Image Segmentation And Clustering Based Approach For Person Re-identification

Ankit Ramchandani, Benton Guess, Ryan Wells
Texas A&M University
{ankit61, bguess10, rawells14}@tamu.edu

Abstract

Person re-identification (PReID) is the problem of finding all instances of a given person in a gallery of images. It has recently gained significant attention due to its applications in intelligent video surveillance and monitoring tasks. Most current methods suffer from problems of occlusion, as they don't explicitly or exactly remove the background or occluding objects. This causes poor performance as models learn features from irrelevant parts. We propose a novel deep learning approach to look at the PReID problem. We view this as supervised clustering problem and suggest two models that can help do the same. To address background clutter and the occlusion problem, we train a segmentation network on low resolution images to blacken everything in the image except the person. This ensures we only learn from relevant parts and makes the job of the clustering network much easier.

1. Introduction

The cost, efficiency, and efficacy of continuously monitoring enormous networks of surveillance cameras is of significant concern to governments and large corporations [25, 11]. As part of monitoring, a common goal is the identification of a particular individual that has been previously seen across a network of cameras. This task, known as Person Re-Identification (PReID), is defined as finding a given person, the query, in a set of images known as the gallery. For a long time, human-based monitoring has been the primary, if not the only solution. However, this can be error prone, inefficient, and highly costly [39]. As the number of cameras operating at a given moment increase, governments and corporations must turn towards an automated approach to PReID to increase accuracy, improve efficiency and of course, reduce costs. Through the increasing availability of large open datasets and recent advancements in machine learning models, specifically, deep learning, vast improvements have been made in the performance for this task.

1.1. Challenges

Figure 1. Challenging Scenarios in Market-1501 dataset [49]. Left: Occlusion. Middle: Low Resolution. Right: Pose Variation

An important challenge with PreID is the occlusion of a person by obstacles within the photo that obscure the features of that person. This not only reduces the amount of information that describes the person, but also increases the amount of irrelevant information that could distract the model from only learning the features of the person.

Another challenge present is training and testing on low resolution and varying sized images. The images have low resolution because the person bounding boxes are extracted from images usually taken by mediocre CCTV cameras. This makes learning facial features and more detailed attributes of a person very difficult, if not, impossible. Thus, methods almost always cannot rely on facial features, as the faces are generally not visible due to pose variation or have very poor resolution when zoomed in. Lastly, since images are usually taken from multiple cameras at different times, the ambient light, person pose, background, and camera perspective can change dramatically. We display some typical examples of these common problems in Figure 1.

One more detail is that PReID can't be naively thought of as a classification problem as the identities seen in training time are completely different from the ones seen at test time. Due to this, models need to learn what makes two people similar/different rather than map images to predetermined identities. This brings PReID closer to the problems in the realm of one-shot learning, though some approaches, with some effort, treat PReID as more similar to classification as

we will touch upon in section 2.

All these problems make PReID quite a difficult challenge, yet of great value as we explore in the next subsection, to solve.

1.2. Importance and Applications

PReID's greatest value is in the field of video surveillance, which has been predicted to increase tenfold from 2015-2020. [20] PReID's applications can thus help improve security for businesses and public safety dramatically by tracking and obtaining additional images of persons of interest. It also has applications in robotics to help robots find owners or known people, in image analysis to improve image tagging on social media, and in security to perform behavioral analysis on where individuals of interest move. [11] [22]

1.3. Paper Structure

In section 2, we talk about some related research in the field of PReID, clustering and segmentation, highlighting some works that inspired ours. Then, in section 3, we explain our idea in detail, explaining strategies used for training and testing. We then 4 touch upon the experimental results, followed by a detailed analysis 5 of those results. Finally, we conclude 6 and present some ideas for future work 7 while also talking about the importance of our ideas in full generality.

2. Related Work

One successful approach to PReID compares discrete images through pairwise or triplet loss[17, 5, 4], often applied over a Siamese network or a similar structure [25]. These methods structure themselves by treating PReID as a clustering problem. Others have approached this as a classification problem, training deep learning networks to identify separate classes representing unique people [18, 26]. Other methods apply re-ranking techniques and use multiple queries of selected images to identify the identity of a single image [34]. Some solutions even make use of multiple frames of video as input to a network [44, 50]. Among other benefits, this can increase the Top K accuracy of the algorithm as it can cleverly combine the best matches for individual frames. This also makes the algorithm more robust to its own errors, as it would be unlikely to make the same error on all input frames.

