a_bn (BatchNormalizatio n)	1152	['block4a_dwconv[0][0]']
block4a_activation (Activation)	0	['block4a_bn[0][0]']
block4a_se_squeeze (GlobalAver agePooling2D)	0	['block4a_activation[0][0]']
block4a_se_reshape (Reshape)	0	['block4a_se_squeeze[0][0]']
block4a_se_reduce (Conv2D)	3468	['block4a_se_reshape[0][0]']
block4a_se_expand (Conv2D)	3744	['block4a_se_reduce[0][0]']
block4a_se_excite (Multiply)	0	['block4a_activation[0][0]', 'block4a_se_expand[0][0]']
block4a_project_conv (Conv2D)	25344	['block4a_se_excite[0][0]']
block4a_project_bn (BatchNorma lization)	352	['block4a_project_conv[0][0]']
block4b_expand_conv (Conv2D)	46464	['block4a_project_bn[0][0]']
block4b_expand_bn (BatchNormal ization)	2112	['block4b_expand_conv[0][0]']
block4b_expand_activation (Act ivation)	0	['block4b_expand_bn[0][0]']

ivation)			
block4b_dwconv (DepthwiseConv2D) (None, 17, 12, 528)	4752	['block4b_expand_activation[0][0]']	
block4b_bn (BatchNormalization) (None, 17, 12, 528)	2112	['block4b_dwconv[0][0]']	
block4b_activation (Activation) (None, 17, 12, 528)	0	['block4b_bn[0][0]']	
block4b_se_squeeze (GlobalAveragePooling2D)	0	['block4b_activation[0][0]']	
block4b_se_reshape (Reshape)	(None, 1, 1, 528)	0	['block4b_se_squeeze[0][0]']
block4b_se_reduce (Conv2D)	(None, 1, 1, 22)	11638	['block4b_se_reshape[0][0]']
block4b_se_expand (Conv2D)	(None, 1, 1, 528)	12144	['block4b_se_reduce[0][0]']
block4b_se_excite (Multiply)	(None, 17, 12, 528)	0	['block4b_activation[0][0]', 'block4b_se_expand[0][0]']
block4b_project_conv (Conv2D)	(None, 17, 12, 88)	46464	['block4b_se_excite[0][0]']
block4b_project_bn (BatchNormalization)	(None, 17, 12, 88)	352	['block4b_project_conv[0][0]']
block4b_drop (Dropout)	(None, 17, 12, 88)	0	['block4b_project_bn[0][0]']
block4b_add (Add)	(None, 17, 12, 88)	0	['block4b_drop[0][0]', 'block4a_project_bn[0][0]']
block4c_expand_conv (Conv2D)	(None, 17, 12, 528)	46464	['block4b_add[0][0]']
block4c_expand_bn (BatchNormalization)	(None, 17, 12, 528)	2112	['block4c_expand_conv[0][0]']
block4c_expand_activation (Activation)	(None, 17, 12, 528)	0	['block4c_expand_bn[0][0]']
block4c_dwconv (DepthwiseConv2D)	(None, 17, 12, 528)	4752	['block4c_expand_activation[0][0]']

block4c_bn (BatchNormalization (None, 17, 12, 528) 2112)		['block4c_dwconv[0][0]']
block4c_activation (Activation (None, 17, 12, 528) 0)		['block4c_bn[0][0]']
block4c_se_squeeze (GlobalAveragePooling2D (None, 528) 0)		['block4c_activation[0][0]']
block4c_se_reshape (Reshape (None, 1, 1, 528) 0)		['block4c_se_squeeze[0][0]']
block4c_se_reduce (Conv2D (None, 1, 1, 22) 11638)		['block4c_se_reshape[0][0]']
block4c_se_expand (Conv2D (None, 1, 1, 528) 12144)		['block4c_se_reduce[0][0]']
block4c_se_excite (Multiply (None, 17, 12, 528) 0)		['block4c_activation[0][0]', 'block4c_se_expand[0][0]']
block4c_project_conv (Conv2D (None, 17, 12, 88) 46464)		['block4c_se_excite[0][0]']
block4c_project_bn (BatchNormalization (None, 17, 12, 88) 352)		['block4c_project_conv[0][0]']
block4c_drop (Dropout (None, 17, 12, 88) 0)		['block4c_project_bn[0][0]']
block4c_add (Add (None, 17, 12, 88) 0)		['block4c_drop[0][0]', 'block4b_add[0][0]']
block4d_expand_conv (Conv2D (None, 17, 12, 528) 46464)		['block4c_add[0][0]']
block4d_expand_bn (BatchNormalization (None, 17, 12, 528) 2112)		['block4d_expand_conv[0][0]']
block4d_expand_activation (Activation (None, 17, 12, 528) 0)		['block4d_expand_bn[0][0]']
block4d_dwconv (DepthwiseConv2D (None, 17, 12, 528) 4752)		['block4d_expand_activation[0][0]']
block4d_bn (BatchNormalization (None, 17, 12, 528) 2112)		['block4d_dwconv[0][0]']
block4d_activation (Activation (None, 17, 12, 528) 0)		['block4d_bn[0][0]']

block4d_se_squeeze (GlobalAveragePooling2D)	(None, 528)	0	['block4d_activation[0][0]']
block4d_se_reshape (Reshape)	(None, 1, 1, 528)	0	['block4d_se_squeeze[0][0]']
block4d_se_reduce (Conv2D)	(None, 1, 1, 22)	11638	['block4d_se_reshape[0][0]']
block4d_se_expand (Conv2D)	(None, 1, 1, 528)	12144	['block4d_se_reduce[0][0]']
block4d_se_excite (Multiply)	(None, 17, 12, 528)	0	['block4d_activation[0][0]', 'block4d_se_expand[0][0]']
block4d_project_conv (Conv2D)	(None, 17, 12, 88)	46464	['block4d_se_excite[0][0]']
block4d_project_bn (BatchNormalization)	(None, 17, 12, 88)	352	['block4d_project_conv[0][0]']
block4d_drop (Dropout)	(None, 17, 12, 88)	0	['block4d_project_bn[0][0]']
block4d_add (Add)	(None, 17, 12, 88)	0	['block4d_drop[0][0]', 'block4c_add[0][0]']
block5a_expand_conv (Conv2D)	(None, 17, 12, 528)	46464	['block4d_add[0][0]']
block5a_expand_bn (BatchNormalization)	(None, 17, 12, 528)	2112	['block5a_expand_conv[0][0]']
block5a_expand_activation (Activation)	(None, 17, 12, 528)	0	['block5a_expand_bn[0][0]']
block5a_dwconv (DepthwiseConv2D)	(None, 17, 12, 528)	13200	['block5a_expand_activation[0][0]']
block5a_bn (BatchNormalization)	(None, 17, 12, 528)	2112	['block5a_dwconv[0][0]']
block5a_activation (Activation)	(None, 17, 12, 528)	0	['block5a_bn[0][0]']
block5a_se_squeeze (GlobalAveragePooling2D)	(None, 528)	0	['block5a_activation[0][0]']
block5a_se_reshape (Reshape)	(None, 1, 1, 528)	0	['block5a_se_squeeze[0][0]']

block5a_se_reduce (Conv2D)	(None, 1, 1, 22)	11638	['block5a_se_reshape[0][0]']
block5a_se_expand (Conv2D)	(None, 1, 1, 528)	12144	['block5a_se_reduce[0][0]']
block5a_se_excite (Multiply)	(None, 17, 12, 528)	0	['block5a_activation[0][0]', 'block5a_se_expand[0][0]']
block5a_project_conv (Conv2D)	(None, 17, 12, 120)	63360	['block5a_se_excite[0][0]']
block5a_project_bn (BatchNormala lization)	(None, 17, 12, 120)	480	['block5a_project_conv[0][0]']
block5b_expand_conv (Conv2D)	(None, 17, 12, 720)	86400	['block5a_project_bn[0][0]']
block5b_expand_bn (BatchNormala lization)	(None, 17, 12, 720)	2880	['block5b_expand_conv[0][0]']
block5b_expand_activation (Act ivation)	(None, 17, 12, 720)	0	['block5b_expand_bn[0][0]']
block5b_dwconv (DepthwiseConv2 D)	(None, 17, 12, 720)	18000	['block5b_expand_activation[0][0] ']
block5b_bn (BatchNormalizatio n)	(None, 17, 12, 720)	2880	['block5b_dwconv[0][0]']
block5b_activation (Activation)	(None, 17, 12, 720)	0	['block5b_bn[0][0]']
block5b_se_squeeze (GlobalAver agePooling2D)	(None, 720)	0	['block5b_activation[0][0]']
block5b_se_reshape (Reshape)	(None, 1, 1, 720)	0	['block5b_se_squeeze[0][0]']
block5b_se_reduce (Conv2D)	(None, 1, 1, 30)	21630	['block5b_se_reshape[0][0]']
block5b_se_expand (Conv2D)	(None, 1, 1, 720)	22320	['block5b_se_reduce[0][0]']
block5b_se_excite (Multiply)	(None, 17, 12, 720)	0	['block5b_activation[0][0]', 'block5b_se_expand[0][0]']
block5b_project_conv (Conv2D)	(None, 17, 12, 120)	86400	['block5b_se_excite[0][0]']

block5b_project_bn (BatchNormaliza lization)	(None, 17, 12, 120) 480	['block5b_project_conv[0][0]']
block5b_drop (Dropout)	(None, 17, 12, 120) 0	['block5b_project_bn[0][0]']
block5b_add (Add)	(None, 17, 12, 120) 0	['block5b_drop[0][0]', 'block5a_project_bn[0][0]']
block5c_expand_conv (Conv2D)	(None, 17, 12, 720) 86400	['block5b_add[0][0]']
block5c_expand_bn (BatchNormaliz ation)	(None, 17, 12, 720) 2880	['block5c_expand_conv[0][0]']
block5c_expand_activation (Act ivation)	(None, 17, 12, 720) 0	['block5c_expand_bn[0][0]']
block5c_dwconv (DepthwiseConv2 D)	(None, 17, 12, 720) 18000	['block5c_expand_activation[0][0]']
block5c_bn (BatchNormalizatio n)	(None, 17, 12, 720) 2880	['block5c_dwconv[0][0]']
block5c_activation (Activation)	(None, 17, 12, 720) 0	['block5c_bn[0][0]']
block5c_se_squeeze (GlobalAver agePooling2D)	(None, 720) 0	['block5c_activation[0][0]']
block5c_se_reshape (Reshape)	(None, 1, 1, 720) 0	['block5c_se_squeeze[0][0]']
block5c_se_reduce (Conv2D)	(None, 1, 1, 30) 21630	['block5c_se_reshape[0][0]']
block5c_se_expand (Conv2D)	(None, 1, 1, 720) 22320	['block5c_se_reduce[0][0]']
block5c_se_excite (Multiply)	(None, 17, 12, 720) 0	['block5c_activation[0][0]', 'block5c_se_expand[0][0]']
block5c_project_conv (Conv2D)	(None, 17, 12, 120) 86400	['block5c_se_excite[0][0]']
block5c_project_bn (BatchNorma lization)	(None, 17, 12, 120) 480	['block5c_project_conv[0][0]']
block5c_drop (Dropout)	(None, 17, 12, 120) 0	['block5c_project_bn[0][0]']

block5c_add (Add)	(None, 17, 12, 120) 0	['block5c_drop[0][0]', 'block5b_add[0][0]']
block5d_expand_conv (Conv2D)	(None, 17, 12, 720) 86400	['block5c_add[0][0]']
block5d_expand_bn (BatchNormal ization)	(None, 17, 12, 720) 2880	['block5d_expand_conv[0][0]']
block5d_expand_activation (Act ivation)	(None, 17, 12, 720) 0	['block5d_expand_bn[0][0]']
block5d_dwconv (DepthwiseConv2 D)	(None, 17, 12, 720) 18000	['block5d_expand_activation[0][0] ']
block5d_bn (BatchNormalization)	(None, 17, 12, 720) 2880	['block5d_dwconv[0][0]']
block5d_activation (Activation)	(None, 17, 12, 720) 0	['block5d_bn[0][0]']
block5d_se_squeeze (GlobalAver agePooling2D)	(None, 720) 0	['block5d_activation[0][0]']
block5d_se_reshape (Reshape)	(None, 1, 1, 720) 0	['block5d_se_squeeze[0][0]']
block5d_se_reduce (Conv2D)	(None, 1, 1, 30) 21630	['block5d_se_reshape[0][0]']
block5d_se_expand (Conv2D)	(None, 1, 1, 720) 22320	['block5d_se_reduce[0][0]']
block5d_se_excite (Multiply)	(None, 17, 12, 720) 0	['block5d_activation[0][0]', 'block5d_se_expand[0][0]']
block5d_project_conv (Conv2D)	(None, 17, 12, 120) 86400	['block5d_se_excite[0][0]']
block5d_project_bn (BatchNorma lization)	(None, 17, 12, 120) 480	['block5d_project_conv[0][0]']
block5d_drop (Dropout)	(None, 17, 12, 120) 0	['block5d_project_bn[0][0]']
block5d_add (Add)	(None, 17, 12, 120) 0	['block5d_drop[0][0]', 'block5c_add[0][0]']
block6a_expand_conv (Conv2D)	(None, 17, 12, 720) 86400	['block5d_add[0][0]']

block6a_expand_bn (BatchNormal ization)	(None, 17, 12, 720) 2880		['block6a_expand_conv[0][0]']
block6a_expand_activation (Act ivation)	(None, 17, 12, 720) 0		['block6a_expand_bn[0][0]']
block6a_dwconv_pad (ZeroPaddin g2D)	(None, 21, 15, 720) 0		['block6a_expand_activation[0][0] ']
block6a_dwconv (DepthwiseConv2 D)	(None, 9, 6, 720) 18000		['block6a_dwconv_pad[0][0]']
block6a_bn (BatchNormalizatio n)	(None, 9, 6, 720) 2880		['block6a_dwconv[0][0]']
block6a_activation (Activation)	(None, 9, 6, 720) 0		['block6a_bn[0][0]']
block6a_se_squeeze (GlobalAver agePooling2D)	(None, 720) 0		['block6a_activation[0][0]']
block6a_se_reshape (Reshape)	(None, 1, 1, 720) 0		['block6a_se_squeeze[0][0]']
block6a_se_reduce (Conv2D)	(None, 1, 1, 30) 21630		['block6a_se_reshape[0][0]']
block6a_se_expand (Conv2D)	(None, 1, 1, 720) 22320		['block6a_se_reduce[0][0]']
block6a_se_excite (Multiply)	(None, 9, 6, 720) 0		['block6a_activation[0][0]', 'block6a_se_expand[0][0]']
block6a_project_conv (Conv2D)	(None, 9, 6, 208) 149760		['block6a_se_excite[0][0]']
block6a_project_bn (BatchNorma lization)	(None, 9, 6, 208) 832		['block6a_project_conv[0][0]']
block6b_expand_conv (Conv2D)	(None, 9, 6, 1248) 259584		['block6a_project_bn[0][0]']
block6b_expand_bn (BatchNormal ization)	(None, 9, 6, 1248) 4992		['block6b_expand_conv[0][0]']
block6b_expand_activation (Act ivation)	(None, 9, 6, 1248) 0		['block6b_expand_bn[0][0]']
block6b_dwconv (DepthwiseConv2 D)	(None, 9, 6, 1248) 31200		['block6b_expand_activation[0][0]

