Person Re-identification using Siamese Networks

Ronak Harish Bhanushali
M.S. Robotics
Northeastern University
Boston, MA, USA
bhanushali.r@northeastern.edu

***Abstract*—Person re-identification poses a unique challenge similar to one-shot learning or few-shot learning tasks, where traditional deep neural networks typically demand substantial data for effective classification. This study delves into the realm of one-shot learning for person re-identification, focusing on implementing Siamese Networks to recognize individuals across diverse camera perspectives. Siamese Networks are particularly useful for this task due to their direct learning of image pair similarities. Training and testing are done using re-ID datasets including Market-1501, DukeMTMC-ReID and LAST, while performance assessment is done on metrics like Cumulative Matching Characteristics (cmc) and mean Average Precision (mAP). A robust baseline model, as proposed in "Bag of Tricks and A Strong Baseline for Deep Person Re-identification," serves as a benchmark for evaluating other deep re-ID models explored in this project. Alternative models, including a lightweight variant featuring [?]MobileNet-v3 as the backbone, and OSNet, are explored and their performances assessed. Code is accessible via https://github.com/ronakhb/PRCV-Final-Project.**

I. Introduction

Person re-identification is a unique and challenging task in computer vision. It involves learning certain features and traits of the subject only by using a single or few images and then comparing the generated feature vector with candidate images to try and re-identify the subject. This problem is exacerbated by the fact that there could be illumination changes, occlusions, changes in the person's pose, amount of background visible in the image due to the bounding box generated, domain shift in datasets, and changes in the background when trying to re-identify the person to state a few. These challenges make it difficult to solve the re-id problem with classical computer vision techniques. To make a robust system, we can use Convolutional Neural Networks (CNNs) and try to solve this problem.

Generally, for object classification or detection tasks, CNNs are trained on datasets with a certain number of unique objects which is the number of classes for the final classification layer. To achieve this, a lot of data is required for each class, and during inference, the model tries to predict the class of the unknown object. In a nutshell, the model tries to predict the class of the object from one of the classes it has seen during training from the training dataset. However, for a re-identification task, the model needs to learn traits in an image like the type of clothing, the color of clothing, and other distinguishing features that make up the subject of that image. It has to identify a person based on completely unseen data. The output of the model in this case is a feature vector instead of a class.

To generate a feature vector for any image, we can take a pre-trained classification model and remove the final classification layer to get a set of features after applying average pooling on the feature maps. Thus, for this task, it is crucial to select a robust model to get good feature vectors and use appropriate comparison metrics to figure out if two feature vectors belong to the same subject or not. Additionally, it is also challenging to train the model as we cannot use conventional classification losses. To solve this problem, this project follows the training tricks mentioned in [1] to speed up training using label loss and center loss for training.

II. Related Works

A. Bag of tricks

In [1], a strong baseline is discussed for the person re-identification task. The paper proposes a model along with a set of training tricks that performs really well on several well-known reID datasets.

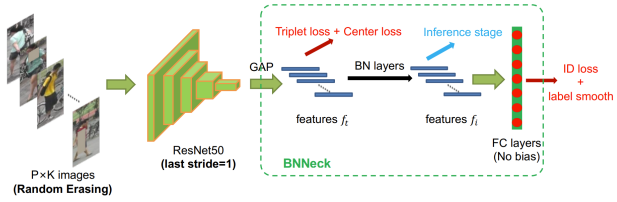

Fig. 1. Baseline model pipeline suggested in Bag of tricks.

The authors use ResNet50 as the backbone for feature extraction.

Fig. 1. shows the pipeline proposed in the paper. Model architecture -

- ResNet pretrained on imagenet used for getting 2048 feature maps
- Gloabal average pooling applied to get a feature vector of length 2048
- Batch norm is applied on these feature vectors. This helps to speed up training.
- A final fully connected layer is used with number of classes equal to number of identities in the dataset. This layer is only used to calculate part of the training loss.

Training tricks -

- Batch size of 64 is used consisting of 16 different people and 4 images of each person
- Image transforms used for training-
  - Resize to 256 x 128
  - Random erasing
  - Random Flipping
  - Convert pixel values to 32 bit floating in [0,1]. Normalize RGB channels by subtracting 0.485, 0.456, 0.406 and dividing by 0.229, 0.224, 0.225, respectively
- Using Adam with a warm up learning rate starting from 3.5×10^{-5}
- Using triplet loss from feature vectors and cross entropy loss from classification and combining both with center loss for training.

