Person Re-Identification Using Pose-Normalized And Occlusion Independent Training

Aditya Kumar
Texas A&M University
aditya30394@tamu.edu

Kamala Akhila Mangipudi
Texas A&M University
akhila1012@tamu.edu

Patel Vihang
Texas A&M University
viky1712@tamu.edu

Abstract

For making sense of the vast quantity of visual data generated by the rapid expansion of large scale distributed multi-camera systems, automated person re-identification is essential[5]. Person re-identification, a tool used in intelligent video surveillance, is the task of correctly identifying individuals across multiple images captured under varied scenarios from multiple cameras. Solving this problem is inherently a challenging one because of the issues posed to it by low resolution images, illumination changes per image, unconstrained pose and occlusions. In this project, we aim at developing a Person re-identification model using Deep Neural Networks (DNN) which can handle variable size input images. Specifically, we aim at implementing two preprocessing techniques, which reduce the chances of over fitting i.e. We aim to make our model robust to occlusion using Random Erasing, a data augmentation technique[20], and reduce the influence of pose variations on features using a Pose normalized Generative Adversarial Network (GAN)[14]. Along with this we also aim to implement and integrate Part-based Convolutional Baseline (PCB) to further improve on the results. [18]. We briefly describe the models trained along with their evaluation results on Market 1501 dataset and provided validation and test sets.

1. Introduction

With safety and security being a growing cause of concern in today's society, large number of cameras are being continuously deployed in public places like airports, railway stations, and office buildings which monitor and record huge amounts of data on a minute to minute basis. Human processing of such huge amounts of data produced is not only error prone but also time consuming and expensive. Intelligent video surveillance systems aims at automating this job and automated analysis of large amounts of data can not only process the data faster but also significantly improve the quality of surveillance[2]. Person re-identification is a fundamental task in automated video surveillance and has been an area of intense research in the past few years. Person re-identification aims to search for the same person across different cameras with a given probe image[12]. It promises enormous potential for a wide range of practical applications. Despite years of efforts, this remains a challenge due to large variations in pose, illumination, background clutter and occlusions.

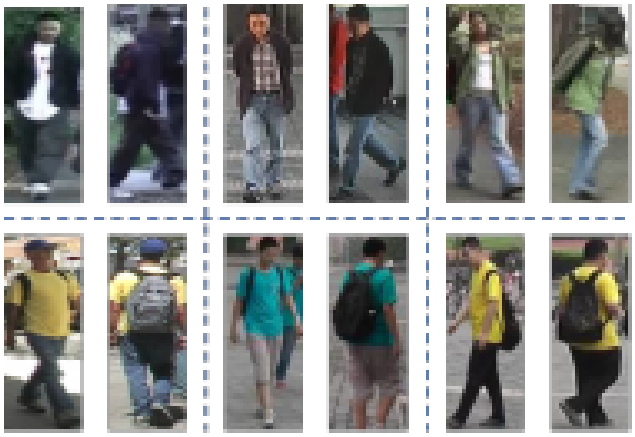

Figure 1. Example images showing that person appearance can change depending upon the camera viewpoint and pose of the person

The primary challenge faced by the re-id systems is due to the undisciplined environment in which the camera networks are deployed. Cameras are not necessarily positioned with overlapping field of views and they have varying distances towards recorded persons. In addition to these variations, the pose of a recorded person is mostly unconstrained. This could be because of the person walking towards or away from the camera, or because of the full body pose (for a walking person, arms and legs are moving)[14]. These pose variations severely affect the appearance of the person and hence in turn also affect the person re-id to a larger extent. Figure 1 shows how the appearance of the person completely changes depending on their pose, and this makes the

learning of features harder. Pose is defined as a combination of viewpoint and body configuration. It is thus also a cause of self-occlusion[14]. Human poses and view points cannot be controlled in real life scenarios and hence hand crafted features may not be robust enough to handle pose and viewpoint variations.

In recent years, deep learning has demonstrated strong model capabilities and obtained very promising performance in many computer vision tasks[16]. The success of Convolutional Neural Networks(CNN) in the field of Computer Vision paved a way for their use in Person Re-identification[1]. Each layer of the CNN generates a set of feature maps in which each pixel of the image corresponds to a specific feature representation. The advent of new datasets like Market-1501[19] and DukeMTMC, which contain more number of labeled person images, has made the usage of deep learning for person re-id more feasible. The ability to generalize is key for any neural network and even though the current deep person re-id models exhibit promising performance, they still suffer the lack of generalizability to new camera networks.

