Improvements Upon the Multiple Granularities Network for Person ReID

Nick McKillip
Texas A&M University
nickmckillip@tamu.edu

Jeffrey Cordero
Texas A&M University
jeffreycordero@tamu.edu

Josiah Coad
Texas A&M University
josiahcoad@tamu.edu

Abstract

In camera surveillance, when presented with a person of interest (query), person reidentification (PRID) returns to us other instances where that particular person has been observed before, be it another place or time. The pictures may be from many angles, lighting and locations. We propose a solution to PRID that uses a Multiple Granularity Network as the baseline and builds upon it using mutual learning, multiple training sets, AlignedReID based local parts alignment, and spatial transformation networks. Our model achieved 85.99% mAP and 94.67% top-1 on a new challenging test set provided by Walmart.

1. Introduction

Here we overview the problems traditionally faced by the PRID retrieval task and present a brief literature review of top performing models such as the Multiple Granularities Network (MGN) and AlignedReID. We also briefly review the various applications of PRID and how a model's performance is quantified.

1.1. Problem Description

"To reidentify a particular, then, is to identify it as (numerically) the same particular as one encountered on a previous occasion" by Alvin Plan Tinga (1961). In camera surveillance, when presented with a person of interest (query), person reidentification (PRID) tell us whether the particular person has been observed before, be it another place or time. PRID seeks to reidentify a person from many angles, lighting and locations. The large variance between possible arrangements of the same identity make this a nontrivial problem.

1.2. PRID Applications

PRID could prove valuable for many applications, including a contribution to increasing demand for public safety by spotting a person-of-interest, and tracking a suspicious person across multiple cameras. The unique ability for a model trained for the task of PRID is the identification of people across many backdrops and locations, in varying poses and activities. PRID can be applied in a theme park, for example, to find a lost child. If the child has no type of locating device on them, the caretakers could provide an image to the security department, and that could be used to retrieve places around the park where the child was seen last.

1.3. Challenges

Traditional challenges of PRID include occlusions, varying scales, low image resolutions, light changes, differing viewpoints, differing poses, similar clothing, tiny faces, and privacy preservation, amongst others. Furthermore, training sets often are too small, leading to overfitting in larger model designs better suited to complex classification.

Figure 1 demonstrates the challenges with PRID to identify people across multiple poses. In the queries shown in Figure 1 it can be seen that query images of people on bikes are hard for the model to learn. This is because people on bikes are assuming poses nontypical to humans. Also, multiple people in the same image present occlusions to each other, melding into difficult cases for a classification network. The figure also presents a visualization of other challenges, including varying viewpoints and similar clothing. Challenges like these make PRID on real life data a nontrivial problem.

1.4. PRID Scoring

We propose an information retrieval model using mean average precision (mAP) as the standard performance metric. Our model uses triplet loss functions (varying between semi-hard and hard) combined with softmax for training. To access our accuracy, we calculated the mean of the average precision for all query images. To do so, we built a distance matrix ranking image similitude. Average precision works by taking the area under the curve of a precision-recall graph. For a given query image, the results are returned in ranked order along with their ground truth labels. Considering only the first returned result as a match for a query image, recall will likely be poor (as their will likely be more than one matched person in the gallery that we are

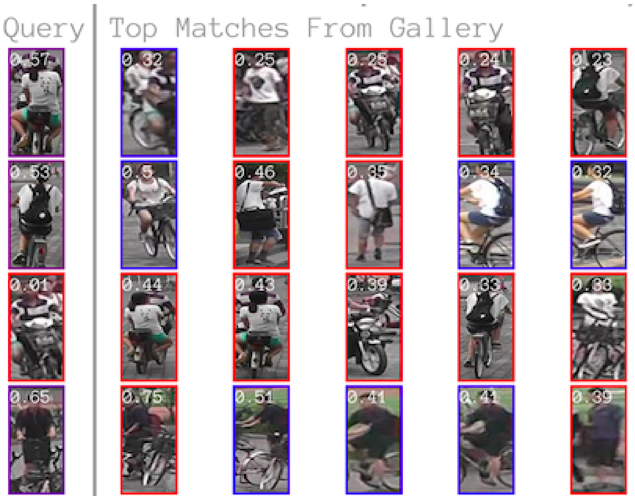


Figure 1. The performance of our model after 47 epochs on Market1501 data, on some especially difficult query images. Red bounding-boxed gallery images represent incorrect person retrievals. For a more detailed explanation of how to interpret the picture, please refer to part 4.6. Predicting Visualization.

not including); but precision could be high (if the top result is a correct match, the precision will be 1). As we consider more top-ranked results, thus increasing recall, our precision will likely decrease due to the inclusion of incorrectly matched identities. Recall and precision are thus often considered to be inversely related. This phenomenon creates a curve, the area under which is called average precision, and which ranges from (0, 1), with 1 being most desirable. The average precision is calculated in this way for each query image and the mean is taken across all query images to find the mean average precision.