This architecture and tricks result in a strong baseline that achieves 94.5% rank-1 and 85.9% mAP on Market1501 [**?**]

B. Omni-Scale Feature Learning

Omni-Scale Feature Learning [2] proposes a novel approach for the problem of person re-identification. Its contribution is the development of a method for learning features at multiple scales to achieve a more robust and discriminative representation of person images. Existing person re-identification

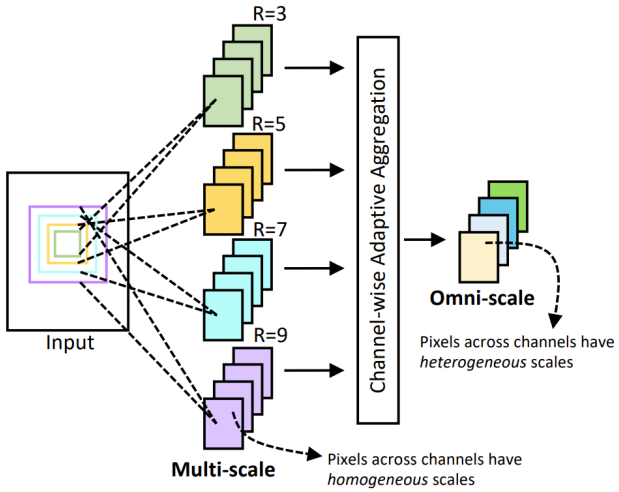

Fig. 2. Multiple Receptive fields used in OSNet

systems usually perform the feature learning on one single scale. It tends to ignore some important features in the images. In contrast, the proposed method, namely Omni-Scale Feature Learning, uses the features at multiple scales to enhance further the representation of person images. The technique uses an attention mechanism to combine features at different scales in a hierarchical way.

Fig. 2. shows the proposed method uses the features at multiple scales to improve the discriminative power of the learned representations. The effectiveness of the proposed method is experimentally supported on benchmark datasets. It shows that the proposed approach achieves better effectiveness than existing methods. OSnet does not use an existing backbone and is designed to be very light weight. It uses depth wise separable convolution to reduce the number of parameters.

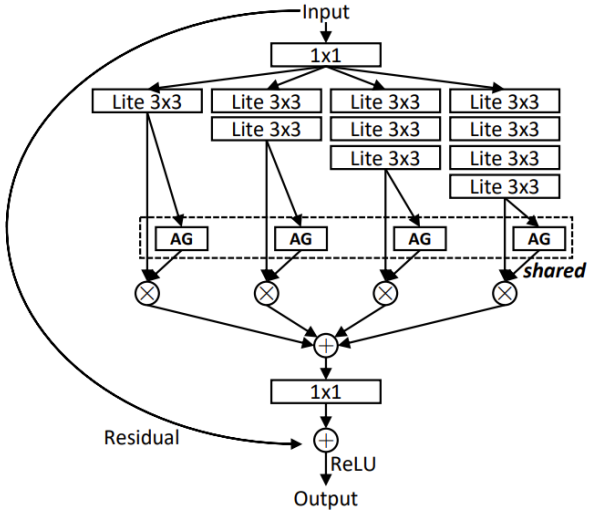

Fig. 3. Residual Block used in OSNet which combines multiple receptive fields

stage	output	OSNet
conv1	128×64, 64 64×32, 64	7×7 conv, stride 2 3×3 max pool, stride 2
conv2	64×32, 256	bottleneck × 2
transition	64×32, 256 32×16, 256	1×1 conv 2×2 average pool, stride 2
conv3	32×16, 384	bottleneck × 2
transition	32×16, 384 16×8, 384	1×1 conv 2×2 average pool, stride 2
conv4	16×8, 512	bottleneck × 2
conv5	16×8, 512	1×1 conv
gap	1×1, 512	global average pool
fc	1×1, 512	fc
# params		2.2M
Mult-Adds		978.9M

