Person Re-Identification

Daniel Pamfil, Carolina Proietti, Tommaso Maldera

1 Introduction

Person Re-Identification has become one of the most interesting and challenging subjects in the field of intelligent video surveillance and is now a commonly known problem studied in Computer Science. It consists of the task of associating images taken from different cameras with disjoint fields of view, usually in a short span of time, to retrieve persons with the same identity.

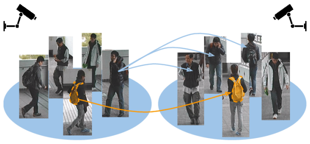


Figure 1: Base idea of person re-identification.

In these years, several methods are developed in order to achieve this goal. Some examples are local feature learning, which use local features to formulate a combined representation for each person image, or Generative adversarial learning, where the image disentangled, representation is used to achieve image-image style transfer or extract invariant features.
However, traditional person Re-ID methods require manual marking of person targets, which consumes a lot of labour cost. With the widespread application of deep neural networks, many deep learning-based person Re-ID methods with innovative approach to retrieve data have emerged.
In this report, we show how the appearance of the person can be used to perform a re-identification process with an unsupervised technique.

2 Techniques and implementation

We approach the task in such manner:

1. We build our custom dataset of 1M images, extracting frames from videos from all over the world.

2. Pre-trained the model in an unsupervised manner on the previous dataset and then validate it on another one.

3. The weights obtained were transferred to the fast-reid model to perform the training process.

4. We did a final test on a new third dataset in order to test the model on unseen and not biased images.

5. At the end, we developed a demo in order to exploit the capabilities of the model.

3 Libraries

In order to build the model, we used the following libraries:

3.1 Pytorch

As DeepLearning platform we used Pytorch, since it's an user-friendly API, furthermore support transfer learning and provide the opportunity to execute the training in a distributed fashion.

3.2 Numpy

NumPy dispense some well structured computing functions, with efficient array operations, and high interpretability with the other libraries used.

3.3 Torchreid

Torchreid is a library for deep-learning person re-identification, written in PyTorch, we used it in order to lighten the implementation of the model and overcome some tedious code blocks.

3.4 LMDB

Lmbd is a is a software library that provides an embedded transactional database in the form of a key-value store. We used it during the building of the dataset, since we needed a smart way to download and to access to the huge amount of images that we stored on the cloud.

3.5 OpenCV

For our image processing needs we used OpenCV, since it provides us a comprehensive and efficient toolkit.

3.6 Gradio

Gradio is a python library for creating and sharing web-based interactive machine learning models that we used in order to create an interactive interface for our demo.

4 Datasets

We used three different datasets, the first for pre-training, the second one both for validation on the pre-training phase and for training, and the last one for testing.

4.1 Artificial unsupervised dataset

Based on this implementation [1], we create a custom unlabelled dataset of 1.095.923 images with 43.454 different identities.
In order to collect the data, we downloaded a pool of YouTube videos from which we extract the images used to accomplish the re-identification process. One of the perks of this method is that address the problem of limited scale Re-ID datasets due to the costly effort required for data annotation. Another advantage is that using images from all over the world will help to develop a less biased model, improving its generalization ability.

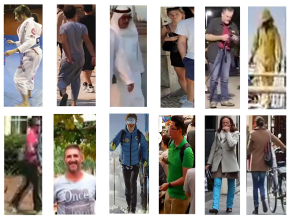


Figure 2: Random samples from our dataset. We can notice that the images very different both in terms of the quality of the pictures and of the subjects.

4.2 Market1501

Market1501 [2] is a large-scale person re-identification dataset that consists of 32,668 annotated images of 1,501 identities, captured by 6 cameras in front of a supermarket in Tsinghua University. This dataset is widely used for evaluating the performance of person Re-ID algorithms, since it provides a challenging test due to the large number of identities, significant variability in clothing, occlusion, and camera viewpoints.