We further measure the performance of our model using top-n scoring, namely top-1, top-5 and top-10. Top-n scoring for a query image is scored as 1 if the matching identity is in the top-n results for that query image, and 0 otherwise. Top-n for the entire data set is thus the average top-n score across all query photos.

2. Existing Person Reidentification Solutions

As person reidentification is a difficult problem, there are a large number of network architectures and further strategies that have been applied.

2.1. Loss Function

Older solutions relied on the use of double branched Siamese networks [1, 6, 9], one branch processing the source image and the other processing the image being compared. These networks typically implement the same base model, trained to have the same weights for both branches. More current solutions implement triplet loss networks [2, 3], a triple branched form of Siamese networks comparing the input image with both a positive and negative sample. These models typically implement a triplet loss function combining the output of each branch. The loss attempts to shorten the distance between the positive pair and increase the distance between the negative pair, thus providing a margin value that determines the classification probability.

Training triplet loss networks requires creating sets of images, with each set containing the base image as well as a positive and a negative sample. However, most triplet sets generated from the training set will be trivial after a small amount of training, an issue addressed by hard sample mining. Online hard sample mining [3, 12, 14, 16] implements some form of hard triplet loss function, selecting triplet sets from the batch after convolutional processing. These loss functions select the most dissimilar positive pair and the most similar negative pair, providing nontrivial training. Margin Sample Mining Loss [14] demonstrated further results using a loss function that maximized the constraints of hard sample mining, although used positive and negative pairs without the common image constraint.

2.2. Multiple Granularities Network

Our solution will expand upon the current state-of-the-art model for PRID, the Multiple Granularities Network (MGN) [12]. MGN is not only currently the state-of-the-art, but it is also relatively simple compared to other comparable networks which rely heavily on more complex image recognition segmentation to achieve their respective results. As MGN not only relied on triplet loss but also on softmax for classification, it built classes for each identity it encountered. To generalize this model to new data that it had never seen before, we made sure that the classes represented a wide range of identities by training on several benchmark datasets. For final classification, MGN applies a combination of a global and two local feature branches. Unlike other approaches, however, MGN does not focus on semantically coherent segmentation or pose estimation techniques. Rather, it segments the images into two and three equally divided horizontal stripes. Thus, to function optimally, the images should already be bounding-box cropped before being fed into the model.

2.3. AlignedReid

To overcome the preprocessing bounding-box assumption, we implement several solutions, the first of which is AlignedReID. AlignedReID [16] tries to align images that aren't necessarily pairwise comparable by first segmenting the images into horizontal strips then using a distance matrix between local slices of two images to align them, although they only use local features in training, finding little benefit to their use in validation testing.

2.4. Spatial Transformer Networks in Person-ReID

There have been previous attempts to apply spatial transformer networks to PRID. Li et al. [7] splits the image into three parts, similar to the third branch of MGN. For each section of the split, a spatial transformer network was inserted before the rest of the branch. Zheng et al. [17] applied a spatial transformer network after the fourth residual layer of a ResNet50 backbone.

2.5. Further Strategies

Many other image recognition strategies have also been applied to the problem of person ReID. As solutions frequently implement ResNet50 for the base model of multi-branch networks, pretraining on ImageNet [10] has been shown to reduce training time and improve overall network performance. [12, 16, 2] Data augmentation is frequently implemented using random erasing [2], removing portions of the image, forcing recognition from other image segments. Horizontal flipping [16, 12] and resolution adjustments [1] are also used. Some papers propose further innovations, AlignedReID [16] found success using mutual learning, and [15] used Domain Guided Dropout, a novel approach which varies dropout patterns corresponding to the current dataset.

