Incremental Learning in Person Re-Identification

Prajjwal Bhargava
SRM University
Chennai
prajjwalin@protonmail.com

Abstract

Person Re-Identification is still a challenging task in Computer Vision due to variety of reasons. On the other side, Incremental Learning is still an issue since Deep Learning models tend to face the problem of overcatastrophic forgetting when trained on subsequent tasks. In this paper, we propose a model which can be used for multiple tasks in Person Re-Identification, provide state-of-the-art results on variety of tasks and still achieve considerable accuracy later on. We evaluated our model on three datasets Market 1501[13], CUHK-03[6], Duke MTMC[12]. Extensive experiments show that this method can achieve Incremental Learning in Person ReID efficiently as well as for other tasks in computer vision as well. The code for this work can be found here

1. Introduction

Deep neural networks have revolutionized the field of computer vision. In recent years, a lot of work has been done in Person Re-Identification ,we've seen a considerable progress but still we face a lot of challenges in terms of getting accurate predictions in real life instances.It plays an important role in many areas, surveillance being one of them. In some sense, it can be compared to other prominent tasks in computer vision like Image Retrieval or Object Detection, where a lot of progress have been made. Moreover, there has been a growing demand of deep learning models that incur low computational cost. Deployment of such models can be cumbersome and may not prove to be much efficient especially if the same task can be carried out with lesser number of parameters. Given a set of images of a person taken from different angles from different camera, our model is required to generate a higher prediction if those images are of the same person and vice versa. The problem is composed by multiple reasons some of which may include background clutter, illumination conditions, occlusion, body pose, orientation of cameras. Numerous methods have been proposed to address some of these issues. So far the models that have been proposed in Person ReID are good in doing well in particular dataset and when tested on quite dissimilar dataset, they struggle to get just right predictions. Unlike other tasks such as Image Classification or Object Detection, we are required to have our model perform well on a large number of classes and all these images are not as much distinctive as other objects do which makes it difficult for neural net to predict. We devise a new method that can be used to create robust Person-ReID systems at lower computational cost that can not only perform well on one tasks, but if trained properly using our techniques, can be well adapted to other tasks as well.

2. Related work

For Incremental Learning, many research work has been carried out. Our work is slightly inspired from LwF[8], which was used for classification purpose. They made use of CIFAR10 and SVHN as the two tasks and then achieved considerable performance. Other closely associated work which build upon it is SeNA-CNN[11], wherein they made the architecture a little more complex by introducing more layers in different pipelines instead of just dealing with fully connected layers. Our work is the first one that tries to tackle the problem of Incremental Learning in Person Re-Identification, unlike image classification where we have relatively lesser number of classes, the number is way more, and this increases the difficulty level for generating accurate predictions. In defense [3] made use of Triplet Loss to show that it can be used to perform end to end deep metric learning. Some work that has been carried out in this incremental learning space also makes use of Distillation[4] of neural networks wherein you train a smaller network to produce close predictions to cumbersome models. But to carry out this task, we are also required to train our cumbersome model first, to be able to train the smaller model which is again a big task. Our proposed method doesn't rely on multiple models,or older data that has been used to train it on earlier task,rather we have multiple pipelines inside one model which aims to resolve this issue.

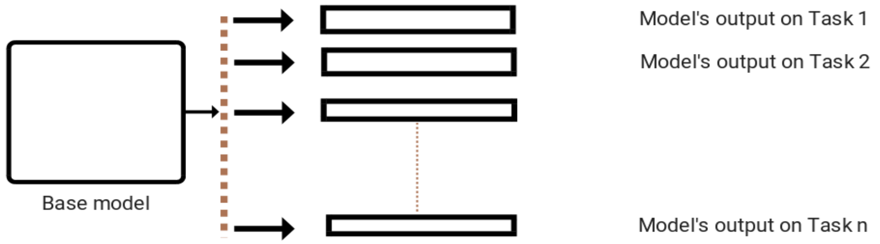

Figure 1. Proposed architecture

3. Our proposed method

We propose a new architecture that's relatively simple as compared to other proposed methods for achieving Incremental Learning along with few techniques that makes convergence faster and increases model's accuracy.