D)

block6b_bn (BatchNormalization)	(None, 9, 6, 1248) 4992	['block6b_dwconv[0][0]'] ']
block6b_activation (Activation)	(None, 9, 6, 1248) 0	['block6b_bn[0][0]']
block6b_se_squeeze (GlobalAver agePooling2D)	(None, 1248) 0	['block6b_activation[0][0]']
block6b_se_reshape (Reshape)	(None, 1, 1, 1248) 0	['block6b_se_squeeze[0][0]']
block6b_se_reduce (Conv2D)	(None, 1, 1, 52) 64948	['block6b_se_reshape[0][0]']
block6b_se_expand (Conv2D)	(None, 1, 1, 1248) 66144	['block6b_se_reduce[0][0]']
block6b_se_excite (Multiply)	(None, 9, 6, 1248) 0	['block6b_activation[0][0]', 'block6b_se_expand[0][0]']
block6b_project_conv (Conv2D)	(None, 9, 6, 208) 259584	['block6b_se_excite[0][0]']
block6b_project_bn (BatchNorma lization)	(None, 9, 6, 208) 832	['block6b_project_conv[0][0]']
block6b_drop (Dropout)	(None, 9, 6, 208) 0	['block6b_project_bn[0][0]']
block6b_add (Add)	(None, 9, 6, 208) 0	['block6b_drop[0][0]', 'block6a_project_bn[0][0]']
block6c_expand_conv (Conv2D)	(None, 9, 6, 1248) 259584	['block6b_add[0][0]']
block6c_expand_bn (BatchNormal ization)	(None, 9, 6, 1248) 4992	['block6c_expand_conv[0][0]']
block6c_expand_activation (Act ivation)	(None, 9, 6, 1248) 0	['block6c_expand_bn[0][0]']
block6c_dwconv (DepthwiseConv2 D)	(None, 9, 6, 1248) 31200	['block6c_expand_activation[0][0] ']
block6c_bn (BatchNormalization)	(None, 9, 6, 1248) 4992	['block6c_dwconv[0][0]']

block6c_activation (Activation (None, 9, 6, 1248) 0)			['block6c_bn[0][0]']
block6c_se_squeeze (GlobalAveragePooling2D)	(None, 1248)	0	['block6c_activation[0][0]']
block6c_se_reshape (Reshape)	(None, 1, 1, 1248)	0	['block6c_se_squeeze[0][0]']
block6c_se_reduce (Conv2D)	(None, 1, 1, 52)	64948	['block6c_se_reshape[0][0]']
block6c_se_expand (Conv2D)	(None, 1, 1, 1248)	66144	['block6c_se_reduce[0][0]']
block6c_se_excite (Multiply)	(None, 9, 6, 1248)	0	['block6c_activation[0][0]', 'block6c_se_expand[0][0]']
block6c_project_conv (Conv2D)	(None, 9, 6, 208)	259584	['block6c_se_excite[0][0]']
block6c_project_bn (BatchNormalization)	(None, 9, 6, 208)	832	['block6c_project_conv[0][0]']
block6c_drop (Dropout)	(None, 9, 6, 208)	0	['block6c_project_bn[0][0]']
block6c_add (Add)	(None, 9, 6, 208)	0	['block6c_drop[0][0]', 'block6b_add[0][0]']
block6d_expand_conv (Conv2D)	(None, 9, 6, 1248)	259584	['block6c_add[0][0]']
block6d_expand_bn (BatchNormalization)	(None, 9, 6, 1248)	4992	['block6d_expand_conv[0][0]']
block6d_expand_activation (Activation)	(None, 9, 6, 1248) 0		['block6d_expand_bn[0][0]']
block6d_dwconv (DepthwiseConv2D)	(None, 9, 6, 1248)	31200	['block6d_expand_activation[0][0]']
block6d_bn (BatchNormalization)	(None, 9, 6, 1248)	4992	['block6d_dwconv[0][0]']
block6d_activation (Activation)	(None, 9, 6, 1248) 0		['block6d_bn[0][0]']
block6d_se_squeeze (GlobalAveragePooling2D)	(None, 1248)	0	['block6d_activation[0][0]']

block6d_se_reshape (Reshape)	(None, 1, 1, 1248)	0	['block6d_se_squeeze[0][0]']
block6d_se_reduce (Conv2D)	(None, 1, 1, 52)	64948	['block6d_se_reshape[0][0]']
block6d_se_expand (Conv2D)	(None, 1, 1, 1248)	66144	['block6d_se_reduce[0][0]']
block6d_se_excite (Multiply)	(None, 9, 6, 1248)	0	['block6d_activation[0][0]', 'block6d_se_expand[0][0]']
block6d_project_conv (Conv2D)	(None, 9, 6, 208)	259584	['block6d_se_excite[0][0]']
block6d_project_bn (BatchNormala lization)	(None, 9, 6, 208)	832	['block6d_project_conv[0][0]']
block6d_drop (Dropout)	(None, 9, 6, 208)	0	['block6d_project_bn[0][0]']
block6d_add (Add)	(None, 9, 6, 208)	0	['block6d_drop[0][0]', 'block6c_add[0][0]']
block6e_expand_conv (Conv2D)	(None, 9, 6, 1248)	259584	['block6d_add[0][0]']
block6e_expand_bn (BatchNormala lization)	(None, 9, 6, 1248)	4992	['block6e_expand_conv[0][0]']
block6e_expand_activation (Act ivation)	(None, 9, 6, 1248)	0	['block6e_expand_bn[0][0]']
block6e_dwconv (DepthwiseConv2 D)	(None, 9, 6, 1248)	31200	['block6e_expand_activation[0][0]']
block6e_bn (BatchNormalizatio n)	(None, 9, 6, 1248)	4992	['block6e_dwconv[0][0]']
block6e_activation (Activation)	(None, 9, 6, 1248)	0	['block6e_bn[0][0]']
block6e_se_squeeze (GlobalAver agePooling2D)	(None, 1248)	0	['block6e_activation[0][0]']
block6e_se_reshape (Reshape)	(None, 1, 1, 1248)	0	['block6e_se_squeeze[0][0]']
block6e_se_reduce (Conv2D)	(None, 1, 1, 52)	64948	['block6e_se_reshape[0][0]']

block6e_se_expand (Conv2D)	(None, 1, 1, 1248)	66144	['block6e_se_reduce[0][0]']
block6e_se_excite (Multiply)	(None, 9, 6, 1248)	0	['block6e_activation[0][0]', 'block6e_se_expand[0][0]']
block6e_project_conv (Conv2D)	(None, 9, 6, 208)	259584	['block6e_se_excite[0][0]']
block6e_project_bn (BatchNormala lization)	(None, 9, 6, 208)	832	['block6e_project_conv[0][0]']
block6e_drop (Dropout)	(None, 9, 6, 208)	0	['block6e_project_bn[0][0]']
block6e_add (Add)	(None, 9, 6, 208)	0	['block6e_drop[0][0]', 'block6d_add[0][0]']
block7a_expand_conv (Conv2D)	(None, 9, 6, 1248)	259584	['block6e_add[0][0]']
block7a_expand_bn (BatchNormala lization)	(None, 9, 6, 1248)	4992	['block7a_expand_conv[0][0]']
block7a_expand_activation (Act ivation)	(None, 9, 6, 1248)	0	['block7a_expand_bn[0][0]']
block7a_dwconv (DepthwiseConv2 D)	(None, 9, 6, 1248)	11232	['block7a_expand_activation[0][0]']
block7a_bn (BatchNormalizatio n)	(None, 9, 6, 1248)	4992	['block7a_dwconv[0][0]']
block7a_activation (Activation)	(None, 9, 6, 1248)	0	['block7a_bn[0][0]']
block7a_se_squeeze (GlobalAver agePooling2D)	(None, 1248)	0	['block7a_activation[0][0]']
block7a_se_reshape (Reshape)	(None, 1, 1, 1248)	0	['block7a_se_squeeze[0][0]']
block7a_se_reduce (Conv2D)	(None, 1, 1, 52)	64948	['block7a_se_reshape[0][0]']
block7a_se_expand (Conv2D)	(None, 1, 1, 1248)	66144	['block7a_se_reduce[0][0]']
block7a_se_excite (Multiply)	(None, 9, 6, 1248)	0	['block7a_activation[0][0]', 'block7a_se_expand[0][0]']

			Assignment_2
block7a_project_conv (Conv2D)	(None, 9, 6, 352)	439296	['block7a_se_excite[0][0]']
block7a_project_bn (BatchNormala lization)	(None, 9, 6, 352)	1408	['block7a_project_conv[0][0]']
block7b_expand_conv (Conv2D)	(None, 9, 6, 2112)	743424	['block7a_project_bn[0][0]']
block7b_expand_bn (BatchNormala lization)	(None, 9, 6, 2112)	8448	['block7b_expand_conv[0][0]']
block7b_expand_activation (Act ivation)	(None, 9, 6, 2112)	0	['block7b_expand_bn[0][0]']
block7b_dwconv (DepthwiseConv2 D)	(None, 9, 6, 2112)	19008	['block7b_expand_activation[0][0] ']
block7b_bn (BatchNormalization)	(None, 9, 6, 2112)	8448	['block7b_dwconv[0][0]']
block7b_activation (Activation)	(None, 9, 6, 2112)	0	['block7b_bn[0][0]']
block7b_se_squeeze (GlobalAver agePooling2D)	(None, 2112)	0	['block7b_activation[0][0]']
block7b_se_reshape (Reshape)	(None, 1, 1, 2112)	0	['block7b_se_squeeze[0][0]']
block7b_se_reduce (Conv2D)	(None, 1, 1, 88)	185944	['block7b_se_reshape[0][0]']
block7b_se_expand (Conv2D)	(None, 1, 1, 2112)	187968	['block7b_se_reduce[0][0]']
block7b_se_excite (Multiply)	(None, 9, 6, 2112)	0	['block7b_activation[0][0]', 'block7b_se_expand[0][0]']
block7b_project_conv (Conv2D)	(None, 9, 6, 352)	743424	['block7b_se_excite[0][0]']
block7b_project_bn (BatchNormala lization)	(None, 9, 6, 352)	1408	['block7b_project_conv[0][0]']
block7b_drop (Dropout)	(None, 9, 6, 352)	0	['block7b_project_bn[0][0]']
block7b_add (Add)	(None, 9, 6, 352)	0	['block7b_drop[0][0]', 'block7a_project_bn[0][0]']

top_conv (Conv2D)	(None, 9, 6, 1408)	495616	['block7b_add[0][0]']
top_bn (BatchNormalization)	(None, 9, 6, 1408)	5632	['top_conv[0][0]']
top_activation (Activation)	(None, 9, 6, 1408)	0	['top_bn[0][0]']
flatten_5 (Flatten)	(None, 76032)	0	['top_activation[0][0]']

Total params: 7,768,569

Trainable params: 7,700,994

Non-trainable params: 67,575

39/39 [=====] - 33s 804ms/step

Part 2. Define a nearest-neighbour function [3 marks]

Write a function `def nearest(img, k)` which accepts an image `img`, and returns the `k` movies in the dataset whose posters are most similar to `img` (as measured in the embedding), ranked by similarity.

```
In [ ]: from sklearn.metrics.pairwise import cosine_similarity
from keras.utils import img_to_array

def k_nearest(img, k):
    ### YOUR CODE HERE

    cosineSimilarities = [] #Storage for similarities

    selectedImage = trainedModel.predict(img) #Get embeddings for selected image

    #Compare each image in the dataset to the selected image (embeddings)
    for i in X:
        cosineSim = cosine_similarity(selectedImage, i.reshape(1, -1))
        cosineSimilarities.append(cosineSim)

    cosineWithID = list(zip(IDs, cosineSimilarities)) #add IDs for pulling posters later
    sortedCosine = sorted(cosineWithID, key= lambda i:i[1]) #Sorting
    sortedCosine.pop(-1) #Remove the 1:1 match, e.g the searched image.

    kNearest = sortedCosine[-k:] #Get last k values (e.g most similar)
    kNearestID = [id[0] for id in kNearest] #Extract just the IDs for return
```

```
    return kNearestID
```

Part 3: Demonstrate your nearest-neighbour function [4 marks]

Choose any movie poster. Call this the query poster. Show it, and use your nearest-neighbour function to show the 3 nearest neighbours (excluding the query itself). This means **call** the function you defined above.

Write a comment: in what ways are they similar or dissimilar? Do you agree with the choice and the ranking? Why do you think they are close in the embedding? Do you notice, for example, that the nearest neighbours are from a similar era?

```
In [ ]: ### YOUR CODE HERE
Q_idx = 1000771 # YOUR VALUE HERE - DO NOT USE MY VALUE

#Year and genre of selected poster
Year = df4.loc[df4.index == Q_idx, 'Year'].item()
Genre = df4.loc[df4.index == Q_idx, 'Genre'].item()

k = 3 #Count of nearest neighbours to find.

#Preprocess image for k_nearest
image = imread(Q_idx) #Read selected poster
image = img_to_array(image) #set to array for use in k_nearest (embedding)
image = np.expand_dims(image, axis=0) #Correct size

nearestPosters = k_nearest(image, k) #where 3 = k neighbors.

print("ID: " + str(Q_idx) + "\nYear: " + str(Year) + "\nGenre: " + str(Genre))

#Display selected poster
imshow(imread(Q_idx))

#For displaying k images
posterCount = len(nearestPosters)

#SubPlot explanation from https://matplotlib.org/stable/gallery/subplots_axes_and_figures/subplots_demo.html
fig, axes = plt.subplots(k) #Create figure + axis based on number of neighbours, in this case 3.

fig.set_figheight(15) #Make posters larger
fig.set_figwidth(15)
```

```
print("K Nearest posters:")

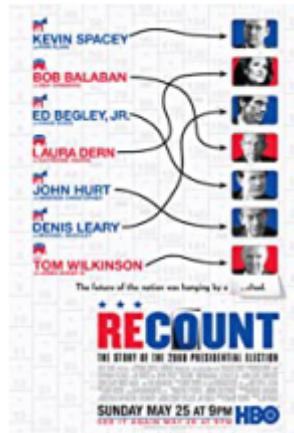
for i, poster in enumerate(nearestPosters):
    axes[i].imshow(imread(poster)) #Read ID + display poster

    #Get posters information for k nearest investigation
    kYear = df4.loc[df4.index == poster, 'Year'].item()
    kGenre = df4.loc[df4.index == poster, 'Genre'].item()
    #Display poster information (Print won't work here)
    axes[i].set_title("ID: " + str(poster) + "\nYear: " + str(kYear) + "\nGenre: " + str(kGenre), loc="left")

    axes[i].axis('off') #Remove irrelevant axis from subplots
```

1/1 [=====] - 0s 112ms/step

ID: 1000771
Year: 2008.0
Genre: Drama|History
K Nearest posters:



ID: 1937449

Year: 2012.0

Genre: Drama|Thriller



ID: 1977002

Year: 2011.0

Genre: Drama|Romance



ID: 63483

Year: 1968.0

Genre: Drama|Romance



Part 3 comments

Note: posters are in descending order, where first of the three posters is 3rd in similarity, and last poster is most similar.