3. Solutions Implementation

Our complete solution was designed to combine the local feature extraction method of AlignedReID with the Multiple Granularities Network. We also implemented a spatial transformer network at the beginning of the model so to help negate classification issues from image misalignment and deformation. Finally, this was combined with random erasing data augmentation, learning rate reduction on plateau, and the use of mutual learning to train models collectively.

3.1. MGN

As the person reidentification challenge involves achieving the highest mAP accuracy, our solution used the Multiple Granularities Network (MGN) [12], as implemented by [11] for our baseline (see Figure 4).

MGN is split into three different branches after ResNet50's third residual block. Copies of ResNet50's fourth and fifth blocks are created for each of the three branches, before being further split corresponding to the shape of later dimensions. Global max pooling is applied to the output of the residual blocks, in order to reduce the dimensionality of the features. The first branch only consists of global feature maps and is trained with softmax loss and triplet loss. Before applying triplet loss, a 1x1 convolution is used to reduce the number of feature maps. Branch two consists of a global feature and a copy of this global feature split into two local features. The loss for the second global branch is identical to the loss of the first branch, including the 1x1 convolution. 1x1 convolutions are also applied to the local feature maps. The split local features each have a fully connected layer with softmax loss. The third branch of MGN is identical to branch two except the global feature is split into three local features.

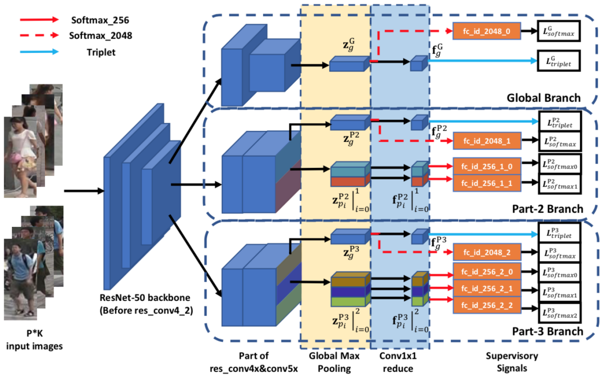


Figure 2. The standard Muliple Granularity Network architecture as presented in [12]. Notice the 1x1 convolutions for dimension reduction and fully connected layers do not share weights.

3.2. Data Augmentation

Overall, we experienced little overfitting due to the relative simplicity of the ResNet50 model for such a complex image classification challenge. As well, any overfitting was mostly alleviated by the inclusion of a second dataset. Therefore, the only data augmentation our solution used was random erasing. This completely removed portions of the training images by replacement with sections of noise, causing difficulty for both the global and local feature extraction layers.

3.3. Learning Rate Reduction on Plateau

During training we decayed the learning rate upon detection of plateauing of training loss. Each time the learning weight was decreased to 1/10th its previous value. A patience of five was used, meaning that we would reduce the learning rate if a minimum loss had not been found in more than five epochs.

3.4. Feature Extraction

Testing the network required generating both gallery and query feature matrices for the validation and test sets. These, as well as the corresponding image labels, were then evaluated using either the provided evaluation script or the test functionality built into the MGN baseline.

The feature matrices were generated by loading image batches through the model so to extract their corresponding features, that is, information extracted from the image and altered via the layers and weights of the network. During matrix construction, batches of 16 images were fed together through the model with each output layer extracting separate features for each image. These separate features were then combined to provide a single feature vector of size 2048 for each image in the batch. After all image batches had been processed, their outputs were combined to create a single matrix of shape (image count)x(feature count). This was finally combined with the labels and either tested with the library functionality or saved in a format accessible by the provided evaluation script.

3.5. Training on Multiple Datasets

In order to best prepare our network for the unseen test data we trained our network using a combination of data from Market1501 and DukeMTMC. We would switch training data batch by batch. During each iteration we would first run an image batch from Market1501 forward and backwards through the network. The same was then done for an image batch from DukeMTMC. Although DukeMTMC had more total images, we would end the epoch when we could no longer generate any more batches from Market1501 as to not bias the network towards either dataset.

3.6. Generalizing the Softmax Features

The softmax classifier, that is CrossEntropy for PyTorch, was applied as the MGN local classification technique. However, as the test set contained unseen subjects, we began by working to prove that the extracted local features could generalize to classify the unseen subjects. If so, a softmax based local feature classifier trained on a large and diverse enough set could therefore be applied successfully to real world use.

