Binary RE-Identification using Convolutional Networks

Introduction

The objective of this project is for you to experience the process of training a deep model using a relatively large dataset. You will be solving a simplified version of the re-identification problem.

Re-identification is a computer vision problem with common applications in surveillance. The goal is to spot a person of interest in shots taken by multiple cameras spread around a venue. Different shots of the same person will vary in illumination, pose and possibly the objects carried by the person. Re-identification is applied to both videos and images.

In the *original* image re-identification problem, one shot of the person of interest is matched against a gallery of shots from neighboring cameras. The goal is to identify the matching shots or to rank the gallery shots according to the probability of matching the key shot[1].

Problem Definition

For this project, we implement **Binary Re-identification**, where a dataset is preprocessed to join two images together in a "pair image". A pair image is a positive match if it includes two shots of the same person, and negative if it includes two shots of different people. Binary Re-identification is a binary classification problem.

[1] Some of the recent papers on the subject:

ICML 2015: https://icml.cc/2015/wp-content/uploads/2015/06/icml_ranking.pdf

NIPS 2018: https://arxiv.org/abs/1803.07293

Dataset

The original dataset CUHK01 (66MB) is available for download from here and described in this paper. CUHK01 has images of 971 subjects, with four shots from different cameras for each subject. Each image is 60x160, hence a pair image is 120x160.

The modified dataset (Binary CUHK01) can be:

1. Downloaded from this Google Drive (download size ~167MB)

   https://drive.google.com/open?id=1ZcdC8tr8a7u2Z1JMvx2nuiDN2YM1niHD

2. Alternatively, it can be generated from the original CUHK01 using provided code.

Environment

You should use CoLab for this project and particularly the GPU or TPU runtimes to speed up the training. For this, you will need the dataset to be uploaded to **your** Google Drive.

Training Time

Using batch normalization and batch size=32, an epoch can take around 4 to 5 minutes using CoLab TPU Runtime, depending on your other settings. You should expect to run around ~30 epochs.

Checkpointing the weights is hence essential, to resume from best weights when session is interrupted.

Your Task

You're provided by a model in Keras, as well as weights that achieve ~85% training accuracy. The model is a modified version of Lab8 model and uses batch normalization. The weights file is available from the same Google Drive above. It was generated using batch size=32.

1- Use Keras' ImageDataGenerator to read the modified dataset and continue training the provided model.

2- Use ImageDataGenerator's parameters to create a training validation split (0.2) and to standardize the images.

3- Use Keras' model.fit_generator to train your model. Choose the appropriate steps_per_epoch and validation_steps value. These values are related to the batch size. Justify your choice.

4- Load the provided weights file and continue the training. Aim for validation accuracy of 92% at least.

5- Use Keras' checkpointing support to save the weights that achieve best validation performance. You will need to submit these.

6- Evaluate your model on the test set provided.

Bonus (0.5% of course grade)

The highest validation and test accuracies using the provided model will receive a bonus.

Double Bonus (1% of course grade)

Modify the model and retrain to achieve a higher performance (test or validation accuracy) than the maximum achieved by the provided model.

Submission

1. Submit a well-formatted Jupyter notebook that describes your results.
2. Submit a link to the weight file that achieves the best validation performance.
3. The notebook should be clear and self-sufficient as it will replace a demo.
4. Please keep the data files with the provided filenames to facilitate testing.
5. Please submit your Jupyter notebook and a shared link to the weights file via email to guc.ml18@gmail.com with the subject line (Project 3- team number).