Learning Deep Convolutional Embeddings for Face Representation Using Joint Sample- and Set-based Supervision

Baris Gecer, Vassileios Balntas, and Tae-Kyun Kim
Department of Electrical and Electronic Engineering,
Imperial College London
{b.gecer,v.balntas15,tk.kim}@imperial.ac.uk

Abstract

In this work, we investigate several methods and strategies to learn deep embeddings for face recognition, using joint sample- and set-based optimization. We explain our framework that expands traditional learning with set-based supervision together with the strategies used to maintain set characteristics. We, then, briefly review the related set-based loss functions, and subsequently we propose a novel Max-Margin Loss which maximizes maximum possible inter-class margin with assistance of Support Vector Machines (SVMs). It implicitly pushes all the samples towards correct side of the margin with a vector perpendicular to the hyperplane and a strength inversely proportional to the distance to it. We show that the introduced loss outperform the previous sample-based and set-based ones in terms verification of faces on two commonly used benchmarks.

1. Introduction

Recently, deep convolutional neural networks (CNNs) have been an important tool that achieves state-of-the-art performances in many computer vision tasks [13]. Its goal is to build a model to address a target problem with a sequence of convolutional layers that are developing from low-level features to more abstract representations. Deep networks can also learn robust representations that are suitable for other task [2, 21, 18, 22, 4]. Deep Distance Metric Learning (DML) approaches explore ways to construct such representations that maintain better similarity/distance measurement for, *e.g.*, verification, retrieval or clustering tasks. While supervision of traditional objective functions (*e.g.* Softmax Loss) yield successful results, comparative loss functions (i.e. Triplet Loss) are shown to be more suitable for semi-supervised deep DML tasks[17].

Beside sample-based supervision which processes each sample individually, one can benefit from the captured information by considering a set of images as a unified entity. An image set is a collection of instances of the same object/person from varying viewpoints, illuminations, poses and exhibits different characteristics. A set contains richer information of the target than a single image and is potentially more useful for problems like object or scene classification, face recognition and action analysis. As the authors of [31] illustrated, set-based supervision can learn discriminative features rather than just separable features like sample-based approaches would learn.

This paper makes the following contributions:

- We propose a novel loss function called Max-Margin Loss that benefits from set-based information by drawing inter-set (inter-class) margins. It improves the separability of learned features by maximizing the maximum possible inter-class margin that is calculated by a support vector machine and address the shortcomings of the existing set-based methods.
- We review existing set-based DML approaches and evaluate them and their combinations together with Max-Margin Loss and Softmax Loss.
- We build a framework where such functions can operate properly jointly with sample-based ones and investigate the strategies to maintain set information during training in the framework.

The rest of the paper is organized as follows: In Section 2, we provide an overview of the related work about sample-,set-based deep metric learning for face recognition. Section 3 describes existing and new set-based loss functions and other strategies. In Section 4, we provide some more information on the set-up and techniques used in the experiments. We then present and discuss some experimental results. Finally, we draw conclusion and elaborate on future works in Section 5.