Occlusion is one factor that has a critical impact on the generalization ability of CNNs. It is desirable that the classification model we develop classify the image from the overall structure of the image even though parts of it are occluded. However, there is a high probability that the samples used for training exhibit very limited occlusion effects. Due to this, when the model is deployed to a new camera network, the test images without occlusions might be properly classified, but the ones which are partially occluded will be erroneous. One work around for this problem could be to add more and more images to the training set that are occluded at various levels, but this is an expensive process and there are high chances that we might still miss few cases after the addition. To address the problems caused by occlusion, we used a data augmentation technique, Random Erasing as illustrated in [20]. As a part of Random erasing, either of the following two operations are performed randomly on the training images: 1. Image is kept unchanged or 2. A rectangular region of arbitrary size is randomly chosen and the pixels in the selected region are either assigned with random values or the mean pixel value of Imagenet[20]. During Operation 2, a random sized mask is applied on the image and therefore various levels of occlusion on training images is achieved as shown in in Figure 3. This is very similar to the techniques, Random cropping[1] and Random flipping[15], but Random erasing in addition also preserves the overall structure of the image by randomly choosing various parts of the image and applying masks on them.

Another factor which is critical to the generalization ability of CNNs is the pose variation. Even with the advent of new datasets which facilitate huge amounts of labeled training data, existing deep re-id models are posed with a challenge of learning pose insensitive features. A person's appearance is determined by a combination of identity sensitive but view-insensitive factors and identity insensitive but view sensitive factors, and these factors are not independent[14]. The appearance of a person depends upon the pose that he is carrying. Learning pose insensitive features is a tough task, as for instance in Example 1, the features learned from a person carrying a backpack should correspond to just a strap as shown in the adjacent block. In this project, we use Pose Normalized Generative Adversarial Network(PN-GAN) to develop models without having to worry about the pose variations.

The heart of the system is a deep image generator, which takes in any image and a desired pose as inputs, and outputs the image of the person in that particular desired pose. For this purpose, 8 cannonical poses which represent the overall pose structure of the training data have been taken and 8 new images have been generated per every training image in the data-set thereby increasing the training data size by 8 fold. These pose normalized images are then used to train a person re-id model which generates features complementary to those generated by the original image.

Also the global features might miss out on small details which might help in re-identification of the person. To avoid this we extract partial features (i.e. localized features) which gives us a fine-grained information. To implement this we use a slightly modified version of the baseline called Part-based Convolutional Baseline (PCB) [18]. For this technique to work, it is of paramount importance that the parts are accurately located. As it is assumed that in Person Re-Identification, the pedestrians in the input images are bound by detected bounding boxes, we can safely use uniform partition strategy to divide the image into stripes and further refining them. Thus by paying attention to the finer details of the image, we can improve the performance of the model for a slightly higher computation cost.

2. Literature Survey

Traditional techniques used for person re-identification used the hand crafted features, which were based on defining a specific descriptor based on color, texture, or biometric features like face, and defining a specific similarity measure between a pair of descriptors[7][4][9][13]. These models concentrated on developing a powerful feature for image representation and learning an effective metric for distance calculation between the features. These hand crafted features are very expensive to develop and also might not take full advantage of the entire information provided by the training image. In recent years, deep learning models started to take over and have obtained impressive performance. This approach is largely inspired by the strong representation auto-learning capacity of deep models

