Person Attributes Recognition and Re-Identification — Deep Learning Project

Simone Caldarella
simone.caldarella@studenti.unitn.it

Federico Pedeni
federico.pedeni@studenti.unitn.it

Gaia Trebucchi
gaia.trebucchi@studenti.unitn.it

University of Trento

Abstract

This document represents the report of the final project for the Deep Learning 2020-2021 course. The goal of the project was to implement a two-stage pipeline in order to solve both Person Attributes Recognition (PAR) and Person Re-Identification (Re-ID) tasks. In this report will be discussed both the theoretical background and the main implementation details followed by an analysis of the results obtained. Finally, the problem of the unbalancing of the dataset will be exposed and the techniques tested in order to cope with it will be presented.

1. Introduction

Person Attribute Recognition and Person Re-Identification are two different tasks that share common points both at semantic and implementation levels. While the first task aims to predict a set of attributes given an image of one or more people (usually pedestrians) the second one aims to find the identity of a person given an image (query) and assuming to have a set of images from where to find its identity (gallery). More specifically, we suppose that the gallery contains a small set of images for each identity available, and based on a similarity measure the query image is associated to a subset of images of which we know the identity. Since images from the same identity should share the same attributes and thus be close to each other, the underlying goal of both task involves the computation of a well discriminative image embedding that pushes together images of the same person while pushing far images of other people.

In the Data section will be discussed the dataset used, with an analysis on labels statistics in order to highlight data correlations and unbalancing.
In the Methods section will be explained the solutions developed to solve the two tasks, with description of the models and their components.
In the Results section will be shown the performances reached in several tests, for both the stages, along with the hyperparameters used, for replication purposes.
Finally, in the Conclusions section will present final remarks about the works done followed by considerations about future modifications that could lead to higher performances.

2. Data

The dataset used for both tasks is Market-1501 [6], which contains 32668 images collected in front of a supermarket, in Tsinghua University, using 6 different cameras. It is composed of 12936 train images, 19732 test images and 3368 *query* images with a total of 1501 different identities, taken at least by two different cameras. It is also important to notice that train and test sets include disjoint sets of identities, respectively 751 and 750. The query set is built using identities taken from test set and the gallery contains all the images available for each identity in the query.
For the first task we split the train set into two folds (60-40), keeping all the images of the same identity in the same split, obtaining a validation set. The separation of identities between the two set is done in order to avoid knowledge contamination and biased performances. Regarding the second task, we relied on the same split produced for the task 1, in order to allow a fair transfer learning from Classification to Re-Identification. Annotations provide a set of visual clues, related to each pedestrian, that range from age to upper and lower body clothing and carried objects.The dataset was also modified by the addition of the multi-color class both to up-color and down-color labels.
During an initial analysis of the dataset we discovered a strong unbalancing between classess of several labels. While the techniques tried to overcome this issue will be explained in the following sections, in this section we would like to show some of the statistics collected from the dataset in order to better assess the level of unbalancing.
As we can see from the Table 4, there are classes, in most of the labels, that have a very low representation in the dataset. For example "Old" and "Young" classes in Age have a very low value in percentage, if compared with "Teenager" (respectively 1.4%, 0.8% and 82%). As another example, regarding Up and Down color distributions, we have unbalancing towards "Black" class in the Downcolors (38.8%) and "White" class in the Upcolors (28,4%). Unbalancing is

present also in binary classes such as Hat, where the negative class ("No Hat") totally overcome the positive one, with a presence of more than 97%. Finally, we decided to employ two data augmentation techniques in order to increase the variability of the samples during the training phase. First, we apply *Random Horizontal Flip* in which images are randomly mirrored; secondly we apply *Random Rotation* with a maximum degrees of $10°$.

3. Methods

In this section will be explained, separately, all the details regarding the pipelines of both tasks along with motivations behind choices of specific components and tricks used to stem the issue of unbalancing. Moreover, a common point between the two tasks relies on the model used as backbone, *ResNet18*, which is a specific implementation of the famous CNN model ResNet [2].