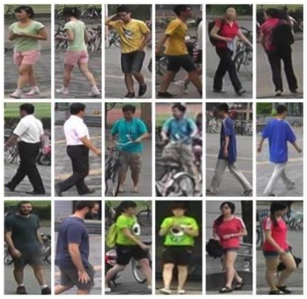


Figure 3: Some identities from Market1501.

4.3 CUHK03

CUHK03 is a re-identification dataset made up of 14,097 images of 1,467 unique individuals . These images were taken using 6 different cameras, with each person being captured by 2 of these cameras. We used this dataset only for a final test of the model.

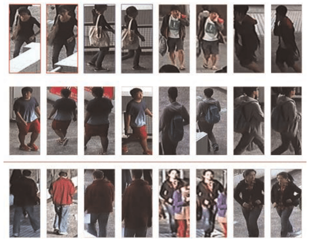


Figure 4: Samples from CUHK03.

5 Creation of the dataset

For the creation of the wide artificial dataset, we used two files provided by [1] for its creation: a text document with all the web addresses of the videos, and a folder of pickle format files containing all the coordinates of the detections in the videos.
First of all, we downloaded all the videos in a folder through the links obtained from the text file. Since all the videos where named with the format *country+city+videoID*, we splitted the string by "+" and then we transferred them in a tree directory according to the following path *country/city/video*. After that, we used the detection files to extract from them all the frames containing a person and we stored all in the same folder structure as before *country/city/video/image.jpg*. Since all these files are very consuming in terms of memory, we did all this process on the cloud. In order to use the dataset for training on our machine, we had to store it in local. In order to do that we

used LMDB, a high-performance key-value embedded database, that allowed us to store all the dataset in a single compressed file and to download it on our device. In total we downloaded approximately 20k videos, and from them we extracted more than 1 million images.

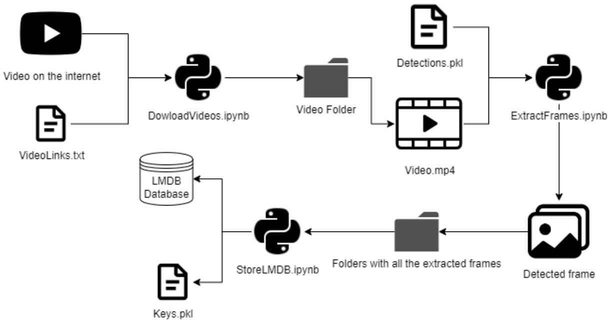


Figure 5: Diagram of the dataset creation process.

6 Pre-traning

The pre-training process is designed as follows: A query image is selected from the created dataset and processed by the encoder network to compute the encoded query image q, then is compared to multiple mini-batch of encoded key images.

The error is formulated by the following contrastive loss function:

$$L_q = -log \frac{exp(qk_+/\tau)}{\sum_{i=1}^{K} exp(qk_i/\tau)} \tag{1}$$

This type of loss allows the model to learn a smaller distance for views from the same image and a larger distance between different images, resembling a softmax-based classifier loss.

Then we adjusted the weights executing the validation on the Market1501 dataset.
All the pre-training process was performed in a "fake" distributed way even if we used only one GPU, this due to the fact that the MoCo model used, works only in this type of execution. However this wasn't totally useless since it speeded up the whole process, allowing us to have our pre-trained weights in a shorter amount of time. The training was divided into several sessions since the huge amount of data. In total the pre-training reached 80 epochs. By using resnet50 as backbone of the model and a mini-batch size of 64, the pre-trained mAP converged around 50%. However, not much importance should be given to precision in this phase since it is only preparatory to the initialization of the weights for the finetuning of the model.

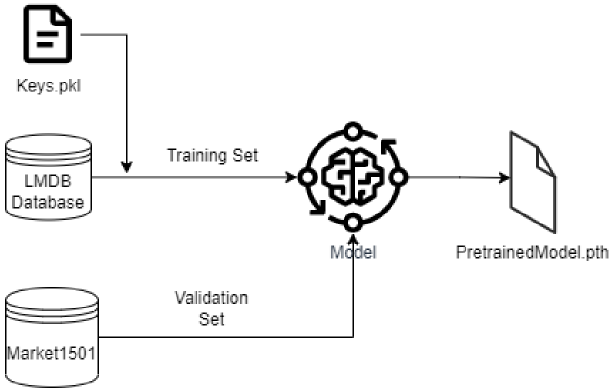


Figure 6: Diagram of the pre-training process.