2.1. Clustering

One approach taken in this paper is to treat PReID as a clustering problem building off of works such as deep embedded clustering (DEC) [19, 45] and improved DEC [14]. Clustering, at least in the context of deep learning, refers to the attempt to try to classify similar samples by observing their relative distance from one another in a feature space. These networks make use of a stacked, denoising autoencoder [42, 43] with KL divergence as the loss function, evaluated by a statistical distribution generated for each cluster through k-means. Furthermore, a work by Shukla et al. [37] also takes into account the KL divergence of the pairwise embeddings and the autoencoder's reconstruction loss. Shukla and Dizaji [8] also support reconstruction loss as a necessary method of preventing overfitting for a given data set. Our work builds upon these ideas by ensuring we only learn features from relevant parts and makes them more concrete by applying them to PReID.

2.2. Siamese Networks

Siamese networks are a class of neural networks which take two inputs at once instead of taking one. They then run both the inputs through two sub-neural networks (generally, with shared weights) to compute two features. These two features are then compared by some distance metric. The idea is to minimize this distance when the two inputs are similar and maximize it otherwise. There are two major advantages of a Siamese network's structure over others: the ability to guarantee that an image will be encouraged to always map closer to the images more similar to itself [23] and that the network can be very useful for one-shot learning problems because of its inherent nature to learn similarly instead of hard labels/classes. These networks were originally applied to situations where similarity was directly being measured, such as with signatures [2] or facial identification [40]. Specifically for PReID, Siamese networks' structures encourage similar identities to map near one another in feature space. Ideally, the loss function is representative of a test time metric used to determine similarity, such as L2 norm. [46, 23]

As would be detailed in section 3, we explore an approach to clustering by making use of a Siamese network to address the problem of identifying person identities using L2 norm as the discriminatory metric. Recently, some modern methods have chosen to use this technique specifically for the purpose of PReID. [6, 35, 30] Siamese networks ideally create a feature space which effectively allows one to identify if a given image has the same person as in the query. [1] Recently, Siamese networks have made use of residual networks as a backbone, such as with Du et al. [9], which is also the backbone used in the network showcased in this paper.

2.3. Image Segmentation

Segmentation relates to PReID by helping address the inherent bias and occlusion that is caused by non-human objects and background present in the frame. [41] One solution is to eliminate the variation by using a generative network to map a person to some ideal pose [28], or taking ad-

vantage of a network that can understand the structure of the pose [47]. Ma et al. go further by proposing a system that uses both pose and image segmentation to fully "disentangle" the person with respect to the background [29]. More recently, Lie et el. have also suggested attention networks as a valid method for eliminating background influence [27]. However, there has been recent, state of the art success in PReID by using image segmentation [21], that has shown to produce impressive results. As other networks show, the background will cause some bias within the final output [41], and eliminating this extraneous information from the bounding box will lead to higher accuracy [21]. Image segmentation, focused on separating human body parts from their backgrounds, has shown great potential in creating effective networks for this task [48, 21]. We expect binary image segmentation to give our clustering-based network a higher accuracy, while reducing background bias.

Specific to our work, we make use of a residual network backbone [16], to address the problem of image segmentation. Recent image segmentation techniques argue that end-to-end, full image samples forwarded through a fully convolutional residual network [31, 3] lead to the best results.

3. Approach

As stated earlier in section 2, we view PReID as a clustering problem for the following two reasons. One, clustering is more general than classification because if we know the clusters, we can say that each cluster can be its own class. Note that this would work with an arbitrary number of classes because we are not trying to put an image in a class, but rather trying to define meaning in the feature space so the features of all similar images are close to each other.

As has been said before in section 2, a vital challenge in the PReID problem is reducing bias and features learned from the background of an image. [21]. One commonly used solution to this problem is to augment data by random erasing, which basically blackens a part of the image and still forces the network to learn the similarity between different images of the same person. In this work, we use a binary image segmentation network to "pre-process" images first to blacken the background in the image, leaving only the person. This is a much richer solution than random erasing as there is no randomness involved.

We also present two models to do clustering. In one model, we try to do supervised clustering with an unknown number of clusters to begin with. This was a two-phase network comprising of a convolutional autoencoder and a regular convolutional network, henceforth called the clustering network. This is similar to DEC [45] in structure, but more general and different in purpose since DEC assumes that the number of clusters are known in advance. The second model used is a Siamese network as it, in principle, can also do clustering. Note that the input to both these models is the segmented image(s). The details of each model will be given in the following subsections.