To test the generalizability of the softmax classification of local parts we experimented with training on two datasets, Market1501 and DukeMTMC. This involved increasing the softmax output dimension, that is the number of classes, to be able to classify features from both. This ensured the ability to successfully learn a larger feature set size and proved successful. When testing, we used the features prior to the fully connected layers to generate our testing matrix.

3.7. Aligned Parts

The local feature extraction of the MGN network relies on horizontal image partitions, that is ”parts”. Using 2-part and 3-part branches, MGN separates the outputs of the first four convolutional ResNet50 layers into 2 and 3 separate sections respectively, feeding each section through horizontal pooling layers as well as 1x1 convolution reductions before softmax classification. This parts-based softmax classification is used to learn features typically observed in the corresponding section of an image, for example, focusing on the head for the upper part, and legs and feet for the lower part classifiers.

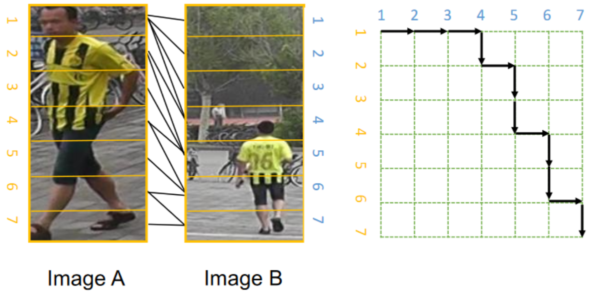


Figure 3. The N-parts alignment scheme proposed by AlignedReID [16]. Image from the AlignedReID paper.

This use of horizontal image partitions therefore presented a natural fit to apply the alignment technology designed by AlignedReID. This alignment solution compares the separate horizontal feature slices from two images, ”aligning” the most similar slices.

The implementation of the new aligned parts branch required both the addition of an N-parts local feature extraction layer to the MGN model along with the novel alignment based triplet loss presented by AlignedReID. However, unlike the original AlignedReID solution, our local parts were extracted like the other MGN local feature parts. Where AlignedReID extracted parts from the output of the final ResNet50 layer, our solution extracted them from the fourth convolutional ResNet50 layer before processing through horizontal max pooling and 1x1 convolutional reduction as outlined above. Unlike either paper, we developed the N-parts addition to the model such that the number of alignment parts was variable, allowing testing with varying numbers of parts.

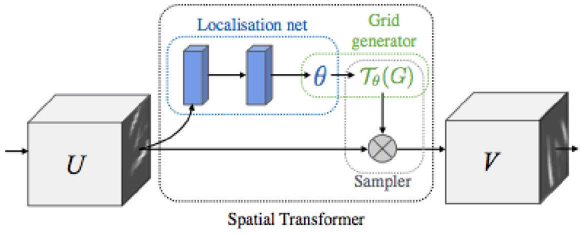
Implementation began by adding support for the aligned branch, that is, the corresponding script options and the ability to average the newly added loss into the existing total loss. The addition of the aligned parts loss function required firstly a custom hard triplet loss function. This was simply a version based on PyTorch’s MarginRankingLoss used to generate a loss value from two distance matrices, first as the distance between the anchor and the positive sample and the second as the distance between the anchor and the negative sample. The second addition was the alignment functionality used from the AlignedReID PyTorch codebase [4]. This alignment function normalized the inputs before generating the distance matrix between each input set. To do so, it calculated the Euclidean distance between every element in the input vector, that is, each part. Shortest distance, applied using dynamic programming, was then implemented to obtain the minimum distance between the features of each part, resulting in the ’alignment’ of most similar parts from each image pair. The aligned distance matrix was finally fed into the AlignedReID hard triplet loss function to acquire the final loss value.

3.8. Spatial Transformer Network

Spatial transformer networks were created to overcome the challenges of scale variation, viewpoint variation, and deformation. They are powerful because they are differentiable and can be inserted into any existing network.

In the global max pooling phase of the Multiple Granularities Network, the image features are split into two distinct parts in the second branch and three distinct parts in the third before a fully connected layer is added on top of each group of split features and used for softmax classification.

In two different images of the same person, these subsections of the image could be misaligned and limit the network's ability to correctly reidentify the person. To try and address this issue we implemented a spatial transformer network (STN) [5] at the beginning of our network. Images were first run through the STN before being fed into the rest of our network. We also experimented with inserting a STN after the last common ResNet50 block. We tried training both networks jointly from end to end but found better results when training our MGN adaptation by itself, then freezing its weights and training those of the STN alone.