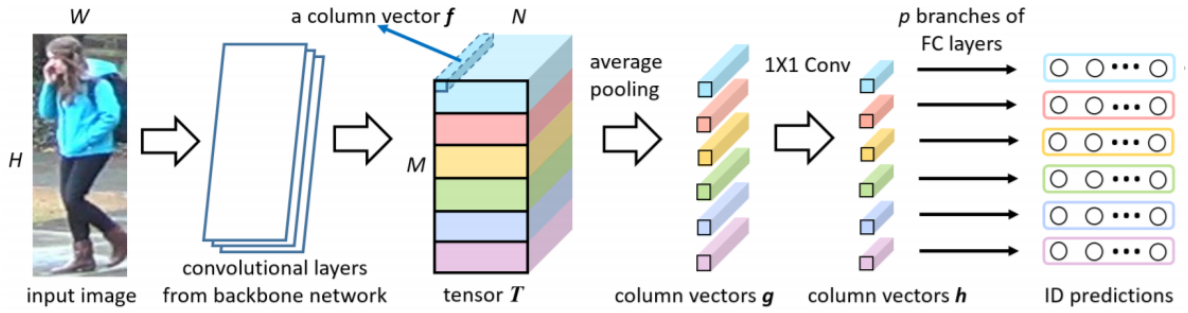

Figure 2. Structure of PCB. The input image goes forward through the stacked convolutional layers from the backbone network to form a 3D tensor T. PCB replaces the original global pooling layer with a conventional pooling layer, to spatially down-sample T into p pieces of column vectors g. A following 1 1 kernel-sized convolutional layer reduces the dimension of g. Finally, each dimension-reduced column vector h is input into a classifier, respectively. Each classifier is implemented with a fully-connected (FC) layer and a sequential Softmax layer. During training, each classifier predicts the identity of the input image and is supervised by Cross-Entropy loss. During testing, either p pieces of g or h are concatenated to form the final descriptor of the input image. [18]

benefiting from large sized labelled training data pools; and the establishment of large person re-id datasets [19][11].

Generally speaking, two kinds of deep learning models have been widely deployed in this research area: 1. Classification model which determines the class an image belongs to and 2. Siamese model based on pairwise or Triplet comparisons which gives a similarity score between the images[10]. Earlier, due to the lack of sufficient data for person re-id task, classification models tend to overfit the data. In such a case, Siamese networks[3] performed better. Even though both Siamese pairwise and triplet models illustrate good performance on datasets with even limited number of training samples[10], we chose to use classification model owing to the probability of failure of Siamese network to converge.

A number of deep learning models focus on solving the problems of pose variation and occlusions. For instance, [12] first detects normalized part regions from a person image and then fuse the features extracted from the original image and the part region images. This project is based on [20] and [14], which address the issues of occlusions and pose variations respectively. We use the ResNet architecture [8] and the standard cross entropy loss for our deep model architecture. GAN[14] uses deep image generator to synthesize whole body images rather than focusing only on specific body parts for pose normalization. This is very similar to the present project, except that we also use Random erasing from [20], because of which we train our model using images of multiple pose variations and multiple levels of occlusion.

For battling the issue of information lost because in global features, we use a network named Part-based Convolutional Baseline (PCB) proposed in [18]. PCB conducts a uniform partition the convolutional layer to obtain the part-level features. Here the image is not explicitly partitioned. PCB takes a whole image as an input and outputs a convolutional feature. PCB can be implemented with a slight modification to the base network (in our case ResNet50). The implementation details are further mentioned in section 3.2.

3. Proposed Model

As explained in the last section, our proposed models build upon the work done in Random Erasing[20] and GAN[14]. We primarily focus on the pre-processing stage to make sure that the learned CNN works with test images that have person in various poses in addition to suffering from varied levels of occlusion.

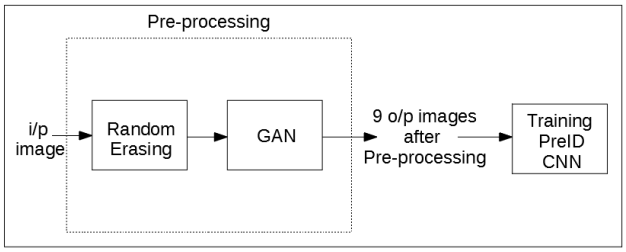

Figure 3. Block Schematic

3.1. Random Erasing[20]

With occlusion critically influencing the generalization capability of a deep re-id model, we use a pre-processing technique called Random Erasure to improve the generalizability. It is a data augmentation technique which is used to train the Convolutional Neural Network. It randomly selects rectangular region in the image and erases pixels with random value. This basically introduces varying degree of occlusion in the training data images. The technique is

easy to implement and integrate with the basic recognition model.

This module takes in an image and performs random erasing on it with a given probability p (i.e. image remains unchanged with probability 1-p). As it is assumed that in Person Re-Identification, the pedestrians in the input images are bound by detected bounding boxes and since there can be occlusion present in any part of the image we randomly select and erase a rectangular region of the image, thereby creating training images with various levels of occlusion.