I think the cosine similarity worked well here. With regards to visuals, the images hold similar colour ways, such as being majority white, with hints of Blues, reds and blacks. Although the last image strays a bit from this, where purple is one of the dominant colours. The original poster does have blue+red (which mixes to purple), but that is a bit of a stretch.

Years are relatively close, except for the 3rd poster, but the genres are all at least dramas. As for romance/thriller/history, these all differ. The fact they are all dramas is a good sign.

Overall I agree with the rankings here, given similar visuals, genre and relatively similar years for 2/3.

Part 4: Year regression [5 marks]

Let's investigate the last question ("similar era") above by running **regression** on the year, ie attempt to predict the year, given the poster. Use a train-test split. Build a suitable Keras neural network model for this, **as a regression head on top of the embedding from Part 1**. Include comments to explain the purpose of each part of the model. It should be possible to make a prediction, given a new poster (not part of the original dataset). Write a short comment on model performance: is it possible to predict the year? Based on this result, are there trends over time?

In []:

```
### YOUR CODE HERE
from sklearn.model_selection import train_test_split
from keras import layers, models
from keras.layers import Dense, Input
```

```
years = df4[["Year"]] #Get just years

#Creating test/train split and normalising pixel values.
X_train, X_test, y_train, y_test = train_test_split(img_array, years, test_size=0.3)
X_train = X_train / 255.0
X_test = X_test / 255.0

#Create an input tensor//input shape for regression that matches our img_array.
inputShape = Input(shape=(268,182,3))
part1Model = trainedModel(inputShape) #Model from part 1
print(part1Model.shape)

#Using the output of our initial model as the input to regression
#Using 4 dense layers, with the last being one neuron to produce one value (e.g the year).
#relu activation function used for vanishing gradients.
regressor = layers.Dense(512, activation='relu')(part1Model)
regressor = layers.Dense(256, activation='relu')(regressor)
regressor = layers.Dense(128, activation='relu')(regressor)
regressor = layers.Dense(1)(regressor)

#Create our full model with the same input tensor and outputs from the regression head.
regressionModel = Model(inputs=inputShape, outputs=regressor)

#using Adam optimizer as it is a common DNN optimizer and highly recognised.
#MSLE to avoid over/under estimates and minimizing error.
#MAE is a common standard for regression and can handle large variations.
regressionModel.compile(optimizer='adam', loss='mean_squared_logarithmic_error', metrics=['mae'])

regressionModel.summary()
```

(None, 76032)
Model: "model_11"

Layer (type)	Output Shape	Param #
<hr/>		
input_12 (InputLayer)	[(None, 268, 182, 3)]	0
model_10 (Functional)	(None, 76032)	7768569
dense_20 (Dense)	(None, 512)	38928896
dense_21 (Dense)	(None, 256)	131328
dense_22 (Dense)	(None, 128)	32896
dense_23 (Dense)	(None, 1)	129
<hr/>		
Total params: 46,861,818		
Trainable params: 46,794,243		
Non-trainable params: 67,575		

In []: `history = regressionModel.fit(X_train,y_train,epochs=12,batch_size=100,validation_split=0.03) #runtime: 20mins`
`regressionModel.save("RegressionModel") #Saving to prevent training over and over.`

```

Epoch 1/12
9/9 [=====] - 102s 11s/step - loss: 13.3163 - mae: 1410.6318 - val_loss: 10.1739 - val_mae: 1909.7960
Epoch 2/12
9/9 [=====] - 94s 10s/step - loss: 0.3343 - mae: 1466.0692 - val_loss: 3.5740 - val_mae: 1690.7823
Epoch 3/12
9/9 [=====] - 94s 10s/step - loss: 0.1292 - mae: 613.3856 - val_loss: 6.1770 - val_mae: 1826.0924
Epoch 4/12
9/9 [=====] - 94s 10s/step - loss: 0.0350 - mae: 303.4424 - val_loss: 4.7557 - val_mae: 1767.0769
Epoch 5/12
9/9 [=====] - 94s 10s/step - loss: 0.0092 - mae: 145.2354 - val_loss: 3.3429 - val_mae: 1672.0077
Epoch 6/12
9/9 [=====] - 94s 10s/step - loss: 0.0061 - mae: 118.2343 - val_loss: 2.9599 - val_mae: 1635.3523
Epoch 7/12
9/9 [=====] - 96s 11s/step - loss: 0.0036 - mae: 90.6178 - val_loss: 1.0927 - val_mae: 1249.7690
Epoch 8/12
9/9 [=====] - 96s 11s/step - loss: 0.0028 - mae: 80.8405 - val_loss: 0.0442 - val_mae: 201.5287
Epoch 9/12
9/9 [=====] - 95s 11s/step - loss: 0.0024 - mae: 75.1877 - val_loss: 0.0217 - val_mae: 297.0335
Epoch 10/12
9/9 [=====] - 95s 10s/step - loss: 0.0019 - mae: 66.4140 - val_loss: 0.0421 - val_mae: 449.9901
Epoch 11/12
9/9 [=====] - 95s 11s/step - loss: 0.0017 - mae: 61.4079 - val_loss: 0.0365 - val_mae: 417.3804
Epoch 12/12
9/9 [=====] - 94s 10s/step - loss: 0.0016 - mae: 59.6979 - val_loss: 0.0292 - val_mae: 369.1561
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op while saving (showing 5 of 115). These functions will not be directly callable after loading.
INFO:tensorflow:Assets written to: RegressionModel\assets
INFO:tensorflow:Assets written to: RegressionModel\assets

```

```
In [ ]: yearPredictModel = keras.models.load_model("RegressionModel")

eval = yearPredictModel.evaluate(X_test,y_test)
print("Loss, MAE", eval)
```

```
12/12 [=====] - 14s 1s/step - loss: 0.0056 - mae: 152.0591
Loss, MAE [0.005579752381891012, 152.05911254882812]
```

```
In [ ]: print(history.history.keys())

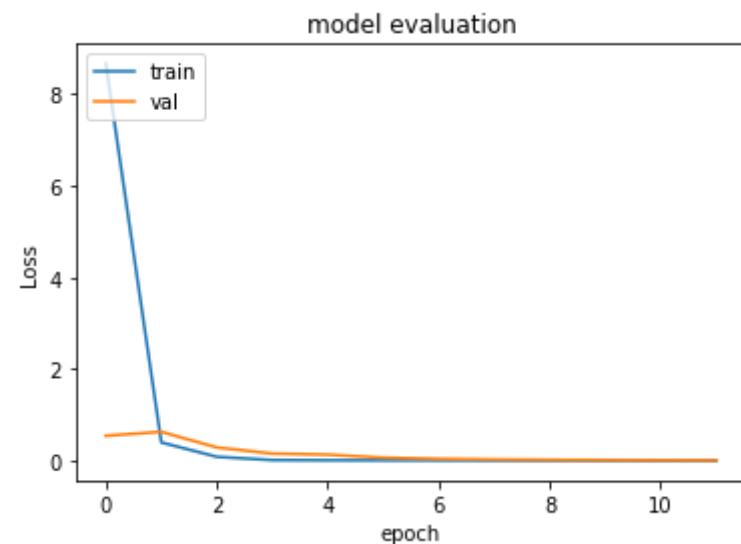
#Plot source: https://stackoverflow.com/questions/41908379/keras-plot-training-validation-and-test-set-accuracy

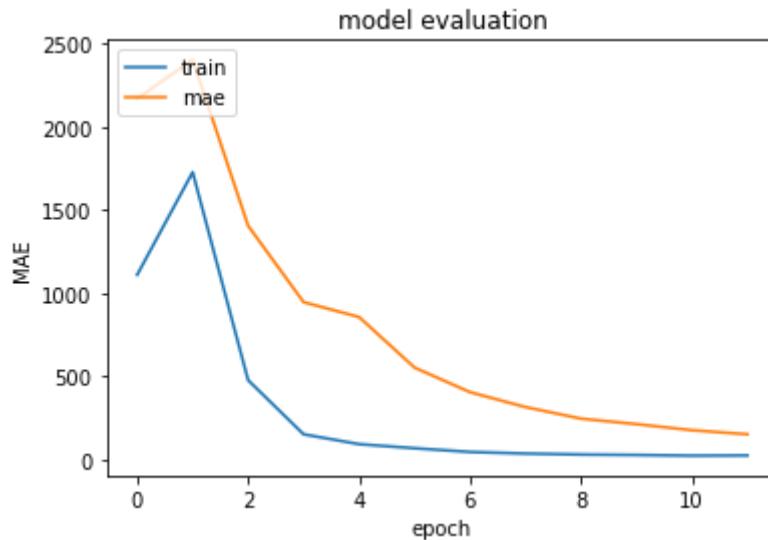
#Loss vs val_Loss
```

```
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model evaluation')
plt.ylabel('Loss')
plt.xlabel('epoch')
plt.legend(['train', 'val'], loc='upper left')
plt.show()

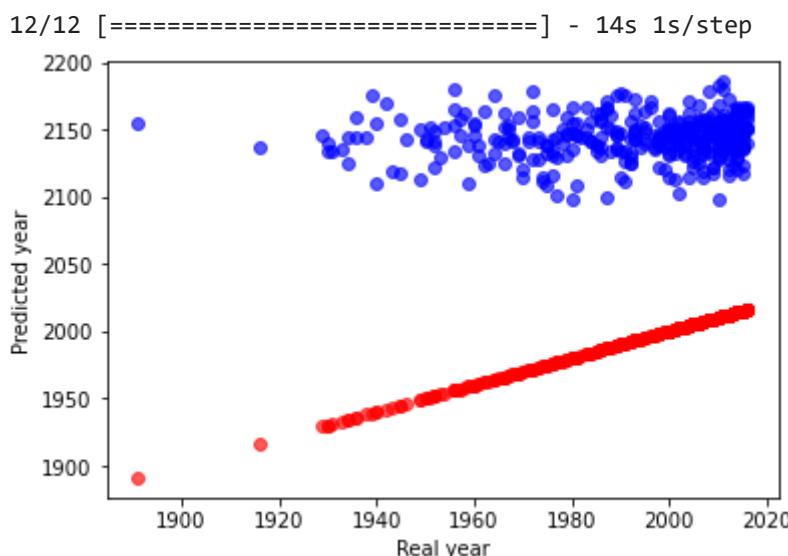
#MAE vs val_mae
plt.plot(history.history['mae'])
plt.plot(history.history['val_mae'])
plt.title('model evaluation')
plt.ylabel('MAE')
plt.xlabel('epoch')
plt.legend(['train', 'mae'], loc='upper left')
plt.show()
```

```
dict_keys(['loss', 'mae', 'val_loss', 'val_mae'])
```





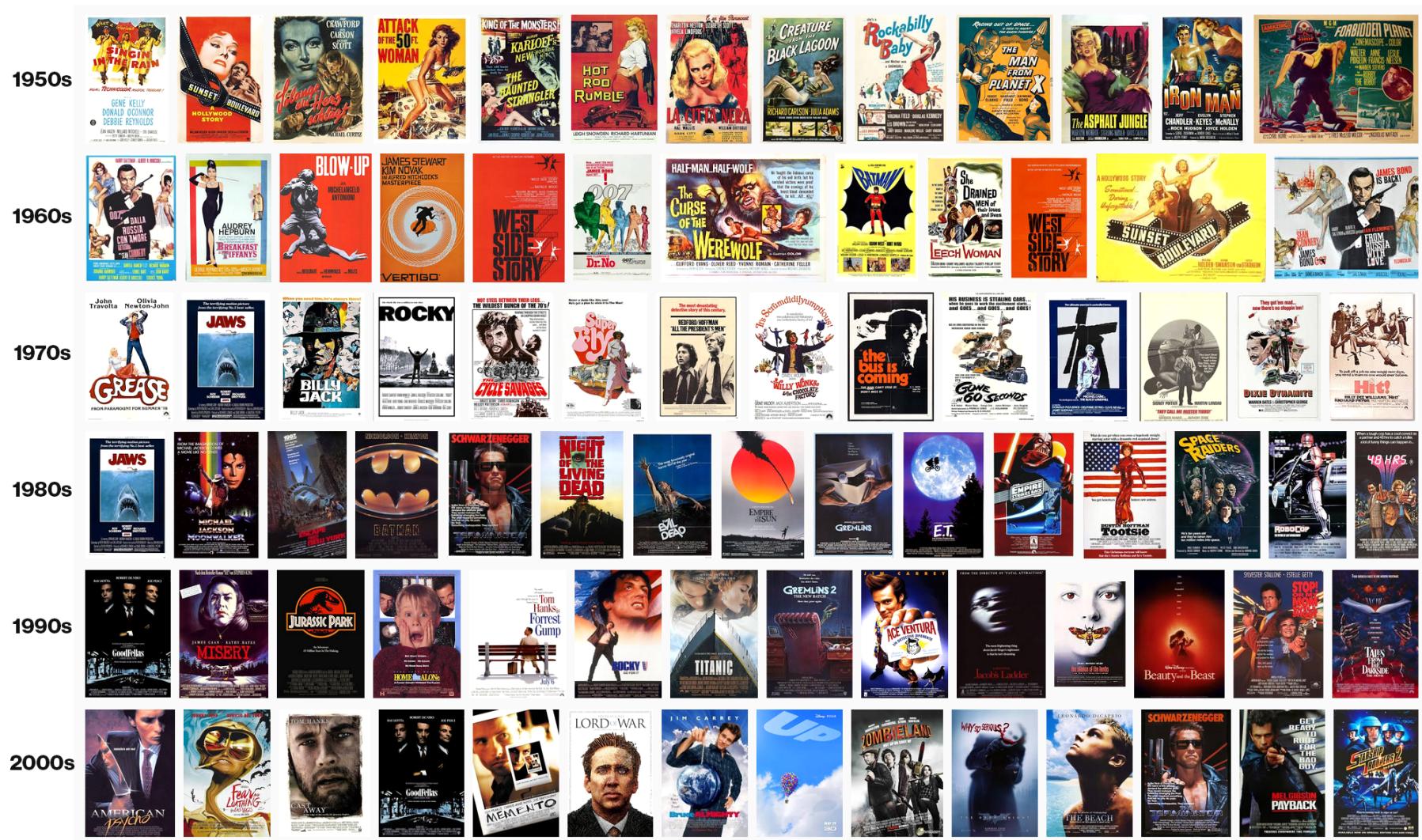
```
In [ ]: y_pred = yearPredictModel.predict(X_test) #you can pass a single poster here via imread(posterID), seen below.  
plt.scatter(y_test, y_pred, c='b', alpha=0.65, label='predicted')  
plt.scatter(y_test, y_test, c='r', alpha=0.65, label='actual')  
plt.xlabel('Real year')  
plt.ylabel('Predicted year')  
plt.show()
```



Part 4 comments

Looking at the MAE and LOSS plots, the model does look to be performing well, with very low values of loss and MAE: [0.0054201362654566765, 150.1516571044922]

In contrast to this though, our predictions are at least 100 years in the future, which isn't ideal. Looking at the scatter plot of predictions vs real years, it is more spread than the real movie years, showing the predictions vary quite a bit. I suppose this does make sense in a way, movie posters are often fairly unique and haven't really deviated much as time went on. Colour printing existed before the first movie on this list, so what the image looks like doesn't really indicate the era it came from. They are of course more stylized as seen below, but I believe it may be hard for regression to associate this stylized poster to a given year/era.



