Deep Learning Course Assignment

Alessio Dellsega Vincenzo Marco De Luca Matteo Zanella

1. Introduction

The first objective of this project is to perform multi-class classifications on multiple attributes of pedestrians taken from the *Market-1501* dataset. The second one is to perform person re-identification (re-ID) on the dataset.

The person re-ID task aims to use a query person image to retrieve all the other pictures of the same person, taken from different cameras angles, from a dataset containing several people and distractors. The training dataset presents the person ID of each picture.

The attributes recognition task is a multi-task classification problem, as the individual attributes to predict (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) are different classification tasks which share the same high-level visual features. The training dataset presents the set of correct attributes of each picture.

2. Proposed Solution

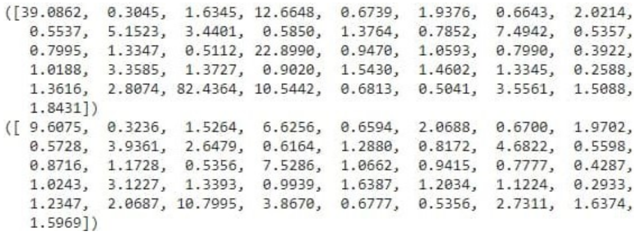
In order to solve the objectives of the project, we investigated the state-of-the-art solutions, to combine the most effective techniques from literature.

Person re-ID is typically addressed by learning image embeddings to be fed into a person ID classifier to encourage the identity recognition during the training phase. Then, in the retrieving phase, the classifier is discarded: each query picture is associated with the nearest images in the embedding space. The multi-task attributes classification is usually solved with a shared backbone that produces high-level embeddings and feed them into multiple parallel classifiers heads, one for each attribute.

Since both person re-ID and attributes recognition tasks are based on similar high-level visual features, it is reasonable to consider the re-ID as an additional task in the multi-task setting: the shared backbone generates a global image embedding that is used by the attributes classifiers and for the re-identification. Multi-task learning does not just improve performances, but also works as a regularization technique that reduces the network overfitting, which represents a severe complication is our scenario where the available samples are fairly limited.

2.1. Dataset preparation

We divided the annotated dataset with a 90/10% split between training and validation. In the random splitting, we put the same-person samples in the same split and adopted a heuristic approach to balance the classes present in each split: splits keep track of unseen classes and the randomly extracted people are added to the split where they maximize the novel classes, resetting the split unseen classes when all classes are seen. Since the dataset was extremely unbalanced, we used a `WeightedRandomSampler` to extract more frequently from the `DataLoader` the samples presenting low-frequency classes, and computed the expected class weights for the cross-entropy. Figure 1 shows that resulting classes are less unbalanced.



```
([39.0862,  0.3045,  1.6345, 12.6648,  0.6739,  1.9376,  0.6643,  2.0214,
   0.5537,  5.1523,  3.4401,  0.5850,  1.3764,  0.7852,  7.4942,  0.5357,
   0.7995,  1.3347,  0.5112, 22.8990,  0.9470,  1.0593,  0.7990,  0.3922,
   1.0188,  3.3585,  1.3727,  0.9020,  1.5430,  1.4602,  1.3345,  0.2588,
   1.3616,  2.8074, 82.4364, 10.5442,  0.6813,  0.5041,  3.5561,  1.5088,
   1.8431])
([ 9.6075,  0.3236,  1.5264,  6.6256,  0.6594,  2.0688,  0.6700,  1.9702,
   0.5728,  3.9361,  2.6479,  0.6164,  1.2880,  0.8172,  4.6822,  0.5598,
   0.8716,  1.1728,  0.5356,  7.5286,  1.0662,  0.9415,  0.7777,  0.4287,
   1.0243,  3.1227,  1.3393,  0.9939,  1.6387,  1.2034,  1.1224,  0.2933,
   1.2347,  2.0687, 10.7995,  3.8670,  0.6777,  0.5356,  2.7311,  1.6374,
   1.5969])
```

Figure 1. The class weights for the unbalanced training set (first tensor) and the balanced one (second tensor)

To cope with the limited dataset we also used data augmentation: we applied to the training images random transformations such as rotation, color jittering, horizontal flipping, erasing, affine, perspective, and sharpness.

The people labels annotations have been transformed in the target form, single integer values in $[0, \mathrm{C})$ representing the correct class for each task. In the annotations, the colors for a single body part were split into several binary attributes. Since we noticed they were mutually exclusive, in the target form we condensed the colors as alternative classes for the body part with an additional multicolor alternative for the case where no color was correct.