The spatial transformer network we implemented consists of three components: a localization network, a grid generator, and a sampler. The localization network generates the parameters of an affine transformation matrix that is applied to the input. The input to this network is a 384x128x3 image, and the output is six parameters, reshaped into a 2x3 matrix. In order to reduce the large image down to six parameters, our localization network consists of six convolutional layers and is followed by three linear layers. We later implemented a second spatial transformer network following residual block three. The input shape to this network is 1024x24x8. This module consist of two convolutional layers followed by three linear layers. The last linear layer outputs six parameters, which are reshaped into a matrix of shape 2x3.

Using the matrix multiplication presented in Table 1, with the output of the localization network as parameters θ, the grid generator produces a set of points in the original image that should be sampled in order to produce the transformed output. X^s and Y^s tell us exactly where to sample our input pixels in order to generate the output pixels of the new image.

$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = \begin{bmatrix} \theta_{11}\theta_{12}\theta_{13} \\ \theta_{21}\theta_{22}\theta_{23} \end{bmatrix} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} \tag{1}$$

In most cases the resulting X^s and Y^s will not be integers and therefore will not correspond to an exact point in the input image. To deal with this the sampler uses bilinear interpolation to generate the output feature map. Bilinear interpolation is an extension of linear interpolation for interpolating functions of two variables on a rectilinear 2D grid. The key idea is to first interpolate in one direction, and then interpolate in the next direction. Bilinear interpolation preserves the differentiability of the network.

3.9. Mutual Learning

Mutual learning describes the process of simultaneously training multiple copies of the same model, two in our case, as to aid in the training process of each. Mutual learning involves the use of mutual loss, specialized loss functions that combine the results of all training models at the given batch so to produce a loss value dependent on the other model. Our version of mutual loss was adapted from that used in AlignedReID, including both mutual probability loss and mutual local feature loss in the computation. This combined the outputs of both models so that the mutual loss was then included in the loss value calculated for each model.

The models used are architecturally identical except for the weights, differing both at initialization and after training. This twin structure means that the models share the same search space of local and global optima that are able to be reached through training. However, the initialization with differing weights causes them to begin at divergent points in the search space and reach differing points throughout training. Mutual learning is therefore applied to aid in preventing both models from getting stuck in local optima, as well as generally improving training efficiency.

4. Results

As validating the improvements from each added technique is difficult when tested together, we present the results of experimenting with each technique separately when added to the baseline MGN model. This provided a clearer understanding of which approaches should be used when developing the final model. In the end, we chose a set of techniques that offered the best trade-offs between training difficulty and model performance when implementing the model used to evaluate features on the test set.

4.1. Generalizing the Softmax Classifier

We found using an additional data set significantly increased our model's performance on the validation set. By adding DukeMTMC data, all metrics improved for the baseline model, as seen in 1.

	mAP	Top-1	Top-5	Top-10
Market	0.8062	0.9133	0.9353	0.9867
Market & Duke	0.8470	0.9333	0.9800	0.9867

Table 1. Results of training the model only on market compared to training the model on Market and Duke.

4.2. Aligned Parts Branch

As we implemented the aligned parts branch using a variable number of parts, we began by testing the accuracy of varying numbers of parts. So to ensure the rest of the model was not affecting the experiment, this test only used the aligned parts branch. The results, presented in Table 2, represent the accuracy after epoch 50 trained on Market1501.

	mAP	Top-1	Top-5	Top-10
2 Parts	0.1229	0.3124	0.4991	0.5790
4 Parts	0.1394	0.3299	0.5190	0.6042
6 Parts	0.1548	0.3637	0.5508	0.6283
8 Parts	0.1447	0.3287	0.5309	0.6087

Table 2. Results of training the model only using varying number of parts in the alignment branch.

While the aligned parts branch did perform as intended when added to the rest of the model, as presented in Table 3, it only provided a small improvement to the overall model accuracy. This is most likely due to the bounding boxes of Market1501 and DukeMTMC not being variable enough to take advantage of part alignment. It is for this reason that six and eight part classification was not much more effective than two or four part in the tests. This inability to successfully train the alignment layer due to low bounding box variability led us to decide to leave it out of the final model.

	mAP	Top-1	Top-5	Top-10
Without A.P.	0.8092	0.9165	0.9406	0.9877
With A.P.	0.8104	0.9221	0.9425	0.9854

Table 3. Results of adding the aligned parts branch with 6 parts to the model.

