Deep Learning Course Assignment

Cunegatti Elia
MSc Computer Science
Student ID: 223825
elia.cunegatti@studenti.unitn.it

Diprima Federico
MSc Artificial Intelligence Systems
Student ID: 224400
federico.diprima@studenti.unitn.it

Farina Matteo
MSc Artificial Intelligence Systems
Student ID: 221252
matteo.farina-1@studenti.unitn.it

Abstract

The aim of this project is to perform Classification and Re-Identification tasks on the famous Market-1501 dataset. We started from some simple and common methods in the Deep Learning community such as finetuning and feature extracting, using some well-known architectures like AlexNet and ResNet-. We then focused on keeping things simple yet introducing some innovative approaches, obtaining surprising results that we will show and discuss throughout this report.*

1. Introduction

Due to the huge advances in the Deep Learning and Computer Vision fields in the recent years, there is now an innumerable amount of tasks that the community can accomplish. In our work we have focused on Attribute recognition and Person Re-Identification, which we will try to introduce in this section.

Attribute recognition (which we will refer to as "*Classification*") consists of correctly recognizing a pre-defined set of attributes when analyzing an image coming from a specific application or domain. It can prove itself critical especially when dealing with pedestrians in video-surveillance scenarios.

In the same scenarios also Person Re-Identification can play a crucial role. With "Person Re-Identification" (which we will refer to as "*Re-ID*") we usually mean the ability, given a query image of a pedestrian, of retrieving every image depicting the exact same person within a large and heterogeneus gallery of images.

Although these may seem unrelated challenges we developed the Re-ID task leveraging the Classification task with clear benefits and thoroughness in the final results.

2. Classification

In this section we will discuss how we tackled the Classification task. Firstly, we will start providing a brief introduction of the Dataset structure and an overview of its main attributes. Secondly, we will explain how we organized that information in a way that was suitable for our needs. Then, we will describe our proposed architecture and solution as well as the underlying motivations that guided us. To conclude we will show and discuss the achieved results with comparisons to some state-of-the-art solutions.

2.1. Dataset

Figure 1. Sample images from Market-1501

Figure 1 presents some sample images from Market-1501. The image resolution is 128x64pixels with the annotations having 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

For additional information see [1].

2.1.1 Train, Validation and Test

We worked on two of the dataset folders: *train* and *test*. The *train* folder has been divided into two subsets: *train_directory* and *validation_directory*. In the process, images depicting the same people have been handled with care: in order to avoid misleading values and overfitting scenarios we imposed that each person-ID must appear in exactly one of the subsets.

Hence, we arbitrarily set a training-split to 80%, being this relative to the overall amount of person-IDs rather than the images themselves. This implies 80% of the people to be assigned to the *train_directory* while the remaining 20% composed the *validation_directory*. Another important implication is represented by the uncertainty concerning the **actual amount of images** in both folders since there is no fixed amount of images for each person-ID.

2.1.2 Data Augmentation

In order to foster the generalization capabilities of our classifier we decided to make use of some Data Augmentation techniques. Specifically the transformations we used are listed below:

- Horizontal Flip ($p = 0.2$)
- Erasing ($p = 0.1$)
- RandomRotation ($angle = 15deg$)
- ColorJitter on Contrast, Brightness and Saturation channels

Before feeding the networks with tensors, the latters were also normalized with the mean and the standard deviation vectors coming from ImageNet[2].

[1] https://github.com/vana77/Market-1501_Attribute

[2] https://image-net.org

2.2. Proposed Solution

We decided to start in a classical way approaching the problem trying to finetune some of the most widely used Deep Learning architectures including AlexNet[4], ResNet18, ResNet34, ResNet50[2] and DenseNet[3]. For each of these we replaced the final output layer with a custom one containing the exact same number of neurons as the attributes to predict.

