Deep Learning Course Assignment

1. Introduction

In this assignment, you will explore the use of neural networks to solve two common computer vision tasks using the PyTorch framework. You are given a video-surveillance dataset containing images of multiple persons each of which is captured multiple times by different cameras along with a set of annotations that specify attributes of each person such as age, gender and clothing. The first part of the assignment consists in building a multi-class classifier to predict such attributes for each image. In the second part of the assignment, you are asked to solve a person re-identification problem where a *query* image of a person is given and all the images of the same person must be retrieved from a collection of images.

2. Data

The dataset that will be used for the project is a version of the Market-1501 person re-identification dataset and can be downloaded here. As you can see in Fig. 1 each image in the dataset corresponds to a tight crop of a pedestrian and the same person appears multiple times in the dataset. Moreover, while the differences between some persons are marked and easy to spot, some other cases are difficult to distinguish. The dataset is structured as follows.

Train directory that contains 12.989 64×128px images of pedestrians for training. Each image is annotated with a *person_id* that uniquely identifies the person in the image. A total of 751 distinct person identities are present in this dataset. Each file is structured as follows *person_id-_camera_id_random_number.jpg*.

Test directory that contains 19.679 64×128px images of pedestrians for testing. A total of 750 distinct person identities are present. Each file is named with a unique test file identification number and no information about the *person_id* is present. No identities from the *train* directory are present. The directory also contains a set of *distractor* and a set of *junk* images depicting partial observations of pedestrians or objects that are not pedestrians. Images of these categories should never be returned as results of a person re-identification query.

Queries directory that contains 2.248 64×128px images of pedestrians to use as queries for testing in the re-identification task. Each file is named with a unique query file identification number and no information about the *person_id* is present. The person identity of a query image is always guaranteed to be found at least once in the *test* directory.

Annotations_train.csv comma-separated file containing 27 annotations for each of the 751 person identities in the train set. Each field in the file follows the structure of the respective one described here. In particular, the *id* field corresponds to the *person_id* and maps each training image to its annotation.

3. Classification Task

As the first task of this assignment, you will build a multi-class classification model. Split the images in the train directory into a train and a validation dataset and use them to train a model that given the image of a pedestrian predicts the following attributes:

- gender
- hair length
- sleeve length
- length of lower-body clothing
- type of lower-body clothing
- wearing hat
- carrying backpack
- carrying bag
- carrying handbag
- age
- color of upper-body clothing
- color of lower-body clothing

The annotations for these attributes are present in the *annotations_train.csv* file and an example is shown in Fig. 2. Evaluate the performance of the model for each attribute using the validation set. Once validation is completed you are required to produce predictions for each image in the *test* directory. Produce a *classification_test.csv* file following the same format as *annotations_train.csv* containing your predictions. Under the *id* field, instead of reporting the *person_id*, which is unknown for the test set, please report the exact name of the file the predictions refer to, eg.

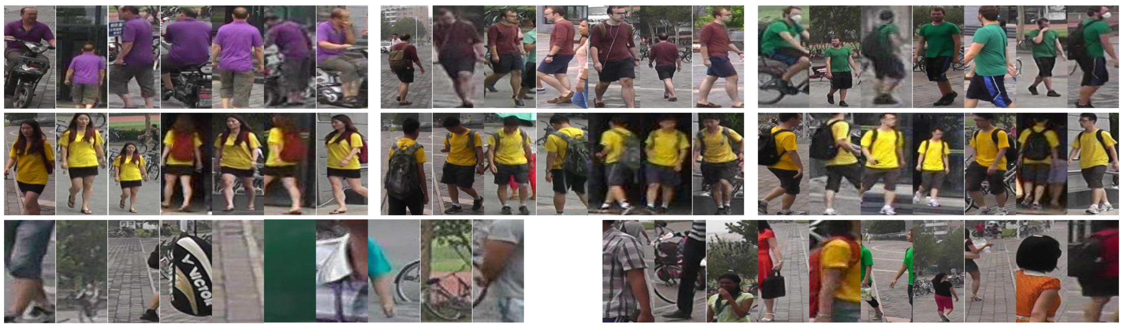
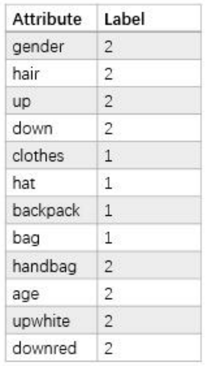


Figure 1: Sample images from the Market 1501 dataset. Each person is captured from multiple cameras in different time instants. In the test set, some distractor and junk images are present as shown in the last row. Image courtesy of [2].



Attribute	Label
gender	2
hair	2
up	2
down	2
clothes	1
hat	1
backpack	1
bag	1
handbag	2
age	2
upwhite	2
downred	2

Figure 2: A sample image with some of the related attributes. Image courtesy of [1].

001609.jpg. Note that the *test* directory also contains *distractor* and *junk* images. Please ignore this fact and produce predictions for them as if they were valid pedestrian images. For bonus points, you may describe how to modify your classifier to automatically detect such instances.