3.1. Person attributes Recognition Task

Person attributes Recognition task aims to recognize a set of attributes, related to an input image, in a multi-label multi-class fashion. As anticipated before, we built a custom model starting from a ResNet18 adding on top of that an ad-hoc multi-branch classifier, composed by a different Sequential layer for each attribute to be classified. In this subsection will be explained our implementation along with the different loss functions used to overcome unbalancing in data.

3.1.1 Standard Pipeline

To build the first pipeline, we started importing the ResNet18 pre-trained model provided by *Pytorch Models* module and keeping all the layers trainable. Then we replace the fully connected module (output layer) with our *Multi-Attribute Classifier*. The custom module (Fig. 1) consists in a Sequential module, composed by a *Dropout* and a *Linear* layer, for each of the attributes to be predicted. In this way we expect that each branch will extract, using its own weight matrix, a more specific set of features rather than using a single branch for all attribute.

3.1.2 Deal with Unbalancing

After having analyzed the dataset, we decided to try several methodologies in order to tame the strong unbalancing.

First of all we compute the percentages representing the relative occurrency frequencies of each class for every label and use their inverses as *weights* for the *Weighted Cross Entropy* (sometimes called balanced) loss. The simple yet effective idea behind WCE is to rescale the loss computed over a sample with the weight associated with its groundtruth (correct) class. In this way, less importance

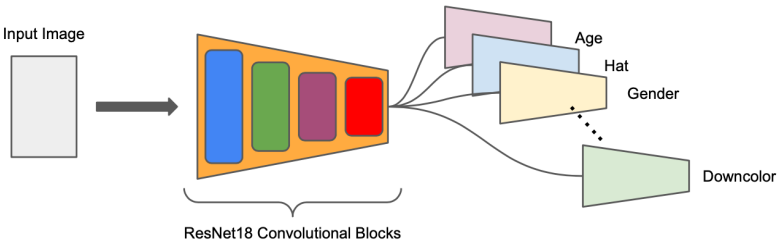


Figure 1: Person Attribute Recognition model

is given to more frequent classes and thus, computing the multi-label loss results in a more balanced value.
Secondly, scraping the literature, we found out about *Focal Loss* [3], which promised to deal robustly with stronger unbalanced data. This new loss is defined as:

$$FL(p_t) = -(1-p_t)^{\gamma}\log(p_t)$$

and the key point that distinguishes it from the Cross Entropy (both weighted and non-weighted) is the addition of the modulating parameter $(1-p_t)^{\gamma}$, with γ as hyperparameter, that allows to reduce the weight for already confident correct predictions, and on the other hand enhance the loss of the lower confident (supposedly wrong) predictions.
As expected, we could also merge the two methods, obtaining the so called α-balanced Focal Loss, defined as:

$$FL(p_t) = -\alpha_t(1-p_t)^{\gamma}\log(p_t)$$

where we combine the weights of Weighted Cross Entropy with the modulating factor of the Focal Loss, in order to obtain a more robust loss.
However, despite the interesting ideas, we will see in the Results section that the above methodologies are not a panacea and cannot totally eradicate the issue. Moreover, looking at the structure of a Focal based loss, we can realize how it cannot deal with overconfident wrong predictions, thus leading to less weight to the worst cases and a bad training regime.

3.2. Person Re-Identification Task

The goal of Person Re-Identification is to find, given a query image, a set of all the images that depict the same person in the query. The database where the set of images is searched is called gallery, and the images in gallery are supposed to be labeled with the identity of the person depicted. A simple use case of Person Re-ID could be trying to indentify a person by retrieving all the images in the database resembling that subject.
Person Re-ID is still a very studied topic and various works have been published in the last five years. To solve the task, we decide to implement a Siamese Network [1] trained minimizing a Triplet Loss in order to force the model to learn embeddings that are close for images depicting the same identity and far for images related to different identities.

