International Journal of Computer Applications (0975 - 8887)
Volume - No.,

Attribute Classification and Re-Identification on Market-1051 Dataset

Ali Akay
Trento University
MSc.Artificial Intelligence Systems
ali.akay@unitn.it

Mert Akkor
Trento University
MSc.Artificial Intelligence Systems
mert.akkor@unitn.it

ABSTRACT

In this assignment, we demonstrate the usage of neural networks to solve two common computer vision tasks using the PyTorch framework. Market-1051 dataset is video-surveillance dataset containing images of multiple persons each of which is captured multiple times by different cameras along with a set of annotations that specify attributes of each person such as age, gender and clothing. The first part of the assignment, we have done multi-class classifier task to predict attributes for each image. In the second part of the assignment, we tried to solve a person re-identification problem.
3

1. INTRODUCTION

Image analysis of pedestrian attributes such as gender, age, hair,clothes color, etc., has become a thriving research field in recent years [1] , on account of its wide range of possible applications, such as person retrieval,person re-identification [2] and so on.
Pedestrian characteristics, such as gender, dark hair, and skirt, have been utilized as soft biometric features in surveillance and have recently received a lot of attention. Person retrieval, subject identification, person recognition, human identifying face verification, and person re-identification, for example, can all benefit from pedestrian characteristics. In many real-world surveillance scenarios, cameras are positioned at a great distance to cover a large area, thus pedestrians are caught with poor resolution, making high-quality face photos difficult to come by. However, pedestrian attributes have a great application potential in such circumstances, since they have been demonstrated to give numerous advantages beyond standard biometrics, such as lighting and contrast invariance. In the categorization of pedestrian attributes, there are three major problems. First, there are significant intra-class differences due to varying apparel looks, lighting circumstances, and camera angles. The backpack annotated examples in the Market-1501 database shot with different cameras have substantial visual differences.
Second, pedestrian attributes have complicated localizing features, meaning that some traits can only be identified in specific or ambiguous local body regions. Long hair, for example, is most noticeable on the head and shoulders. As a result, extracting pedestrian attribute features is quite challenging. Third, because pedestrian attributes are not mutually exclusive, pedestrian attribute classification is a multi-label classification issue rather than a multi-class classification problem. As a result, the majority of existing multi-class classification algorithms are inefficient, and multi-label classification has its own set of challenges. The most popular approach for attribute prediction is the one using hand-crafted features and SVM based individual attribute classifier, which cannot solve the above-mentioned challenges successfully because hand-crafted features have limited representation ability for large intra-class variations, and independent SVM classifiers cannot investigate interactions between attributes. We solve the multi-attribute classification problem with a multi-output convolutional neural network. The multi-output convolutional neural network is trained from raw pixels rather than hand-crafted features and is able to simultaneously recognize multiple attributes. Furthermore, combining attribute distances and low-level feature distances between pairs of person images leads to a better person re-identification method than existing approaches, thanks to the improvement of the proposed attribute classification method. Lastly, on the second part of our project, we remove classifier from network and create a feature map for each image. We look for cosine similarity for these images at last.

2. DATASET

For the dataset, we annotate 27 attributes for Market-1501 dataset. The original dataset contains 751 identities for training and 750 identities for testing. The attributes are annotated in the identity level, thus the file contains 28 x 751 attributes for training and 28 x 750 attributes for test.You can check the dataset with **this link**.

As an addition for color of upper (8) and color of lower (9) attributes, only one color is labeled as yes for an identity. So, for multi-colored upper and lower, we created new attributes called ‘multicolorup’ and ‘multicolordown’ to make identification better.