Source: <https://www.quora.com/Has-anyone-noticed-that-in-the-1980s-the-movie-posters-are-hand-drawn-why-is-this>

Part 5: Improvements [5 marks]

Propose a possible improvement. Some ideas are suggested below. The chosen improvement must be notified to the lecturer at least 1 week before submission and **must be approved by the lecturer to avoid duplication with other students**. Compare the performance between your

original and your new model (the proposed improvement might not actually improve on model performance -- that is ok). Some marks will be awarded for more interesting / challenging / novel improvements.

Ideas:

- Try a different pretrained model for creating the embedding
- Alternative ways of reducing the pretrained model's output to a flat vector for the embedding
- Gather more data (see the csv file for URLs)
- Add different architectural details to the regression head
- Fine-tuning
- Training an end-to-end convnet of your own design (no pretraining)
- Improve the embedding by training a multi-headed model, eg predicting both genre and year
- Create a good visualisation of the embedding.

```
In [ ]: ### YOUR CODE HERE
from sklearn.manifold import TSNE

years = []
genres = []

#Get years and genre for grouping colours within t-sne plots later.
for i, data in enumerate(IDs):
    tempYear = df4.loc[df4.index == data, 'Year'].item()
    years.append(tempYear)
    tempGenre = df4.loc[df4.index == data, 'Genre'].item()
    genres.append(tempGenre)

#Used for rounding years down to their respective decade, credit: https://stackoverflow.com/questions/13082698/rounding-down-int
def roundDown(number, divisor):
    return number - (number % divisor)

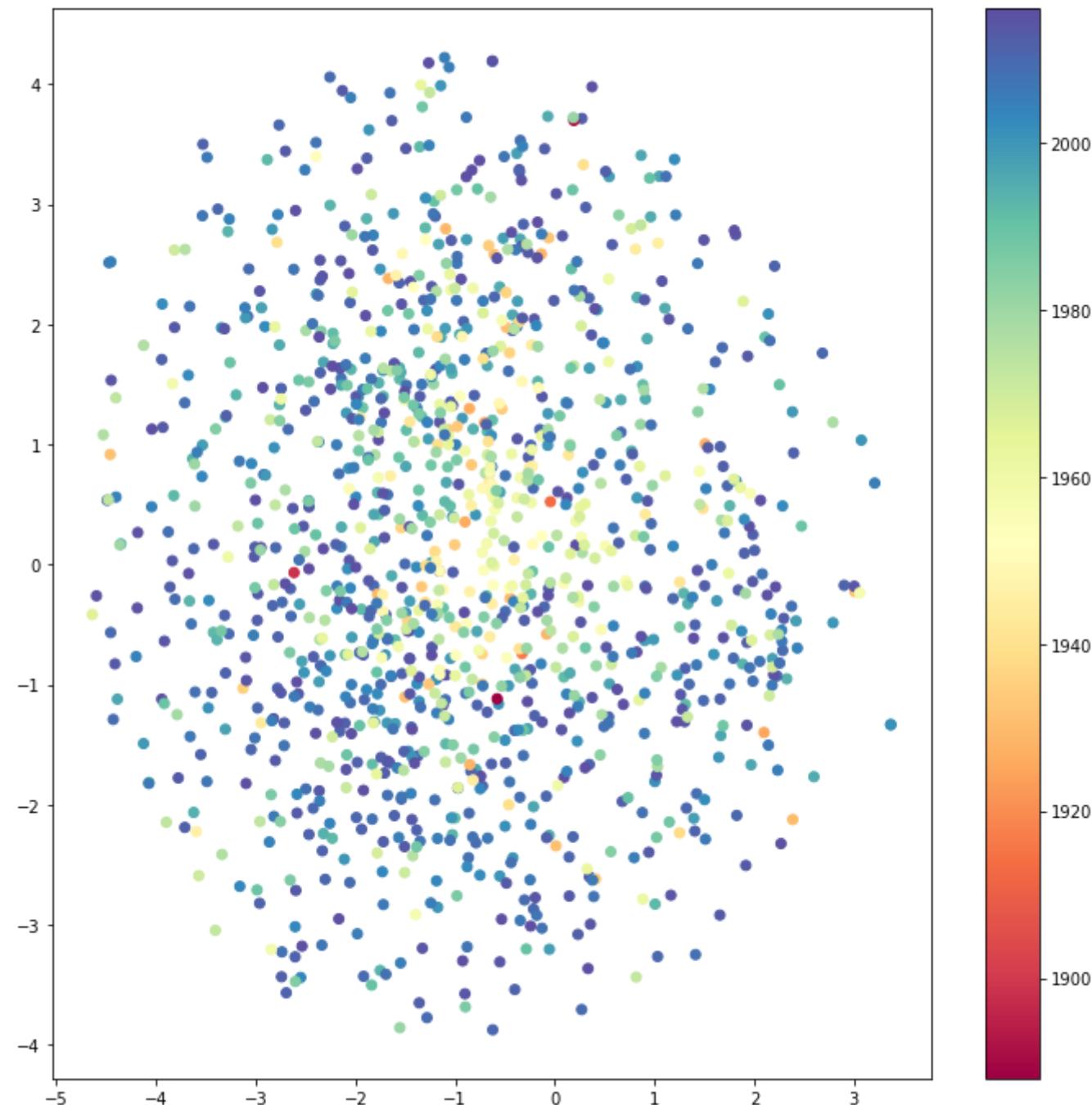
decades = [roundDown(int(value),10) for value in years]

#embeddingsWithGenre = list(zip(genres, X))
#embeddingsWithYear = list(zip(years, X))

#t-sne work
tsneYear = TSNE(n_components=2, random_state=0, perplexity=100, n_iter=5000)
tsneOutputYear = tsneYear.fit_transform(X)
```

```
c:\Users\CAOLA\anaconda3\lib\site-packages\sklearn\manifold\_t_sne.py:780: FutureWarning: The default initialization in TSNE will change from 'random' to 'pca' in 1.2.  
    warnings.warn(  
c:\Users\CAOLA\anaconda3\lib\site-packages\sklearn\manifold\_t_sne.py:790: FutureWarning: The default learning rate in TSNE will change from 200.0 to 'auto' in 1.2.  
    warnings.warn(
```

```
In [ ]: plt.figure(figsize=(12,12))  
plt.scatter(tsneOutputYear[:, 0], tsneOutputYear[:, 1], c=years, cmap='Spectral')  
plt.colorbar()  
plt.show()
```

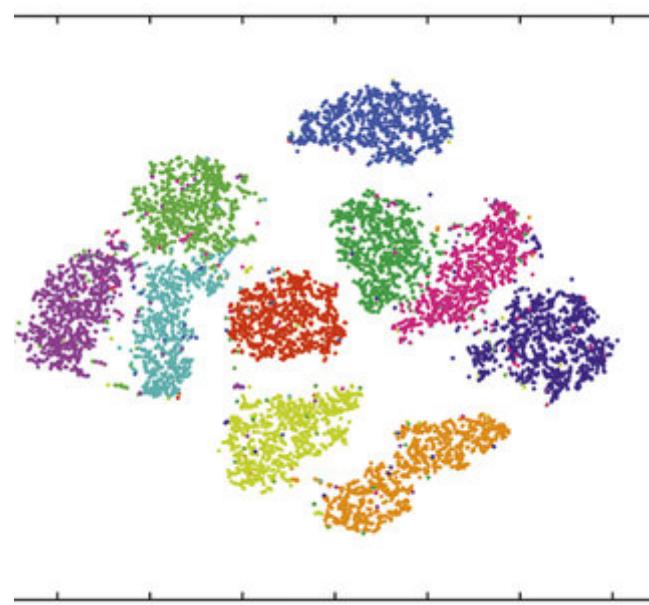


Part 5 comments

I used t-sne (T-distributed stochastic neighbor embedding) which is a dimensionality reduction algorithm, which can let us pull underlying patterns from our data (see the image at the bottom of this markdown for a visual of the Fashion MNist dataset, as seen in assignment 1.)

With this we can potentially pull out separations in our data to see how our embeddings correlate, and therefore let us understand if they are great or poor at future tasks involving said embeddings.

The plot above shows the t-sne, grouped by year. Each unique year is given a unique colour. We can see from the above plot, there is a vague pattern showing, with older movies congregating in the centre (mostly) and more recent movies (2000's and up) spreading to the further bounds of the plot. This could hint that the posters embeddings are potentially separable based on the year they came out, this could be due to the old stylized format movie posters has in the early days, compared to modern posters that make more use of photographs of actors/actresses on their posters.

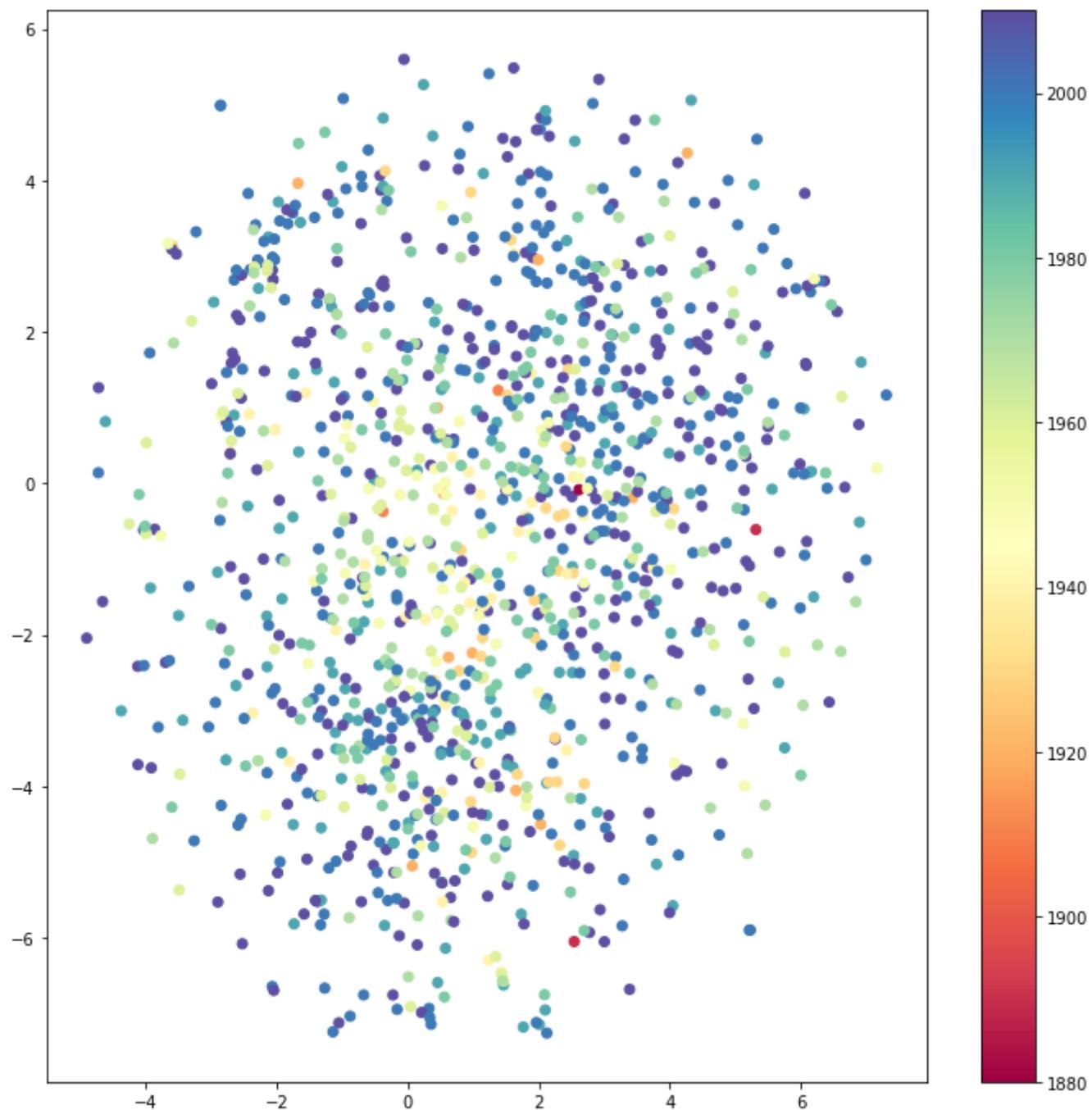


I also wanted to see if they were more separable on decade, which you can see in the below plot.

```
In [ ]: tsneDecade = TSNE(n_components=2, random_state=0, perplexity=50, n_iter=5000)
tsneOutputDecade = tsneDecade.fit_transform(X)
```

```
c:\Users\CAOLA\anaconda3\lib\site-packages\sklearn\manifold\_t_sne.py:780: FutureWarning: The default initialization in TSNE will change from 'random' to 'pca' in 1.2.
  warnings.warn(
c:\Users\CAOLA\anaconda3\lib\site-packages\sklearn\manifold\_t_sne.py:790: FutureWarning: The default learning rate in TSNE will change from 200.0 to 'auto' in 1.2.
  warnings.warn(
```

```
In [ ]: plt.figure(figsize=(12,12))
plt.scatter(tsneOutputDecade[:, 0], tsneOutputDecade[:, 1], c=decades, cmap='Spectral')
plt.colorbar()
plt.show()
```



