Medium Article Link: https://medium.com/@adityatushar.wadnerkar/pruning-methods-for-person-re-identification-a-survey-860cf8a987cf?sk=bd06ef85f37ae7e105350426acad55dd

SlideShare Presentation Link: https://www.slideshare.net/AdityaWadnerkar1/pruning-methods-for-person-reidentification-a-survey

Pruning Methods for Person Re-identification: A Survey

Introduction

We have witnessed tremendous increase in deep learning architectures proposed in recent times for visual based recognition such as person re-identification in which people are identified with help of distributed shots on several cameras. We will discuss about the survey of state-of-the-art pruning techniques that are suitable for compressing deep Siamese networks applied to person re-identification.

The computational complexity of CNNs hinders the deployment of Deep Siamese networks on platforms with lesser resource, though they've improved accuracy, but cannot be used in applications with real time data constraints, and thus we can compress these without losing accuracy.

There are various techniques which could be effective for the compression of the networks which are analysed and compared based on their strategy and pruning criterion, in different design scenarios fine-tuning networks by applying pruning methods for targeted applications.

Pruning can drastically reduce the complexities in network according to experimental outcomes from Siamese networks with ResNet feature extractors and keeping track of good accuracy. This reduces the number of FLOPS required by ResNet feature extractor by half when dealing with large pre-training and fine-tuning datasets, while maintaining good accuracy. Pruning can improve the performance while training a larger CNNs than fine-tuning smaller ones.

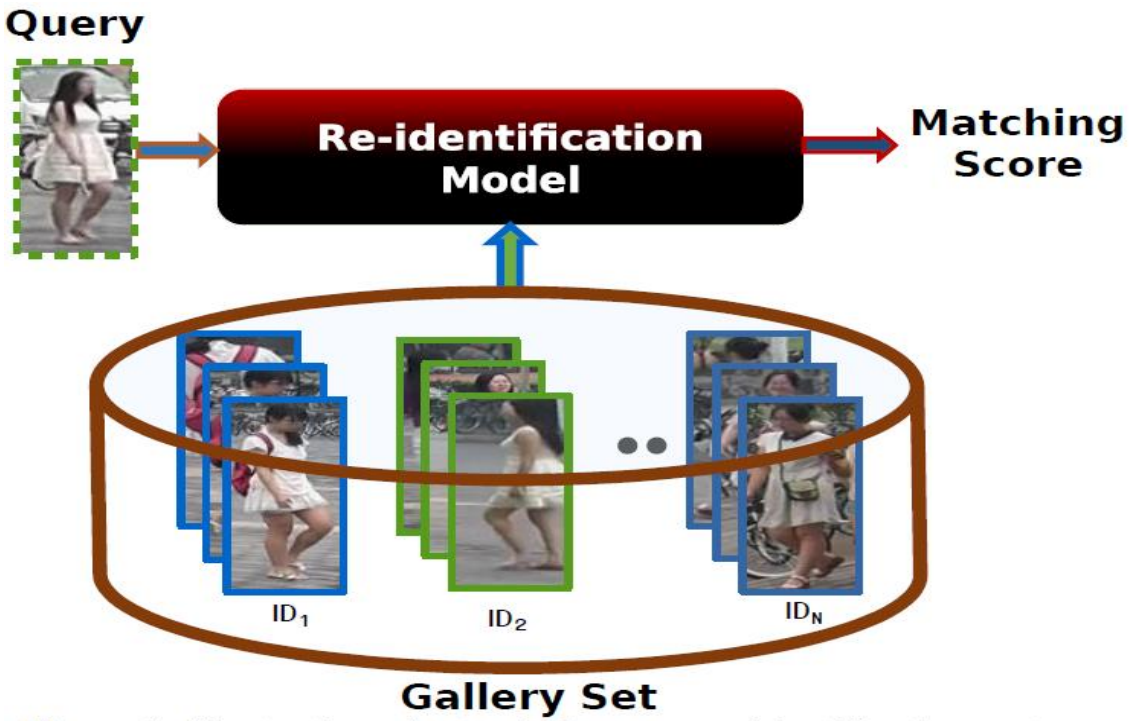


Figure 1: Illustration of a typical person re-identification system.