Figure 4. Random Occlusion [17]

Random Erasing algorithm randomly selects a rectangular region I_e and erase it by assigning random values to the pixels. The area of training image be $S = W \times H$, where W and H are dimensions of the image. We randomly select area of rectangle region I_e to be S_e, such that $\frac{S_e}{S}$ is bounded by minimum S_l and maximum S_h. We also randomly initialize aspect ratio r_e such that it is within values r_l and $1/r_l$. We get the dimension of I_e as $H_e = \sqrt{S_e \times r_e}$ and $W_e = \sqrt{\frac{S_e}{r_e}}$. Now the initial point $(x_e.y_e)$ in the training image is randomly selected and if $x_e + W_e \leq W$ and $y_e + H_e \leq H$, then we set the selected rectangle region $I_e = \{x_e, y_e, x_e + W_e, y_e + H_e\}$, if above condition fails then repeat the above process until the condition is satisfied. Now that we have the randomly selected region I_e, the value of pixels in it are randomly assigned. Fig 4 shows an example of the output obtained from random erasing module.

Thus by introducing random occlusions, random erasing helps the model grasp overall features and thereby helps in reducing the tendency of a model to over-fit to the training set and makes it robust against partial occlusions.

3.2. Part-based Convolutional Baseline (PCB)[18]

A traditional CNN based network will take in effect the global features of the input image. But the results can be improved by considering the partial features of an image and combining them. This is exactly what PCB aims to achieve.

We can reshape the base network with slight modification to attain a PCB, as illustrated in fig 2. Here the structure of the base network before the Global Average Pooling (GAP) layer is preserved as such. The GAP and what follows are all removed. The output of this remaining base network is a 3D tensor T of activations. Using conventional average pooling PCB then partitions the tensor T into *p* horizontal stripes. Then it averages all the column vectors into a same stripe into a single part-level column vectors ***g***. The dimensions of ***g*** are further reduced by a 1×1 convolutional layer to output a new vector ***h***. The dimensions of this reduced column vector ***h*** is set to be 256. Each of these ***h*** is finally input into a classifier., which is implemented by a fully connected (FC) layer followed by a Softmax function, to predict the identity of the input. We use sum of cross entropy losses over the p pieces of identity prediction to optimize the PCB. During training this pieces are concatenated to form the final descriptor $\boldsymbol{G} = [g_1\ g_2\ g_3\ ....\ g_p]$ or $\boldsymbol{H} = [h_1\ h_2\ h_3\ ....\ h_p]$.

Parameters such as the input size i,e, W $\times$ H and the number of pooled column vectors *p* are very important for performance of PCB. The optimized parameters used in our experimentation are: **(1)** Input image size (384×192), i.e. height to width ratio of 2:1. **(2)** 6 pooled column vectors *p* are used. i.e. **T** is divided into 6 horizontal stripes.

From the results it can be observed that this ***G*** achieves a better result than a traditional base network.

3.3. Deep Image Generator (Generative adversarial net)[14][6]

To enhance the generalizability of the re-id model, we use pose normalization[14], to learn discriminative identity sensitive features under various pose variations. The image generator model is based on generative adversarial network (GAN) designed specifically for pose normalization. It consists of a generator and a discriminator to synthesize images. Given an input image I and a desired pose P, the generator tries to generate an image J, of the person present in I, having pose P. The discriminator is used to reduce the error in the generator. Fig 5 represents the schematic of this GAN model. The pose estimation in the model is done using pre-trained off-the-shelf model - OpenPose. For every image, the model is capable of generating images in 8 different poses. These 8 different images gets added to the training set along with the original image and thereby helps in building generalizable re-id model free of pose variations.

3.3.1 Pose Estimation

The pose estimation in the model is done using pre-trained off-the-shelf model - OpenPose, which is trained without using any re-id benchmark. We used this library to estimate the pose of each and every training image. Next, we did

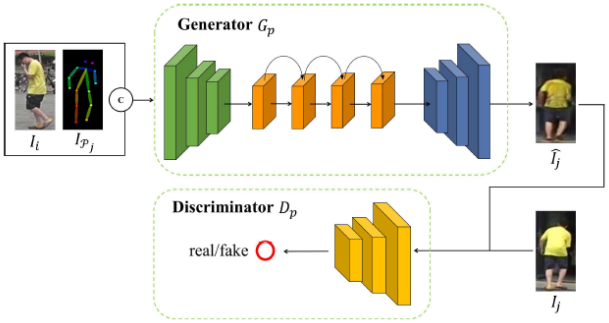

Figure 5. PN-GAN Schematic[14]

a manual pre-processing step to remove images which did not have realistic poses. Because of poor quality of some images in training some estimated pose images just had leg, hand, no head, etc. Our initial clustering result on the poses before this pre-processing step was creating clusters that represented noisy images. Some of these noisy images can be seen in Fig 7

Figure 6. Sample of a training image and its corresponding pose generated using OpenPose

Figure 7. Noise in pose estimation that resulted in improper clustering results in our experiments

3.3.2 Generator

The main usage of GANs[6] is in creating data from the scratch using two different deep networks that get trained together - Generator and Discriminator. GANs are mostly used for images but it has been used in other domains like music too.