7 Momentum Contrast

As neural network model for the pre-training, we used MoCo, a model that uses the Momentum Contrast [3] technique as method of training. It is a self-supervised learning method that fits perfectly in the task of pre-training. The hypothesis is that good features can be learned by a large dictionary that covers a rich set of negative samples. This is implemented augmenting an image sample in two ways, in order to create two different samples of it. One of the samples is used as query (q) and the other one as positive sample (k). The rest of samples not augmented from the same image are used as negative sample.

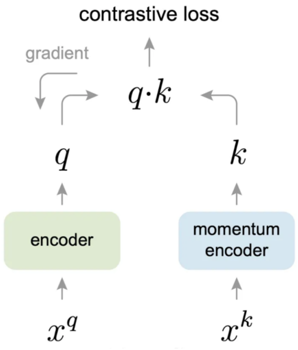


Figure 7: An explanation of how query image q and key k are processed.

8 Transfer learning and training

Transfer learning consists of retraining a neural network on which a large data set has already been inserted, usually for the purpose of classifying large-scale images. Transfer learning has several benefits, but the main advantages are saving training time, better performance of neural networks, improving both skill usage and performance outcomes. After the pre-training process, we trained the fast-reid [4] model with the new transferred weights on the Market-1501 dataset for 24237 iterations. For this task we used the train set and the validation set of the Market1501 dataset.

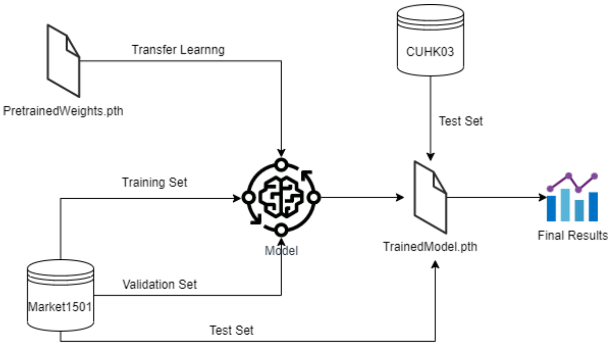


Figure 8: Diagram of the model training process.

9 Evaluation

The model was finally evaluated both on the testing set of Market1501 and the CUHK03 dataset. As metrics to evaluate its performances we used the followings:

9.1 mean Average Precision

To have a general view and a good starting point for comparison we used mAP as one of the evaluation metrics for our model. So, given a set of queries, each one of them is compared to a set of the gallery, and the model is expected to retrieve the correct identity from it.

9.2 CMC Rank

The second evaluation metric that we used to evaluate the model is the Cumulative Match Characteristic rank, one of the most used in the field of person re-identification. This metric calculates the percentage of queries for which the correct match appears into the top-k objects into our gallery. In our case we set the k to 1, 5 and 10. This metric gives us a view about how well our model can retrieve the correct match at different rank positions.

9.3 ROC

We used the Receiver Operating Characteristic in order to have a visual representation about how well the model is able to distinguish true positive pairs from false positive ones.

10 Results

After the evaluation we noticed, as expected, that the performances on the Market1501 dataset are better than the ones on CUHK03 dataset. This can be easily explained through the fact that even if the model has never seen the testing images of the first dataset, it learned some biases during the training, since the images provided in the test set are captured in similar conditions of the ones of the training and the validation set. On the other hand, even if we expected slightly worst results on the never seen dataset, the actual results, worried us a little bit. After that, however, also observing other works in the literature, we noticed that it is quite common for the models to have worse results on the CUHK03 than on the Market1510 where they are both used for comparison.