2.2. Attribute Recognition

In our first attempts, we tried to solve the attribute recognition task without considering the re-ID part, training from scratch a MultiHead DenseNet architecture. The multi-task paradigm has been implemented with a shared backbone composed of DenseBlocks with specialized DenseBlocks heads for each sub-task on top of it. Each head was composed of a linear classification layer used to predict the correct attribute class. To improve the results, we also implemented Squeeze-and-Excitation blocks [4]. Despite our initial attempts, the results were unsatisfying because the small dataset and the large number of classes did not permit us to learn features representative enough.

To improve the results, we moved to pretrained solutions and started to perform also the re-ID with the same network. After some testing, we decided to use pretrained Resnet as multi-task backbone. The first `JointNet` had simple heads made of fully connected layers, later evolved in `JointConvNet` with heads composed of a 2D convolution followed by two linear layers.

2.2.1 Cost functions

For every attribute classification sub-task in the heads, we initially used a simple weighted Cross Entropy loss, later improved with a Focal Loss [6] to better handle the class unbalance. The class weights are computed during the dataset instantiation.

In the multi-task scenario, the sub-tasks losses need to be aggregated. The simplest solution is summing them together, like we did with the `UniformMultitask` wrapper. This approach does not allow assigning different weights to each loss based on the task difficulty. To overcome this problem, we implemented other wrappers from the literature: Loss-Balanced Task Weighting [8], Dynamic Task Prioritization [3] and Task Uncertainty Weighting [5]. Each of them adopts a different strategy to decide which task to prioritize in the loss backpropagation. After an empirical analysis, we found out that Task Uncertainty Weighting leads to the best results. This wrapper in particular weights the losses with some learnable parameters that are added to the optimizer.

2.3. Re-identification

In `JointConvNet`, the pretrained Resnet backbone produces the embeddings necessary for this task.

Many works suggest enhancing the backbone embeddings by concatenating them with the attributes predictions, and use the enriched embeddings for the re-ID task. We also tried using the Attributes Re-weighting module suggested in a recent paper [7], which multiplies the attribute prediction scores with the sigmoid of their linear transformation. Despite the literature evidence, we obtained the best results by simply using the normal embeddings instead of rich ones.

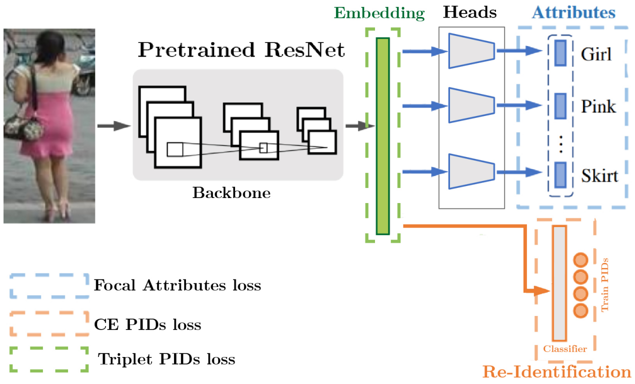


Figure 2. Final architecture

During the training, the embeddings are also fed into a linear classifier used to predict the PID of the person in the picture. This classifier is discarded in the validation and test phases, because its only objective is to reward embeddings with a recognizable PID. For the re-ID task, we produce the embeddings of the queries and the test set. Then we associate to the queries embeddings the test embeddings with a cosine similarity above a certain threshold. For setting the threshold value, we tried to take the average minimum similarity within each PID cluster, but at the end we resorted to a simpler fixed-value threshold.

In order to further improve the predictions, the presence of distractor and junk images in the test set should be taken into account. We suggest using a pre-filtering operation with a pretrained state-of-the-art image classifier like CoAtNet [2] to remove from the test set images which aren't classified as people.

2.3.1 Cost functions

The fundamental cost function for this task is a simple Cross Entropy loss applied to the predictions of the PID classifier.

The second auxiliary loss we applied with success is a Triplet Margin loss where the classes are the PIDs.

We also tried using a Centroid Triplet Margin loss [10], which computes the triplet loss with the PIDs centroids instead of directly using the embeddings samples. Another tried loss is the Center loss [9], which penalizes the euclidean distances between same-PID embeddings, and its evolution Island loss [1], which also penalizes the cosine similarity between the different-PID clusters. These last losses are not used in the final work because they weren't producing satisfying results.

Following the Task Uncertainty Weighting approach used for the attributes recognition, all the losses (Attributes Focal Multi-task, PIDs Cross Entropy, PIDs Triplet Margin) have been weighted with learned parameters.

3. Training

During training we used an **early stopping** procedure that updates the number of epochs in which the network does not improve and, if this number is greater than a patience threshold, the training is stopped and it is returned the network with the best performances before overfitting.

During the empirical analysis of the network, the performances have been visualized by means of Tensorboard for logging the metrics described in Section 4.

In addition to everything mentioned above, we also tried a **scheduler** offered by PyTorch in order to have a dynamic learning rate throughout the epochs. In our case, this procedure had not influenced the network performance positively.