3.1. The segmentation network

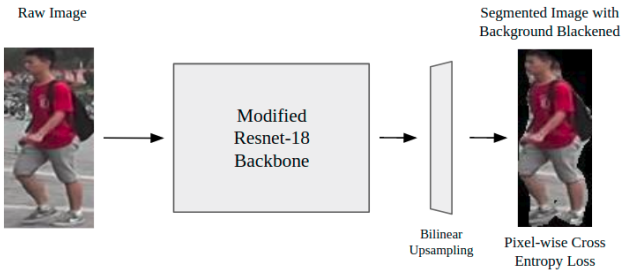

Figure 2. The architecture of the Segmentation Network Submodule

We used a segmentation model with the backbone of a modified ResNet-18 from a previous work by Pakhomov D. et al. [32] in binary image segmentation. Their network differs slightly from the usual ResNet-18: they use a different dropout per layer, extra stride, and an extra up-sampling layer after the final layer. Regardless, their network performs very well for medical image segmentation given both single class and multi-class segmentation queries, while using minimal resources in terms of GPU memory and processing requirements.

We modified their loss function such that we penalize the network more if it erroneously classifies a person as background, but not that much if background is classified as a person (equation 1). The reason behind this is that we do not, in any circumstance, want the image segmentation network to cut the parts of a person and lose useful information, but we can sustain the network not getting the sharp borders of a person. Simply put, a false positive is much more tolerable than a false negative. The loss function is as follows:

$$L = -\frac{\alpha p log(x) + \beta(1-p) log(1-x)}{\alpha + \beta} \quad (1)$$

Where α and β represent the relative weights of the foreground and background class respectively, with the values of 3.0 and 1.0 respectively. p represents the current class (foreground versus background).

With this modified loss function and just the ResNet-18 backbone, the segmentation network surprisingly does a very good job. Results are shown in section 4.

3.2. CAE and Clustering Network

The CAE and the clustering network are designed to perform supervised clustering with an arbitrary number of clusters. The CAE's architecture comprises of a ResNet-18 encoder and a decoder which is symmetric to the encoder, except all convolution layers are replaced by transpose convolution layers.

The clustering network takes the hidden layer of the CAE as input. We attempted to use many different clustering networks, ranging from a simple 8 layer convolution network to ResNet-101.

The reason for adding the clustering network is as follows. We want to first force the network to compress information of an image (done via the CAE), to remove any redundant or non useful information. We could have (and actually even tried initially but observed terrible performance) applied the clustering loss on the latent space of the CAE directly without using the clustering network, but decided against it because we don't want the final feature space to have all the information about the image. We want it to have the most prominent and useful information about the person. This is the job of the clustering network. The clustering network takes a compressed, but complete representation of the image and retains and finds the most useful information.

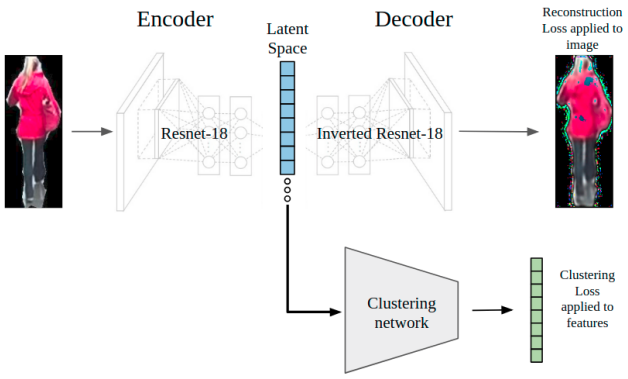

Figure 3. The architecture of the CAE + Clustering

3.2.1 Training Phase

It turns out that training this network was incredibly difficult. The reason may be easier to see after looking at the loss function. Basically, the total loss function of the network was the sum of clustering loss and reconstruction loss.

Mathematically, assume that the CAE is currently processing I_j^i, the j^{th} image of the i^{th} person, and has processed $\forall_{k\in[1,j)} I_k^i$. Let the feature output of the clustering network for I_j^i be C_j^i and the reconstructed image (by decoder) be R_j^i. Then, the total loss function is:

$$L = |R_j^i - I_j^i| + ||C_j^i - \bar{C}^i||_F^2 \tag{2}$$

where,

$$\bar{C}^i = 1/j \sum_{k=1}^{(} j-1) C_k^i \tag{3}$$

$|.|$ is the L1 norm [1] and $||.||_F^2$ is the squared frobenius norm. Note that the first term is the reconstruction loss which should implicitly maximize distance between images in latent space so the decoder's job is easy, the next is clustering loss which minimizes intraclass distance.