Datasets	Rank-1	Rank-5	Rank-10	mAP
Market1501	93.91%	98.07%	98.93%	82.68%
CUHK03	72.88%	88.40%	93.83%	68.58%

There instead, we can see the ROC results:

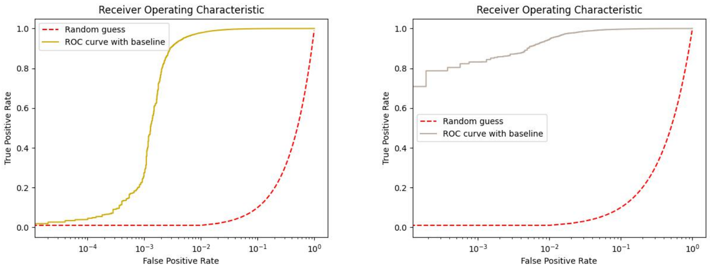


Figure 9: Respetctively the ROC performed on Market1501 and CUHK03

We plotted them using the semilogx function of Matplotlib instead of the classical plot one. This in order to enhance the difference between them, since with the other one the plotted curve was very similar since the way in which sklearn computes the FPR.

11 Demo

Our demo gives the opportunity to upload a video to an interface, implemented using Gradio, a python library that consent to develop graphic content and associate them function and external call.
The User interface of our demo is structured in multiple tabs that represent a phase of a biometric system pipeline : Enrollment, Verification and Identification.

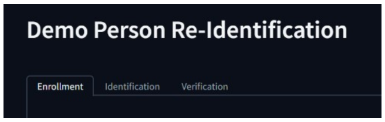

Demo Person Re-Identification

Figure 10: Main tabs of UI

11.1 Extract persons from sequences of frames

Using the following object detection model [5], we extract each frame in which a person occurs, cropping it in its bounding box. Then we built a folder structure containing a directory for each video that represent our subject, and relabelled the identities according to the following format:

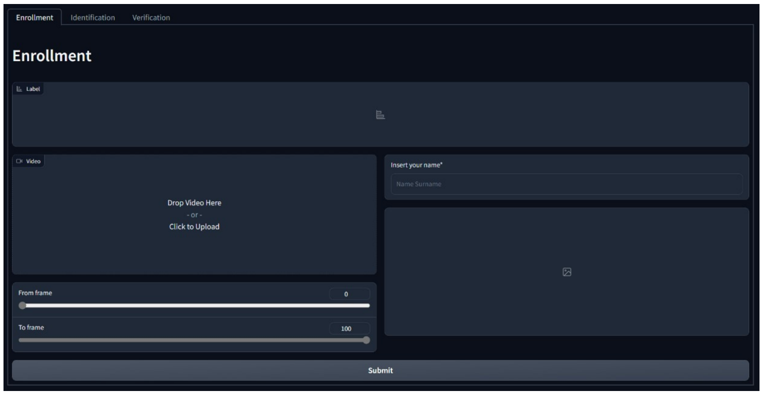
$$NumFrame\_BB.jpg \tag{2}$$

- **NumFrame**: the number of frame, eight character that represent where the subject is captured.
- **BB**: Bounding Box, two characters that represent the number of people detected into the frame in case of multiple detection.

11.2 Extract features from an image

To make use of the trained model for person re-identification, we first utilized its knowledge to extract features from each frame in the query and gallery images. These images were in the same format as the training and evaluation phase, allowing us to effectively leverage the model's capabilities. Next, we made predictions on the extracted features and used the results to calculate the cosine similarity between the query and gallery images. The cosine similarity score is a measure of the similarity between two vectors and can be used to determine the degree of similarity between the query and gallery images. Finally, we used the results of the cosine similarity and a predefined threshold to determine whether or not there was a match between the query and gallery images. This process allowed us to efficiently and accurately identify individuals within the images.