Part 5 comments

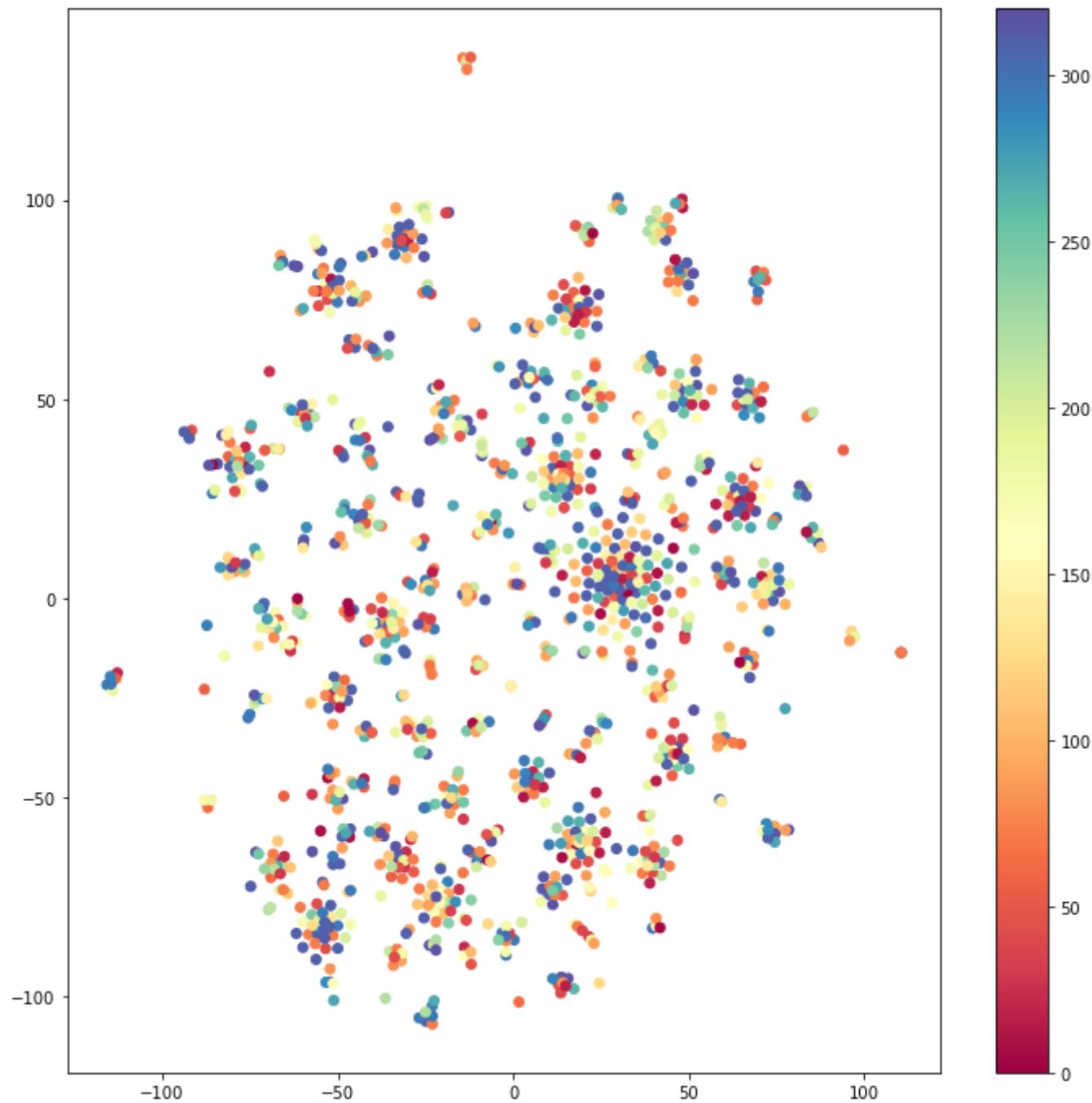
The above plot of t-sne, colour grouped by decade looks to reinforce the observations from the original t-sne based on year, where older film posters look to group together (mostly) and as you get to the outer reaches of the plot, you see more and more newer films, which also happen to be more spread out. This could be due to more modern posters including more unique and creative design, wider varieties of colour and more unrealistic visuals on the poster (such as monsters or robots in sci-fi films), that weren't really present on many posters from before the 1960's.

```
In [ ]: tsneGenre = TSNE(n_components=2, random_state=0, perplexity=2, n_iter=5000, n_iter_without_progress=1250)
tsneOutputGenre = tsneGenre.fit_transform(X)

c:\Users\CAOLA\anaconda3\lib\site-packages\sklearn\manifold\_t_sne.py:780: FutureWarning: The default initialization in TSNE will change from 'random' to 'pca' in 1.2.
    warnings.warn(
c:\Users\CAOLA\anaconda3\lib\site-packages\sklearn\manifold\_t_sne.py:790: FutureWarning: The default learning rate in TSNE will change from 200.0 to 'auto' in 1.2.
    warnings.warn(
```

```
In [ ]: #Plot doesn't accept list of strings as valid colours, so convert to numbers:
genreNumbers = {genre: i for i, genre in enumerate(set(genres))}
genreColours = [genreNumbers[genre] for genre in genres]

plt.figure(figsize=(12,12))
plt.scatter(tsneOutputGenre[:, 0], tsneOutputGenre[:, 1], c=genreColours, cmap='Spectral')
plt.colorbar()
plt.show()
```



Part 5 comments

Using multiple different hyperparameters for t-sne calculations (primarily perplexity, which is related to the number of nearest neighbors), most gave seemingly random/scattered results. Lowering perplexity to 2 gave the above plot, which shows there is no real link in embeddings between Genre and the posters. Each genre can have wildly varying posters and the embeddings would essentially be no good at differentiating them, as seen by the clusters above where multiple genres are grouped together, in contrast to the pattern in the previous plot.

```
In [ ]: #Testing another models embeddings on t-sne:  
  
#The "v2" of the original model used.  
from keras.applications import EfficientNetV2B2  
  
newModel = EfficientNetV2B2(include_top = False, weights = 'imagenet', input_shape=(268,182,3))  
newFlatLayer = layers.Flatten()(newModel.output) #GlobalAveragePooling2D  
  
newTrainedModel = Model(inputs=newModel.input,outputs=newFlatLayer)  
  
newTrainedModel.summary()  
  
newEmbeddingVectors = newTrainedModel.predict(img_array)  
  
newX = np.vstack(newEmbeddingVectors)
```

Model: "model_12"

Layer (type)	Output Shape	Param #	Connected to
input_13 (InputLayer)	[None, 268, 182, 3 0)]		[]
rescaling_6 (Rescaling)	(None, 268, 182, 3) 0		['input_13[0][0]']
normalization_6 (Normalization)	(None, 268, 182, 3) 0)		['rescaling_6[0][0]']
stem_conv (Conv2D)	(None, 134, 91, 32) 864		['normalization_6[0][0]']
stem_bn (BatchNormalization)	(None, 134, 91, 32) 128		['stem_conv[0][0]']
stem_activation (Activation)	(None, 134, 91, 32) 0		['stem_bn[0][0]']
block1a_project_conv (Conv2D)	(None, 134, 91, 16) 4608		['stem_activation[0][0]']
block1a_project_bn (BatchNorma lization)	(None, 134, 91, 16) 64		['block1a_project_conv[0][0]']
block1a_project_activation (Ac tivation)	(None, 134, 91, 16) 0		['block1a_project_bn[0][0]']
block1b_project_conv (Conv2D)	(None, 134, 91, 16) 2304		['block1a_project_activation[0][0]']
block1b_project_bn (BatchNorma lization)	(None, 134, 91, 16) 64		['block1b_project_conv[0][0]']
block1b_project_activation (Ac tivation)	(None, 134, 91, 16) 0		['block1b_project_bn[0][0]']
block1b_drop (Dropout)	(None, 134, 91, 16) 0		['block1b_project_activation[0][0]']
block1b_add (Add)	(None, 134, 91, 16) 0		['block1b_drop[0][0]', 'block1a_project_activation[0][0]']
block2a_expand_conv (Conv2D)	(None, 67, 46, 64) 9216		['block1b_add[0][0]']

block2a_expand_bn (BatchNormal (None, 67, 46, 64) 256 ization)		['block2a_expand_conv[0][0]']
block2a_expand_activation (Act (None, 67, 46, 64) 0 ivation)		['block2a_expand_bn[0][0]']
block2a_project_conv (Conv2D) (None, 67, 46, 32) 2048		['block2a_expand_activation[0][0] ']
block2a_project_bn (BatchNorma (None, 67, 46, 32) 128 lization)		['block2a_project_conv[0][0]']
block2b_expand_conv (Conv2D) (None, 67, 46, 128) 36864		['block2a_project_bn[0][0]']
block2b_expand_bn (BatchNormal (None, 67, 46, 128) 512 ization)		['block2b_expand_conv[0][0]']
block2b_expand_activation (Act (None, 67, 46, 128) 0 ivation)		['block2b_expand_bn[0][0]']
block2b_project_conv (Conv2D) (None, 67, 46, 32) 4096		['block2b_expand_activation[0][0] ']
block2b_project_bn (BatchNorma (None, 67, 46, 32) 128 lization)		['block2b_project_conv[0][0]']
block2b_drop (Dropout) (None, 67, 46, 32) 0		['block2b_project_bn[0][0]']
block2b_add (Add) (None, 67, 46, 32) 0		['block2b_drop[0][0]', 'block2a_project_bn[0][0]']
block2c_expand_conv (Conv2D) (None, 67, 46, 128) 36864		['block2b_add[0][0]']
block2c_expand_bn (BatchNormal (None, 67, 46, 128) 512 ization)		['block2c_expand_conv[0][0]']
block2c_expand_activation (Act (None, 67, 46, 128) 0 ivation)		['block2c_expand_bn[0][0]']
block2c_project_conv (Conv2D) (None, 67, 46, 32) 4096		['block2c_expand_activation[0][0] ']
block2c_project_bn (BatchNorma (None, 67, 46, 32) 128 lization)		['block2c_project_conv[0][0]']

block2c_drop (Dropout)	(None, 67, 46, 32)	0	['block2c_project_bn[0][0]']
block2c_add (Add)	(None, 67, 46, 32)	0	['block2c_drop[0][0]', 'block2b_add[0][0]']
block3a_expand_conv (Conv2D)	(None, 34, 23, 128)	36864	['block2c_add[0][0]']
block3a_expand_bn (BatchNormal ization)	(None, 34, 23, 128)	512	['block3a_expand_conv[0][0]']
block3a_expand_activation (Act ivation)	(None, 34, 23, 128)	0	['block3a_expand_bn[0][0]']
block3a_project_conv (Conv2D)	(None, 34, 23, 56)	7168	['block3a_expand_activation[0][0]']
block3a_project_bn (BatchNorma lization)	(None, 34, 23, 56)	224	['block3a_project_conv[0][0]']
block3b_expand_conv (Conv2D)	(None, 34, 23, 224)	112896	['block3a_project_bn[0][0]']
block3b_expand_bn (BatchNormal ization)	(None, 34, 23, 224)	896	['block3b_expand_conv[0][0]']
block3b_expand_activation (Act ivation)	(None, 34, 23, 224)	0	['block3b_expand_bn[0][0]']
block3b_project_conv (Conv2D)	(None, 34, 23, 56)	12544	['block3b_expand_activation[0][0]']
block3b_project_bn (BatchNorma lization)	(None, 34, 23, 56)	224	['block3b_project_conv[0][0]']
block3b_drop (Dropout)	(None, 34, 23, 56)	0	['block3b_project_bn[0][0]']
block3b_add (Add)	(None, 34, 23, 56)	0	['block3b_drop[0][0]', 'block3a_project_bn[0][0]']
block3c_expand_conv (Conv2D)	(None, 34, 23, 224)	112896	['block3b_add[0][0]']
block3c_expand_bn (BatchNormal ization)	(None, 34, 23, 224)	896	['block3c_expand_conv[0][0]']

block3c_expand_activation (Act (None, 34, 23, 224) 0 ivation)		['block3c_expand_bn[0][0]']
block3c_project_conv (Conv2D) (None, 34, 23, 56) 12544		['block3c_expand_activation[0][0] ']
block3c_project_bn (BatchNorma (None, 34, 23, 56) 224 lization)		['block3c_project_conv[0][0]']
block3c_drop (Dropout) (None, 34, 23, 56) 0		['block3c_project_bn[0][0]']
block3c_add (Add) (None, 34, 23, 56) 0		['block3c_drop[0][0]', 'block3b_add[0][0]']
block4a_expand_conv (Conv2D) (None, 34, 23, 224) 12544		['block3c_add[0][0]']
block4a_expand_bn (BatchNormal (None, 34, 23, 224) 896 ization)		['block4a_expand_conv[0][0]']
block4a_expand_activation (Act (None, 34, 23, 224) 0 ivation)		['block4a_expand_bn[0][0]']
block4a_dwconv2 (DepthwiseConv (None, 17, 12, 224) 2016 2D)		['block4a_expand_activation[0][0] ']
block4a_bn (BatchNormalization (None, 17, 12, 224) 896)		['block4a_dwconv2[0][0]']
block4a_activation (Activation (None, 17, 12, 224) 0)		['block4a_bn[0][0]']
block4a_se_squeeze (GlobalAver (None, 224) 0 agePooling2D)		['block4a_activation[0][0]']
block4a_se_reshape (Reshape) (None, 1, 1, 224) 0		['block4a_se_squeeze[0][0]']
block4a_se_reduce (Conv2D) (None, 1, 1, 14) 3150		['block4a_se_reshape[0][0]']
block4a_se_expand (Conv2D) (None, 1, 1, 224) 3360		['block4a_se_reduce[0][0]']
block4a_se_excite (Multiply) (None, 17, 12, 224) 0		['block4a_activation[0][0]', 'block4a_se_expand[0][0]']
block4a_project_conv (Conv2D) (None, 17, 12, 104) 23296		['block4a_se_excite[0][0]']