2.2.1 Output Neurones and Losses

Here, it is important to point out the meaning of each of the output neurons. Since we captured different independent classification sub-tasks the output neurons can be interpreted by grouping them together as follows:

- **Multiclass Classification - Age Attribute**: the first four output neurons represented *young, teenager, adult, old* respectively. Since this is multiclass we decided to apply a Cross-Entropy loss on them.
- **Multilabel Classification - Independent Attributes**: the next nine neurons represented *backpack, bag, handbag, clothes, down, up, hair, hat, gender* respectively. Since this is multilabel we decided to apply a Binary-Cross-Entropy loss on them.
- **Multiclass Classification - Color of Upperbody (C.up) Attribute**: the following nine attributes represented *upblack, upwhite, upred, uppurple, upyellow, upgray, upblue, upgreen, upmulticolor* respectively. Since this is multiclass we decided to apply a Cross-Entropy loss on them.
- **Multiclass Classification - Color of Lowerbody (C.down) Attribute**: the following ten attributes represented *downblack, downwhite, downpink, downpurple, downyellow, downgray, downblue, downgreen, downbrown, downmulticolor* respectively. Since this is multiclass we decided to apply a Cross-Entropy loss on them.

Another critical aspect of the implementation concerned how to combine the aforementioned losses. Empirically, after having tried several weighting configurations, we noticed that a straightforward average was the most effective combination. Nevertheless, we were not satisfied enough with the results, especially with the *C.up* and *C.down* values, so we strived to come up with an original solution.

After some subjective observations about the current implementation as well as having reviewed some of the recent literature such as the paper by Wang et al. [9], we identified a few weaknesses. At first we understood that perhaps our architecture needed more capacity in order to cope with the different classification tasks it had to face.

Furthermore, we discovered that *C.up* and *C.down* were by far the most challenging classification tasks and would need some additional effort. For this reason we decided to implement a custom architecture based on ResNet34 that will be described in the next section.

2.2.2 Custom Architecture

Starting from ResNet34 as the backbone, we then appended four heads dedicated to the different classification tasks that we highlighted in the previous section. This was mainly to increase the network capacity, as anticipated previously.

For what concerns the *age* and *independent* attributes the respective heads are relatively simple, being made up of two linear layers each. ReLU was used as the activation function while both batch normalization and dropout were used for generalization purposes.

A bit more of an interesting design choice can be observed in the *C.up* and *C.down* branches. In order to face these challenging classification tasks we thought that including an Attention mechanism could bring several benefits. Our implementation is partially inspired by F. Wang et al. [10].

Attention Maps are built with the joint use of Transposed Convolutional Layers and Traditional ones, and are then combined with the output of the backbone via the Hadamard product (element-wise). Another interesting factor to underline is that Attention Maps are summed with 1 before performing the Hadamard product. This could seem unrelevant, but it has at least two very important implications:

- with this trick they do not serve as filters rather as feature selectors;
- avoiding zeros inside the Maps allows them to be stacked multiple times as they will essentially be **residual**[10].

Needless to say, this network showed significant improvements with respect to any other network that we have experimented with.

Figure 2 describes our implementation. To better view it, follow this link.