The convolutional neural network (CNN) and other deep learning architectures have achieved state of art accuracy in wide range of visual recognition tasks, but also exponentially increasing the complexity due to deeper and wider neural networks. Hence, to deploy such complex neural network we need to work on reducing the memory complexity and energy consumption consequently being able to speed up and contract them.

Siamese Networks

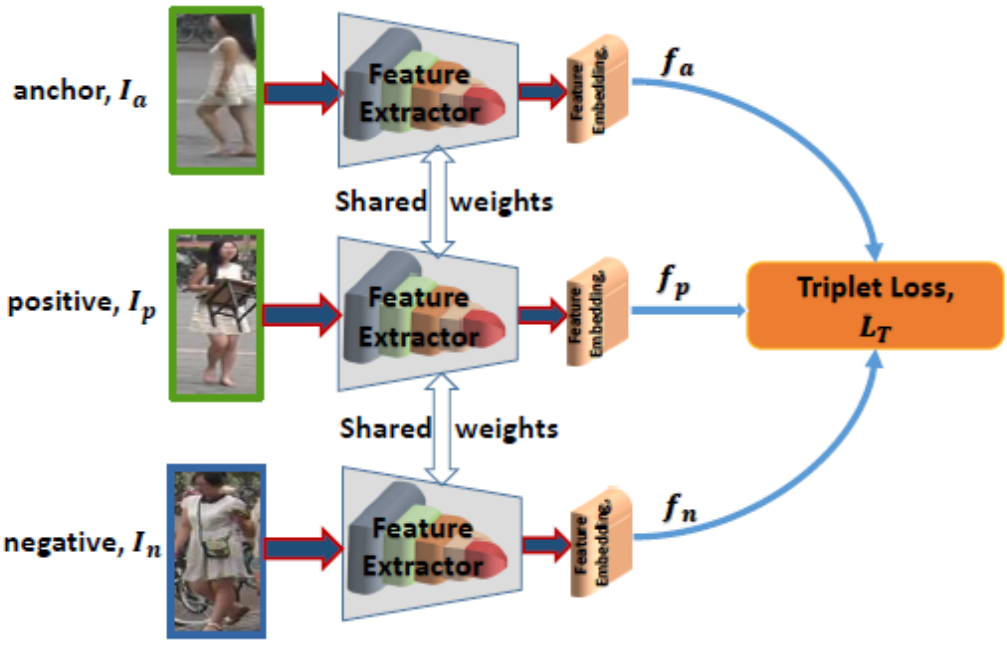


Figure 2: Triplet Training Architecture. Anchor and positive samples are same individual, whereas negative sample is different individual. These triplet set is fed through three identical networks. The triplet loss function optimizes the network parameters in such a way that minimizes the intra-class distances while maximizing the inter-class distance.

The Siamese networks are used for biometric authentication where two sub networks share weights encode feature embeddings matching between query and reference images. They are trained with labelled data extracting features from input images and performing pairwise matching.

The networks VGG, Inception, ResNet and DenseNet can be used as feature extractor with great accuracy. In contrast, ResNet18 and ResNet34 being shallow CNNs provide lower re-identification accuracy. In person ReID applications, most state-of-the-art methods use pre-trained CNNs since they outperformed training CNN feature extractors from scratch.

During training, for a given mini-batch with labels, we randomly sample a triplet {Ia, Ip, In}, where, (Ia, Ip) is a pair of images of the same individual, and (Ia, In) is that of different individual. The corresponding features from the backbone networks are fa, fp and fn. The most common form of triplet loss is as follows:

$$\mathcal{L}_{\mathrm{T}} = \frac{1}{N_T} \sum_{\substack{\mathrm{a,p,n} \\ y_a = y_p \neq y_n}} \left[m + d\left(\mathbf{f}_a, \mathbf{f}_p\right) - d\left(\mathbf{f}_a, \mathbf{f}_n\right) \right]_+$$