For the clustering loss, note that $\bar{C}^i$ changes every time after I_j^i is processed. This means that the loss landscape keeps changing after every update (similar to the case of GANs [13]). This makes training such a network so hard because the loss function is so dynamic and unstable. To work around this, we used lazy updates. So basically, $\bar{C}^i$ does not get updated after every batch, but only after a certain number of batches, 130 in our case. We also noted that as the network trains for many epochs, the estimate of $\bar{C}^i$ becomes inaccurate because the very initial (when network was relatively untrained) C_j^is are also contributing to the value of $\bar{C}^i$. To work around this, we reset $\bar{C}^i$ every few epochs, 20 in our case. This ensures that a trained network does not get penalized for the mistakes it made when it was untrained.

We also observed that it was really hard to train the entire network (CAE and clustering network) end to end. After many many tries, we couldn't make either the CAE or the clustering network converge when both were trained end to end. So, we decided to train just the CAE with only the reconstruction loss first and this worked very well for the CAE (Figure 4.2). To train the clustering network, we froze the weights of the encoder and trained just the clustering network. Even after relentless effort and attempts, we observed that we could not make the clustering network converge with input being the hidden layer of our CAE. We analyze the reasons that the clustering network may not have converged in section 5.

3.2.2 Testing Phase

In the testing phase, the CAE's decoder is discarded. Given the query image, I_q, we run it through the segmentation network to remove background and then through the CAE and clustering network to get the final features (same as output

[1] In context of this paper L1 norm refers to the sum of the absolute values of all elements and not the maximum L1 norm of columns, which is indeed the technically correct definition of an L1 norm for matrices.

of the last layer of clustering), C_q. We also find the features of all images in the gallery. We return our top k predictions by finding the k nearest neighbors (distance measured by mean squared error, which is same as the one used while training) of the query image in this clustering space.

An interesting point is that we can also run an unsupervised clustering algorithm like DBScan [10] after converting all images in the gallery to their clustering space representation to find all people in the data set in one shot. Ideally, all images in one cluster should correspond to the same person. This is possible due to the use of clustering loss in training time.

3.3. Siamese networks

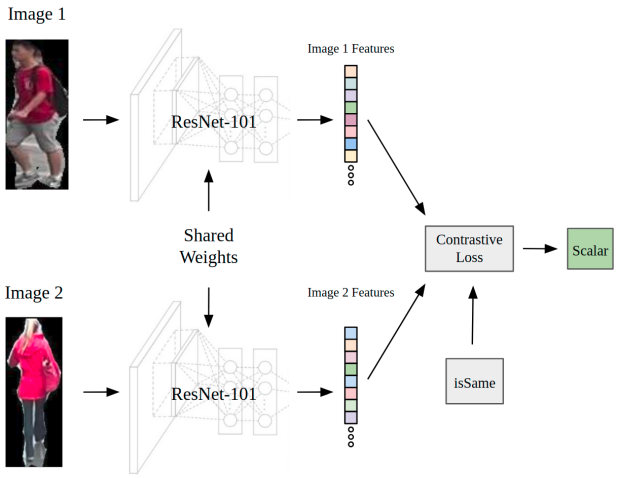

Figure 4. The architecture of the Siamese Network

We chose to go with Siamese networks because they also implicitly do clustering and have proven to be useful for one shot learning tasks. [24] We tried ResNet-101 (without the average pooling and fully connected layer) as the backbone of the Siamese network. The architecture can be seen in Figure 4.

3.3.1 Training phase

For training, we used contrastive loss as described by Hadsell, Chopra, and LeCun [15]. One problem with Siamese nets was to ensure that we get a large, representative sample of the data set as it was practically impossible to train on the entire dataset. The fundamental reason is because a Siamese net requires two images as input. So for a data set of size N, there could be $\binom{N}{2}$ possible inputs, which is a lot to practically go over every epoch.

To work around this, we picked a manageable number k and randomly chose $0.9k$ image pairs of different people and $0.1k$ image pairs of the same person. This ensured we saw enough samples and the sample of the training set given to the network was diverse. We also wanted the network to see enough images of the same pair of people to ensure depth in addition to breadth in the sample of the training set. For that, we pick a number c. Then, for every pair, $(p1, p2)$, where $p1, p2$ are person IDs, we pick c images of both $p1$ and $p2$ and generate c pairs by putting images of $p1, p2$ in a pair. Thus, in total, we fed ck images to the network. With this method, we could decide how much breadth and depth we want in our sample of the training set by tuning c and k. The higher the c, the more the depth and the higher the k, the higher the breadth. An important point to note is that we purposely gave the network a new set of ck images every epoch so it gets to see a wide variety of images.

3.3.2 Testing phase

The testing phase is almost exactly the same as the testing phase of our previous design of the CAE and clustering network. We again choose the k nearest neighbors (distance measured by mean square error) to the features of the query image. This is a valid way because it is easy to show that the contrastive loss is basically trying to minimize the mean squared error (assuming batch size is constant) when it gets features of two images of the same class/person.