Fig. 4. OSNet Architecture

Fig. 3. shows how multiple receptive fields are achieved by stacking Depth wise convolutions. ”AG” seen in the image stands for aggregation gate which assigns different weights to different receptive fields and combines them. The input is combined with the output to form this omni-scale residual block. Fig. 4. shows the architecture in detail. We see that OSNet only has 2.2 Million parameters compared to the bag of tricks baseline which has more than 24 Million parameters

C. Evaluation of Distance Measures

In the inference stage of person reidentification, we should decide whether feature vectors extracted from two images

correspond to the same person. Though this is usually done using the Euclidean distance between these feature vectors, there are also cases when this would not be the best method. One limitation of Euclidean distance, for example, is that Euclidean distance assumes a Euclidean feature space, which might not fully capture the underlying structure, especially in high-dimensional representations as in deep learning.

Moreover, Euclidean distance is a non-invariant metric to feature scale and cannot model nonlinearity and inter-features relations in data. This would result in errors in similarity measurements for complex distributions of features or noisy data.

The consideration of intrinsic properties of the feature space, accounting for scale variation, nonlinearity, and data distributions, these metrics would offer a more robust and meaningful comparison of feature vectors, increasing the overall effectiveness of a reidentification algorithm. In [3] the authors talk about possible metrics for evaluating feature vectors -

- **Cityblock distance**: Also known as Manhattan distance, it measures the distance between two points in a grid based on the sum of the absolute differences of their coordinates. In the context of image feature comparison, it evaluates dissimilarity by considering the path length between corresponding pixels, offering insights into spatial arrangement.
- **Euclidean distance**: This is perhaps the most commonly used metric for measuring the dissimilarity between feature vectors. It calculates the straight-line distance between two points in feature space, providing a straightforward measure of dissimilarity that considers both magnitude and direction of differences.
- **Cosine distance**: It computes the cosine of the angle between two feature vectors in multidimensional space. By normalizing the dot product of the vectors, it measures dissimilarity while considering only the direction, not the magnitude, of the vectors. This is particularly useful when comparing feature vectors with varying magnitudes.
- **Minkowski distance**: This is a generalization of both Manhattan (Cityblock) and Euclidean distances. It allows for tuning the sensitivity to different dimensions by adjusting the exponent parameter. When the exponent is 1, it becomes equivalent to the Cityblock distance, and when it's 2, it becomes equivalent to the Euclidean distance.
- **Correlation distance**: It evaluates the correlation between feature points by considering their mean and standard deviation. It measures how features change together and is particularly useful for capturing similarities in feature distribution. It calculates dissimilarity based on how well one feature vector can be represented as a linear function of another.

In the study they conclude that cosine distance followed by correlation give the best results. Cosine distance works well because it measures dissimilarity from the angle in feature vectors. It was also found to be robust against variations in scale and sensitive to feature orientation. Correlation distance, on the other hand, estimates dissimilarity based on the linear relationship among feature vectors, making it robust against variations in feature distribution. The combined effectiveness of the two metrics in dissimilarity assessment produces the better overall performance of the study.

III. Methodology

In this project, three models have been trained and tested on two reID datasets. All models use a common training pipeline define by the bag of tricks baseline. Every model is trained for 120 epochs on Market1501 dataset and [**?**]Duke MTMC (Duke Multi-Tracking Multi-Camera) datasets.

A. Models

Input	Operator	exp size	$\#out$	SE	NL	s
$224^2 \times 3$	conv2d	-	16	-	HS	2
$112^2 \times 16$	bneck, 3x3	16	16	-	RE	1
$112^2 \times 16$	bneck, 3x3	64	24	-	RE	2
$56^2 \times 24$	bneck, 3x3	72	24	-	RE	1
$56^2 \times 24$	bneck, 5x5	72	40	✓	RE	2
$28^2 \times 40$	bneck, 5x5	120	40	✓	RE	1
$28^2 \times 40$	bneck, 5x5	120	40	✓	RE	1
$28^2 \times 40$	bneck, 3x3	240	80	-	HS	2
$14^2 \times 80$	bneck, 3x3	200	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	184	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	184	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	480	112	✓	HS	1
$14^2 \times 112$	bneck, 3x3	672	112	✓	HS	1
$14^2 \times 112$	bneck, 5x5	672	160	✓	HS	2
$7^2 \times 160$	bneck, 5x5	960	160	✓	HS	1
$7^2 \times 160$	bneck, 5x5	960	160	✓	HS	1
$7^2 \times 160$	conv2d, 1x1	-	960	-	HS	1
$7^2 \times 960$	pool, 7x7	-	-	-	-	1
$1^2 \times 960$	conv2d 1x1, NBN	-	1280	-	HS	1
$1^2 \times 1280$	conv2d 1x1, NBN	-	k	-	-	1