3. CLASSIFICATION TASK

We have used Resnet-50 pre-trained model to predict 12 attributes of the image.ResNet-50 is 50 layers deep and is trained on a million images of 1000 categories from the ImageNet database. So the output feature of the model is 1000. We added 12 fully connected linear layer instead of classifier of the model.Using AdaptiveAvgPool2 we can use any image size and our model

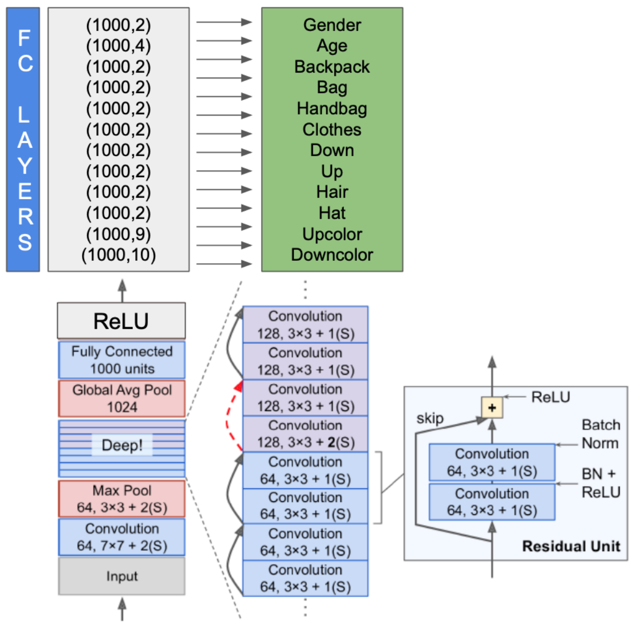

Fig. 1. ResNet Architecture

will work. You can see our architecture of the resnet model [Fig. 1].

You can reach our model graph in our tensorboard graph session with **this link**.

3.1 Data Preparation

For the data preparation task,in the dataset labels start with 1 (No) and 2 (Yes) for the binary attributes.We convert all labels starting from 0 for the resnet.

In the original dataset, there are 8 for upcolor and 9 for downcolor labels but there is some problem for the color of the tshirt or skirts if he/she wear something with multicolor, it is a problem for classification.So,we also added into multicolor label for both attributes.

To prevent overfitting, we use data augmentation methods for instance normalization,horizontal flip.We tried also random erasing but we didn't see any improvement in our model so we decided to not use it.

We splitted our data into train and validation with custom way.Same person images shouldn't be in both train and validation data, there could be a overfitting problem for the model.

3.2 Loss Function

We use cross entropy loss for each each attribute.The cross entropy formula takes in two distributions, p(x) the true distribution, and q(x), the estimated distribution, defined over the discrete variable x and is given by

$$H(p,q) = -\sum_{\forall x} p(x) \log(q(x)) \qquad (1)$$

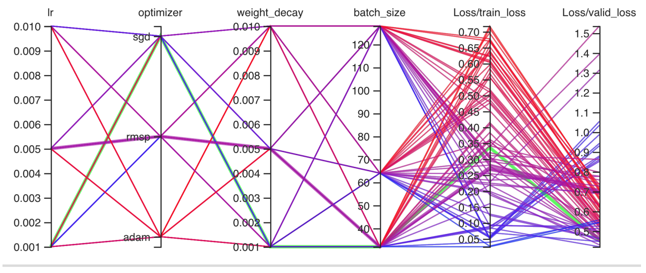

Fig. 2. Hyperparameter Tuning

$$H(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{\mathbf{i}} \mathbf{y_i} \log_{\mathbf{e}}(\hat{\mathbf{y}}_\mathbf{i}) \qquad (2)$$

Where $yhat$ is the predicted probability vector (Softmax output), and y is the ground-truth vector.The reason we use natural log is because it is easy to differentiate and the reason we do not take log of ground truth vector is because it contains a lot of 0's which simplify the summation.

We calculate loss separately for each attribute and then we sum all losses for each attribute then divide in to attribute number then give it to the optimizer which we used SGD with momentum.