First, we sample some random noise $z \sim N(0, 1)$ using normal distribution and then using z as input to the generator G, we create image $G(z)$. Here, z represents the features of the image. But a generator alone would only create random noise and therefore it is helped by another network called Discriminator.

3.3.3 Discriminator

The discriminator looks at the target image and the output of generator separately and tries to distinguish between the two images. The output $D(X)$ is the probability that the input image X is real image. This network gets trained as a classifier which ideally outputs $D(X) = 1$ when X is real and $D(X) = 0$ otherwise. During the training phase, the discriminator learns features that contribute to real image. Since the objective of GAN is to have the generator produce X such that $D(X) = 1$ this value is back propagated to GAN so that it generates images that the discriminator thinks is real. Therefore, these two components constantly play a two-player minimax game with each other.

The objective for the discriminator D is :

$$\max_D V(D) = \mathbb{E}_{x \sim p_data(x)}[\log(D(x))] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))] \quad (1)$$

And for the Generator, the objective is

$$\min_G V(G) = \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(x)))] \quad (2)$$

In other words, D and G play the following two-player minimax game with value function $V(G, D)$ [6]:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_data(x)}[\log(D(x))] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))] \quad (3)$$

With respect to our GAN model[14], following equation govern the training:

$$\min_G \max_D V(D, G) = \mathbb{E}_{I_j \sim p_data(I_j)}[\log(D(I_j))] + \log(1 - D(G(I_i, I_{p_j}))) \quad (4)$$

where the pair (I_i, I_{p_j}) is input to generator. I_i is the image of a person, I_{p_j} is the desired pose and I_j is the real target image. Refer to Figure 5 for the schematic diagram of GAN networks.

3.3.4 Pose Normalization

We need to obtain a set of 8 cannonical poses which are representative of typical view-point and body configurations exhibited by people. To accomplish this, we first obtained poses of all the images in the training data set and applied K Means clustering on them to get 8 cannonical poses. We used VGG-19[1] pre-trained on the ImageNet ILSVRC-2012 dataset for feature extraction. We extracted features at the 10th layer of VGG-19 which are of dimension (256,32,16), and clustered them using KMeans algorithm into 8 different clusters. The images closest to the center of each cluster have been finally chosen as the 8 cannonical pose images. The result of our clustering on poses derived from Market 1501 training data is shown in Figure 8.

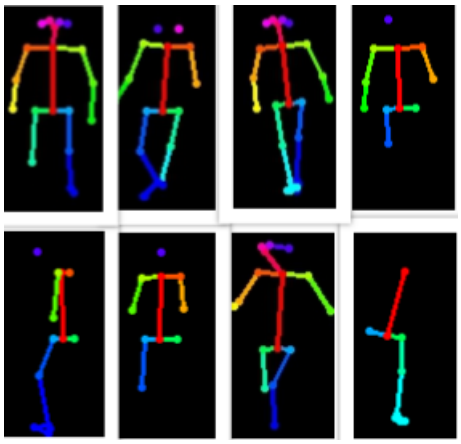

Figure 8. 8 Cannonical poses obtained as a result of Pose Normalization

4. Experiments

4.1. Person Re-ID with Random Erasing and PN-GAN Integrated

The previous two pre-processing steps - Random erasing and pose normalization, generate a total of 9 additional images - 1 from Random erasor and 8 from PN-GAN described above. In our proposed model, we serialize these pipelines and train our ResNet-50 based person re-id model. Fig 3 shows the high level design of our model.

4.2. Training Dataset and Train Image Model

Market1501 image dataset is being used for training the model. ResNet-50 is chosen as the train image model as it provides a good trade-off between accuracy and training complexity/convergence time. Following is the breakdown of images in Market1501 dataset.

Subset	# ids	# images
train	751	12936
query	750	3368
gallery	751	15913
total	1501	32217

We also measured the performance of some models against the validation set which was introduced in the course.