The core idea is to form batches by randomly sampling a person, and then sampling number of images of each person. For each sample in the batch, it selects the hardest positive and the hardest negative samples within the batch when forming the triplets for computing the loss:

$$\mathcal{L}_{\mathrm{TBH}} = \frac{1}{N_s} \sum_{a=1}^{N_s} \left[m + \max_{y_p = y_a} d\left(\mathbf{f}_a, \mathbf{f}_p\right) - \min_{y_p \neq y_a} d\left(\mathbf{f}_a, \mathbf{f}_n\right) \right]_+$$

CNN Pruning Techniques

The objective of pruning is to remove unnecessary parameters from a neural network. For channel pruning, the objective is to remove all the parameters of a channel (output or input). Removing these parameters is done to reduce the complexity of network while trying to maintain a comparable accuracy.

Table 1: A Taxonomy of techniques according to pruning strategyto reduce chanels.

Pruning Strategy	Methods
Prune Once	Hao Li[31] Redundant Channels[40] Entropy[32]
Iterative Pruning	Molchanov[30] Play and Prune[41] FPGM[42]
Pruning using regularization	Auto-Balance[43] Play and Prune[41]
Pruning by minimizing reconstruction error	ThiNet[44] Channel Pruning[33]
Progressive Pruning	PSFP[34]

Pruning neural networks comes with many challenges. The first major challenge is the pruning criteria. The criteria need to be able to discern the parameters that contribute to the accuracy and the ones that do not. The second major challenge is finding an optimal pruning compression. This compression ratio is essential since we need to find a compromise between the reduction of complexity for model and the loss of accuracy. The third and last challenge is the retraining and pruning schedule of the model. The pruning could be done in one iteration, but the damage done to the network will be considerable.

Comparing Algorithms

The PSFP pruning scheme is very interesting since the model keeps its original dimension during the retraining phase. The authors also proposed to add a progressive pruning scheme where at each pruning iteration, the compression ratio is increased in order to get a shallower network. Once these iterations of pruning and retraining are complete, they do a last channel ranking using a pruning criterion and they discard the lowest channels depending on the compression ratio. Their pseudo code for the progressive soft pruning scheme can be viewed in algorithm 1.

Algorithm 1 Algorithm Description of PSFP

Input: training data: $\mathbf{X}$
Input: pruning rate: P_i, pruning rate decay D
Input: the model with parameters $\mathbf{M} = \{\mathbf{M}^{(i)}, 0 \leq i \leq L\}$
  Initialize the model parameter $\mathbf{M}$
  for $epoch = 1; epoch \leq epoch_{max}; epoch++$ **do**
    Update the model paramters $\mathbf{M}$ based on X
    for $i = 1; i \leq L; i++$ **do**
      Calculate the l_2-norm for each channel
      Calculate the pruning rate P' at this epoch using P_i and D
      Select the N lowest l_2-norm depending on the pruning rate
      Zeroize the weights W of the selected channels
    end for
  end for
  Obtain the compact model with parameters **M'** from $\mathbf{M}$
Output: Compact model with parameters M'

Here, they used the L1 or L2 norm of the weights as a pruning criterion which means this method could be categorized as a weight-based method. The L represents the number of layers in the model, i represents the layer number, W represents the weights of a channel and N is the number of channels to pruned. The pruning rate P0 is calculated at each epoch using the pruning rate goal Pi for the corresponding layer i and the pruning rate decay D.

FPGM is a new technique that focuses on using geometric median to prune away output channels. The algorithm of FPGM for the progressive soft pruning scheme can be viewed in algorithm 2. can be summarised as Play and Prune, an adaptive output channel pruning technique, that, instead of focusing on a criterion, tries to find an optimal number of output channels that can pruned away given an error tolerance rate.