11.3 Enrollment



From frame 0

To frame 100

Figure 11: Enrollment interface

As we can see in the above picture, the main parameter of the form is the **Video** input; thanks to this upload field we can choose the media where we want to extract the frame regarding a new subject to store into the gallery.
We added a feature that consent us to define a time interval to specify a single part of the entire video; to use this functionality we must specify a **From Frame** and a **To Frame** value. This integration involves a reduction of time and power of extraction.
After uploading the source, a name must be specified to associate all the new frames to the corresponding identity. When all the required information are provided, the code can be run using the submit button and execute the enrollment process.
At the end of execution, the user interface return two response:

- Completed message, to notify the user about the correct enrollment
- Gallery image, containing a sub-set of all captured frames

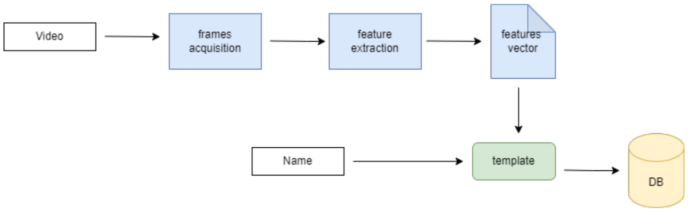


Figure 12: Scheme of the enrollment flow.

11.4 Verification

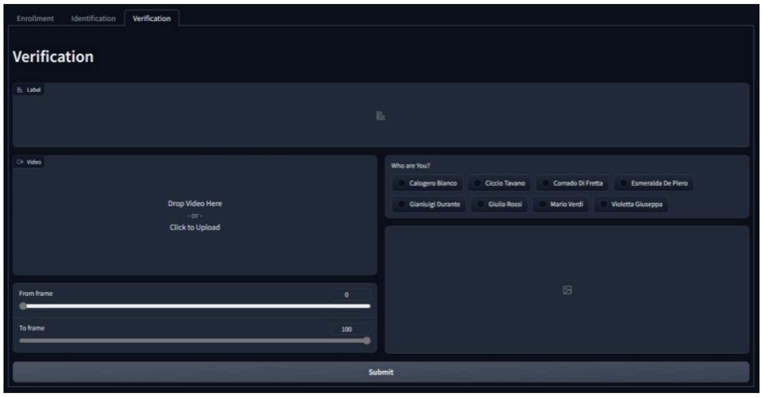


Figure 13: Verification interface

Verification Interface is very similar to the Enrollment Interface, the first difference that we can see is the input field that suggest all the current Identities stored in the dataset.
The user, given a video content, claim an identity selecting the interested representative radio-box.
The first feedback that will be shown will be a set of samples belonging to the claimed identity. Also in this case the process can be run using the Submit button and, at the termination of it, will show a massage that notify about the correct or failed matching.

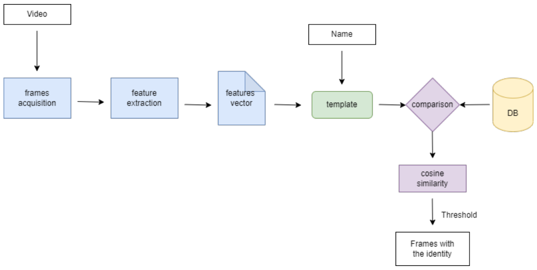
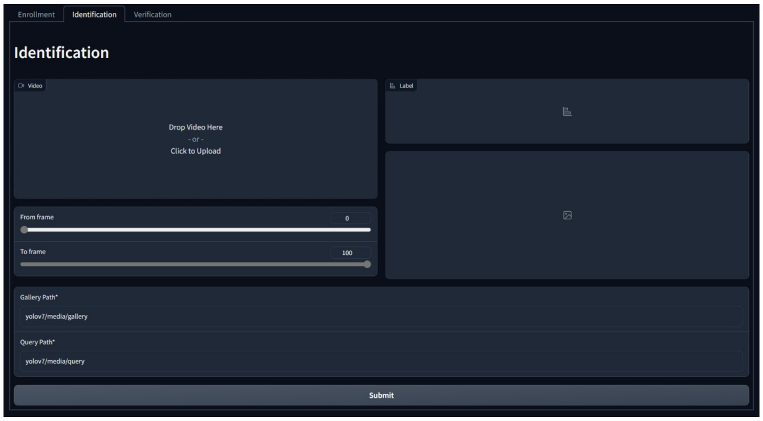