3.2.1 Triplet-Loss

Triplet Loss is a metric introduced in the work by Schroff et al. [5] for face recognition task. The idea behind this loss is to force a Siamese Network to learn embeddings keeping in account their relative distances. The concept of creating distance aware embeddings is already well known in fields such as Natural Language Processing, where words are mapped into a learned embedding space in which they are arranged to be closer if semantically related. In this case, the difference is given by the nature of the formulation of the problem. Indeed we are not explicitly creating clusters, but instead we force the model to keep closer samples of the same semantic category and push away samples of different semantic categories. More formally, Triplet Loss is expressed as:

$$L = \sum_{i}^{N} \left[\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha \right]_+$$

where, for each iteration i we have x_i^a, called *Anchor*, x_i^p, called *Positive*, x_i^n, called *Negative*. Our goal is to ensure the closeness between Anchor and Positive, while moving away Negative from Anchor. From the above formula we can see that we want to minimize the difference between the relative distances of Positive and Negative from Anchor. Doing so, the loss will be zero if the Positive embedding is closer than the Negative one to the Anchor; otherwise the loss will be equal to the difference between the distances. Moreover, a parameter $alpha$ is introduced to avoid the network to learn trivial embeddings, such as vectors full of zeros for every sample.

3.2.2 Clustered Dataset

Despite the effectiveness of the Triplet Loss, a key point in training with such loss is the choice of each *triplet* given in input to the model. To enforce the learning process is fundamental to create a dataset of "enough complex" triplets to be learnt. It is trivial to notice that, for very different images, an average model would easily learn embeddings in order to minimize the loss, without having learnt the right semantics, thus reducing the generalization power. To ensure this step, we decided to build a custom dataset, that creates clusters of images in the dataset based on image histograms; in this way we easily create coarse clusters of visually similar images. With this simple trick we were able to construct triplets such that after having chosen a random image as anchor, and having obtained one positive, we can sample the negative in the same cluster of the anchor, thus making the learning process harder.

3.2.3 Siamese Network Model

A Siamese Neural Network (SNN) is a type of Neural Network that process in tandem two inputs in a shared fashion. It is generally used for comparisons of couple of embedded input in order to generate clustering in a supervised setting. We decided to develop our custom Siamese Model starting from the model trained for the first task, but only up to the last classification layer, in order to keep a higher dimensional embedding space. Then, we added on top of this backbone a ReLU and a Fully Connected layer to better control the final dimension of the embeddings.
Finally, to train the model minimizing the Triplet Loss function, we trivially employed the same model at each step to compute both Anchor-Positive and Anchor-Negative couple embeddings.

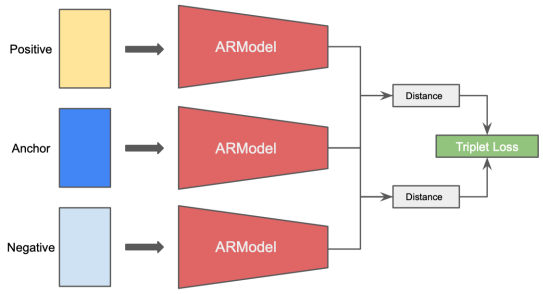


Figure 2: Person Re-ID Model

4. Results

In this section will be shown the results obtained in the performed tests, for both tasks, along with comments on performance variations related to the different parameters and architectural choices.

4.1. Person attributes Recognition

Regarding the first task, beyond the standard multi-label recognition with vanilla Cross-Entropy, we tested 3 different loss functions: Balanced Cross-Entropy, Focal Loss and α-Balanced Focal Loss. Moreover, for comparison purposes, we choose the best set of hyperparameters found empirically and keep them equals for all the tests. Respectively we did the tests with a learning rate of $3e-4$ and a linear schedular with warmup equal to 10% of the total number of training steps for 15 epochs (with early stopping). The metrics taken in account are Accuracy and F1.
Despite the motivation behind the different losses, explained in the section 3, from the results tables (2) and (3) we can see that the model obtained higher performances when using the two versions of the Cross Entropy. However, two considerations should be taken in account to better understand the results. First, Focal Loss and its variant are probably facing the issue of overconfidence of network predictions, thus reducing their power in treating unbalanced data. Second, validation set is sampled randomly from the original train set, thus it keeps the unbalance itself. Moreover, looking at the results obtained by the two Cross En-

tropy variants, we can also see that they tend to underline the original distribution of the dataset.
In light of the results obtained, we can state that we didn't overcome the unbalancing problem, probably because of the strong unabalancing also present in validation mixed with the overconfidence of the network, but it would be interesting to validate our trained model on a more balanced test/validation set.