Algorithm 2 Algorithm Description of FPGM

Input: training data: $\mathbf{X}$
Input: pruning rate: P
Input: the model with parameters $\mathbf{M} = \{\mathbf{M}^{(i)}, 0 \leq i \leq L\}$
  Initialize the model parameter $\mathbf{M}$
  for $epoch = 1$; $epoch \leq epoch_{max}$; $epoch++$ **do**
    Update the model parameters $\mathbf{M}$ based on X
    for $i = 1$; $i \leq L$; $i++$ **do**
      Select the $n_{out} \times P$ of W_i channels that satisfy Equation 22
      Zeroize the selected channels
    end for
  end for
  Obtain the compact model with parameters $\mathbf{M}$' from $\mathbf{M}$
Output: Compact model with parameters M'

This technique is min-max game of two modules, The Adaptive Filter Pruning (AFP) module and the Pruning Rate Controller (PRC). The goal of the AFP is to minimize the number of output channels in the model while the PRC tries to maximize the accuracy of the remaining set of output channels. This technique considers a model M can be partitioned into two set of important channels I and unimportant channels U.

Table 3: Comparison of rank-1 accuracy and network complexity analysis in term of GFLOPS and Parameters taken from the literature.

Dataset	CIFAR10					
Feature Extractor	ResNet56					
Algorithm	Original			Pruned		
	R-1 (%)	GFLOPS	Parameters (M)	R-1 (%)	FLOPS (G)	Parameters (M)
Hao Li [31]	93.04	0.125	0.85	93.06	0.091	0.73
Auto-Balanced [43]	93.93	0.142	N/D	92.94	0.055	N/D
Redundant channel [40]	93.39	0.125	0.85	93.12	0.091	0.65
PP [41]	93.39	0.125	0.85	93.09	0.039	N/D
FPGM [42]	93.39	0.125	0.85	92.73	0.059	N/D

Dataset	ImageNet					
Feature Extractor	VGG16					
Algorithm	Original			Pruned		
	R-1 (%)	GFLOPS	Parameters (M)	R1 (%)	GFLOPS	Parameters (M)
ThiNet [44]	90.01	30.94	138.34	89.41	9.58	131.44
Molchanov [30]	89.30	30.96	N/D	87	11.5	N/D
HaoLi [31]	90.01	30.94	138.34	89.13	9.58	130.87
Channel Pruning [33]	90.01	30.94	138.34	88.1	7.03	131.44

Dataset	ImageNet					
Feature Extractor	ResNet50					
Algorithm	Original			Pruned		
	R-1 (%)	GFLOPS	Parameters (M)	R-1 (%)	GFLOPS	Parameters (M)
Entropy [32]	72.88	3.86	25.56	70.84	2.52	17.38
ThiNet [44]	75.30	7.72	25.56	72.03	3.41	138
FPGM [42]	75.30	7.72	25.56	74.83	3.58	N/D

The main difference between weight-based methods and feature map-based methods is that weight-based methods are not dependent of the dataset since weight statistic do not depend on output of a convolutional neural network. Whereas, feature map-based methods need a dataset in order to compute whether the output of convolution layer or its gradients.

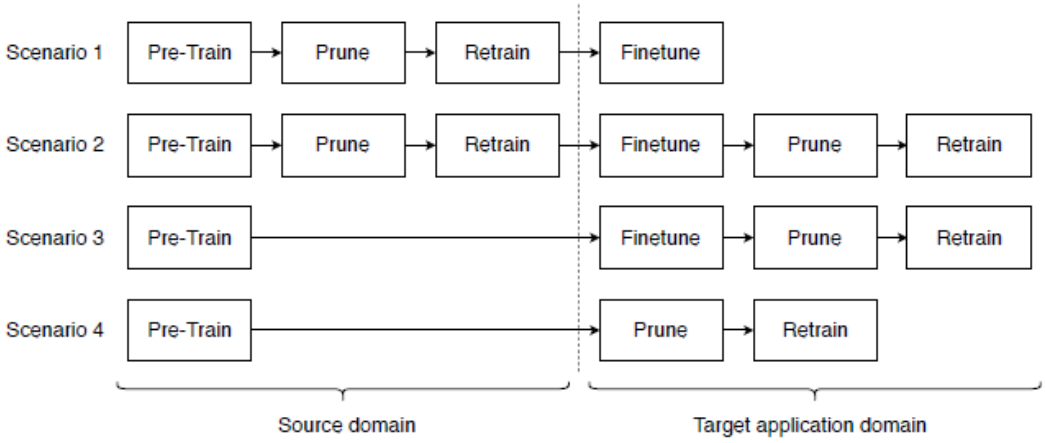


Figure 3: Scenarios for pruning and training a CNN.