4. Experiments

4.1. Segmentation

We began experimenting with segmentation using the base code made available by Pakhomov et al. [33] for their experimentation on the Endovis 2017 data set.

We trained the image segmentation network on the LIP dataset [12], which contains low resolution images, perfect for the PReID task. The data set contains pixel-wise labels for 19 human body parts, but we preprocessed the labels to keep only foreground (person) and background (everything else) classes. Given the availability of our training resources, we were only able to reliably train a network similar to ResNet 34 for image segmentation before the CUDA memory errors became too common for reliable training.

Our current network was run for about 50 epochs and we validated on 1000 images randomly selected from the training set (that were not trained on). The best model was trained with Adam optimizer with learning rate of 0.00001 and weight decay of 0.0005. The results are shown in Table 1.

To test if segmentation actually helps, we used a PReID model [38] and trained it by feeding it regular images for 59 epochs. We found that the model reached an mAP of 0.53 in that case. However, when we trained the same model for 59 epochs again by feeding it segmented images and keeping everything else the same, we observed that the mAP rose to 0.59. A 6% rise in performance clearly shows that even a

Table 1. Segmentation Results

Backbone	MIoU
Dilated ResNet 9	.724
Dilated ResNet 18	.821
Dilated ResNet 34	.848

Table 2. Performance comparison of a PReID network trained on segmented images with the same network trained on regular images

Segmented	rank1	rank5	rank10	mAP
False	0.77	0.91	0.94	0.53
True	0.82	0.92	0.95	0.59

segmentation network with the simple backbone of ResNet-18 can significantly boost performance. The precise results are in Table 2

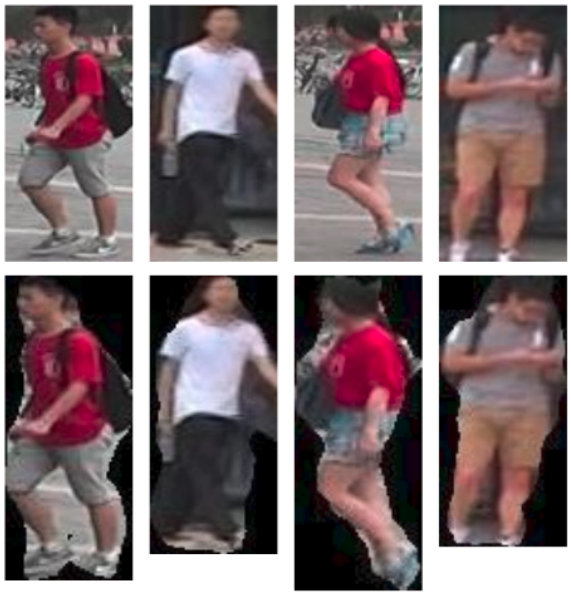

Figure 5. Examples results of the segmentation network on Market-1501 using ResNet-18 [49]

4.2. CAE and Clustering Network

We first experimented with different loss functions for the CAE to see which loss function results in the sharpest images. We found that using L2 loss yields blurry images, but using L1 loss gets some quite good images. We also developed one of our own loss functions which was equal to the maximum difference between corresponding pixels of the input and the reconstructed image to make the loss function more interpret-able. We found that this worked better than L2 loss, but not as good as L1 loss.

As can be seen in figure 4.2, the network produces strikingly similar images to the original ones. Most of the visual error comes from the "blurry-ness" of the reconstructed image. We also observe some error just outside the person boundary, but that is not a great concern as we only care about the hidden space having a complete representation of the person. If it has some extra useless information, the clustering network should be able to remove that.

Our experimentation with the clustering network was not successful. Despite multiple attempts to force the network to converge through changing loss functions, network architecture and hyper-parameters, we were not able to make it converge in any meaningful way. There are no results to report in this section.

We trained both the CAE and clustering network on the DukeMTMC data set [36].

Figure 6. Examples of some of the segmented images being reconstructed by the CAE when using L1 loss

4.3. Siamese Network

One important hyper-parameter in training this network was the margin used in the contrastive loss function. For an ideally trained network, the margin determines the minimum distance between two images of different classes. We tried different values for margin in the range of 1 to 100 and finally chose 100. We wanted margin to be very high because our test time method of finding the nearest neighbors was very delicate and susceptible to outliers.

We also tried different optimizers like SGD (with different values of momentum, weight decay and learning rate) and Adam to see what gives best results. We found out that Adam gave us best results with no weight decay and learning rate of 0.001.