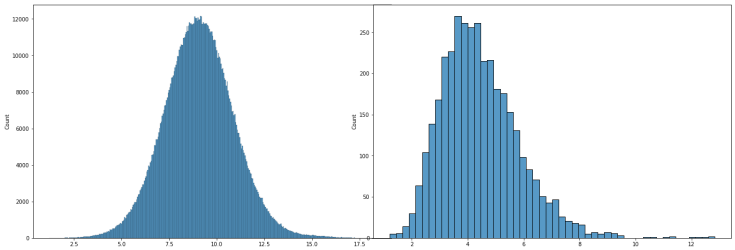
4.2. Person Re-Identification

Concerning the Person Re-ID, we decided to test it in the two variants we implemented it. In the first case, we use a pre-trained ResNet18 without any information from the first task. In the second case, we trained the Siamese Model using as backbone a modified version of the model trained for the first task. In this way, thanks to the knowledge sharing between the two learnings, better results are expected. The metric used to assess the results is the so called *mAP*, which stands for mean-Average Precision, that will be explained in the next subsection.

4.2.1 Mean-Average Precision and selection heuristics

Mean-Average precision is a metric mostly used in computer vision applications for object detection and segmentation tasks. Formally, it represents the mean between all the Average Precisions for each class, where the Average Precision represents the area under the Precision-Recall curve. The idea behind the mAP applied to the Re-ID task relies on the fact that during the retrieval phase the order on which images are retrieved should play a fundamental role on evaluation. For example, we could retrieve all the images in the gallery set, but having the correct ones in the first positions would be certainly better that having them in the last positions. In this way, mAP introduces a weight inversely proportional to the position of the correct retrieved images thus leading to higher values when they are retrieved before than the wrong ones.
Moreover, in order to obtain better results, filtering algorithms could be employed in order to have less images retrieved, reducing the decreasing of the weight in the mAP computation. To do so, we implemented two different techniques to select the images to be retrieved. First, we add a distance based threshold choosed from the statistics of the distances for all the galleries and the queries. More specificly, we empirically found out that the value of distances of all the correct images from each query where averagely less than the mean of all the distances from the query. This reasoning could be easily spotted in the Fig. 3, where the Gaussian representing the distances of the correct images in the gallery from every query lies in the left part of the Gaussian representing all the distances from every query.

Second, we apply a filtering based on a sort of Hamming distance related to number of wrong attributes predicted.



(a) Total (b) Corrects

Figure 3: Total and Corrects distances distributions

In this way, assuming that images from the same identities should share the same attributes, we employ a model trained on the task 1, in order to predict attributes for each element of the gallery and at each step for the query. So, if the distance between the query and the gallery image is greater than a parameter, to be fine-tuned, the gallery image is rejected.

4.2.2 Re-ID results

For the task, we trained a model using transfer learning from the first task, while another model without employing it. Both model were trained for 10 epochs, using early stopping, a linear scheduler with warmup equal to 10% of the total number of training steps and with a learning rate of $2e-5$. As we can se from the Tab. 1, the impact of transfer learning is astonishing, leading the network that employs it to reach a mAP two times higher than the one that doesn't use it.

W/out Transfer Learning	W/ Transfer Learning
21.85	44.78

Table 1: mAP for the two configurations.

5. Conclusions

We implemented a two-stage pipeline that try to solves first Person Attribute Recognition task and second Person Re Identification. As seen in the Results, the obtained performances are encouraging but method still struggle to reach a better generalization in the unbalanced setting, despite the methodologies tested.
Future implementations could be the introduction of convolution operations for each branch in the task 1 and detection methods in order to feed specific branches with more refined ROI. Finally, as suggested in the work [4] by Lin et al., train the two tasks jointly could increase the process of information sharing, thus leading to higher performances in both tasks.

