Artificial Intelligence Project: CelebA dataset

Context

A popular component of computer vision and deep learning revolves around identifying faces for various applications from logging into your phone with your face or searching through surveillance images for a particular suspect. The **CelebA** dataset is great for training and testing models for face detection, **particularly for recognising facial attributes such as finding people with brown hair, are smiling, or wearing glasses**. Images cover large pose variations, background clutter, diverse people, supported by a large quantity of images and rich annotations. This data was originally collected by researchers at MMLAB, The Chinese University of Hong Kong (specific reference in Acknowledgment section).

Content

Overall

- **202,599 number of face images of various celebrities**
- 10,177 unique identities, but names of identities are not given
- **40 binary attribute annotations per image**
- 5 landmark locations

Data Files

- **img_align_celeba.zip**: All the face images, cropped and aligned
- list_eval_partition.csv: Recommended partitioning of images into training, validation, testing sets. Images 1-162770 are training, 162771-182637 are validation, 182638-202599 are testing
- list_bbox_celeba.csv: Bounding box information for each image. "x_1" and "y_1" represent the upper left point coordinate of bounding box. "width" and "height" represent the width and height of bounding box
- list_landmarks_align_celeba.csv: Image landmarks and their respective coordinates. There are 5 landmarks: left eye, right eye, nose, left mouth, right mouth
- **list_attr_celeba.csv**: Attribute labels for each image. There are 40 attributes. "1" represents positive while "-1" represents negative

Project description

In this project, you are asked to build and train a convolutional neural network model able to detect the presence or not of **one specific human feature** in face images. For example, you can chose to build a model able to detect whether the person in the image is smiling or not, or has brown hair or not, has a big nose or not etc. There are 40 different attributes that are recorded in this database and you are free to choose the attribute that you want.

The project should be realized in groups of 4. The code as well as a short report must be provided at the end of the project. The data pre-processing, model building (architecture, optimization technique, activation functions etc.), model training(epochs, size of batches, image augmentation etc.) and testing as well as any improvement or tuning of the model must be explained in the report. Your choices and analysis must be also clearly stated in the report.

Steps to follow in building the model

1. **Download the database** from the following link:
   https://www.kaggle.com/jessicali9530/celeba-dataset
   - Unzip the folder **img_align_celeba**. This folder contains the celebrities' images.
   - Inspect the **list_attr_celeba.csv**. This file contains the attribute (human feature) for each image. There are 40 attributes. "1" represents positive while "-1" represents negative.
2. **Data pre-processing**:
   This step consists on preparing your data for the training and the testing of the model. The idea is to split the images into **2 folders**: **Training** and **Testing** such as 80% of the images are copied to the Training folder and 20% of the images are copied to the Testing folder. In each folder, you should have **2 sub folders**: **Presence_of_feature** and **Absence_of_feature**. For example, Smile and NoSmile if the feature you are detecting is a smile. The first sub folder contains only images of people where the specific feature is present (1) and the second sub folder contains only images of people where it is not present (-1).

   Write the piece of code in python that allows you to copy the images into these folders.

   Hint: The os and shutil libraries can be very helpful.
   Example:
   - *src=os.path.join('Folder1','Folder2') → To concatenate paths : Folder1\\Folder2*
   - *shutil.copy(src,dest) → To copy a src file to a dest folder*

   Tip: Querying the dataset allow you to know the total of images that contain a specific feature. Comparing the number of images that were copied to that total allows you to make sure that splitting the images was successful.
   Example:
   - *dataset_filtered=dataset.query('Smiling==1')*

3. Build the CNN using Keras.
4. Print the summary of the model that you have just built using the *summary()* function. Explain the numbers that appear for each layer in the OutputShape and #Params columns.
   Example: classifier.summary()
5. Generate images using the image augmentation technique.
6. Train the model.
7. Save the model using the following instruction the save function.
   Example: classifier.save('my_CelebA_model.h5')

To load the model use the load_model function from keras.models
Example: from keras.models import keras.models
classifier=load_model('my_CelebA_Smile_model.h5')

8. Test the model to find the accuracy and the confusion matrix.
9. Test your model on a single prediction using an image from outside the Celeba dataset.
10. Based on the accuracy that you get, try to improve or to tune the model that you have created.
11. Plot some interesting results for example: the prediction of some the test set images.
12. Bonus: Compare the results that you have obtained with another image classification algorithm.

Any other idea or creativity to enhance the project is highly appreciated (e.g. real-time human face feature detection, training plots, etc.)