Fig. 5. MobilenetV3 Large Architecture

1) Baseline: The baseline model selected for this project is the model proposed by [1]. As explained in II.A), this model uses ResNet50 pretrained on Imagenet as a feature extractor. The model generates logits as well as feature vectors. During inference, only the feature vectors are returned, whereas during training, feature vectors as well as logits are returned to calculate loss. The feature vector generated by this model is of length 2048. The baseline model has 25.050927 million trainable parameters.

2) OSNet: This model is explained in section II.B). As previously explained, this model uses multiple receptive fields to try and capture the small and large features. The use of depth wise convolution helps to cut down on parameters. Along with depth wise convolution, they also use a width multiplier which reduces the number of output channels and helps to reduce the number of parameters. Detailed architecture is discussed in II.B). The feature vector generated by OSNet

is 1000 elements long. The model used with a 0.25 width multiplier has 1.017543 million trainable parameters.

3) Custom Lightweight model with MobileNetV3: To compare against the bag of tricks baseline and OSNet, a custom model was designed which used a pretrained MobileNetV3 Large as backbone for feature generation.

Fig. 5. shows the model architecture for MobilenetV3 Large. The bottleneck uses a special inverted residual block. Mobilenet also uses a depth wise convolution approach to keep the model light weight. For the custom model, we take the feature vector after pooling the output from conv2d, 1x1 layer. The feature vector length is 960 and the total number of trainable parameters in the model is 6.206663 million. This model lies in between the mentioned baseline and OSNet in terms of model size.

B. Datasets

The datasets used for this project were [**?**]Market1501, Duke MTMC-reID and [**?**]LAST datasets. All these datasets are standard datasets used for evaluating reID model performance. Duke MTMC is known to be relatively challenging to train on. LAST dataset is the most diverse and largest of the three.

Fig. 6. Datasets used for the project

1) Market1501: Market-1501 is the largest dataset for person re-identification, and it contains 32,643 fully annotated bounding boxes of 1501 pedestrians. Every person is captured in at most six camera views, and their bounding boxes are detected using the state-of-the-art Deformable Part Model (DPM) detector. The dataset is randomly divided into training and testing sets, to have 750 and 751 identities, respectively.

2) Duke MTMC: The DukeMTMC-reID dataset, released from the larger DukeMTMC dataset, is an image-based person re-identification dataset. It includes high-resolution video data captured by 8 cameras and has been characterized by manually bounding box cropped images of pedestrians. The resulting dataset thus comprises 16,522 training images representing 702 different persons, 2,228 query images from another 702 identities, and 17,661 gallery images. The dataset follows the evaluation protocol of a source.

3) LAST: LaST is a massive dataset containing over 228,000 images of pedestrians, which facilitates the exploration of scenarios where pedestrians are involved in various activities for a longer duration. Although the dataset is drawn from movies, the dataset undergoes meticulous curation, and therefore, the careful selection and labeling of frames make it reliable. In the training set, other than the identity label, clothing details are also annotated for the pedestrians. The training subset contains 5,000 identities and 71,248 images, and the validation subset contains 56 identities with 21,379 images. The test subset contains 5,806 identities and 135,529 images, which proof as an extensive and rich resource for research in pedestrian behavior analysis and other related fields.

C. Loss Criteria

Multiple losses are used for training the three reID models namely triplet loss, center loss and ID loss (Cross entropy loss) with label smoothing. We also use a technique called "Hard Negatives" to get good loss while training

1) Triplet Loss: Triplet loss is calculated using the feature vectors that we get from the network. This loss is also called margin loss. The formula for this is given as -

$L_{\text{Triplet}} = \max(d(a,p) - d(a,n) + \alpha, 0)$

- L represents the loss function.
- $d(a,p)$ denotes the distance between the anchor (a) and the positive sample (p).
- $d(a,n)$ denotes the distance between the anchor (a) and the negative sample (n).
- α is a margin hyperparameter.
- $\max$ denotes the maximum function, ensuring that the loss is non-negative.