Figure 14: Scheme of the verification flow

11.5 Identification



From frame: 0

To frame: 100

Gallery Path*: yolov7/media/gallery

Query Path*: yolov7/media/query

Figure 15: Identification interface

Also for the Identification Interface the main parameter of form is the **Video** input; thanks to this upload field we can choose the media that we want to split into different frame, collecting the detected people.
The paths to the **Query** and **Gallery** can be inserted in the form below, both required but compiled by default with the pre-defined location.
After this acquiring process, which consent to extract from the video uploaded all the people detected and collects them storing in the declared dataset, the identification process starts comparing the new frames and the already existing gallery ones.
When the program ends, prints in the output sections a response that could be of two different types:

- Detected response
- Failed response

The first one occurs when, giving a threshold, some matching return a similarity higher than this value; we can assume this situation like a good response that express a Correct Detected. With this response the User Interface will show a ”Good news message” and the new frames extracted. On the other side, will be show a ”Bad news message” and a failure image in the output gallery.

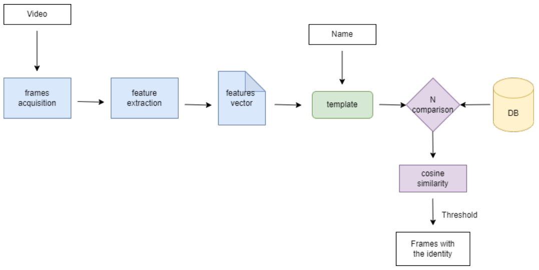


Figure 16: Scheme of the identification flow.

12 Conclusions

The aim of this report was to show how an unsupervised technique, applied to a very large scaled dataset, can be used to improve the results of the current person re-identification methods developed. Experiments demonstrated the effectiveness and generalization ability of our pre-training model, to address the limited size of the existing dataset.

One of the difficulties that we have found is surely the high computational power required by this type of process. We were able to overcome a part of these problems using cloud systems like Collab pro, that provide some powerful GPU, shared notebooks and unlimited runtime execution, despite we encounter some issues for long-term training even with this solution.
Certainly, there are lots of improvements that could be done to our implementation, for instance, performing the validation executed during the pre-training process on a split of another dataset, other than those already mentioned. A further step in a better direction could testing the transferred weights to other models, such as TRIP [6] or IDE [7].

References

[1] D. Fu et al., *Unsupervised Pre-training for Person Re-identification* 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 2021, pp. 14745-14754, doi: 10.1109/CVPR46437.2021.01451.

[2] Zheng, Liang and Shen, Liyue and Tian, Lu and Wang, Shengjin and Wang, Jingdong and Tian, Qi *Scalable Person Re-identification: A Benchmark* 2015, IEEE International Conference on Computer Vision

[3] K. He, H. Fan, Y. Wu, S. Xie and R. Girshick, *Momentum Contrast for Unsupervised Visual Representation Learning* 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 2020, pp. 9726-9735, doi: 10.1109/CVPR42600.2020.00975.

[4] He, Lingxiao and Liao, Xingyu and Liu, Wu and Liu, Xinchen and Cheng, Peng and Mei, Tao, *FastReID: A Pytorch Toolbox for General Instance Re-identification* 2020, arXiv preprint arXiv:2006.02631.

[5] Wang, Chien-Yao and Bochkovskiy, Alexey and Liao, Hong-Yuan Mark, *Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors* 2022, arXiv preprint arXiv:2207.02696.

[6] Dengpan Fu, Bo Xin, Jingdong Wang, Dongdong Chen, Jianmin Bao, Gang Hua, and Houqiang Li. *Improving person re-identification with iterative impression aggregation.* IEEE Transactions on Image Processing, 29:9559–9571, 2020.

[7] Liang Zheng, Hengheng Zhang, Shaoyan Sun, Manmohan Chandraker, Yi Yang, and Qi Tian. *Person re-identification in the wild.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1367–1376,2017.