$loss = (loss1 + loss2 + loss3 + loss4 + loss5 + loss6 + loss7 + loss8 + loss9 + loss10 + loss11 + loss12)/12$
Backward pass
$loss.backward()$
Update parameters
$optimizer.step()$
Resets the gradients
$optimizer.zero_grad()$

3.3 Results

3.3.1 hyperparameter Tuning. Before we decide the model, we have done hyperparameter tuning with some optimizer and parameters for 10 epoch because of the computational issues. In [Fig 2] you can see that some models had overfit based on the training loss is small while validation loss still high. In terms of the graph, the worst model came with "Rmsp" optimizer and the best models came with "SGD" optimizers. Finally we choose the "Green" model in the graph as a best model which is has the parameters;

LR:0.001,Optimizer:SGD, Weight Decay: 0.001,Batch Size:32
Weight decay is seems so high but with this value we could able to prevent overfitting.
You can reach our hyperparameter tuning tensorboard with **this link**.

Here in the table you can see our best models.Then we run this models with 30 epoch.We realized that after 15 epoch it goes to overfit because while training loss is decreasing,validation loss stable after some epoch. To prevent from this issue we run our training with 15 epoch.

In the end,our average accuracy reached to 84 percent for validation test.For some attributes like "Hat", it has mostly 0 rather

Lr	Optimizer	Weight Decay	Batch size	Valid loss
0.005	sgd	0.01	128	0.452
0.001	sgd	0.001	32	0.456
0.001	sgd	0.01	32	0.461
0.01	sgd	0.01	32	0.477

then 1 so our accuracy for that attribute is 97 percent but it is because of imbalanced data.Other observation is about upcolor and downcolor attributes,our model still continue to learn for these attributes but we stop our training after 15 epoch.So here you can see our validation results for each attributes;

Valid loss: 0.4568
Validation average Acc: 84.05
Gender: 85.46
Age: 89.50
Backpack: 81.74
Bag: 77.33
Handbag: 90.12
Clothes: 91.94
Down: 91.34
Up: 95.98
Hair: 82.15
Hat: 97.83
Upcolor: 69.72,
downcolor: 55.53

You can reach our training and validation loss and accuracies for each attributes per epoch in our tensorboard with **this link**.

You can reach our google colab repository with **this link**.

4. RE-IDENTIFICATION TASK

We used a ResNet-50 again and for this time we remove the final classification layer and we get feature space for each image and then we train our network on the training set by imposing the triplet loss on such feature space. After this process,we got feature space for each images and for each query image features that we extracted from validation set, we computed the 15 images between the validation set that are closest in terms of cosine similarity.Finally wee evaluate our model with mean average precision metrics for query images.

4.1 Data Preparation

We did not do any specifics data preparation for this part of the project.We used data augmentation methods such as normalization,horizontal flip and random erasing.
We used the same training and validation size with the classification part of the project.

To calculate mAP, we splitted our validation set to 2 dataset which is query and test after we got the feature space for whole validation set from the model.

As you can see in the [Fig. 3] the average number for each person is 16.But the maximum image for a person is 58 images. Based on the our model, we should also give some error margin for the prediction.So we predicted 80 images per person for each query images based on the histogram. Because mAP metric does not penalize the wrong labels.

```
maximum image of a person:  58
minimum image of a person:  1
average image of a person:  16.638613861386137
<matplotlib.axes._subplots.AxesSubplot at 0x7f502a3cd110>
```