3.1. Overall architecture

We use a ResNet50[2] which has been pretrained on ImageNet[9] . We remove the last two layers i.e Fully connected layer and Average Pooling layer. We then introduce pipelines. Main goal behind keeping ResNet is to performs the task for feature extraction effectively, it acts a base model which contains global common features extracted from the data. Then these Pipelines have been introduced to generate task specific predictions. These pipelines can be modified as per use case to better adapt to given task. In our case, since the two tasks were same, we decided to keep them identical.

3.2. Multiple Pipelines

We introduced two pipelines after ResNet, one is meant to work on Market1501 and the other dataset can be used either for DukeMTMC or CUHK-03. Each pipeline consists of two convolutional blocks followed by a Fully connected layer. Each convolutional block consists of convolutional layer which takes in *ni* input channels with kernel size 1,stride 1 and outputs *ni/2* channels. This is followed by Batch Normalization[5] and usage of Leaky ReLU[1] activation. Another block takes in *ni/2* input channels and outputs *ni* channels with kernel size 3 and stride kept to 1. So in this process dimensionality is not changed. Later the input is then fed to Fully connected layer, to generate the prediction vector depending upon the number of classes we require.

We tried adding residual blocks within pipelines to learn residual mapping rather than underlying mapping but it didn't improve performance much. Since our pipeline consist of two convolutional blocks, adding residual connection didn't much help but adding this connection in case of several layers is bound to help and would also reduce the number of parameters by a greater amount.

3.3. Optimizer

We tried many optimizers. We initially tried with Adam[7], that gave an accuracy of 74% on Rank 1 on Market1501 dataset, then we introduced weight decay that helped us achieve higher accuracy. We then tried Cyclical Learning Rate[10],we were able to achieve much higher accuracy. We saw an increment of more than 10% on Rank 1 on Market 1501 to reach 89.3%. This clearly shows that there were issues with non convex optimization and not enough gradients were being generated to get out out of saddle points. We use the triangular variant with default values as suggested. We restricted our batch size to 32 as it provided the best results, Keeping a higher batch size would lead to less frequent weight updates. Since the learning rate becomes variable with CLR, it can take advantage of it's behaviour of making LR variable wherever necessary in a more effective manner as our experiments have showed.

No.	Batch Size	Rank1 (Market1501)
1	32	79.2%
2	64	89.3%

3.4. Using Covariance loss for contrastive feature learning

We are proposing a new addition to our loss function, whose main aim is make positive targets (images of same person, taken with different camera) closer and negative targets (images of different person) far away in embedding space. We take feature maps which we get from second convolution block from both the pipelines during second phase .This is going to optimize embedding space such that data points with same identity are closer to each other than those with different identities. We are required to take feature maps of positive targets and negative targets, we then have to perform the following operation:

$$A = \lambda * (\alpha * \sum((P_{i+1} - P_i) - (N_{i+1} - N_i)) - \beta) \quad (1)$$