As mentioned in section 3, we gave the network a differ-

Table 3. Siamese Performance

Backbone	Training Data Set	Top1	Top10	mAP
ResNet 101	Duke	.59	.83	.50

ent sample of the training set every epoch due to the problem of not being able to give the entire training set at once. We ensured that for every 9 different pairs, we give 1 same pair to the network. (By different/same pairs, we mean the two input images to the network having either the different/same person.)

We trained on Duke [36], while validating on the data set provided to us for this challenge by Dr. Wang. No identities in the validation set were included in the training data. The best results on Dr. Wang's validation data set are listed in Table 3.

5. Analysis

In this section, we analyze the experimental results in more detail, attempting to go into the root causes of some parts that did not work as expected.

5.1. Why clustering network may not have worked?

There are many reasons that the clustering network may not have worked as expected. The primary one may be that we were attempting to solve a very general and challenging problem. Note that, the idea of making the CAE and clustering network work is not in any way specific to the PReID problem. In many ways, we were attempting to make a very general supervised clustering algorithm with an arbitrary number of clusters. To our knowledge, there is no successful network for such a task. Often, clustering is viewed as an unsupervised learning problem. The generality of our idea can be appreciated by realizing that if we would have successfully converged this network, we could use the same network with no change and train it on ImageNet [7] and have it do classification. So this may be the main reason: we were trying to solve a very general problem and that may be much harder than we thought.

The second big reason may be our unconventional loss function for the clustering network, which made it really hard to decide how to tune hyper parameters. While training the clustering network, we observed some of the strangest loss curves. This was because the loss function was updating every few batches and basically resetting every few epochs. Because of the novelty of the loss function, we could not find resources on how to interpret such loss curves and traditional ideas to tune parameters based on the way loss curves look were not valid for us.

Another (unlikely, but theoretically possible) reason may come from the following subtle analysis of the loss function. Note that the clustering loss only tries to minimize intraclass distance (distance between features of images of same person). It does not explicitly deal with interclass distance (distance between features of images of different people). An extreme theoretical possibility is that the mean features of all classes are the same and that individual image features of all classes are very close to this "grand mean". This would make the loss function take very low values and we would think the network is doing a great job, but when in reality it may not be. To address this problem, we did change the loss function to account for interclass distance too, but we couldn't make that network converge either. It is unlikely that the situation described above is what happened, but something similar and less extreme is likely to have happened.

5.2. Could Siamese network have done better?

Even though the Siamese networks clearly converged they failed to give a high accuracy. The following are speculations and explanations as to why this happened.

The most important reason may be the way Siamese networks are dealt with at test time. As explained in section 3, we simply try to find the k nearest neighbors at test time. This approach has three issues. One, it assumes that all features in the feature space of the given two images are equally important. In other words, it doesn't weight some features differently than others. Two, it assumes that the features don't have complex relationships. This means that feature i of image A and feature j of image B don't have any interactions/relations. This is because the mean square error would only calculate error between feature i of image A and feature i of image B. Three, it is highly prone to errors due to outliers. It may just coincidentally happen that the top 10 closest features to the features of the query image belong to just 10 images that are outliers of their own clusters. This is actually not as unlikely as one may think: the probability of something similar happening depends on the number of images in the gallery. The more the number of images in the gallery, the higher the probability.

6. Conclusion

In this paper, we present a novel clustering based approach for the PReID task. We propose two models to do clustering. We also suggest how we can use binary image segmentation to actually get very good results even with the type of low resolution images commonly used in the PReID task. We also show that binary segmentation can also be used to improve performance of existing models. We also do an in-depth analysis of the reasons it was hard to make the clustering network converge.

7. Future Work

There are many directions this work could be extended and improved, some of which are currently being explored by us. In this section, we point out some ideas to improve upon this work. As will be clear, the ideas talked about actually are very general and extend much beyond the scope of PReID. We list two ways that this work can be extended.

One, to the best of our knowledge, this is among the earliest work in the area of deep supervised clustering with undetermined number of clusters. We believe that the ability to do supervised clustering can have very interesting and useful applications. We believe , clustering is a much more general task than classification. An example may help clear this out. Assume a problem of classification with a 1000 classes. In that case, a classification model can't extend beyond these 1000 classes. So, if we give it an image of something that doesn't belong to the 1000 classes, the behavior of the network would be undefined (commonly known as erroneous). However, if we treat the same problem as a clustering problem without assuming the number of classes in our clustering algorithm, then the objective of our model is learning the most important features that make a data point have a specific class (this class does not have to be among the 1000 classes). In other words and in context of images, we are trying to teach the model what makes two images similar and how is similarity even defined instead of naively forcing it to remember features of a particular class. Thus, if we give the clustering model an image from a class not in the training set, we can be more confident that images of that class would cluster together. Thus, supervised clustering can be used to do classification (without labelling the class) on data from classes that are not in the training set, which is surprisingly powerful. We have proposed a general loss function in our work to do supervised clustering. We believe that a more in-depth study of loss functions for such tasks and experimental studies to compare different ways to measure distance in clustering problems would immensely benefit the research community and may even take us closer to more human-like learning.