4.3. GAN training

To generate the training set for our GAN, we used openpose to estimate the pose of all the training images in Market1501 dataset. Next, we took combinations of two images per person and feed them as input to this GAN. We had a total of 13265 such pairs. In our experiment, we found the training of GAN to be computationally very expensive. Each epoch took approximately 11 hours and we were able to complete 12 epochs before we ran out of resources. Figure 9 and 10 shows the results of trained GAN after 10 and 12 epochs respectively. First two images from the left are input to the generator whereas the third one is real image in the desired input pose(image 2). Fourth image is the final generated output.

Figure 9. GAN generated image after 10 epoch

Figure 10. GAN generated image after 12 epoch

4.4. Training environments

For training various models, we used HPRC Terra clusters that have Intel Xeon E5-2680 v4 2.40GHz 14-core, 1 NVIDIA K80 Accelerator and 128GB RAM running CentOS 7. During our project we completely exhausted the HPRC quota available to us and thus we also used amazon aws's EC2 Deep Learning AMI (Ubuntu) Version 19.0 with two configurations - p2.xlarge instance that has 61GB GPU memory and g3.4xlarge instance that has 122GB GPU memory.

4.5. Evaluation metric

We used two different evaluation metrics to quantify the performance of our various re-id models. The first one is pair of Rank-1 and Rank-5 accuracy whereas the second one is mean Average Precision (mAP).

5. Results

All of our re-id models have ResNet-50 as their backbone. Majority of our models have been built upon the work done in this github repo -`https://github.com/KaiyangZhou/deep-person-reid`. We used this as baseline for our Pose-Normalized GAN based(PN-GAN) re-id model. We also experimented with ResNet-50 based models presented in layumi repo - `https://github.com/layumi/Person_reID_baseline_pytorch`. We wanted to see the results of PCB[18] which was integrated in this repo.

Epochs	Top1	Top5	mAP
55	64.67%	83.33%	44.12%
60	68.67%	82.67%	46.6%
65	64%	82.67%	45.86%
70	66.67%	82%	45.79%
80	62.67%	82.67%	44.58%

Table 1. Results of our proposed model PNGAN

Table 1 summarizes the performance of our PN-GAN model[14]. In all these experiments, we have used Random Erasing[20] and the results are on the validation set provided during the semester. It can be seen that training for higher number of epochs doesn't really helps in improving the performance on this compact network. We found that 60 epochs is the ideal number of epochs for this network too - which is the case with a lot of different re-id models.

Table 2 summarizes the performance seen on other ResNet-50 based models with various configurations. Here again, we observed that training for large number of epochs in not giving any performance boost. A simple inclusion of Random erasing[20], keeping other parameters same, improves the performance by 2%-3%. However, the highest performance was achieved with the inclusion of both Random Erasing(RE[20]) and PCB[18] - outperforming any similar sized model by a good margin of 9%-10%.

Model	Epochs	Test Set	Top1	Top5	mAP
ResNet-50 with RE	60	Market test set	86%	94.1%	69.1%
ResNet-50 without RE	100	Market test set	84.5%	94%	67.2%
Layumi baseline with RE	60	Course Validation set	84.67%	94.67%	66.93%
Layumi baseline + PCB + RE	60	Market test set	93.2%	97.54%	80.84%
ResNet-50 without RE	60	Course Validation set	79.33%	88%	65.17%
ResNet-50 with RE	60	Course Validation set	82%	91.33%	65.10%

Table 2. Results of various ResNet50 based models

6. Conclusion and Future work

Our model serialized random erasing and pose normalization one after another in the pipeline. Our low top-1 and mAP score on the model suggests that this very serialization might have prevented our model from learning sufficient good distinguishing features per image. Random erasing blacks out random parts of images and our GAN was never trained on such input and therefore we feel this would have been a major factor in our low score. In future, we could experiment training the GAN-integrated re-id model with the original image, without random erasing, and document the results obtained. The training of GAN-integrated re-id model took more than 24 hours for 60 epochs compared to usual 3-5 hours which prevented this experiment given that our GAN itself took a lot time.