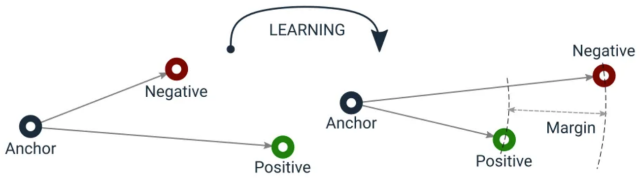

Fig. 7. Triplet loss

Fig. 7. shows how the triplet loss works. If the positive and negative samples lie closer to the anchor in feature space, the margin helps to generate a higher loss and tries to push them farther from each other

2) Center Loss: Triplet loss, denoted as L_{Triplet}, computes the difference between the feature distances of a positive pair (d_p) and a negative pair (d_n), with a margin of α. The margin ensures that the loss only contributes when the negative pair's distance exceeds the positive pair's distance. However, triplet loss neglects the absolute values of d_p and d_n, leading to similar loss values for cases with different absolute distances. Center loss, on the other hand, addresses this limitation by simultaneously learning class centers and penalizing distances between deep features and their respective centers. Formulated as L_C, center loss increases intra-class compactness by minimizing the distances to class centers, thereby effectively capturing intra-class variations.

The model combines these loss functions as $L = L_{\text{ID}} + L_{\text{Triplet}} + \beta L_C$, where L_{ID} represents the identification loss, β is a balancing weight, and L_C denotes the center loss. By integrating identification, triplet, and center losses, the model

aims to enhance feature discrimination while promoting intra-class compactness. This combined loss formulation contributes to more effective feature representation learning, facilitating improved performance. β used in this project is set to 0.0005

3) ID loss with Label Smoothing: Combining identification and label smoothing improves the training process by preventing overfitting and enhancing the generalization capability of the model. Instead of using hard labels, 0 or 1, for training examples, label smoothing uses soft probabilities, for instance, instead of setting the probability of the correct class to 1 and all the other classes to 0, label smoothing might spread the probability mass more uniformly among classes: 0.9 for the correct class and 0.1 equally divided among the other classes. This regularization method forces the model to not be too confident in its predictions and learn more robust and generalizable features. The loss function used for ID loss in this case is binary cross entropy loss

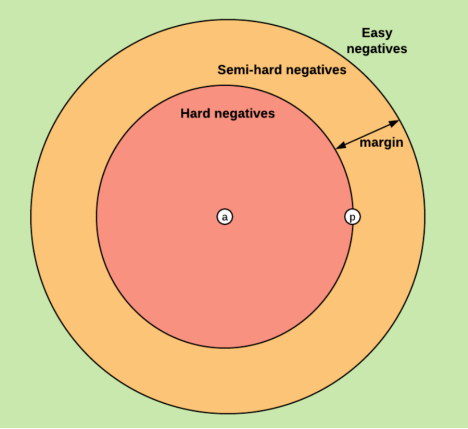

Fig. 8. Hard Negatives

4) Hard Negatives: The concept of Hard negatives is used while training to ensure a robust model. In Fig. 8. the center of the circle is the anchor and the first concentric circles show the distance between the anchor and a positive sample. Any negative sample with distance less than this circle is considered a hard negative. During training, we smartly choose such negatives from the batch to maximize the triplet loss generated and make the inner circle smaller. The figure also shows how the margin separates the negatives from the positives.

D. Evaluation Criteria

Evaluation criteria used for this project are CMC (Cumulative Matching Characteristic) and mAP (mean Average Precision). In evaluating person re-identification models, CMC (Cumulative Matching Characteristic) and mAP (mean Average Precision) serve as crucial metrics. CMC measures the probability of finding the correct match within the top-K ranked images in the gallery, reflecting the model's ranking performance. On the other hand, mAP assesses the precision and recall across all possible rank thresholds, providing an overall measure of the model's accuracy and ranking quality. Together, these metrics offer valuable insights into the effectiveness of ReID models in accurately identifying and ranking persons, guiding the development and assessment of robust ReID systems.