Finally, we took advantage of the PyTorch **Automatic Mixed Precision**, which automatically chooses the precision for GPU operations to improve performance while maintaining accuracy.

4. Results

The performance of the network has been evaluated with three different metrics.

The Attribute Recognition task has been evaluated through both accuracy and mean average class accuracy. The first metric is the average accuracy of all the tasks using the default formula for accuracy both globally and for each sub-task, while the second metric we used is the mean average class accuracy (mAcc) to properly measure the model capacity to deal with unbalanced classes.

Whereas the evaluation of the re-ID task has been realized as the Mean Average Precision (mAP) metric.

The best performance in the re-ID evaluation reached by the network during validation is: 0.563 Validation mAP, at the same time the validation accuracy was 0.827 and Validation Average mAcc 0.668.

See Table 1 for full results for the accuracy on the attribute recognition task.

Task	Acc	mAcc
Age	77.8	41.7
Backpack	81.9	73.9
Bag	72.2	54.1
Handbag	91.7	55.4
Cloth	92.6	79.1
Down	93.9	93.1
Up	92.3	60.2
Hair	80.4	83.4
Hat	96.5	62.5
Gender	79.8	81.1
UpColor	69.0	66.2
DownColor	64.6	51.0
Average	**82.7**	**66.8**

Table 1. More precise report of the accuracies on attribute recognition

Task	mAP
Re-ID	**56.3**

Table 2. Person Re-ID Validation mAP

5. Conclusions

In this work, we analyzed the attribute and identity learning paradigm realized by means of a multi-task mechanism.

The most common feature extractors had been empirically analyzed with and without the help of pre-trained networks.

As we expected, feature extractors trained from scratch lead us to face overfitting phenomenon, addressed with pretrained extractors.

Despite this kind of backbone, the dataset dimension and its strongly unbalanced nature have led us to face overfitting again, at that point we simplified as much as possible our backbone, then we moved to less deep extractor (Resnet-18) and we explored multiple regularization techniques (e.g. weight decay, dropout, penalization based on class frequency and penalization based on class difficulty).

Furthermore we empirically explored multiple loss variations, as an improved version of Cross Entropy loss (FocalLoss) in Attribute Recognition task; whereas in re-ID task, we firstly attempt to use just TripletMarginLoss that did not produce the expected results, this motivates us to integrate attribute reweight to produce a richer embedding, Island Loss and Centroid Triplet Loss but, these two losses even more worsened the results despite the way we handled the order of their magnitudes.

A relevant improvement in re-ID and regularization of the two tasks has been introduced by the pid classifier and its associated loss (a simple Cross Entropy).

The final result we get has motivated us to think that most of the possible improvements have to be localized in the computation of the embedding similarity in the multi-task loss.

This conclusion is due to the fact that our network does not overfit soon, motivating us to think that the regularization techniques we integrated are able to handle the tasks but it seems like the right embedding distances are not learned as well and as fast as expected by us.

Another interesting idea we had at an architectural level to improve the performances is the Self-attention mechanism on the heads of both attribute recognition and re-ID task.

In addition to this, to improve the part related to the in-put data, we though to append GAN-based techniques for Domain Adaptation to inject newer data coming from other datasets in the network.

References

[1] Jie Cai, Zibo Meng, Ahmed-Shehab Khan, Zhiyuan Li, James O'Reilly, and Yan Tong. Island loss for learning discriminative features in facial expression recognition. *CoRR*, abs/1710.03144, 2017. 2

[2] Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. *arXiv preprint arXiv:2106.04803*, 2021. 2

[3] Michelle Guo, Albert Haque, De-An Huang, Serena Yeung, and Li Fei-Fei. Dynamic task prioritization for multitask learning. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 282–299, Cham, 2018. Springer International Publishing. 2

[4] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 2

[5] Lukas Liebel and Marco Körner. Auxiliary tasks in multitask learning. *CoRR*, abs/1805.06334, 2018. 2

[6] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *CoRR*, abs/1708.02002, 2017. 2

[7] Yutian Lin, Liang Zheng, Zhedong Zheng, Yu Wu, and Yi Yang. Improving person re-identification by attribute and identity learning. *CoRR*, abs/1703.07220, 2017. 2

[8] Shengchao Liu, Yingyu Liang, and Anthony Gitter. Loss-balanced task weighting to reduce negative transfer in multitask learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):9977–9978, Jul. 2019. 2

[9] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 499–515, Cham, 2016. Springer International Publishing. 2

[10] Mikolaj Wieczorek, Barbara Rychalska, and Jacek Dabrowski. On the unreasonable effectiveness of centroids in image retrieval. *CoRR*, abs/2104.13643, 2021. 2