4.3. Spatial Transformer Network

We were able to find a slight increase in performance when using the spatial transformer network. We found that inserting it before the rest of the network performed best. The results in Table 4 come from training on both Market1501 and DukeMTMC and are evaluated on the Walmart validation set. We believe results could be improved by using multiple spatial transformer network blocks at the beginning of our pipeline, in order to allow the network to more easily learn larger and more complex transformations.

	mAP	Top-1	Top-5	Top-10
STN at front	0.8491	0.9300	0.9867	0.9933
STN after conv3	0.8453	0.9250	0.9800	0.9867
Without STN	0.8470	0.9267	0.9867	0.9933

Table 4. Results on the validation set with inserted spatial transformer networks.

4.4. Mutual Learning

We used the baseline MGN model with Market1501 when testing the effects of training with mutual learning, so as to ensure the clearest results. As mutual learning involves training two models simultaneously, this required the use of a P5000 GPU, whereas a P4000 had been used in previous training.

	mAP	Top-1	Top-5	Top-10
Without ML	0.8092	0.9165	0.9406	0.9877
With ML	0.8115	0.9290	0.9481	0.9861

Table 5. Results of training the baseline with mutual learning.

While mutual learning did present slightly improved results on the validation set, we elected not to use it in the final model. This was because mutual learning over doubled training time due to our implementation not being parallelized across multiple GPUs. Therefore, to allow for running more tests, mutual learning was discarded for future tests.

4.5. Test Set Results

Table 6 represents our results on the provided test set using the final model design. As stated before, the techniques used were chosen as they provided the best trade-off between accuracy and training difficulty.

	mAP	Top-1	Top-5	Top-10
Test Set	0.8599	0.9467	0.9867	1.000

Table 6. Test results on the provided test set.

The final design implemented both global features and parts based local features with twice the global loss contribution than that of the local feature loss, with a margin value of 1.2 for hard triplet loss. The model included an STN module, trained from scratch after training the rest of the network by freezing all non-STN weights. The model was trained on both Market1501 and DukeMTMC, training the softmax classifiers on features from both sets. During training the learning rate was reduced to 1/10th its previous value on plateau. Finally, to ensure best accuracy, the network was trained for 300 epochs but was configured to save a trained model every time validation loss decreased. In the end, the model trained to 180 epochs preformed best and was used.

4.6. Prediction Visualization

Our model returns a ranked list of possible person matches for a given query image. As one would expect, these returned results become more accurate with more training. Figures 4 and 5 show this training in action. Shown on the leftmost of the figures are four query images from the Market1501 data set. The bounding box around the query images, ranging from red to blue, represent the performance the model had on that query image, with red being the worst performance. To the right of each query image are the top five results retrieved for that query from the gallery set. (There may be more matches from the gallery that are omitted due to space constraints.) For the gallery photos on the right side, the red/blue bounding box represents the ground truth about the image, with red representing an incorrect match to the query. Finally, the small number at the top of each query image is the average precision for that query; and the small number at the top of each gallery image can be interpreted as the confidence of our model on that image being a match. It can be seen that our model retrieval performance improves significantly from epoch 1 to epoch 50.

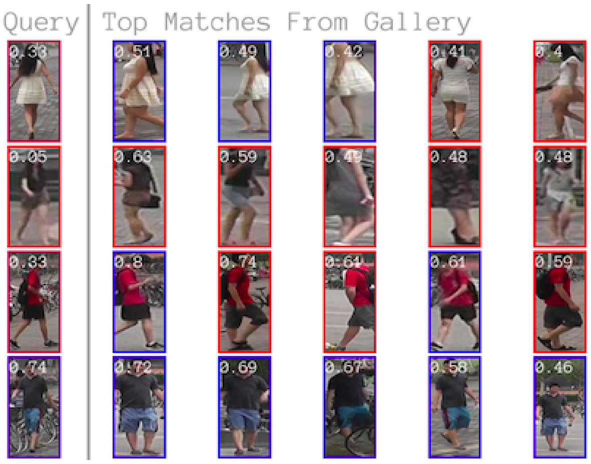


Figure 4. The performance of our model after 1st epoch of training on Market1501 data. The mAP after the first epoch was .295.

5. Potential Future Works

Going forward there are a set of changes we would be interested in investigating. Firstly, we would design and implement a global feature-based, mutual loss component, based off the global features of each network. This was not added to our current solution, as the method presented by AlignedReID didn't fit our global extraction method.