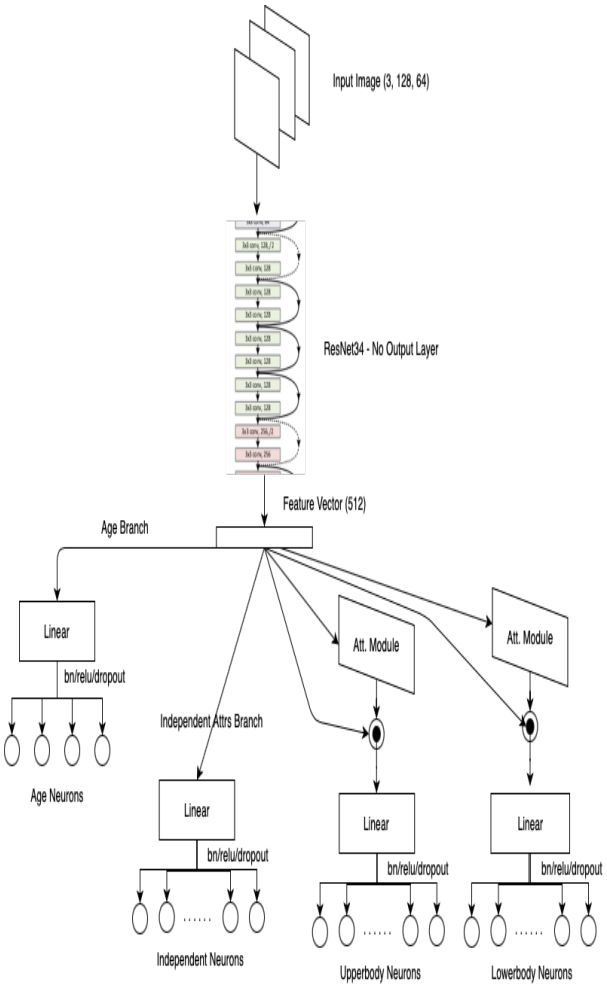

Figure 2. Custom Architecture

2.3. Results

Table 1 showcases the performances of our classification architecture. Before analyzing them it is very important to clarify the evaluation protocol of *C.up* and *C.down*. Regardless of the Multiclass approach used when training for these attributes, accuracy has been computed as in common Multilabel classification tasks. This is because we found a lot of uncertainty both in the literature and the repositories we reviewed concerning how to evaluate them. We went for the Multilabel evaluation approach as it was the most popular one, so that we could compare our results with the majority of others' works. The same accounts for the Age Attribute.

3. Re-Identification

In this section we will discuss how we approached the second task of this assignment. The aim of this task is to create a person re-identification model to retrieve from the test directory of the Market-1501 dataset all the images depicting the same person identity appearing in a given *query* image.

3.1. Dataset

The images in test and queries directory are unlabeled. We have no information about the person-ID, so we have to deal with an unsupervised task. For this reason, in order to have an idea of the performance of our solution, we started by creating our own set of queries from the validation set that we had previously used for the classification task, which contains information about the identities.

From now on, we will call *Gallery* the set of the remaining images of the validation set that have not been chosen to be query images. To measure the performance we decided to use the Mean Average Precision (mAP) that is the most common metric to evaluate Re-ID models. mAP compares the predictions to the given ground truth and in this type of problem it refers to the mean of the AP (Average Precision) over all queries.

In the next section we will explore in a detailed way our proposed solution.

3.2. Proposed Solution

We decided to start in a pretty simple way, performing a clustering on the identities of our *Gallery* directory. The first thing to do was to extract the features from each image to have the most compact representation as possible. To do that we decided to use our pre-trained classifier and to retrieve the 512 features obtained from the ResNet34 backbone for each *Gallery* image.

Ideally we thought that similar images should hold similar features. For this reason we initially performed a K-Means clustering on the feature vectors of the *Gallery*, setting K as the number of identities contained in our *Gallery*. In this way, we would hopefully obtain one cluster for each different person.

The next step was to extract features from the query images in the exact same way we had done for the *Gallery* images, so that we could perform the cosine similarity between each query feature vector and the centroid of each cluster. Each image belonging to the cluster with the highest similarity is then retrieved as a person-ID match for the given query. Computing the mAP we noticed that we obtained acceptable results, but they could still be improved.

Hence, we understood we could act on two main factors:

- The Clustering algorithm;
- extracting more significant features from the images.

Concerning the first point we tried different types of clustering and after various attempts we realized that a hierarchical Agglomerative Clustering (bottom-up) led us to better results.

Working on the second point, on the other hand, we thought that in order to embed more information inside our feature vectors we could also make use of the 32 predictions coming from the classification task. Therefore, we decided to use both the 512 features obtained from the fine-tuned Resnet34 backbone and the 32 attribute predictions, concatenating them together into a final feature vector with dimensionality equal to 1 and size 544 (512+32). This implementation was ispired by Lin at el. [5].

With this configuration we projected the feature vectors in a higher dimensional space, leading us to obtain surprising results in terms of mAP. We believe that the primary reason this worked lies in the increased dimensionality of our space: the more the dimensions, the more the distances between points. Thus, any clustering procedure would find it easier to group data points.

Once we were satisfied with the achieved performances, we applied this clustering algorithm on the test folder, to produce predictions for each *query* image in the queries folder. We knew that a total of 750 distinct person identities are present in the test set, so we decided to set to the number of clusters to 751, with the additional one aiming at collecting *junk* and *distractor* images.

3.3. Results

Table 2 compares our results to the latest SOTA techniques. We have been able to incrementally improve our results up to comparing them to Lin at el. [5] which was the baseline that guided us in the first place.

The mAP has been computed in two different ways: we first followed the assignment guidelines, using a portion of the validation set. We then tested our implementation in a more complex scenario using the whole training set as the *Gallery*.

The latter led to more inaccurate performances in the clustering procedure due to the increased amount of both person-IDs and *Gallery* images.