block4a_project_bn (BatchNormaliza lization)	(None, 17, 12, 104) 416	['block4a_project_conv[0][0]']
block4b_expand_conv (Conv2D)	(None, 17, 12, 416) 43264	['block4a_project_bn[0][0]']
block4b_expand_bn (BatchNormalizat ion)	(None, 17, 12, 416) 1664	['block4b_expand_conv[0][0]']
block4b_expand_activation (Acti vation)	(None, 17, 12, 416) 0	['block4b_expand_bn[0][0]']
block4b_dwconv2 (DepthwiseConv 2D)	(None, 17, 12, 416) 3744	['block4b_expand_activation[0][0] ']
block4b_bn (BatchNormalizat ion)	(None, 17, 12, 416) 1664	['block4b_dwconv2[0][0]']
block4b_activation (Activation)	(None, 17, 12, 416) 0	['block4b_bn[0][0]']
block4b_se_squeeze (GlobalAver agePooling2D)	(None, 416) 0	['block4b_activation[0][0]']
block4b_se_reshape (Reshape)	(None, 1, 1, 416) 0	['block4b_se_squeeze[0][0]']
block4b_se_reduce (Conv2D)	(None, 1, 1, 26) 10842	['block4b_se_reshape[0][0]']
block4b_se_expand (Conv2D)	(None, 1, 1, 416) 11232	['block4b_se_reduce[0][0]']
block4b_se_excite (Multiply)	(None, 17, 12, 416) 0	['block4b_activation[0][0]', 'block4b_se_expand[0][0]']
block4b_project_conv (Conv2D)	(None, 17, 12, 104) 43264	['block4b_se_excite[0][0]']
block4b_project_bn (BatchNorma lization)	(None, 17, 12, 104) 416	['block4b_project_conv[0][0]']
block4b_drop (Dropout)	(None, 17, 12, 104) 0	['block4b_project_bn[0][0]']
block4b_add (Add)	(None, 17, 12, 104) 0	['block4b_drop[0][0]', 'block4a_project_bn[0][0]']
block4c_expand_conv (Conv2D)	(None, 17, 12, 416) 43264	['block4b_add[0][0]']

block4c_expand_bn (BatchNormal ization)	(None, 17, 12, 416) 1664	['block4c_expand_conv[0][0]']
block4c_expand_activation (Act ivation)	(None, 17, 12, 416) 0	['block4c_expand_bn[0][0]']
block4c_dwconv2 (DepthwiseConv 2D)	(None, 17, 12, 416) 3744	['block4c_expand_activation[0][0] ']
block4c_bn (BatchNormalization)	(None, 17, 12, 416) 1664	['block4c_dwconv2[0][0]']
block4c_activation (Activation)	(None, 17, 12, 416) 0	['block4c_bn[0][0]']
block4c_se_squeeze (GlobalAver agePooling2D)	(None, 416) 0	['block4c_activation[0][0]']
block4c_se_reshape (Reshape)	(None, 1, 1, 416) 0	['block4c_se_squeeze[0][0]']
block4c_se_reduce (Conv2D)	(None, 1, 1, 26) 10842	['block4c_se_reshape[0][0]']
block4c_se_expand (Conv2D)	(None, 1, 1, 416) 11232	['block4c_se_reduce[0][0]']
block4c_se_excite (Multiply)	(None, 17, 12, 416) 0	['block4c_activation[0][0]', 'block4c_se_expand[0][0]']
block4c_project_conv (Conv2D)	(None, 17, 12, 104) 43264	['block4c_se_excite[0][0]']
block4c_project_bn (BatchNorma lization)	(None, 17, 12, 104) 416	['block4c_project_conv[0][0]']
block4c_drop (Dropout)	(None, 17, 12, 104) 0	['block4c_project_bn[0][0]']
block4c_add (Add)	(None, 17, 12, 104) 0	['block4c_drop[0][0]', 'block4b_add[0][0]']
block4d_expand_conv (Conv2D)	(None, 17, 12, 416) 43264	['block4c_add[0][0]']
block4d_expand_bn (BatchNormal ization)	(None, 17, 12, 416) 1664	['block4d_expand_conv[0][0]']
block4d_expand_activation (Act ivation)	(None, 17, 12, 416) 0	['block4d_expand_bn[0][0]']

ivation)		
block4d_dwconv2 (DepthwiseConv 2D)	(None, 17, 12, 416) 3744	['block4d_expand_activation[0][0]']
block4d_bn (BatchNormalization)	(None, 17, 12, 416) 1664	['block4d_dwconv2[0][0]']
block4d_activation (Activation)	(None, 17, 12, 416) 0	['block4d_bn[0][0]']
block4d_se_squeeze (GlobalAveragePooling2D)	(None, 416) 0	['block4d_activation[0][0]']
block4d_se_reshape (Reshape)	(None, 1, 1, 416) 0	['block4d_se_squeeze[0][0]']
block4d_se_reduce (Conv2D)	(None, 1, 1, 26) 10842	['block4d_se_reshape[0][0]']
block4d_se_expand (Conv2D)	(None, 1, 1, 416) 11232	['block4d_se_reduce[0][0]']
block4d_se_excite (Multiply)	(None, 17, 12, 416) 0	['block4d_activation[0][0]', 'block4d_se_expand[0][0]']
block4d_project_conv (Conv2D)	(None, 17, 12, 104) 43264	['block4d_se_excite[0][0]']
block4d_project_bn (BatchNormalization)	(None, 17, 12, 104) 416	['block4d_project_conv[0][0]']
block4d_drop (Dropout)	(None, 17, 12, 104) 0	['block4d_project_bn[0][0]']
block4d_add (Add)	(None, 17, 12, 104) 0	['block4d_drop[0][0]', 'block4c_add[0][0]']
block5a_expand_conv (Conv2D)	(None, 17, 12, 624) 64896	['block4d_add[0][0]']
block5a_expand_bn (BatchNormalization)	(None, 17, 12, 624) 2496	['block5a_expand_conv[0][0]']
block5a_expand_activation (Activation)	(None, 17, 12, 624) 0	['block5a_expand_bn[0][0]']
block5a_dwconv2 (DepthwiseConv 2D)	(None, 17, 12, 624) 5616	['block5a_expand_activation[0][0]']

block5a_bn (BatchNormalization (None, 17, 12, 624) 2496)		['block5a_dwconv2[0][0]']
block5a_activation (Activation (None, 17, 12, 624) 0)		['block5a_bn[0][0]']
block5a_se_squeeze (GlobalAveragePooling2D) (None, 624) 0		['block5a_activation[0][0]']
block5a_se_reshape (Reshape) (None, 1, 1, 624) 0		['block5a_se_squeeze[0][0]']
block5a_se_reduce (Conv2D) (None, 1, 1, 26) 16250		['block5a_se_reshape[0][0]']
block5a_se_expand (Conv2D) (None, 1, 1, 624) 16848		['block5a_se_reduce[0][0]']
block5a_se_excite (Multiply) (None, 17, 12, 624) 0		['block5a_activation[0][0]', 'block5a_se_expand[0][0]']
block5a_project_conv (Conv2D) (None, 17, 12, 120) 74880		['block5a_se_excite[0][0]']
block5a_project_bn (BatchNormalization (None, 17, 12, 120) 480)		['block5a_project_conv[0][0]']
block5b_expand_conv (Conv2D) (None, 17, 12, 720) 86400		['block5a_project_bn[0][0]']
block5b_expand_bn (BatchNormalization (None, 17, 12, 720) 2880)		['block5b_expand_conv[0][0]']
block5b_expand_activation (Activation (None, 17, 12, 720) 0)		['block5b_expand_bn[0][0]']
block5b_dwconv2 (DepthwiseConv2D) (None, 17, 12, 720) 6480		['block5b_expand_activation[0][0]']
block5b_bn (BatchNormalization (None, 17, 12, 720) 2880)		['block5b_dwconv2[0][0]']
block5b_activation (Activation (None, 17, 12, 720) 0)		['block5b_bn[0][0]']
block5b_se_squeeze (GlobalAveragePooling2D) (None, 720) 0		['block5b_activation[0][0]']
block5b_se_reshape (Reshape) (None, 1, 1, 720) 0		['block5b_se_squeeze[0][0]']

block5b_se_reduce (Conv2D)	(None, 1, 1, 30)	21630	['block5b_se_reshape[0][0]']
block5b_se_expand (Conv2D)	(None, 1, 1, 720)	22320	['block5b_se_reduce[0][0]']
block5b_se_excite (Multiply)	(None, 17, 12, 720)	0	['block5b_activation[0][0]', 'block5b_se_expand[0][0]']
block5b_project_conv (Conv2D)	(None, 17, 12, 120)	86400	['block5b_se_excite[0][0]']
block5b_project_bn (BatchNormala lization)	(None, 17, 12, 120)	480	['block5b_project_conv[0][0]']
block5b_drop (Dropout)	(None, 17, 12, 120)	0	['block5b_project_bn[0][0]']
block5b_add (Add)	(None, 17, 12, 120)	0	['block5b_drop[0][0]', 'block5a_project_bn[0][0]']
block5c_expand_conv (Conv2D)	(None, 17, 12, 720)	86400	['block5b_add[0][0]']
block5c_expand_bn (BatchNormala lization)	(None, 17, 12, 720)	2880	['block5c_expand_conv[0][0]']
block5c_expand_activation (Act ivation)	(None, 17, 12, 720)	0	['block5c_expand_bn[0][0]']
block5c_dwconv2 (DepthwiseConv 2D)	(None, 17, 12, 720)	6480	['block5c_expand_activation[0][0]']
block5c_bn (BatchNormalizatio n)	(None, 17, 12, 720)	2880	['block5c_dwconv2[0][0]']
block5c_activation (Activation)	(None, 17, 12, 720)	0	['block5c_bn[0][0]']
block5c_se_squeeze (GlobalAver agePooling2D)	(None, 720)	0	['block5c_activation[0][0]']
block5c_se_reshape (Reshape)	(None, 1, 1, 720)	0	['block5c_se_squeeze[0][0]']
block5c_se_reduce (Conv2D)	(None, 1, 1, 30)	21630	['block5c_se_reshape[0][0]']
block5c_se_expand (Conv2D)	(None, 1, 1, 720)	22320	['block5c_se_reduce[0][0]']

block5c_se_excite (Multiply)	(None, 17, 12, 720)	0	['block5c_activation[0][0]', 'block5c_se_expand[0][0]']
block5c_project_conv (Conv2D)	(None, 17, 12, 120)	86400	['block5c_se_excite[0][0]']
block5c_project_bn (BatchNormala lization)	(None, 17, 12, 120)	480	['block5c_project_conv[0][0]']
block5c_drop (Dropout)	(None, 17, 12, 120)	0	['block5c_project_bn[0][0]']
block5c_add (Add)	(None, 17, 12, 120)	0	['block5c_drop[0][0]', 'block5b_add[0][0]']
block5d_expand_conv (Conv2D)	(None, 17, 12, 720)	86400	['block5c_add[0][0]']
block5d_expand_bn (BatchNormala lization)	(None, 17, 12, 720)	2880	['block5d_expand_conv[0][0]']
block5d_expand_activation (Act ivation)	(None, 17, 12, 720)	0	['block5d_expand_bn[0][0]']
block5d_dwconv2 (DepthwiseConv 2D)	(None, 17, 12, 720)	6480	['block5d_expand_activation[0][0] ']
block5d_bn (BatchNormalizatio n)	(None, 17, 12, 720)	2880	['block5d_dwconv2[0][0]']
block5d_activation (Activation)	(None, 17, 12, 720)	0	['block5d_bn[0][0]']
block5d_se_squeeze (GlobalAver agePooling2D)	(None, 720)	0	['block5d_activation[0][0]']
block5d_se_reshape (Reshape)	(None, 1, 1, 720)	0	['block5d_se_squeeze[0][0]']
block5d_se_reduce (Conv2D)	(None, 1, 1, 30)	21630	['block5d_se_reshape[0][0]']
block5d_se_expand (Conv2D)	(None, 1, 1, 720)	22320	['block5d_se_reduce[0][0]']
block5d_se_excite (Multiply)	(None, 17, 12, 720)	0	['block5d_activation[0][0]', 'block5d_se_expand[0][0]']
block5d_project_conv (Conv2D)	(None, 17, 12, 120)	86400	['block5d_se_excite[0][0]']

block5d_project_bn (BatchNormaliza lization)	(None, 17, 12, 120) 480	['block5d_project_conv[0][0]']
block5d_drop (Dropout)	(None, 17, 12, 120) 0	['block5d_project_bn[0][0]']
block5d_add (Add)	(None, 17, 12, 120) 0	['block5d_drop[0][0]', 'block5c_add[0][0]']
block5e_expand_conv (Conv2D)	(None, 17, 12, 720) 86400	['block5d_add[0][0]']
block5e_expand_bn (BatchNormaliz ation)	(None, 17, 12, 720) 2880	['block5e_expand_conv[0][0]']
block5e_expand_activation (Act ivation)	(None, 17, 12, 720) 0	['block5e_expand_bn[0][0]']
block5e_dwconv2 (DepthwiseConv 2D)	(None, 17, 12, 720) 6480	['block5e_expand_activation[0][0]']
block5e_bn (BatchNormalizatio n)	(None, 17, 12, 720) 2880	['block5e_dwconv2[0][0]']
block5e_activation (Activation)	(None, 17, 12, 720) 0	['block5e_bn[0][0]']
block5e_se_squeeze (GlobalAver agePooling2D)	(None, 720) 0	['block5e_activation[0][0]']
block5e_se_reshape (Reshape)	(None, 1, 1, 720) 0	['block5e_se_squeeze[0][0]']
block5e_se_reduce (Conv2D)	(None, 1, 1, 30) 21630	['block5e_se_reshape[0][0]']
block5e_se_expand (Conv2D)	(None, 1, 1, 720) 22320	['block5e_se_reduce[0][0]']
block5e_se_excite (Multiply)	(None, 17, 12, 720) 0	['block5e_activation[0][0]', 'block5e_se_expand[0][0]']
block5e_project_conv (Conv2D)	(None, 17, 12, 120) 86400	['block5e_se_excite[0][0]']
block5e_project_bn (BatchNorma lization)	(None, 17, 12, 120) 480	['block5e_project_conv[0][0]']
block5e_drop (Dropout)	(None, 17, 12, 120) 0	['block5e_project_bn[0][0]']

block5e_add (Add)	(None, 17, 12, 120) 0	['block5e_drop[0][0]', 'block5d_add[0][0]']
block5f_expand_conv (Conv2D)	(None, 17, 12, 720) 86400	['block5e_add[0][0]']
block5f_expand_bn (BatchNormal ization)	(None, 17, 12, 720) 2880	['block5f_expand_conv[0][0]']
block5f_expand_activation (Act ivation)	(None, 17, 12, 720) 0	['block5f_expand_bn[0][0]']
block5f_dwconv2 (DepthwiseConv 2D)	(None, 17, 12, 720) 6480	['block5f_expand_activation[0][0] ']
block5f_bn (BatchNormalization)	(None, 17, 12, 720) 2880	['block5f_dwconv2[0][0]']
block5f_activation (Activation)	(None, 17, 12, 720) 0	['block5f_bn[0][0]']
block5f_se_squeeze (GlobalAver agePooling2D)	(None, 720) 0	['block5f_activation[0][0]']
block5f_se_reshape (Reshape)	(None, 1, 1, 720) 0	['block5f_se_squeeze[0][0]']
block5f_se_reduce (Conv2D)	(None, 1, 1, 30) 21630	['block5f_se_reshape[0][0]']
block5f_se_expand (Conv2D)	(None, 1, 1, 720) 22320	['block5f_se_reduce[0][0]']
block5f_se_excite (Multiply)	(None, 17, 12, 720) 0	['block5f_activation[0][0]', 'block5f_se_expand[0][0]']
block5f_project_conv (Conv2D)	(None, 17, 12, 120) 86400	['block5f_se_excite[0][0]']
block5f_project_bn (BatchNorma lization)	(None, 17, 12, 120) 480	['block5f_project_conv[0][0]']
block5f_drop (Dropout)	(None, 17, 12, 120) 0	['block5f_project_bn[0][0]']
block5f_add (Add)	(None, 17, 12, 120) 0	['block5f_drop[0][0]', 'block5e_add[0][0]']
block6a_expand_conv (Conv2D)	(None, 17, 12, 720) 86400	['block5f_add[0][0]']