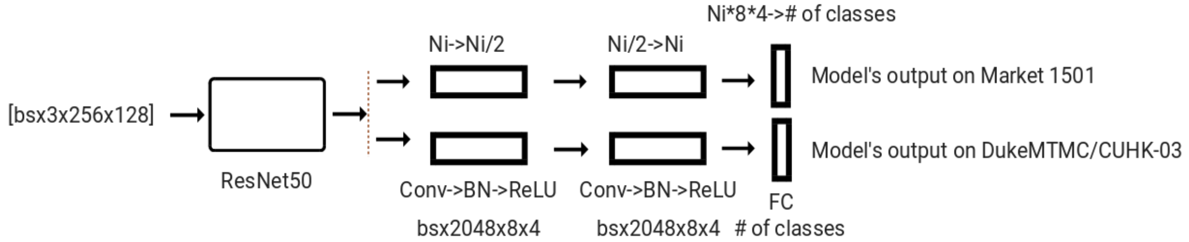

Figure 2. Our architecture

where P and N denotes positive and negative targets respectively and *i* corresponding indexes. λ, α and β are three hyperparameters. Finding the most optimum value for these hyperparameters is an exhaustive process since we are required to let our model train for considerable amount of epochs. Higher value of λ introduces fluctuations in overall loss function since the value changes rapidly and may cause instability. We found that these set of values worked best in our case

Hyperparameter	Value
λ	1
α	1e-9
β	0

To perform this type of task we either can keep track of positive targets and negative targets before feeding them to the model or we can create a mask which can indicate which feature maps to choose. We use the second approach. Mask outputs a consecutive vectors in pair, one for positive target and the other for negative targets that indicates which feature map to pick out of multiple maps. Dimension of feature map is $[batch\_size * 2048 * 8 * 4]$. To perform the operation of subtraction,we flatten the feature map of both positive and negative targets. Then we perform subtraction, giving us a Tensor of $[batch\_size * 2048]$, add another axis which gives us $[batch\_size * 2048 * 1]$, then require a transpose of this matrix as well, giving us another matrix of dimension $[batch\_size * 1 * 2048]$. We are required to create co-occurence embedding matrix whose sum of elements is going to give us an indicator to improve model's predictions.

$$X = \mathbf{A} * \mathbf{A}^\intercal \tag{2}$$

We tried different values of λ to get the best accuracy possible. The value being computed by covariance loss introduces fluctuations on overall loss since cross entropy reduces approximately monotonically in the initial and mid course of the training,but that;s not the case with covariance loss since weights of feature maps are changing rapidly relatively. So it's recommeneded to keep both the loss values in the same range. Therefore the value of λ plays a great role in determining the overall performance of the model itself.

Value of λ	Accuracy after 100 epochs(Rank 1)
0.7	83.3
0.8	83.1
0.9	83.1
1	**84.6**
1.3	82.9

3.5. Training methodology

There are few ways to train these pipelines, we divided the training into two phases. In the first phase, our model along with the first pipeline was trained on Market 1501 with the other pipeline kept frozen and predictions being taken from the first pipeline itself. There can also be slight variation after first phase, where in some sections of the model can be kept frozen and other pipelines be trained in a different manner (fully task specific). In the second phase, we freeze the first pipeline and then train the base model along with second pipeline on CUHK-03 and Duke MTMC alternatively and take predictions accordingly. Similar procedure can be repeated for n pipelines for n tasks as well.

3.6. Objective Function

Our loss function has two critical components now. We are using cross entropy as our classification loss along with covariant loss. Cross Entropy is given as:

$$H_{y'}(y) := -\sum_i y'_i \log(y_i) \tag{3}$$

where y_i is the predicted probability value for class i and y'_i is the true probability for that class. Our final loss is the sum of cross entropy and covariance loss.

$$A + H_{y'}(y) \tag{4}$$

4. Experiments

4.1. Datasets

We used three datasets for this work. Although other datasets can be used, but as of now these three are most widely used and have the most number of images as compared to other prevalent datasets. Market1501 contains 32668 images of 1501 persons split into train/test sets of 12,936/19,732. It has bounding boxes from a person detector which have been selected based on their intersection-over-union overlap with manually annotated bounding boxes. CUHK-03 contains 13164 images of 1360 identities which have been manually cropped. Duke MTMC has 16,522 training images of 702 identities, 2,228 query images of the other 702 identities and 17,661 gallery images (702 ID + 408 distractor ID). Proposed models are bound to perform much better if it's trained on more data. Considering how deep commonly used models are, these datasets are not large enough to effectively train such number of parameters. So our model is very unlikely to overfit.