4. Person Re-Identification Task

In the second task of your assignment, you will build a person re-identification model to retrieve from the *test* directory all the images depicting the same person identity appearing in a given *query* image. More formally, for each image in the *queries* directory, produce an ordered list of images from the *test* directory that you believe correspond to the same person identity depicted in the query image. Each query image has at least a corresponding image in the *test* directory, but the number of corresponding images varies from query to query. The *test* directory also contains a set of *distractor* and *junk* images that should never be returned as the result of a query. Start by creating your own set of queries from the validation set that you previously used for the classification task and train your person re-identification model. Use the set of queries extracted from the validation set to evaluate the performance of your model. Mean average precision (mAP) is a commonly used metric to evaluate performance in person re-identification and should be included in your analysis. A reference python implementation of mAP is available here. Once you are satisfied with the performance of your model, you are required to produce predictions for each *query* image in the *queries* folder. For each query image with filename q_i, produce a list of images in the *test* folder with filenames $t_{i,1}, t_{i,2}, ..., t_{i,m_i}$ that you believe depict the same person as q_i. Note that the number of images m_i associated with q_i is variable. Produce a file containing your predictions named *reid_test.txt* with 2.248 rows, one for each *query* image. The i^{th} row has the format $q_i : t_{i,1}, t_{i,2}, ..., t_{i,m_i}$. For example, the row you would write for the results corresponding to the *query* image *000616.jpg* may be "*000616.jpg*: *001249.jpg*, *000316.jpg*, *000924.jpg*, *009847.jpg*".

5. Evaluation

Produce a report of minimum 2, maximum 4 pages, bibliography excluded, where you describe:

- The technical solution you adopted. In this section, you should describe the architecture you developed and the training losses using proper references to the literature. Highlight all your original contributions. A diagram showcasing your architecture and losses may be used to aid the discussion.
- Training details and evaluation. Describe in detail the procedure you followed to obtain a validation set and how your model was trained. Describe the metrics you chose for evaluation of the classification and person re-identification tasks and show the obtained results. If original technical contributions are adopted you may consider showing an ablation table where the performance increase obtained using your contribution is evaluated. In this section, you may also discuss cases in which your model fails, showing qualitative results if necessary.
- A description of the main classes and modules in your source code

To prepare your report use the CVPR 2021 paper template available here. The use of LaTeXis recommended, but you can also use Word to produce your report as long as you respect the provided template. As the authors of the paper include all the members of your group and specify the student id of each.

The project will be evaluated based on:

- Originality of the solution
- Thoroughness of the results
- Clarity of the report
- Performance measured on the provided *classification_test.csv* and *reid_test.txt* files
- Quality of the code

Submit a single archive file named *student_id_1_student_id_2.zip* to willi.menapace@unitn.it with mail title "[DL2020-2021 Assignment Submission]" with the following content:

- Your report named *student_id_1_student_id_2_report.pdf*
- Predictions for the classification task named *classification_test.csv*
- Predictions for the person re-identification task named *reid_test.txt*
- Your complete source code in a folder named *source_code*. The folder should contain a *README.md* file with detailed instructions on how to execute your code. If your code is intended to run in a Google Colab notebook please include your *.ipynb* files. If your code is not intended to run in a Google Colab notebook include all your source files and please provide a Conda environment to execute your code on a Linux machine.

The deadline for the submission of the project is 23:59 CET of tenth day before the exam. As an example, if the exam is on July 11, the deadline for project submission will be 23:59 CET July 1.

6. Group Registration

The project is intended for groups of 2 or 3 people. You can register your group by compiling the form available here before April 10. Groups of 1 person are not allowed unless exceptional circumstances are present.

If you have difficulties in finding a group you may use the shared document available here. Please list yourself in the document and proceed to autonomously contact the other listed students. Please remove your name once you find a group.

7. Collaboration and GitHub policy

The aim of the project is to learn both how to design and how to implement a solution. You may discuss the project with other groups as long as no source code is shared. Your solution has to be implemented from scratch or starting from the material of the laboratory sessions. Starting your project from an existing GitHub repository is not allowed since it would prevent you from making practice with all the aspects involved in the implementation of a solution. However, small portions of code from public GitHub repositories may be used in your project.

Your code may be checked against an automatic code plagiarism detection tool. Should a violation of this policy be found, actions will be taken for both the group that copied and the group where code was copied from. It is the responsibility of each group to keep their code private.

8. Suggestions

- Perform a review of the well-established methods for person re-identification. You may start with a survey such as https://arxiv.org/abs/2001.04193 and take inspiration from the approaches explained in 'Closed-World Person Re-Identification'. You are not required to implement a complex solution obtaining state-of-the-art performance, but rather strive to design a solution with an original component.
- Be careful when producing your train and validation splits. All images of the same person should be either in the train or in the validation dataset, otherwise, your validation performance will be higher than test performance.
- Make use of a logging framework such as *Tensorboard* or *Weights and Biases* to monitor the training of your models and compare different versions of them.
- Start your experiments with a small backbone to speed-up the initial experiments and increase the capacity only when your solution is stable.
- You may think of ways to use the classifier learned in the first task to improve the performance of the person re-identification task
- You may think of ways to use person re-identification to improve the performance in the classification task
- If you use Google Colab calibrate your solution to use a maximum of 8GB GPU memory. Google Colab will make an effort to provide the best available GPU with up to 16GB memory, but a GPU with 8GB may be assigned in the worst case. Using more than 8GB of memory will not guarantee you to be able to execute your code if a different GPU is assigned to the current session.

References

[1] Yutian Lin, Liang Zheng, Zhedong Zheng, Yu Wu, Zhilan Hu, Chenggang Yan, and Yi Yang. Improving person re-identification by attribute and identity learning. *Pattern Recognition*, 2019.

[2] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian. Scalable person re-identification: A benchmark. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1116–1124, 2015.