block6a_expand_bn (BatchNormal ization)	(None, 17, 12, 720) 2880		['block6a_expand_conv[0][0]']
block6a_expand_activation (Act ivation)	(None, 17, 12, 720) 0		['block6a_expand_bn[0][0]']
block6a_dwconv2 (DepthwiseConv 2D)	(None, 9, 6, 720) 6480		['block6a_expand_activation[0][0] ']
block6a_bn (BatchNormalizatio n)	(None, 9, 6, 720) 2880		['block6a_dwconv2[0][0]']
block6a_activation (Activation)	(None, 9, 6, 720) 0		['block6a_bn[0][0]']
block6a_se_squeeze (GlobalAver agePooling2D)	(None, 720) 0		['block6a_activation[0][0]']
block6a_se_reshape (Reshape)	(None, 1, 1, 720) 0		['block6a_se_squeeze[0][0]']
block6a_se_reduce (Conv2D)	(None, 1, 1, 30) 21630		['block6a_se_reshape[0][0]']
block6a_se_expand (Conv2D)	(None, 1, 1, 720) 22320		['block6a_se_reduce[0][0]']
block6a_se_excite (Multiply)	(None, 9, 6, 720) 0		['block6a_activation[0][0]', 'block6a_se_expand[0][0]']
block6a_project_conv (Conv2D)	(None, 9, 6, 208) 149760		['block6a_se_excite[0][0]']
block6a_project_bn (BatchNorma lization)	(None, 9, 6, 208) 832		['block6a_project_conv[0][0]']
block6b_expand_conv (Conv2D)	(None, 9, 6, 1248) 259584		['block6a_project_bn[0][0]']
block6b_expand_bn (BatchNormal ization)	(None, 9, 6, 1248) 4992		['block6b_expand_conv[0][0]']
block6b_expand_activation (Act ivation)	(None, 9, 6, 1248) 0		['block6b_expand_bn[0][0]']
block6b_dwconv2 (DepthwiseConv 2D)	(None, 9, 6, 1248) 11232		['block6b_expand_activation[0][0] ']
block6b_bn (BatchNormalizatio n)	(None, 9, 6, 1248) 4992		['block6b_dwconv2[0][0]']

)			
block6b_activation (Activation (None, 9, 6, 1248) 0)			['block6b_bn[0][0]']
block6b_se_squeeze (GlobalAveragePooling2D)	0		['block6b_activation[0][0]']
block6b_se_reshape (Reshape) (None, 1, 1, 1248) 0			['block6b_se_squeeze[0][0]']
block6b_se_reduce (Conv2D) (None, 1, 1, 52) 64948			['block6b_se_reshape[0][0]']
block6b_se_expand (Conv2D) (None, 1, 1, 1248) 66144			['block6b_se_reduce[0][0]']
block6b_se_excite (Multiply) (None, 9, 6, 1248) 0			['block6b_activation[0][0]', 'block6b_se_expand[0][0]']
block6b_project_conv (Conv2D) (None, 9, 6, 208) 259584			['block6b_se_excite[0][0]']
block6b_project_bn (BatchNormalization (None, 9, 6, 208) 832 lization)			['block6b_project_conv[0][0]']
block6b_drop (Dropout) (None, 9, 6, 208) 0			['block6b_project_bn[0][0]']
block6b_add (Add) (None, 9, 6, 208) 0			['block6b_drop[0][0]', 'block6a_project_bn[0][0]']
block6c_expand_conv (Conv2D) (None, 9, 6, 1248) 259584			['block6b_add[0][0]']
block6c_expand_bn (BatchNormalization (None, 9, 6, 1248) 4992 iization)			['block6c_expand_conv[0][0]']
block6c_expand_activation (Activation (None, 9, 6, 1248) 0 ivation)			['block6c_expand_bn[0][0]']
block6c_dwconv2 (DepthwiseConv2D) (None, 9, 6, 1248) 11232			['block6c_expand_activation[0][0]']
block6c_bn (BatchNormalization (None, 9, 6, 1248) 4992)			['block6c_dwconv2[0][0]']
block6c_activation (Activation (None, 9, 6, 1248) 0)			['block6c_bn[0][0]']

block6c_se_squeeze (GlobalAveragePooling2D)	(None, 1248)	0	['block6c_activation[0][0]']
block6c_se_reshape (Reshape)	(None, 1, 1, 1248)	0	['block6c_se_squeeze[0][0]']
block6c_se_reduce (Conv2D)	(None, 1, 1, 52)	64948	['block6c_se_reshape[0][0]']
block6c_se_expand (Conv2D)	(None, 1, 1, 1248)	66144	['block6c_se_reduce[0][0]']
block6c_se_excite (Multiply)	(None, 9, 6, 1248)	0	['block6c_activation[0][0]', 'block6c_se_expand[0][0]']
block6c_project_conv (Conv2D)	(None, 9, 6, 208)	259584	['block6c_se_excite[0][0]']
block6c_project_bn (BatchNormalization)	(None, 9, 6, 208)	832	['block6c_project_conv[0][0]']
block6c_drop (Dropout)	(None, 9, 6, 208)	0	['block6c_project_bn[0][0]']
block6c_add (Add)	(None, 9, 6, 208)	0	['block6c_drop[0][0]', 'block6b_add[0][0]']
block6d_expand_conv (Conv2D)	(None, 9, 6, 1248)	259584	['block6c_add[0][0]']
block6d_expand_bn (BatchNormalization)	(None, 9, 6, 1248)	4992	['block6d_expand_conv[0][0]']
block6d_expand_activation (Activation)	(None, 9, 6, 1248)	0	['block6d_expand_bn[0][0]']
block6d_dwconv2 (DepthwiseConv2D)	(None, 9, 6, 1248)	11232	['block6d_expand_activation[0][0]']
block6d_bn (BatchNormalization)	(None, 9, 6, 1248)	4992	['block6d_dwconv2[0][0]']
block6d_activation (Activation)	(None, 9, 6, 1248)	0	['block6d_bn[0][0]']
block6d_se_squeeze (GlobalAveragePooling2D)	(None, 1248)	0	['block6d_activation[0][0]']
block6d_se_reshape (Reshape)	(None, 1, 1, 1248)	0	['block6d_se_squeeze[0][0]']

				Assignment_2
block6d_se_reduce (Conv2D)	(None, 1, 1, 52)	64948		['block6d_se_reshape[0][0]']
block6d_se_expand (Conv2D)	(None, 1, 1, 1248)	66144		['block6d_se_reduce[0][0]']
block6d_se_excite (Multiply)	(None, 9, 6, 1248)	0		['block6d_activation[0][0]', 'block6d_se_expand[0][0]']
block6d_project_conv (Conv2D)	(None, 9, 6, 208)	259584		['block6d_se_excite[0][0]']
block6d_project_bn (BatchNormala lization)	(None, 9, 6, 208)	832		['block6d_project_conv[0][0]']
block6d_drop (Dropout)	(None, 9, 6, 208)	0		['block6d_project_bn[0][0]']
block6d_add (Add)	(None, 9, 6, 208)	0		['block6d_drop[0][0]', 'block6c_add[0][0]']
block6e_expand_conv (Conv2D)	(None, 9, 6, 1248)	259584		['block6d_add[0][0]']
block6e_expand_bn (BatchNormala lization)	(None, 9, 6, 1248)	4992		['block6e_expand_conv[0][0]']
block6e_expand_activation (Act ivation)	(None, 9, 6, 1248)	0		['block6e_expand_bn[0][0]']
block6e_dwconv2 (DepthwiseConv 2D)	(None, 9, 6, 1248)	11232		['block6e_expand_activation[0][0]']
block6e_bn (BatchNormalizatio n)	(None, 9, 6, 1248)	4992		['block6e_dwconv2[0][0]']
block6e_activation (Activation)	(None, 9, 6, 1248)	0		['block6e_bn[0][0]']
block6e_se_squeeze (GlobalAver agePooling2D)	(None, 1248)	0		['block6e_activation[0][0]']
block6e_se_reshape (Reshape)	(None, 1, 1, 1248)	0		['block6e_se_squeeze[0][0]']
block6e_se_reduce (Conv2D)	(None, 1, 1, 52)	64948		['block6e_se_reshape[0][0]']
block6e_se_expand (Conv2D)	(None, 1, 1, 1248)	66144		['block6e_se_reduce[0][0]']
block6e_se_excite (Multiply)	(None, 9, 6, 1248)	0		['block6e_activation[0][0]',

				'block6e_se_expand[0][0]'
block6e_project_conv (Conv2D)	(None, 9, 6, 208)	259584		['block6e_se_excite[0][0]']
block6e_project_bn (BatchNormala lization)	(None, 9, 6, 208)	832		['block6e_project_conv[0][0]']
block6e_drop (Dropout)	(None, 9, 6, 208)	0		['block6e_project_bn[0][0]']
block6e_add (Add)	(None, 9, 6, 208)	0		['block6e_drop[0][0]', 'block6d_add[0][0]']
block6f_expand_conv (Conv2D)	(None, 9, 6, 1248)	259584		['block6e_add[0][0]']
block6f_expand_bn (BatchNormala lization)	(None, 9, 6, 1248)	4992		['block6f_expand_conv[0][0]']
block6f_expand_activation (Act ivation)	(None, 9, 6, 1248)	0		['block6f_expand_bn[0][0]']
block6f_dwconv2 (DepthwiseConv 2D)	(None, 9, 6, 1248)	11232		['block6f_expand_activation[0][0] ']
block6f_bn (BatchNormalization)	(None, 9, 6, 1248)	4992		['block6f_dwconv2[0][0]']
block6f_activation (Activation)	(None, 9, 6, 1248)	0		['block6f_bn[0][0]']
block6f_se_squeeze (GlobalAver agePooling2D)	(None, 1248)	0		['block6f_activation[0][0]']
block6f_se_reshape (Reshape)	(None, 1, 1, 1248)	0		['block6f_se_squeeze[0][0]']
block6f_se_reduce (Conv2D)	(None, 1, 1, 52)	64948		['block6f_se_reshape[0][0]']
block6f_se_expand (Conv2D)	(None, 1, 1, 1248)	66144		['block6f_se_reduce[0][0]']
block6f_se_excite (Multiply)	(None, 9, 6, 1248)	0		['block6f_activation[0][0]', 'block6f_se_expand[0][0]']
block6f_project_conv (Conv2D)	(None, 9, 6, 208)	259584		['block6f_se_excite[0][0]']
block6f_project_bn (BatchNorma lization)	(None, 9, 6, 208)	832		['block6f_project_conv[0][0]']

lization)			
block6f_drop (Dropout)	(None, 9, 6, 208)	0	['block6f_project_bn[0][0]']
block6f_add (Add)	(None, 9, 6, 208)	0	['block6f_drop[0][0]', 'block6e_add[0][0]']
block6g_expand_conv (Conv2D)	(None, 9, 6, 1248)	259584	['block6f_add[0][0]']
block6g_expand_bn (BatchNormal ization)	(None, 9, 6, 1248)	4992	['block6g_expand_conv[0][0]']
block6g_expand_activation (Act ivation)	(None, 9, 6, 1248)	0	['block6g_expand_bn[0][0]']
block6g_dwconv2 (DepthwiseConv 2D)	(None, 9, 6, 1248)	11232	['block6g_expand_activation[0][0]']
block6g_bn (BatchNormalizatio n)	(None, 9, 6, 1248)	4992	['block6g_dwconv2[0][0]']
block6g_activation (Activation)	(None, 9, 6, 1248)	0	['block6g_bn[0][0]']
block6g_se_squeeze (GlobalAver agePooling2D)	(None, 1248)	0	['block6g_activation[0][0]']
block6g_se_reshape (Reshape)	(None, 1, 1, 1248)	0	['block6g_se_squeeze[0][0]']
block6g_se_reduce (Conv2D)	(None, 1, 1, 52)	64948	['block6g_se_reshape[0][0]']
block6g_se_expand (Conv2D)	(None, 1, 1, 1248)	66144	['block6g_se_reduce[0][0]']
block6g_se_excite (Multiply)	(None, 9, 6, 1248)	0	['block6g_activation[0][0]', 'block6g_se_expand[0][0]']
block6g_project_conv (Conv2D)	(None, 9, 6, 208)	259584	['block6g_se_excite[0][0]']
block6g_project_bn (BatchNorma lization)	(None, 9, 6, 208)	832	['block6g_project_conv[0][0]']
block6g_drop (Dropout)	(None, 9, 6, 208)	0	['block6g_project_bn[0][0]']
block6g_add (Add)	(None, 9, 6, 208)	0	['block6g_drop[0][0]',