IV. Experiments and Results

All three models mentioned above were trained on Market1501 and DukeMTMC. LaST dataset was used to evaluate the performance of the models. Given the timeline of this project, the following experiments were carried out -

- Training baseline on Market1501 and DukeMTMC for 120 epochs
- Training OSNet with 0.25 multuplier on Market1501 and DukeMTMC for 120 epochs
- Training Custom MobileNetV3 network on Market1501 and DukeMTMC for 120 epochs
- Fine tuning baseline trained on Market1501 on DukeMTMC and vice versa for 10 epochs
- Fine tuning custom model trained on Market1501 on DukeMTMC and vice versa for 10 epochs

TABLE I
Performance comparison of different models on Market1501, Duke, and LaST datasets.

Model	Market1501		Duke		LaST	
	r1	map	r1	map	r1	map
Baseline	89.4%	75.5%	80.4%	66.1%	35.0%	6.9%
OSNET 0.25x	77.3%	56.3%	69.9%	50.8%	23.0%	4.3%
Custom model	77.0%	56.3%	51.4%	31.8%	25.0%	5.2%

Table 1 shows the Rank1 accuracy and mean Average Precision of each model for each of the datasets. OSNet being the most lightweight model still outperforms the custom model and performs very close to the baseline model. It should be noted that the datasets are quite different from each other with LaST having the most variation for the same ID. All models perform well for Market1501, followed by Duke MTMC and LaST at the last place. The custom model seems to be underperforming for all the datasets.

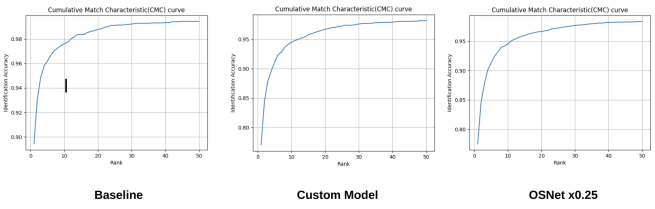

Fig. 9. CMC Plot for models trained on Market1501

The figures show cmc plots for all the models. The accuracy is more than 90% at rank 10 for Market1501. For Duke MTMC, baseline and OSNet cross 90% at rank 10 but the custom model seems to underperform. All three models perform badly for LaST. Fine-tuning seems to work well enough. Without fine-tuning, the r1 accuracy is 10%-15% less. The fine-tuning is done for only 10 epochs. With more epochs, we

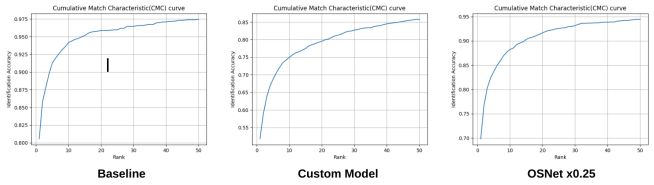

Fig. 10. CMC Plot for models trained on Duke MTMC

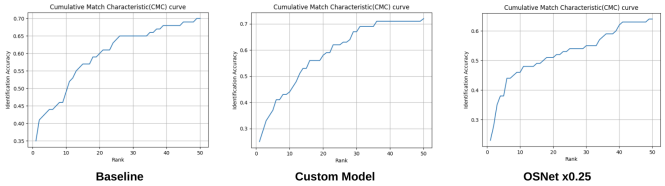

Fig. 11. CMC Plot for models trained on LaST

could easily transfer the learning and save training time across datasets.

To avoid inserting a lot of images, the training plots for all the runs can be found on the GitHub page of the project.

Different distance metrics like cosine distance and correlation distance were also tried out to see if they changed the performance. Please note that due to time limitations, the models were not retrained using these metrics. Only the inference part used these metrics

- Cosine distance - Cosine distance gave the same cmc and mAP results for all models and datasets. It affects the output distance that we get which is useful when the model is used as a standalone module with a distance threshold. Since the distance matrix generated for cmc and mAP only takes into account the relation between the distances and not the absolute values, the cmc and mAP values looked the same
- Correlation distance - Correlation distance gave slightly better results for cmc and mAP with increments of 0.1% which is not significant but still a small improvement.

