A Multi-Task Approach to Person Attribute Recognition and Re-Identification

Azzolin Steve
University of Trento
Artificial Intelligence Systems
steve.azzolin@studenti.unitn.it

Destro Matteo
University of Trento
Artificial Intelligence Systems
matteo.destro@studenti.unitn.it

Masina Gabriele
University of Trento
Computer Science
gabriele.masina@studenti.unitn.it

Abstract

This is the report of the final Deep Learning 2020-2021 course project. Attribute recognition and person re-identification are respectively the problems of predicting a set of attributes for an individual from a set of images, and retrieving from a possibly very large collection of images all the elements depicting a certain individual. We propose a multi-task learning approach to this problems by using a single CNN with multiple heads for attribute recognition and re-ID. We combine the AR cross-entropy loss and the re-ID triplet loss using a pair of weights that are automatically learned by the model. The training is performed using a custom batch sampler to favor the generation of useful triplets. The results show that the proposed models are able to learn a common internal feature representation to solve both tasks with a common backbone.

1. Introduction

Attribute recognition and person re-identification are two independent but complementary tasks. The former deals with the prediction of the physical attributes of an individual, like age, gender, hair style, etc. The latter instead deals with the retrieval of images depicting the same individual of a query image from a (possibly very) large database of samples, captured from multiple non-overlapping cameras. These two tasks are related to each other, since we can enhance the re-ID performances by filtering all the images with some specific attributes, or we may use re-ID to find better view-points from which to predict the attributes.

Person re-ID is essential for intelligent surveillance systems, in which for example given a query person-of-interest, the goal is to determine whether this person has appeared in another place at a distinct time captured by a different camera, or even the same camera at a different time instant. Unfortunately, this is a very challenging problem, due to the intrinsic distributed nature of the problem, low-resolution surveillance cameras, occlusions, different light conditions and unreliable bounding box generations.

In recent years, advancements in Deep Learning techniques have led to their adoption in many fields of science, from brain stimuli reconstruction [9] to person re-ID included. A broad overview of Deep Learning approaches for the problem of person re-ID, along with a taxonomy of its various sub-tasks, is presented in a survey by Mang et al. [10].

In our work, we took inspiration from the current state-of-the-art in Computer Vision and from previous works [10] in order to jointly address the problem of attribute recognition and person re-ID. Our contribution is structured as follows: we will first briefly review the Market-1501 dataset [11], then we will present some architectures to deal with attribute recognition and person re-ID, and finally we will present our results and final conclusions.

2. Dataset: Market-1501

Market-1501 [11] is a collection of 32,668 images taken in front of a supermarket in Tsinghua University and depicting 1,501 different identities. Each annotated identity is captured by at least two different cameras. Moreover, in addition to these images, a number of distractors (junk images) are present. Lately, this dataset was augmented by attribute labels information [5] making it suitable for a joint benchmark of attribute recognition and person re-ID.

The special version of the dataset that we used in our experiments (provided by the teachers[1]) contains a training and testing split of 12,989 and 19,679 samples respectively. To make our experiments more robust, we decided to partition the training split into 2 folds, in a 70-30 manner: 70% of samples for training and 30% for validation. This split was done in such a way that no identities are both in the training and in the validation split.

[1] https://drive.google.com/drive/folders/142DAXGIXxV3ALFc_wjBiDtx4wxULcWT1?usp=sharing

3. Models

This section describes our models for attribute recognition and re-identification. Since the pre-trained ResNet model used as backbone for both tasks was trained on images cropped to 224x224, and our dataset has 64x128 images, we decided to change the stride values of the first convolutional layer and maxpool layer in order to obtain a resulting feature map with a size similar to the one used to train the hidden ResNet layers. In particular, we changed the first convolutional layer's stride from (2, 2) to (2, 1), to account for the rectangular aspect ratio, and the maxpool stride from (2, 2) to (1, 1), to account for the lower resolution. The experimental results showed that this modification brings an average performance gain of $\sim 2\%$ and $\sim 4\%$ for attribute recognition and re-identification respectively.