On closer inspection of our GANs performance, we found that there was still a lot of room for improvement in it's training. The GAN generated images are quite blurry and therefore might have affected the performance of our ResNet-50 based re-id model. In future, we could experiment training our GAN for a sufficiently larger time - say 20-30 epochs and observe the difference in performance of

re-id model. One common theme with models that have performed extremely well was the usage of multiple datasets during training and ensemble technique - combining the predictions from multiple heterogeneous models. We have a few trained standalone models and if given an opportunity, we could try and experiment with multiple datasets and combining various models. We could also in-fact try fusion on the output from last layers of different model and check for performance on the combined output. The authors of original paper[14] have reported improvements in such a scheme.
Finally, we could experiment with the backbone of our re-id model itself. We persisted with ResNet-50 owing to its compact size and hence better practical applicability. In future, we could experiment with deep networks like ResNet-101, ResNet-152, DenseNet-121, DenseNet-161, etc.

7. Individual contributions

Although the number of commits per member to the github repo might be different but everyone has contributed equally towards this project. Many a times we all solved issues/bugs together whereas only one person pushed the changes. However, if we look from a broader perspective then ownership of various parts can be divided as follows:

- Vihang : Integrating the Random Erasing code to our baseline model(it was not present), PCB training and corresponding parameter tuning. Running the second baseline (layumi).
- Aditya : Integrating the GAN model with our baseline, it's training and parameter tuning. Generation of pose images using openpose API.
- Akhila : Pose Normalization (clustering), codes for validation and mat file generation, and testing code changes.
- Everyone has contributed equally in this report.

References

[1] G. E. H. A. Krizhevsky, I. Sutskever. Imagenet classification with deep convolutional neural networks. *In: Advances in neural information processing systems, pp. 10971105*, 2012.

[2] A. Bedagkar-Gala and S. K. Shah. A survey of approaches and trends in person re-identification. *Image and Vision Computing 32, no. 4*, 2014.

[3] J. B. et al. Signature verification using a siamese time delay neural network. *Int. J. Pattern Recognit. Artif. Intell., vol. 7, no. 4, pp. 669688*, 1993.

[4] L. B. A. P. V. M. Farenzena, Michela and M. Cristani. Person re-identification by symmetry-driven accumulation of local features. *Computer Vision and Pattern Recognition (CVPR), IEEE Conference*, 2010.

[5] M. C. C. C. L. Gong, Shaogang and T. M. Hospedales. The re-identification challenge. *In Person re-identification, pp. 1-20. Springer, London*, 2014.

[6] 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.

[7] D. Gray and H. Tao. Viewpoint invariant pedestrian recognition with an ensemble of localized features. *European conference on computer vision*, 2018.

[8] X. Z. S. R. He, Kaiming and J. Sun. Deep residual learning for image recognition. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[9] P. M. R. Hirzer, Martin and H. Bischof. Person re-identification by efficient impostor-based metric learning. *Advanced Video and Signal-Based Surveillance (AVSS), IEEE Ninth International Conference*, 2012.

[10] M. F. S. Lavi, Bahram and I. Ullah. Survey on deep learning techniques for person re-identification task. 2018.

[11] R. Z. T. X. Li, Wei and X. Wang. Deepreid: Deep filter pairing neural network for person re-identification. *mage and Vision Computing 32, no. 4*, 2014.

[12] X. C. Z. Z. Li, Dangwei and K. Huang. Learning deep context-aware features over body and latent parts for person re-identification. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[13] Y. S. Ma, Bingpeng and F. Jurie. Covariance descriptor based on bio-inspired features for person re-identification and face verification. *Image and Vision Computing*, 2014.

[14] Y. F. T. X. W. W. J. Q. Y. W. Y.-G. J. Qian, Xuelin and X. Xue. Pose-normalized image generation for person re-identification. 2017.

[15] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *In ICLR*, 2015.

[16] J. L. S. Z. J. X. W. G. Su, Chi and Q. Tian. Pose-driven deep convolutional model for person re-identification. *In Computer Vision (ICCV), IEEE International Conference*, 2017.

[17] Z. K. Z. S.-J. H. D.-S. Wu, Di. Random occlusion-recovery for person re-identification. 2018.

[18] Y. Y.-Q. T. S. W. Yifan Sun, Liang Zheng. Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline). 2018.

[19] L. S. L. T.-S. W. J. W. Zheng, Liang and Q. Tian. Scalable person re-identification: A benchmark. *In Proceedings of the IEEE International Conference on Computer Vision*, 2015.

[20] L. Z. G. K.-S. L. Zhong, Zhun and Y. Yang. Random erasing data augmentation. 2017.