No.	Dataset	Num of identities
1	Market 1501	751
2	CUHK-03	1360
3	Duke MMTC	704

4.2. Ensembling

Ensembling has often given improved results in various computer vision tasks. This often works really well when predictions are being taken from multiple models. Here we tried ensembling amongst these pipelines .The first phase was performed as usual. The second phase was tried with ensembling using the mentioned methods. But we decided not to include this in our proposed architecture as the model converged faster relatively and accuracy was saturated to a lower max value.

Ensembling method	Rank1(epochs)
Base model,second pipeline	84.5(100),87.7(500)
Both pipelines	84.1 (100)

4.3. Results

Since our main goal is bring generalization into our model and avoid over catastrophic forgetting, we first train the first pipeline, then we evaluated the predictions coming from the last FC layer of first pipeline. Then we train the second pipeline, evaluated it. In the last phase, we don't do any training and just evaluate it on the first task our model was made to perform. These results are reported after the model has converged.

No.	Dataset	Rank1	Rank20	MaP
1	Market 1501	89.3%	98.3%	71.8%
2	DukeMTMC	80.0%	93.7%	60.2%
3	Market 1501	69.5%	92.8%	40.3%

No.	Dataset used	Rank1	Rank20	MaP
1	Market 1501	89.3%	98.3%	71.8%
2	CUHK-03	-	-	-
3	Market 1501	-	-	-

We achieve state-of-the-art accuracy on both the tasks, and yet achieve considerable accuracy on the first task again.

5. Effectiveness of proposed method

Our work indicates that we now have a simple method that can achieve state-of-the-art results when trained on Person Reidentification tasks and yet achieve considerable accuracy on older tasks without losing much information and doesn't rely on older data after it has been used for training it. This is a big step because very often in real world, we don't have access to old data and this would reduce the robustness of our model otherwise. Our architecture and discussed methods can be applied to other computer vision tasks as well. This method is bound to work with tasks which have less variations in domain. For tasks that are similar, it seems to outperform other methods.

6. Conclusion

In this paper, we have shown that we can achieve incremental learning in Person ReID tasks with simpler methods yet achieving state-of-the art results. We also propose a new loss that can be used to bring positive targets closer and vice versa. We hope that future work in ReID community would be to build more better and robust incremental learning systems that can be further adapted to other domains as well thus increasing real life usage of such systems.

References

[1] A. F. Agarap. Deep learning using rectified linear units (relu). *CoRR*, abs/1803.08375, 2018.

[2] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[3] A. Hermans, L. Beyer, and B. Leibe. In defense of the triplet loss for person re-identification. *CoRR*, abs/1703.07737, 2017.

[4] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015.

[5] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015.

[6] S. Karanam, M. Gou, Z. Wu, A. Rates-Borras, O. I. Camps, and R. J. Radke. A comprehensive evaluation and benchmark for person re-identification: Features, metrics, and datasets. *CoRR*, abs/1605.09653, 2016.

[7] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[8] Z. Li and D. Hoiem. Learning without forgetting. In *European Conference on Computer Vision*, pages 614–629. Springer, 2016.

[9] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li. Imagenet large scale visual recognition challenge. *CoRR*, abs/1409.0575, 2014.

[10] L. N. Smith. No more pesky learning rate guessing games. *CoRR*, abs/1506.01186, 2015.

[11] A. Zacarias and L. Alexandre. Sena-cnn: Overcoming catastrophic forgetting in convolutional neural networks by selective network augmentation. In *8th IAPR TC3 Workshop on Artificial Neural Networks in Pattern Recognition*, volume 11081 of *LNAI*, Siena, Italy, September 2018. Springer.

[12] Z. Zhang, J. Wu, X. Zhang, and C. Zhang. Multi-target, multi-camera tracking by hierarchical clustering: Recent progress on dukemtmc project. *CoRR*, abs/1712.09531, 2017.

[13] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian. Scalable person re-identification: A benchmark. In *Computer Vision, IEEE International Conference on*, 2015.