3.1. Attribute recognition

We first describe our solution to the problem of attribute recognition (AR). We started with a baseline approach based on a pretrained ResNet18 [1] predicting a single global embedding vector v, as for standard object classification. This embedding vector is then used as input feature for 12 densely connected attribute-specific branches, ending with the final output prediction: by doing this we encourage the extraction of specialized features in the last layer. The loss is computed by taking the average of the cross-entropy losses computed separately for each of the 12 attributes, therefore the final loss magnitude does not depend on the number of attributes. Since the embedding vector v encodes global features of the input image, we then decided to give more freedom to the attribute-specific branches, allowing each of them to extract some attribute-specific visual features from the latent representation. Therefore, we decided to convolutionalize the densely connected attribute-specific branches, feeding them with feature maps extracted from the output of the last convolution in the third layer of ResNet, after Batch Normalization and ReLU activation function. This specific layer was chosen after an empirical analysis. We will refer to this model as *ConvHead*, a graphical representation is shown in Figure (1).

At this point we decided to take advantage of a popular technique in Deep Learning, widely used both in the Computer Vision and NLP community: Attention. Attention is a technique that mimics cognitive attention, enhancing the important parts of the input data and fading out the rest. Despite the fact that in the literature there are many variants of attention, one specific for each problem, we focused just on *Multi-Head Attention* [7]. Specifically, we modified the ConvHead architecture in order to make each attribute-specific branch output an embedding w_i, obtained as a densely connected layer on top of the flattened attribute-specific convolutional feature maps. These attribute-specific embeddings w_i, along with the global embedding v, are then passed through a two-head attention layer, which produces as output the same number of input embeddings, where each embedding is obtained by a weighted combination of all the other embeddings. The rationale is to encourage feature reuse across attributes, making them helping each other, allowing also each attribute to take information from the global feature representation if needed. Finally, the attended embeddings are passed throughout Layer Normalization and a final dense layer. We will refer to this model as *Self-Attention* (*SA*).

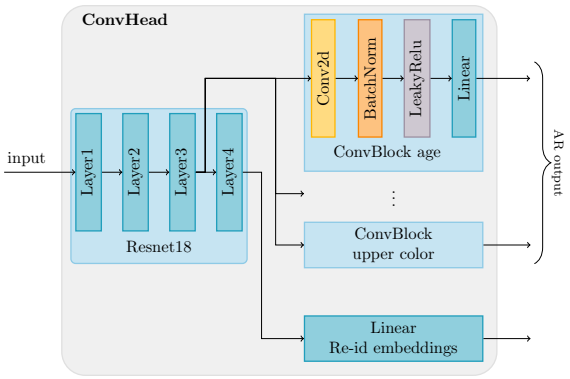

Figure 1: Graphical representation of the ConvHead model.

3.2. Person Re-Identification

For the task of person re-identification we decided to use a Siamese Network [2] in combination with the attribute recognition task, in a multi-task learning setting. The rationale is to learn an embedded representation of each image s.t. images from the same identity have similar representations, where the similarity is computed as the Euclidean distance between the embeddings. We took the global feature representation v described in Section (3.1), and we added a further dense layer that provides the final embedding (in the case of the ResNet baseline we included also Batch Normalization and LeakyReLU).

To obtain a final embedding able to encode the identity information, *Triplet loss* [8] was used to train the model:

$$L_{id}(\theta) = \frac{1}{N} \sum_{\substack{a,p,n \\ y_a=y_p\neq y_a}} [m + D_{a,p} - D_{a,n}]_+ \qquad (1)$$

This loss makes sure that given an *anchor* sample x_a, a *positive* sample x_p (from the same identity) and a *negative* sample x_n (from a different identity), the distance $D_{a,p}$ between the embeddings of the anchor and the positive samples is lower by a predefined margin m than the distance between the anchor and the negative $D_{a,n}$. By training over this loss function, the model learns to "pull" together the embedding representations of x_a and x_p while pushing away the representations from different identities.

Figure 2: Example of re-identification, using the first image as query. The green boxes denote a correct image, i.e. an image depicting the same identity of the query, while the red ones denote a wrong prediction.

The main issue with this approach is that the number of triplets is cubic w.r.t. the size of the dataset, making it too computationally expensive to process all the possible triplets. Moreover, the model usually learns to correctly separate most trivial triplets (e.g. people with different clothes colors), making them useless. For these reasons, sampling randomly the triplets to use for training leads to a very slow convergence, since most triplets are trivial and provide no useful information, having a zero loss. To solve this issue, Schroff et al. [6] proposed to extract a set of *hard* triplets, i.e. triplets that result in a positive loss, for which $m + D_{a,p} > D_{a,n}$. To do so efficiently, we implemented a custom batch sampler: for each batch it samples randomly P identities, and for each identity K images, thus resulting in a batch size of PK from which a total of $PK(K-1)(PK-K)$ valid triplets can be generated. During training, we first compute the embeddings for each of the PK samples, then the pairwise distances $D_{i,j}$ are computed as the Euclidean distance between each pair of samples. From these distances, we extract the losses of every valid triplet which also provides a positive loss, i.e. we exclude from the loss computation the triplets that have a zero loss, in order to avoid obtaining a very small value.