Two, in section 5, we talked about how calculating just the mean squared error between two features may not be ideal as it assumes that all features are equally important and there is no complicated relation between features. As one of many potential solutions to this problem, we are currently planning to concatenate the final features we get from both input images and then have a fully connected layer from this concatenated tensor to a scalar. This scalar would denote the probability of the input images to be of the same class. Needless to say, the fully connected layer takes care of both the (incorrect) assumptions of all features being equally weighted and features not having complex relationships. We can now train the network with a binary cross entropy loss and put a sigmoid non-linearity at the end. [2]

8. Contributions

In this section, we list the contribution of each team member in five brief bulleted points.

Ankit did the following:

- Proposed the entire idea of CAE, clustering net and Siamese nets after reading state of the art research methods
- Designed and trained the CAE and the clustering network
- Designed the novel loss function of the clustering network
- Designed and trained the Siamese Network
- Wrote the Approach, Analysis, Conclusion and Future Work in the final paper

Benton did the following:

- Set up the server and GPU, did all of the system administration and accessibility for the server
- Modified the binary segmentation script and LIP data set to train the segmentation network for human segmentation
- Helped design and create the Siamese network
- Wrote the Related Work and Experiments sections of the paper, along with small contributions to other sections

Ryan did the following:

- Attended Ye's seminar regarding the use of triplet loss which later inspired our clustering philosophy
- Proposed the idea of a pre-processing segmentation network
- Trained and tested our baseline pytorch model on the output of our segmentation network
- Installed and deployed tensorboardX on Benton's server so we could read our pytorch metrics in the format of a familiar tensorboard
- Made the architecture diagrams, graphics, and wrote the introduction for the proposal and final paper

References

[1] E. Ahmed, M. J. Jones, and T. K. Marks. An improved deep learning architecture for person re-identification. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3908–3916, 2015. 2

[2] J. Bromley, I. Guyon, Y. Lecun, E. Sackinger, and R. Shah. Signature verification using a siamese time delay neural network. In J. Cowan and G. Tesauro, editors, *Advances in neural information processing systems (NIPS 1993)*, volume 6. Morgan Kaufmann, 1993. 2

[2] This seemed like a promising approach and so we have currently completely coded this idea and are in the process of experimenting.

[3] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *arXiv e-prints*, June 2016. 3

[4] W. Chen, X. Chen, J. Zhang, and K. Huang. Beyond triplet loss: a deep quadruplet network for person re-identification. *ArXiv e-prints*, Apr. 2017. 2

[5] D. Cheng, Y. Gong, S. Zhou, J. Wang, and N. Zheng. Person re-identification by multi-channel parts-based cnn with improved triplet loss function. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2

[6] D. Chung, K. Tahboub, and E. J. Delp. A two stream siamese convolutional neural network for person re-identification. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 2

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. Ieee, 2009. 7

[8] K. G. Dizaji, A. Herandi, and H. Huang. Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5747–5756, 2017. 2

[9] W. Du, M. Fang, and M. Shen. Siamese convolutional neural networks for authorship verification. *cs231n.stanford.edu*, 2018. 2

[10] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996. 5

[11] A. Gala and S. K. Shah. A survey of approaches and trends in person re-identification. *Image Vision Comput.*, 32:270–286, 2014. 1, 2

[12] K. Gong, X. Liang, X. Shen, and L. Lin. Look into person: Self-supervised structure-sensitive learning and A new benchmark for human parsing. *CoRR*, abs/1703.05446, 2017. 5

[13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 4

[14] X. Guo, L. Gao, X. Liu, and J. Yin. Improved deep embedded clustering with local structure preservation. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 1753–1759, 2017. 2

[15] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *null*, pages 1735–1742. IEEE, 2006. 5

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. 3

[17] A. Hermans, L. Beyer, and B. Leibe. In defense of the triplet loss for person re-identification. *CoRR*, abs/1703.07737, 2017. 2

[18] M. Hirzer, C. Beleznai, P. M. Roth, and H. Bischof. Person re-identification by descriptive and discriminative classification. 2

[19] P. Huang, Y. Huang, W. Wang, and L. Wang. Deep embedding network for clustering. In *2014 22nd International Conference on Pattern Recognition (ICPR)*, volume 00, pages 1532–1537, Aug. 2014. 2