Even though we slightly improved our baseline we are still far away from the current SOTA performances.

4. Final considerations

We are satisfied with the performances on both tasks. Specifically we observed that our Attention Map implementation contributed to an improvement of $\sim$4%/5% of the *C.up* and *C.down* accuracies. We could also observe that without further training procedures decent Re-ID models can be implemented exploiting the former classifier, showing that, as anticipated at the beginning of the report, these are not independent tasks but can be tackled together. In the next section we will briefly explore where could we go next starting from this work.

5. Possible Improvements

- It could be interesting to exploit the residual Attention Maps, combining many of them at different layers from the ground-up;
- another possibility could be to jointly improve the classification and the Re-ID with a training pipeline that leverages also the ”ID” label, in order to make the network learn discriminant features while recognizing attributes;
- it would also be appealing to learn a re-weighting function to allow for feature weighting when computing

any pre-defined distance metric for the Re-ID task;

6. Code

We organized our work by means of a GitHub repository. If you want to have a look at the code, you can find it here.

References

[1] Yunpeng Gong. A general multi-modal data learning method for person re-identification, 2021. 6

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. 2

[3] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2017. 2

[4] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2. 2

[5] Yutian Lin, Liang Zheng, Zhedong Zheng, Yu Wu, and Yi Yang. Improving person re-identification by attribute and identity learning. *Pattern Recognition*, 95, 03 2017. 4, 6

[6] Hao Liu, Jingjing Wu, Jianguo Jiang, Meibin Qi, and Bo Ren. Sequence-based person attribute recognition with joint ctc-attention model, 2018. 6

[7] Kilian Pfeiffer, Alexander Hermans, István Sárándi, Mark Weber, and Bastian Leibe. *Visual Person Understanding Through Multi-task and Multi-dataset Learning*, pages 551–566. 10 2019. 6

[8] Chenxin Sun, Na Jiang, Lei Zhang, Yuehua Wang, Wei Wu, and Zhong Zhou. *Unified Framework for Joint Attribute Classification and Person Re-identification: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part I*, pages 637–647. 09 2018. 6

[9] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification, 2017. 2

[10] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6450–6458, 2017. 3

[11] Guangcong Wang, Jianhuang Lai, Peigen Huang, and Xiaohua Xie. Spatial-temporal person re-identification, 2018. 6

[12] Mikolaj Wieczorek, Barbara Rychalska, and Jacek Dabrowski. On the unreasonable effectiveness of centroids in image retrieval, 2021. 6

	Gender	Age*	Hair	L.slv	L.low	S.chlt	B.pack	H.bag	Bag	Hat	Avg	C.up*	C.low*	Avg
Sun et al.[8]	88.9	84.8	78.3	93.5	92.1	84.8	85.5	88.4	67.3	97.1	86.1	87.5	87.2	87.0
APR[5]	86.5	87.1	83.7	93.7	93.3	91.5	82.8	89.0	75.1	97.1	88.0	73.4	69.9	85.3
JCM[6]	89.7	87.4	82.5	93.7	93.3	89.2	85.2	86.2	86.9	97.2	89.1	92.4	93.1	89.7
Pfeiffer et al [7]	92.9	87.0	89.7	93.6	94.8	94.6	88.0	89.4	79.7	98.0	90.8	79.4	71.9	88.2
Our baseline	85.9	89.4	84.6	95.9	89.1	92.4	81.0	84.1	73.6	96.0	**87.2**	93.2	92.5	**88.1**

Table 1. "L.slv", "L.low", "S.clth", "B.pack", "H.bag", "C.up", "C.low" denote length of sleeve, length of lower-body clothing, style of clothing, backpack, handbag, color of upper-body clothing and color of lower-body clothing, resp.
* As previously mentioned, these values are computed as if their neurons were dedicated to a Multilabel classification task.

Methods	mAP
CTL Model [12]	**98.3**
RGT&RGPR [1]	95.6
st-ReID [11]	95.5
APR [5]	66.9
Our baseline	81.3*

Table 2. Source https://paperswithcode.com/sota/person-re-identification-on-market-1501.
* This is achieved after averaging several attempts with a gallery composed of ∼2k images and ∼150 person-IDs coming from the validation_set of the classification task. A single run on the entire train_set (∼12k images and 751 person-IDs) showed a mAP of 70.8