The chosen criteria usually depend on the desire to simplify the pruning steps for a loss of accuracy compared to some more complex criteria that requires a lot more computations to be able to keep a high accuracy. If training and pruning time is an issue, i.e. in an environment that requires fast deployment, it is more adapted to choose some simple criteria like L1 and L2 norm. But if there is no time constraints, some more complex pruning criteria like the minimization in the difference of activation or cost function seems to outperform the simple criteria but will require a lot more computations and time.

Datasets

Four publicly available datasets are considered for the experiments, namely Imagenet, Market1501, DukeMTMC-reID and CUHK03-NP. Imagenet, a larg-scale dataset, is used as pre-trained dataset and rest of the other datasets (small-scale) are used for the experiments of person re-identifications.

ImageNet (ILSVRC2012) is composed of two parts. The first part is used for training the model and the second part is used for validation/testing. There is 1.2M images for training and 50k for validation. The ILSVRC2012 dataset contains 1000 classes of natural images.

Market-1501 is one of the largest public benchmark datasets for person re-identification. It contains 1501 identities which are captured by six different cameras, and 32,668 pedestrian image bounding-boxes obtained using the Deformable Part Models (DPM) pedestrian detector.

CUHK03-NP consists of 14,096 images of 1,467 identities. Each person is captured using two cameras on the CUHK campus and has an average of 4.8 images in each camera. The dataset provides both manually labeled bounding boxes and DPM-detected bounding boxes.

DukeMTMC-reID is constructed from the multi-camera tracking dataset DukeMTMC. It contains 1,812 identities. We follow the standard splitting protocol proposed in where 702 identities are used as the training set and the remaining 1,110 identities as the testing set.

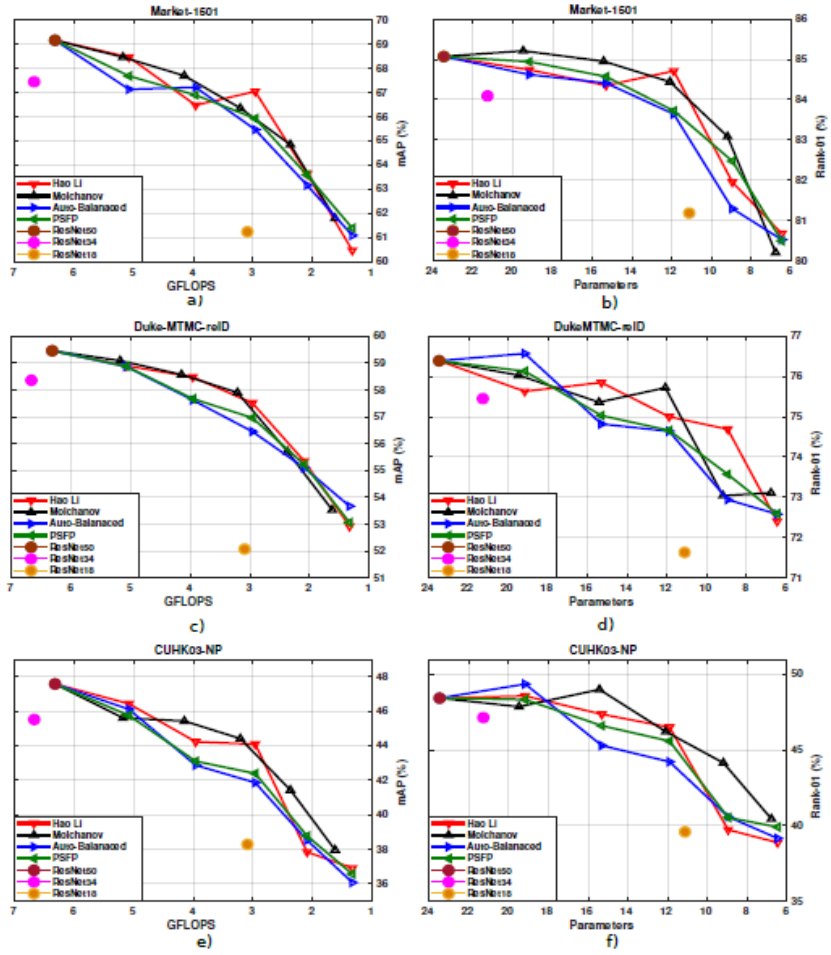
Performance Analysis

Table 5: Accuracy and complexity of baseline and pruning Siamese networks on ReID datasets. Mean average precision (mAP) and rank-01 accuracy (R-1) are shown in percentage (%).