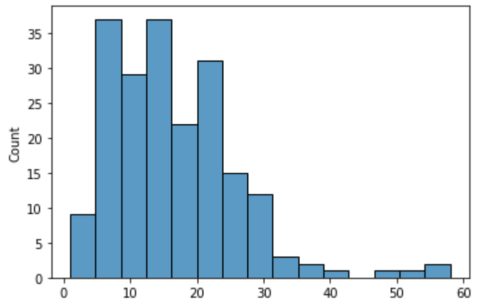

Fig. 3. Number of image per person

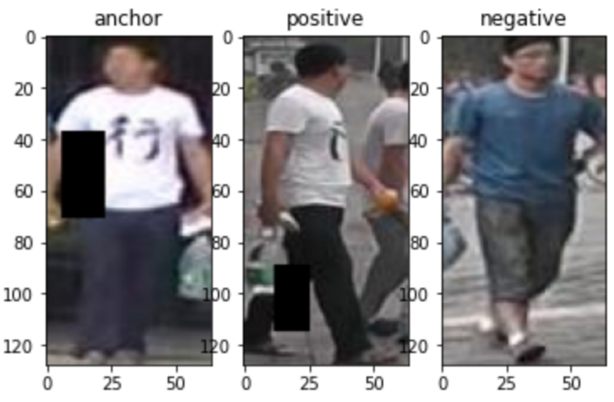

Fig. 4. Triplet loss

4.2 Triplet loss

The cost function for Triplet Loss is as follows:

$$L(a, p, n) = max(0, D(a, p)|D(a, n) + margin)$$

where D(x, y): the distance between the learned vector representation of x and y. As a distance metric L2 distance or (1 - cosine similarity) can be used. The objective of this function is to keep the distance between the anchor and positive smaller than the distance between the anchor and negative. [Fig 2]

For the triplet loss,we 3 identical networks having the same our model and they should share weights.As a output we have 3 vector representation for anchor,positive and negative image that are passed through their respective network and during back propagation weight vectors are updated using shared architecture.

4.3 Mean average precision Evaluation Metric

Mean average precision (mAP) is a commonly used metric to evaluate performance in person re-identification.Compute the mAP of the predictions with respect to the given ground truth.In person re-identification mAP refers to the mean of the Area under curve AUC(AP) over all queries.The AP for a query is the area under the precision-recall curve obtained from the list of predictions consid-

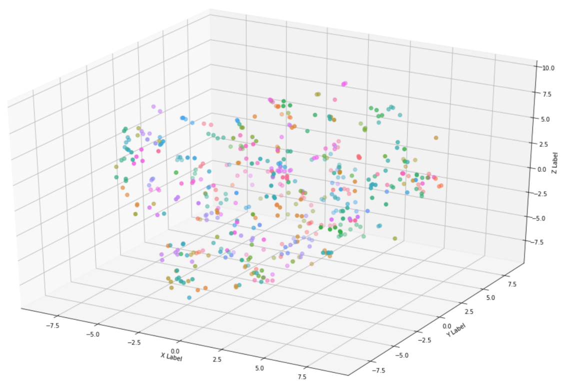

Fig. 5. T-SNE Visualization of feature space

ering the ground truth elements as positives and the other ones as negatives.

4.4 Results

We use this parameters for our Resnet-50 model

LR:0.001,Optimizer:SGD, Weight Decay: 0.01,Batch Size:32
Our model tend to be overfit,to prevent this we use very high weight decay.

You can reach our training and validation loss per epoch in our tensorboard with **this link**.

We got 2048 vector for each validation images as a feature space. After that we used the T-SNE method for sample of image to visualize in 3-dimension. As you can see in the [Fig. 5] some images which has the same id are very close each other.But still our model is not perfect to discriminate for each id. (Each color represent person id)

For the next step we use cosine similarity to compute first 80 images which has higher similarity with each query images and this is our prediction of the query images.

You can reach our google colab repository with **this link**.

5. CONCLUSION

First of all,we were able to use Pytorch framework in this project.We have changed to demonstrate neural networks and CNN architectures and implement it for the real world problem.While classification results made us satisfied, re-identification problem have quite low mAP results. We could'nt achieve a reference mAP result in the end.We should have improved our model by using combination of ID Loss and Triplet loss could have increased the mAP result.

Finally,we haven't done this kind of CNN project before,so It was very instructive and helped us to gain practice about this topic.

6. REFERENCES

[1] J. Zhu, S. Liao, D. Yi, Z. Lei, and S. Z. Li *Multi-label cnn based pedestrian attribute learning for soft biometrics,*. 2015.

[2] R. Layne, T. M. Hospedales, S. Gong, et al. *Person re-identification by attributes*. 2012.