Secondly, we believe that the spatial transformer network is not preforming to its full potential. Therefore, we would like to investigate possible improvements by replacing our single deep spatial transformer block with multiple, but shallower, spatial transformer blocks. Such stacked smaller STNs may provide better results than the single, large STN block.

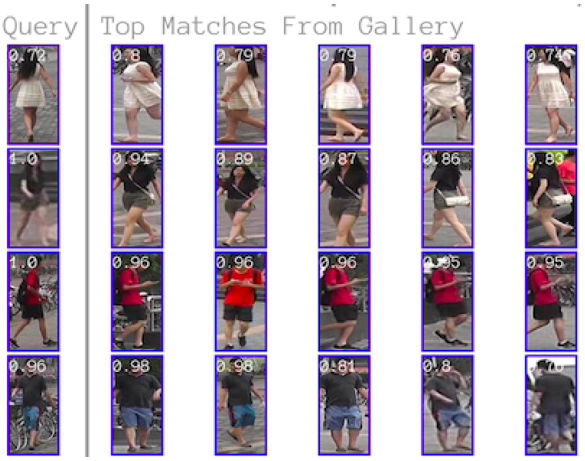


Figure 5. The performance of our model after 50 epochs on Market1501 data. The mAP after the 50th epoch was .807.

As ResNet50 is considered a relatively small model for image classification problems, we believe that switching to a larger convolutional model would provide a major increase in classification accuracy. This would involve testing ResNet101, ResNet152, DenseNet121, DenseNet169 and DenseNet201 and would also require increasing the training set size via addition of the CUHK03 [8] and/or MSMT17 [13] datasets to prevent overfitting due to size.

References

[1] E. Ahmed, M. J. Jones, and T. K. Marks. An improved deep learning architecture for person re-identification. In *CVPR*, pages 3908–3916. IEEE Computer Society, 2015.

[2] J. Almazán, B. Gajic, N. Murray, and D. Larlus. Re-id done right: towards good practices for person re-identification. *CoRR*, abs/1801.05339, 2018.

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

[4] H. Huang. Alignedreid-re-production-pytorch, Dec. 2017.

[5] M. Jaderberg, K. Simonyan, A. Zisserman, and k. kavukcuoglu. Spatial transformer networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2017–2025. Curran Associates, Inc., 2015.

[6] G. Koch, R. Zemel, and R. Salakhutdinov. Siamese neural networks for one-shot image recognition. 2015.

[7] D. Li, X. Chen, Z. Zhang, and K. Huang. Learning deep context-aware features over body and latent parts for person re-identification. *CoRR*, abs/1710.06555, 2017.

[8] W. Li, R. Zhao, T. Xiao, and X. Wang. Deepreid: Deep filter pairing neural network for person re-identification. In *CVPR*, 2014.

[9] X. Qian, Y. Fu, Y. Jiang, T. Xiang, and X. Xue. Multi-scale deep learning architectures for person re-identification. *CoRR*, abs/1709.05165, 2017.

[10] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[11] D. Wang. Mgn-pytorch, 2018.

[12] G. Wang, Y. Yuan, X. Chen, J. Li, and X. Zhou. Learning discriminative features with multiple granularities for person re-identification. *CoRR*, abs/1804.01438, 2018.

[13] L. Wei, S. Zhang, W. Gao, and Q. Tian. Person trasfer gan to bridge domain gap for person re-identification. In *Computer Vision and Pattern Recognition, IEEE Conference on*, 2018.

[14] Q. Xiao, H. Luo, and C. Zhang. Margin sample mining loss: A deep learning based method for person re-identification. *CoRR*, abs/1710.00478, 2017.

[15] T. Xiao, H. Li, W. Ouyang, and X. Wang. Learning deep feature representations with domain guided dropout for person re-identification. *CoRR*, abs/1604.07528, 2016.

[16] X. Zhang, H. Luo, X. Fan, W. Xiang, Y. Sun, Q. Xiao, W. Jiang, C. Zhang, and J. Sun. Alignedreid: Surpassing human-level performance in person re-identification. *CoRR*, abs/1711.08184, 2017.

[17] Z. Zheng, L. Zheng, and Y. Yang. Pedestrian alignment network for large-scale person re-identification. *CoRR*, abs/1707.00408, 2017.