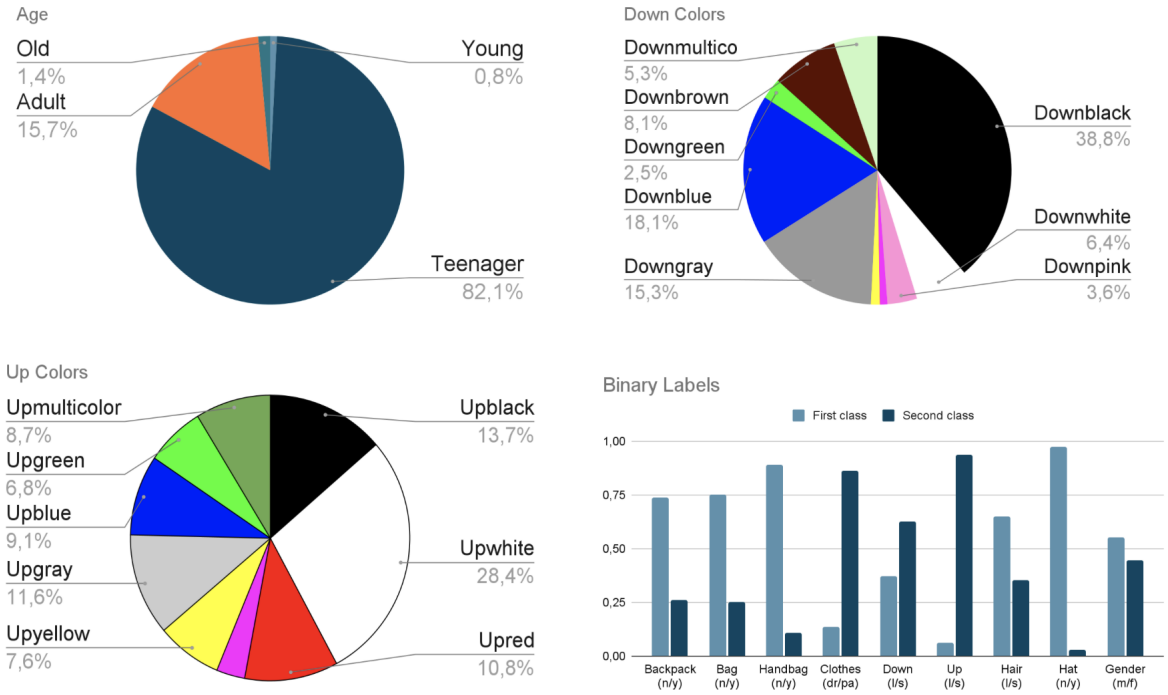


Figure 4: Dataset statistics

Label	Vanilla CE	Weighted CE	Focal	AlphaFocal
Age	79.84	75.72	80.31	55.54
Gender	86.45	87.32	85.32	86.54
Hair	81.66	84.88	81.77	82.91
Up	93.28	92.47	91.64	86.89
Down	87.19	90.13	88.40	89.29
Clothes	92.78	92.82	90.60	91.90
Hat	94.84	95.13	96.34	88.96
Backpack	82.83	81.79	80.26	81.94
Bag	72.40	70.02	70.10	70.11
Handbag	85.68	81.79	83.10	78.90
Upcolor	68.96	70.22	71.87	69.49
Downcolor	65.65	63.28	58.97	13.13
Global	82.63	82.13	81.56	74.63

Table 2: Accuracy scores (%) for different losses.

Label	Vanilla CE	Weighted CE	Focal	AlphaFocal
Age	36.86	36.66	43.20	42.35
Gender	86.16	87.06	85.29	86.40
Hair	79.12	83.09	80.60	81.02
Up	63.72	59.75	59.33	60.99
Down	86.74	89.58	87.39	88.70
Clothes	81.70	83.93	79.37	82.34
Hat	63.84	67.77	62.56	58.83
Backpack	78.34	77.51	74.31	77.81
Bag	60.63	61.14	60.08	62.36
Handbag	52.83	57.29	54.78	54.86
Upcolor	64.77	63.29	65.97	62.10
Downcolor	50.24	56.62	40.85	11.29
Global	67.08	68.64	66.14	64.09

Table 3: F_1 scores for different losses.

References

[1] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546 vol. 1, 2005. 2

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 2

[3] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection, 2018. 2

[4] 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*, 95:151–161, Nov 2019. 4

[5] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2015. 3

[6] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *Computer Vision, IEEE International Conference on*, 2015. 1