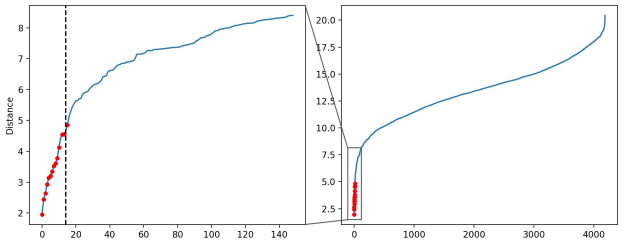

Figure 3: Example of the distances obtained w.r.t a query sample, sorted from nearest to furthest sample. Red dots represent samples from the same identity of the query. The dashed vertical line corresponds to the number of samples of the ground truth.

Figure (2) shows an example of the re-identification results based on these distances to retrieve images of the same identity, while Figure (3) shows an example of the computed distances from the embedding of one sample to the others.

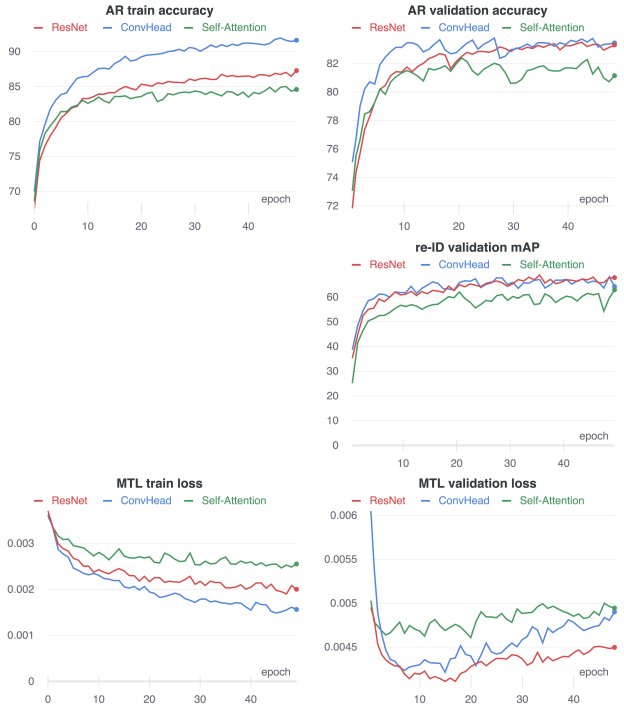

Figure 4: Training and validation results for attribute recognition (accuracy) and re-identification (mAP) and the final values of the combined losses.

3.3. Multi-task Loss Function

Once the losses for attribute recognition and re-identification have been computed, we need to combine them into a single value. A simple sum may not be optimal, since the loss magnitudes of the two tasks may be very different: attribute recognition uses an average of cross entropy losses, while for re-ID the loss magnitude depends on the Euclidean distance, i.e. it's strictly related to the embeddings size and also on the margin value used in the triplet loss. To account for this issue, we implemented a weighted summation, as proposed by Kendall et al. [3]. The final loss is computed as:

$$L_{mtl}(\theta) = \frac{1}{\sigma_{ar}^2} L_{ar}(\theta) + \frac{1}{2\sigma_{id}^2} L_{id}(\theta) + \log \sigma_{ar} + \log \sigma_{id} \quad (2)$$

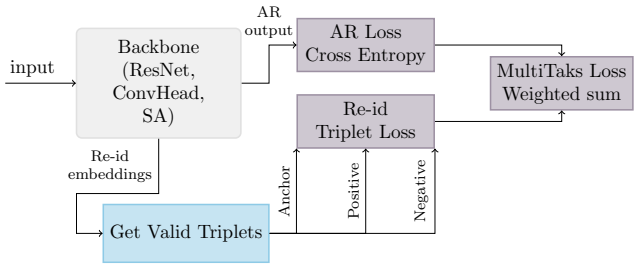

Figure 5: Final architecture combining Attribute recognition and Re-identification tasks.

where L_{ar} and L_{id} are respectively the attribute recognition and the re-identification loss functions, while σ_{ar} and σ_{id} are two parameters that determine the final weighting. Both σ_{ar} and σ_{id} are parameters learned by the network: a large value decreases the corresponding loss contribution, but the term $\log \sigma$ acts as regularization. In practice, to avoid numerical instability and division by zero, the model instead learns $s = \log \sigma$. The final loss therefore becomes:

$$L_{mtl}(\theta) = \frac{1}{e^{s_{ar}}} L_{ar}(\theta) + \frac{1}{e^{2s_{id}}} L_{id}(\theta) + s_{ar} + s_{id} \quad (3)$$

where s_{ar} and s_{id} are initialized to zero and learned by the network.