				'block6f_add[0][0]']
block6h_expand_conv (Conv2D)	(None, 9, 6, 1248)	259584		['block6g_add[0][0]']
block6h_expand_bn (BatchNormal ization)	(None, 9, 6, 1248)	4992		['block6h_expand_conv[0][0]']
block6h_expand_activation (Act ivation)	(None, 9, 6, 1248)	0		['block6h_expand_bn[0][0]']
block6h_dwconv2 (DepthwiseConv 2D)	(None, 9, 6, 1248)	11232		['block6h_expand_activation[0][0] ']
block6h_bn (BatchNormalization)	(None, 9, 6, 1248)	4992		['block6h_dwconv2[0][0]']
block6h_activation (Activation)	(None, 9, 6, 1248)	0		['block6h_bn[0][0]']
block6h_se_squeeze (GlobalAver agePooling2D)	(None, 1248)	0		['block6h_activation[0][0]']
block6h_se_reshape (Reshape)	(None, 1, 1, 1248)	0		['block6h_se_squeeze[0][0]']
block6h_se_reduce (Conv2D)	(None, 1, 1, 52)	64948		['block6h_se_reshape[0][0]']
block6h_se_expand (Conv2D)	(None, 1, 1, 1248)	66144		['block6h_se_reduce[0][0]']
block6h_se_excite (Multiply)	(None, 9, 6, 1248)	0		['block6h_activation[0][0]', 'block6h_se_expand[0][0]']
block6h_project_conv (Conv2D)	(None, 9, 6, 208)	259584		['block6h_se_excite[0][0]']
block6h_project_bn (BatchNorma lization)	(None, 9, 6, 208)	832		['block6h_project_conv[0][0]']
block6h_drop (Dropout)	(None, 9, 6, 208)	0		['block6h_project_bn[0][0]']
block6h_add (Add)	(None, 9, 6, 208)	0		['block6h_drop[0][0]', 'block6g_add[0][0]']
block6i_expand_conv (Conv2D)	(None, 9, 6, 1248)	259584		['block6h_add[0][0]']
block6i_expand_bn (BatchNormal alization)	(None, 9, 6, 1248)	4992		['block6i_expand_conv[0][0]']

ization)			
block6i_expand_activation (Act (None, 9, 6, 1248) 0 ivation)			['block6i_expand_bn[0][0]']
block6i_dwconv2 (DepthwiseConv 2D (None, 9, 6, 1248) 11232			['block6i_expand_activation[0][0]']
block6i_bn (BatchNormalization (None, 9, 6, 1248) 4992)			['block6i_dwconv2[0][0]']
block6i_activation (Activation (None, 9, 6, 1248) 0)			['block6i_bn[0][0]']
block6i_se_squeeze (GlobalAveragePooling2D (None, 1248) 0			['block6i_activation[0][0]']
block6i_se_reshape (Reshape (None, 1, 1, 1248) 0			['block6i_se_squeeze[0][0]']
block6i_se_reduce (Conv2D (None, 1, 1, 52) 64948			['block6i_se_reshape[0][0]']
block6i_se_expand (Conv2D (None, 1, 1, 1248) 66144			['block6i_se_reduce[0][0]']
block6i_se_excite (Multiply (None, 9, 6, 1248) 0			['block6i_activation[0][0]', 'block6i_se_expand[0][0]']
block6i_project_conv (Conv2D (None, 9, 6, 208) 259584			['block6i_se_excite[0][0]']
block6i_project_bn (BatchNormalization (None, 9, 6, 208) 832			['block6i_project_conv[0][0]']
block6i_drop (Dropout (None, 9, 6, 208) 0			['block6i_project_bn[0][0]']
block6i_add (Add (None, 9, 6, 208) 0			['block6i_drop[0][0]', 'block6h_add[0][0]']
block6j_expand_conv (Conv2D (None, 9, 6, 1248) 259584			['block6i_add[0][0]']
block6j_expand_bn (BatchNormalization (None, 9, 6, 1248) 4992 ization)			['block6j_expand_conv[0][0]']
block6j_expand_activation (Activation (None, 9, 6, 1248) 0 ivation)			['block6j_expand_bn[0][0]']

block6j_dwconv2 (DepthwiseConv 2D)	(None, 9, 6, 1248)	11232	['block6j_expand_activation[0][0]']
block6j_bn (BatchNormalization)	(None, 9, 6, 1248)	4992	['block6j_dwconv2[0][0]']
block6j_activation (Activation)	(None, 9, 6, 1248)	0	['block6j_bn[0][0]']
block6j_se_squeeze (GlobalAveragePooling2D)	(None, 1248)	0	['block6j_activation[0][0]']
block6j_se_reshape (Reshape)	(None, 1, 1, 1248)	0	['block6j_se_squeeze[0][0]']
block6j_se_reduce (Conv2D)	(None, 1, 1, 52)	64948	['block6j_se_reshape[0][0]']
block6j_se_expand (Conv2D)	(None, 1, 1, 1248)	66144	['block6j_se_reduce[0][0]']
block6j_se_excite (Multiply)	(None, 9, 6, 1248)	0	['block6j_activation[0][0]', 'block6j_se_expand[0][0]']
block6j_project_conv (Conv2D)	(None, 9, 6, 208)	259584	['block6j_se_excite[0][0]']
block6j_project_bn (BatchNormalization)	(None, 9, 6, 208)	832	['block6j_project_conv[0][0]']
block6j_drop (Dropout)	(None, 9, 6, 208)	0	['block6j_project_bn[0][0]']
block6j_add (Add)	(None, 9, 6, 208)	0	['block6j_drop[0][0]', 'block6i_add[0][0]']
top_conv (Conv2D)	(None, 9, 6, 1408)	292864	['block6j_add[0][0]']
top_bn (BatchNormalization)	(None, 9, 6, 1408)	5632	['top_conv[0][0]']
top_activation (Activation)	(None, 9, 6, 1408)	0	['top_bn[0][0]']
flatten_6 (Flatten)	(None, 76032)	0	['top_activation[0][0]']

=====

Total params: 8,769,374

Trainable params: 8,687,086

Non-trainable params: 82,288

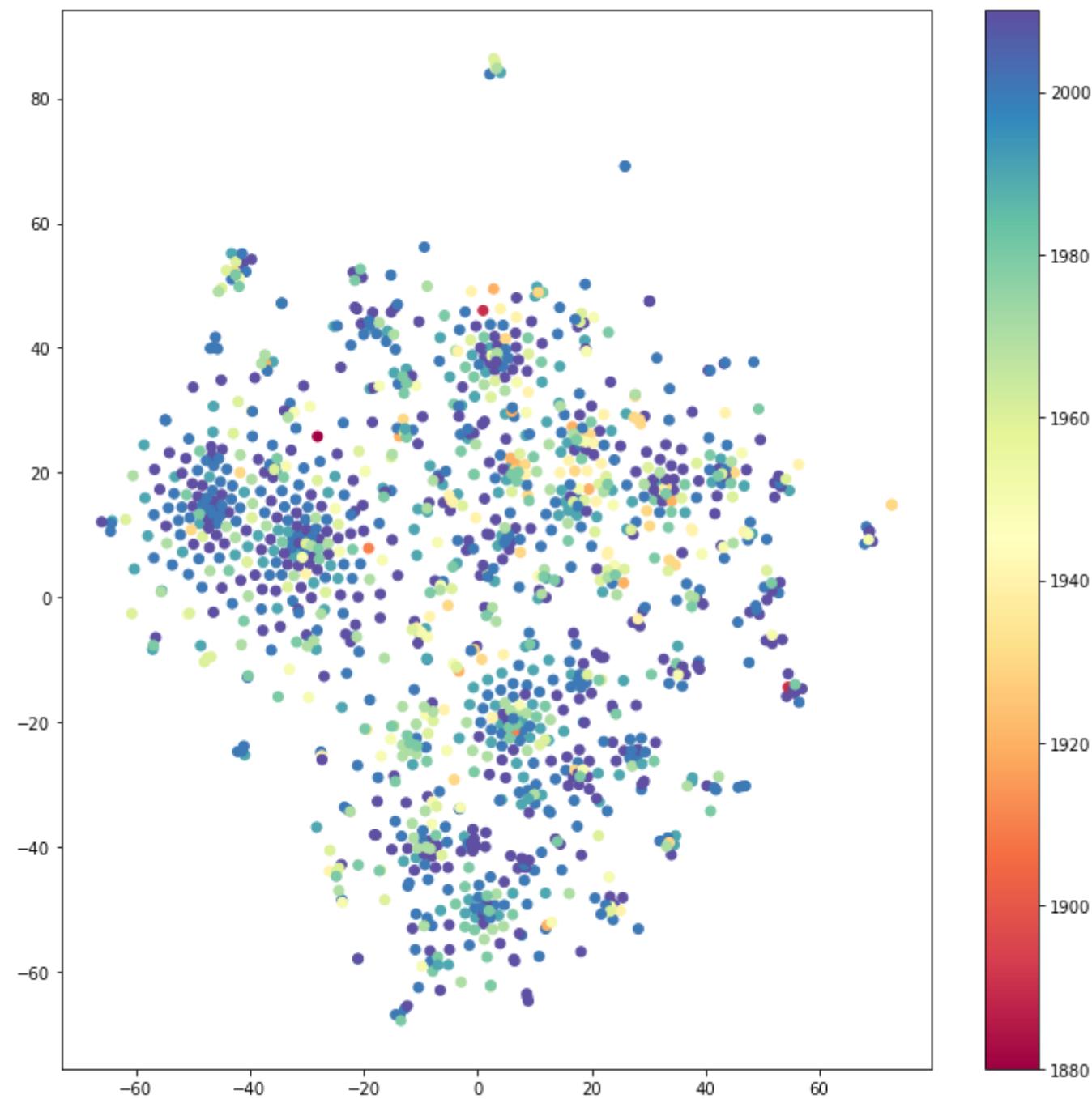
```
39/39 [=====] - 42s 1s/step
```

```
In [ ]: #Testing on decades
```

```
tsneDecadeV2 = TSNE(n_components=2, random_state=0, perplexity=3, n_iter=5000)
tsneOutputDecadeV2 = tsneDecadeV2.fit_transform(newX)
```

```
c:\Users\CAOLA\anaconda3\lib\site-packages\sklearn\manifold\_t_sne.py:780: FutureWarning: The default initialization in TSNE will change from 'random' to 'pca' in 1.2.
  warnings.warn(
c:\Users\CAOLA\anaconda3\lib\site-packages\sklearn\manifold\_t_sne.py:790: FutureWarning: The default learning rate in TSNE will change from 200.0 to 'auto' in 1.2.
  warnings.warn(
```

```
In [ ]: plt.figure(figsize=(12,12))
plt.scatter(tsneOutputDecadeV2[:, 0], tsneOutputDecadeV2[:, 1], c=decades, cmap='Spectral')
plt.colorbar()
plt.show()
```



Part 5 comments

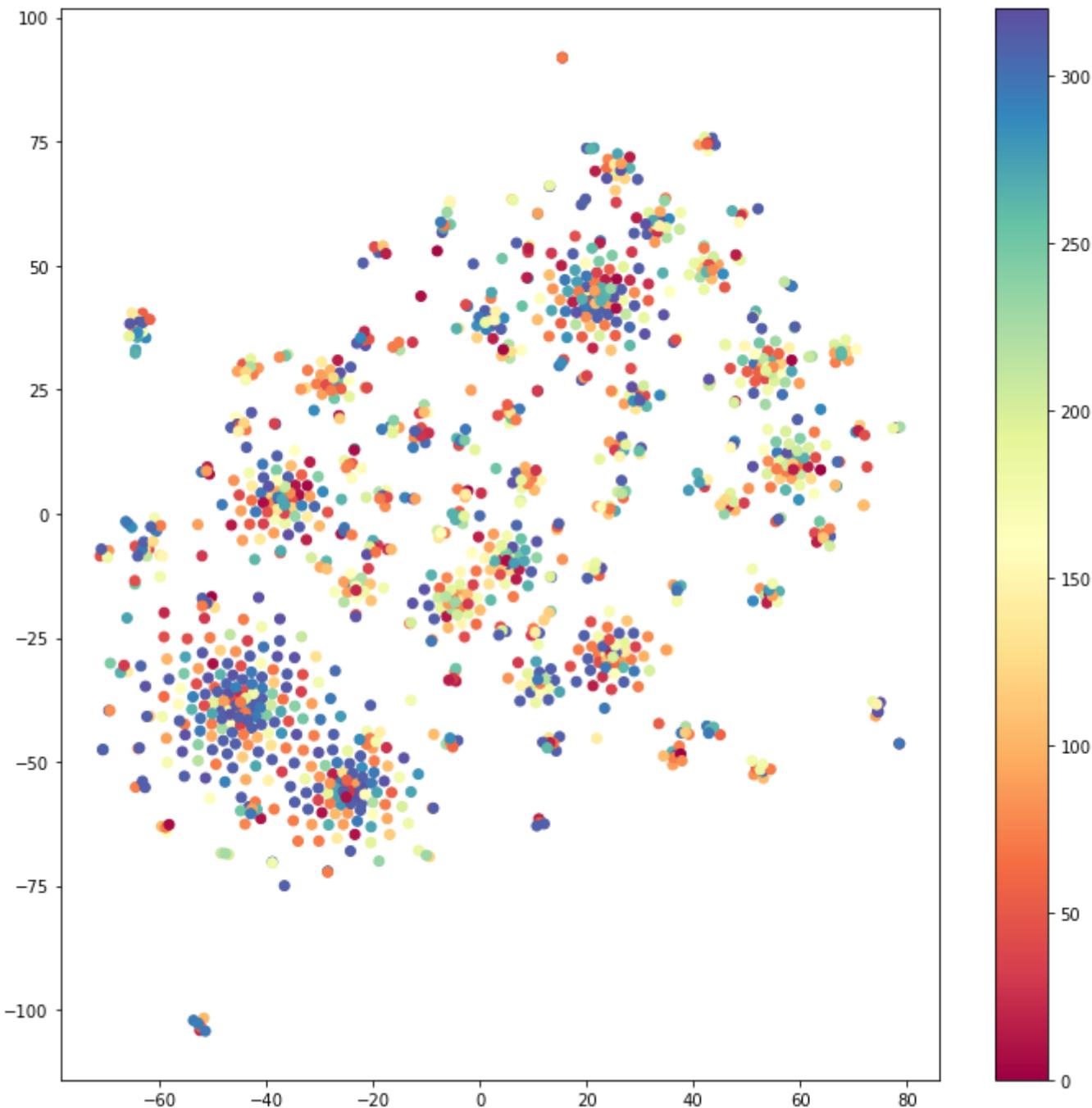
We can see similar behaviour with the same perplexity value, but if we change this. What originally looked random, now has a drastically different look, as seen in the above plot. Clusters have begin to form, where the largest clusters are from the relatively same era, e.g the large cluster in the central-left, is posters mostly from 1970s and up, with the most modern films concentrated at the centre. A lot of these posters hold photographs of the characters in the centre of these posters, which could be a reason for this clustering.

```
In [ ]: tsneGenreV2 = TSNE(n_components=2, random_state=0, perplexity=2, n_iter=5000, n_iter_without_progress=1250)
tsneOutputGenreV2 = tsneGenreV2.fit_transform(newX)

c:\Users\CAOLA\anaconda3\lib\site-packages\sklearn\manifold\_t_sne.py:780: FutureWarning: The default initialization in TSNE will change from 'random' to 'pca' in 1.2.
    warnings.warn(
c:\Users\CAOLA\anaconda3\lib\site-packages\sklearn\manifold\_t_sne.py:790: FutureWarning: The default learning rate in TSNE will change from 200.0 to 'auto' in 1.2.
    warnings.warn(
```



```
In [ ]: plt.figure(figsize=(12,12))
plt.scatter(tsneOutputGenreV2[:, 0], tsneOutputGenreV2[:, 1], c=genreColours, cmap='Spectral')
plt.colorbar()
plt.show()
```



Part 5 comments

We can see a similar results using the new model on the genre t-sne, but the clusters are larger, with certain genres (colours) closer together. This does also reinforce the observation that genre and posters don't correlate well, with plenty of different colours mixed within these clusters.