[20] C. S. Inc. Cisco visual networking index: Forecast and methodology, 2015/2020,. April 2016. 2

[21] M. M. Kalayeh, E. Basaran, M. Gokmen, M. E. Kamasak, and M. Shah. Human semantic parsing for person re-identification. *CoRR*, abs/1804.00216, 2018. 3

[22] 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. 2

[23] G. Koch. Siamese netowrks for one-shot image recognition. Master's thesis, University of Toronto, Toronto, Canada, 2015. 2

[24] G. Koch, R. Zemel, and R. Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, volume 2, 2015. 5

[25] B. Lavi, M. F. Serj, and I. Ullah. Survey on deep learning techniques for person re-identification task. *CoRR*, abs/1807.05284, 2018. 1, 2

[26] W. Li, X. Zhu, and S. Gong. Person re-identification by deep joint learning of multi-loss classification. *CoRR*, abs/1705.04724, 2017. 2

[27] W. Li, X. Zhu, and S. Gong. Harmonious attention network for person re-identification. *CoRR*, abs/1802.08122, 2018. 3

[28] L. Ma, X. Jia, Q. Sun, B. Schiele, T. Tuytelaars, and L. V. Gool. Pose guided person image generation. *CoRR*, abs/1705.09368, 2017. 2

[29] L. Ma, Q. Sun, S. Georgoulis, L. V. Gool, B. Schiele, and M. Fritz. Disentangled person image generation. *CoRR*, abs/1712.02621, 2017. 3

[30] N. McLaughlin, J. M. Rincon, and P. Miller. Recurrent convolutional network for video-based person re-identification. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 00, pages 1325–1334, June 2016. 2

[31] F. Milletari, N. Navab, and S. Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. *CoRR*, abs/1606.04797, 2016. 3

[32] D. Pakhomov, V. Premachandran, M. Allan, M. Azizian, and N. Navab. Deep residual learning for instrument segmentation in robotic surgery. *arXiv preprint arXiv:1703.08580*, 2017. 3

[33] D. Pakhomov, V. Premachandran, M. Allan, M. Azizian, and N. Navab. Deep residual learning for instrument segmentation in robotic surgery. *CoRR*, abs/1703.08580, 2017. 5

[34] B. Prosser, W. shi Zheng, S. Gong, T. Xiang, Q. Mary, and V. Laboratory. Person reidentification by support vector ranking. In *In British Machine Vision Conference*, 2010. 2

[35] R. Rama Varior, M. Haloi, and G. Wang. Gated Siamese Convolutional Neural Network Architecture for Human Re-Identification. *arXiv e-prints*, July 2016. 2

[36] E. Ristani and C. Tomasi. Features for multi-target multi-camera tracking and re-identification. *arXiv preprint arXiv:1803.10859*, 2018. 6, 7

[37] A. Shukla, G. S. Cheema, and S. Anand. Semi-supervised clustering with neural networks. 2018. 2

[38] Y. Sun, L. Zheng, W. Deng, and S. Wang. Svdnet for pedestrian retrieval. 2017. 5

[39] K. Tahboub. *Person Re-identification and Intelligent Crowdsourcing with Applications in Public Safety*. PhD thesis, 2017. 1

[40] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, pages 1701–1708, Washington, DC, USA, 2014. IEEE Computer Society. 2

[41] M. Tian, S. Yi, H. Li, S. Li, X. Zhang, J. Shi, J. Yan, and X. Wang. Eliminating background-bias for robust person re-identification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2, 3

[42] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 1096–1103, New York, NY, USA, 2008. ACM. 2

[43] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.*, 11:3371–3408, Dec. 2010. 2

[44] T. Wang, S. Gong, X. Zhu, and S. Wang. Person re-identification by video ranking. 2

[45] J. Xie, R. B. Girshick, and A. Farhadi. Unsupervised deep embedding for clustering analysis. *CoRR*, abs/1511.06335, 2015. 2, 3

[46] D. Yi, Z. Lei, and S. Z. Li. Deep Metric Learning for Practical Person Re-Identification. *arXiv e-prints*, July 2014. 2

[47] H. Zhao, M. Tian, S. Sun, J. Shao, J. Yan, S. Yi, X. Wang, and X. Tang. Spindle net: Person re-identification with human body region guided feature decomposition and fusion. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 3

[48] L. Zhao, X. Li, J. Wang, and Y. Zhuang. Deeply-learned part-aligned representations for person re-identification. *CoRR*, abs/1707.07256, 2017. 3

[49] 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. 1, 6

[50] L. Zheng, Y. Yang, and A. G. Hauptmann. Person re-identification: Past, present and future. *CoRR*, abs/1610.02984, 2016. 2