In Figure (5) we show a diagram of the whole architecture that combines the two tasks.

Attrib.	MFC	ResNet	ConvHead	SA
age	78.61	78.63	83.09	74.50
backpack	76.16	81.83	82.86	75.98
bag	73.35	73.97	70.08	75.36
handbag	88.28	90.47	90.19	90.38
clothes	85.43	89.47	92.10	92.36
down	63.68	89.26	87.08	85.86
up	94.13	93.94	93.62	93.10
hair	64.32	84.10	84.03	84.12
hat	96.81	96.35	96.47	96.51
gender	55.42	87.80	89.37	82.14
uppercolor	27.16	69.96	69.25	65.74
lowercolor	45.52	66.14	64.30	62.89
Average	70.70	83.49	83.53	81.58

Table 1: Attribute recognition accuracies on the validation set.

4. Experiments and results

Each network was trained with Adam optimizer [4] for 50 epochs. All the hyper-parameters, like learning rate, weight decay or triplet loss margin, were chosen after an empirical analysis.

ResNet	ConvHead	SA
67.86	69.93	62.93

Table 2: Re-identification mAP on the validation set.

To evaluate the performances of our model in the attribute recognition task we stuck to the accuracy metric since it is the one used by many previous works. Table (1) shows the detailed per-attribute accuracy scores for the three models, with the addition of a Most Frequent Classifier baseline (MFC) to assess any class-imbalance related biases. The results of MFC show that actually class-imbalance, for some attributes, is severe. For re-ID, we used the *mean average precision* (mAP), computed over the samples sorted by increasing Euclidean distance between their embeddings and the one of the query image. Finally, Figure (4) depicts the training statistics for the aforementioned models.

We do not deal directly with the presence of junk images in our test set, for which attribute predictions are made, but here we report some ideas that can help to spot these distractors:

- If the network is properly trained, then the prediction's confidence scores for junk images should be low for every attribute.
- Run a person-detector pretrained model over the test set before running the attribute recognition module.
- Add junk images also in the training set and make the network predicting whether the input image is a person or not, in a Multi-Task Learning setting.

With a closer inspection of *MTL validation loss* in Figure (4) we can note an overfitting regime starting at around epoch 10-15, despite successfully reducing it by empirical tunings of both the architecture and hyper-parameters. Two more ways to contrast overfitting, blocking it from degrading test results, are to use EarlyStopping and/or selecting the best model w.r.t. a certain metric, like validation loss minimization or validation metric maximization. Due to our MTL settings, defining a unique metric to be maximized/minimized is not trivial. In fact, a simple minimization of the validation loss may not be sound, due to the continuous rescaling of the loss dictated by Eq. (3). Moreover, metric maximization should account for both AR accuracy and re-ID mAP, resulting again in something not trivial. For the aforementioned reasons, we decided to provide our final submission file with the predictions made by the Siamese network with ResNet backbone, since is the one obtaining more stable asymptotic behaviour as shown in Figure (4).

References

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

[2] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification, 2017. 2

[3] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics, 2018. 3

[4] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. 4

[5] 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. 1

[6] 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

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017. 2

[8] Kilian Weinberger, J. Blitzer, and L. Saul. *Distance Metric Learning for Large Margin Nearest Neighbor Classification*, volume 10. 01 2006. 2

[9] Haiguang Wen, Junxing Shi, Yizhen Zhang, Kun-Han Lu, Jiayue Cao, and Zhongming Liu. Neural Encoding and Decoding with Deep Learning for Dynamic Natural Vision. *Cerebral Cortex*, 28(12):4136–4160, 2017. 1

[10] Mang Ye, Jianbing Shen, Gaojie Lin, Tao Xiang, Ling Shao, and Steven C. H. Hoi. Deep learning for person re-identification: A survey and outlook, 2021. 1

[11] 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