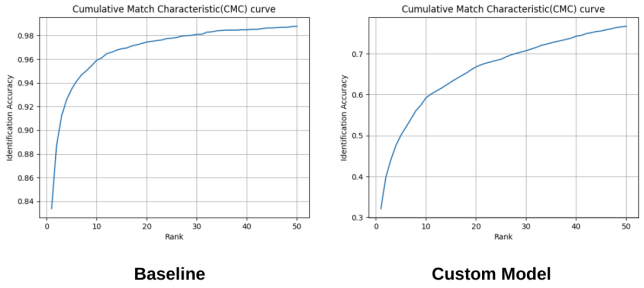

Fig. 12. CMC Plot for models trained on Market1501 and fine tuned on Duke MTMC

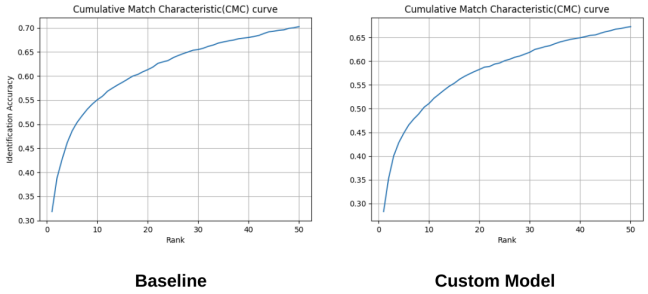

Fig. 13. CMC Plot for models trained on Duke MTMC and fine tuned on Market1501 MTMC

V. Discussion and Summary

This project explored new techniques specific to the task of person reidentification and the effects of the techniques on accuracy and training times. The results in this project as well as comparable work done in [1],[2],[3] show that there is a lot more to be explored in the reidentification models and few-shot learning field. One of the significant findings is the fact that there is a big domain shift when we go from one dataset to another and it shows that there is no one-size-fits-all model for this. Models trained on one dataset perform poorly on other datasets due to differences in camera angles, surroundings, and image quality.

Another finding is that the model need not be extremely bulky for it to do well. OSNet is a great example of such a lightweight model that gave comparable results to the baseline which had 20 times more parameters.

The distance metrics also are a crucial part of this project. The three distance metrics tested in this project gave similar results, however, that could change if the models are trained using those distance metrics while training. A future scope for this project which was not completed due to time restrictions is using a fully connected layer at the end to compare the feature vectors. The feature vectors can be concatenated and passed to the final layer which would have only one binary

Acknowledgment

I would like to thank Prof Bruce Maxwell for the great lectures and notes. Projects and exams in the course helped to make this project possible in a short time. I would also like to thank the TAs for their guidance and help throughout the course

References

[1] Hao Luo, Youzhi Gu, Xingyu Liao, Shenqi Lai, and Wei Jiang. ”Bag of Tricks and a Strong Baseline for Deep Person Re-Identification.” In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, June 2019.

[2] Kaiyang Zhou, Yongxin Yang, Andrea Cavallaro, and Tao Xiang. ”Omni-Scale Feature Learning for Person Re-Identification.” In ICCV, 2019.

[3] K. Kavitha, B. Thirumala Rao, and B. Sandhya. ”Evaluation of Distance Measures for Feature-based Image Registration using AlexNet.” International Journal of Advanced Computer Science and Applications (IJACSA), vol. 9, no. 10, 2018, pp. 284

[4] Xiujun Shu, Xiao Wang, Xianghao Zang, Shiliang Zhang, Yuanqi Chen, Ge Li, and Qi Tian. "Large-Scale Spatio-Temporal Person Re-identification: Algorithm and Benchmark." arXiv preprint arXiv:1812.02806, 2018.

[5] Lin Wu, Yang Wang, Junbin Gao, and Dacheng Tao. "Deep Co-attention based Comparators For Relative Representation Learning in Person Re-identification." arXiv preprint arXiv:2003.11539, 2020.

[6] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, and Hartwig Adam. "Searching for MobileNetV3." arXiv preprint arXiv:1905.02244, 2019.

[7] Mang Ye, Jianbing Shen, Gaojie Lin, Tao Xiang, Ling Shao, and Steven C. H. Hoi. "Deep Learning for Person Re-identification: A Survey and Outlook." IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 8, pp. 1945-1966, 2020. DOI: 10.1109/TPAMI.2019.2940538

[8] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. "Scalable Person Re-identification: A Benchmark." In IEEE International Conference on Computer Vision (ICCV), 2015.