Networks	Parameters	GFLOPS	Market-1501		DukeMTMC		CUHK03-NP	
			mAP	R-1	mAP	R-1	mAP	R-1
ResNet50	23.48	6.32	69.16	85.07	59.46	76.39	47.57	48.43
ResNet34	21.28	6.67	67.44	84.09	58.36	75.45	45.51	47.14
ResNet18	11.12	3.09	61.23	81.18	52.07	71.63	38.27	39.57
HaoLi	11.90	2.96	67.04	84.71	57.51	75.00	44.08	46.50
Molchanov	12.09	3.21	66.35	84.44	57.90	75.72	44.40	46.21
AutoBalanced	11.90	2.96	65.46	83.64	56.45	74.64	41.85	44.21
Entropy	11.90	2.96	65.16	82.39	56.64	74.64	42.44	44.07
PSFP	11.90	2.96	65.92	83.72	56.96	74.66	42.38	45.58

Table 5 reports the results for Market-1501, DukeMTMC-reID and CUHK03 NP re-identifications. The reported results are for Scenario 1. Molchanov has higher FLOPS and a higher number of parameters than the other method which would probably lead to a slower model and more consuming in terms of memory. Out of the 5 methods, the Hao Li method seems to be working the best by having the best or close to the best on the three datasets.

To get a more global view of these results, we refer to the supplementary material to see the complete result tables for each pruning iteration. Plus, the graphics in Figures 4 also shows us visually which models are better where the optimal placement would be top right and the worse would be bottom left. There are two graphics for each dataset where the first one presents the mAP vs FLOPS and the second one presents Rank1 vs Parameters.



The experiment shows that we could limit the effects of pruning by using a layer by layer approach and freezing the other layers to regain the accuracy. The problem with this scheme is that it's not very effective time wise since it's a tedious task to prune and retrain to the desired compression ratio for each layer instead of doing the whole model in one pass.

Conclusion

We discussed about different state-of-art pruning approaches suitable for compressing Siamese networks for person Re-identification application for the criteria selecting channels, and of strategies to reduce channels. Along with that the we are proposing different pipelines for integrating a pruning method during deployment of network for the application.

Experimental evaluations on multiple benchmarks source and target datasets indicate that pruning can considerably reduce network complexity (number of FLOPS and parameters) while maintaining a high level of accuracy. Moreover, pruning larger CNNs can also provide a significantly better performance than

fine tuning the smaller ones. A key observation from the scenario based experimental evaluations is that both fine tuning and pruning should be performed in the same domain.

#Deep Learning #Convolutional #Neural Networks #Siamese Networks #Complexity #Pruning #Domain #Adaptation #Person #Re-identification

References

Original Paper - https://arxiv.org/abs/1907.02547

J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, K. Murphy, Speed/accuracy trade-offs for modern convolutional object detectors, in: CVPR 2017.

E. Ahmed, M. Jones, T. K. Marks, An improved deep learning architecture for person re-identi_cation, in: CVPR, 2015.

A. Hermans, L. Beyer, B. Leibe, In defense of the triplet loss for person reidentification, arXiv, 2017.

R. R. Varior, M. Haloi, G. Wang, Gated siamese convolutional neural network architecture for human re-identification, in: ECCV, 2016.

W. Chen, X. Chen, J. Zhang, K. Huang, Beyond triplet loss: a deep quadruplet network for person re-identification, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

M. Geng, Y. Wang, T. Xiang, Y. Tian, Deep transfer learning for person reidentification, arXiv, 2016.

D. Cheng, Y. Gong, S. Zhou, J. Wang, N. Zheng, Person re-identification by multi-channel parts-based cnn with improved triplet loss function, in: CVPR, 2016.