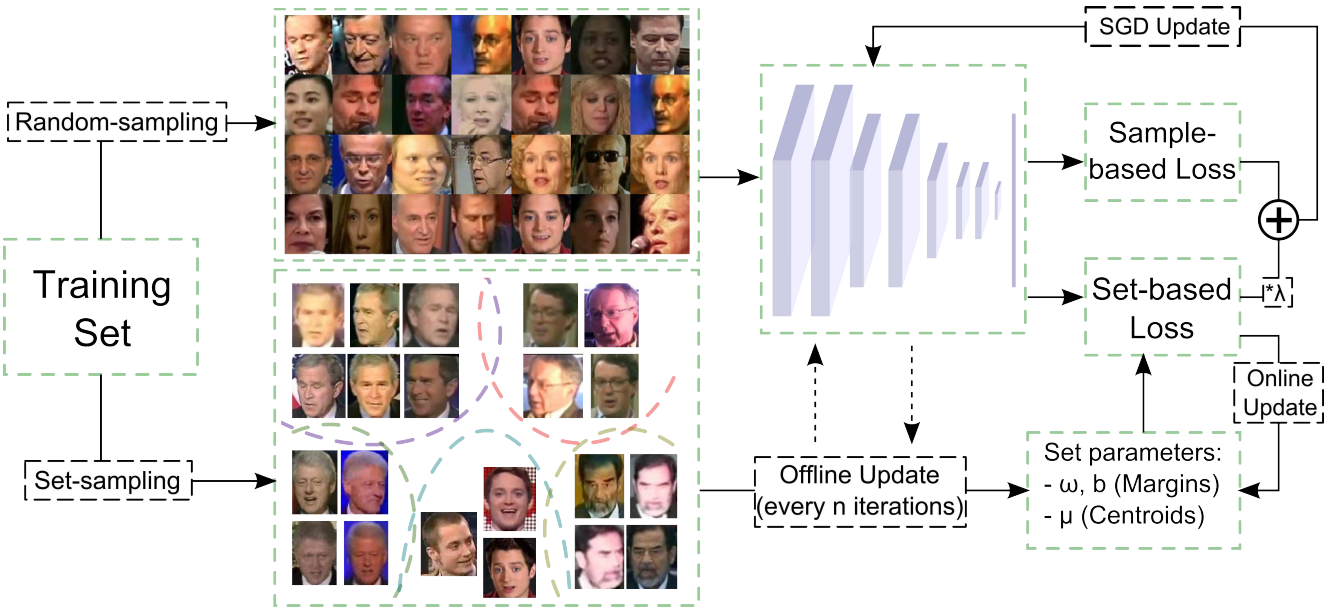


Figure 1: Overview of joint sample-based set-based learning. Random face images sampled from the training images in traditional fashion to train a CNN. **Offline Update:** In every n iterations, set of face images are sampled that consist significant amount of images from each identities and fed into the network while training is paused. Resulting feature vectors are used to calculate set parameters whose way is specific to set-based loss used. **Online Update:** While the training is going, set parameters are updated with a small weight by the parameters calculated with current random batches.

2. Related Work

While the **traditional embedding** approaches include Neighbourhood Component Analysis [5], Large Margin Nearest Neighbour [30] and Nearest Class Mean [15], state-of-the-art performances are usually achieved by **deep DML** networks. Contrastive Loss [6] is one such approach where the features are learned with supervision of a loss computed with (positive or negative) pairs of samples. Triplet Loss [30] optimizes the relative difference between a positive and a negative pair. Both functions share the goal to minimize the distances between the samples from the same class and to maximize the distances between the samples from different classes. Several extensions were proposed such as lifted structured embedding [16] where an advanced hard sample mining introduced within mini-batches for efficiency, and quadruplet embedding [8] that employs local similarity awareness.

There have been many methods developed for **set-based recognition** such as CCA [11] , Manifold-Manifold Distance [28], Sparse Approximated Nearest Points [7], Simultaneous Feature and Dictionary Learning [14], Discriminant-Analysis on Riemannian Manifold of Gaussian Distribution [29]. Yet, recent **set-based deep DML studies** show excellent performance, such as Rippel *et al.* [19] proposed magnet loss that achieve local discrimination by penalizing class distribution overlap and Feng *et al.* [3] combined set presentations (mean, variance, min, max, vlad features) with hashing in a single network for end-to-end learning of binary code of sets. Wen *et al.* [31] did the first attempt to combine sample-based loss functions (*e.g.* softmax, contrastive, triplets) with a set-based term called center loss which minimizes the distance of each sample with its corresponding class center.

Those studies come with their strategies to compute set parameters (*e.g.* clusters, centroids, margins) on-the-go as well. Rippel *et al.* [19] pause training periodically to cluster samples on the new feature space, Wen *et al.* [31] calculates class centroids with vanilla update with momentum in every iteration. After every iteration of the ongoing learning, feature space is being bended and therefore above approximations should be biased. While the first uses the same cluster indices until next refreshment, in the latter momentum update would lead to aggregation of parameter vectors of different feature spaces. Although they are still good approximations, using both ideas together should yield less biased approximations as we do in our experiments.

Most of above sample- or set-based deep DML studies revolve around learning features by pulling positive samples and pushing negative samples. In fact, more discriminative features can be learned by increasing the inter-class distances without forcing to pull all the samples to the same

point (i.e. centroid). Although Wen *et al*. [31] claims to learn discriminative features rather than just separable ones, Center loss keep pulling and pushing samples no matter how distinct they are. Although Magnet Loss take care of such intra-class variation tolerance with its multi-cluster models, its sophisticated sampling procedure make it difficult to combine it with sample-based objectives. The proposed Max-Margin Loss, on the other hand, cover these problems by calculating inter-class separating hyperplanes and pushing all the samples to the correct side of the margin accordance with their proximity to the margin. This procedure eventually increase maximum possible margin between sets without distorting the intra-class distribution.

Tang [26] attempted to learn with margin-based optimization by minimizing squared hinge loss for classification. Yet, ignoring the weight term(w) in the differentiation appears to be penalizing only slack variables rather than increasing the maximum achievable margin by SVM. Further, the study is not clear about integration of SVM with SGD in the loss layer. [1] is another related SVM based study where SVMs are used during the testing time for template adaptation rather than supervising the network to learn better embedding as in our case.

3. Proposed Set-based Learning Framework

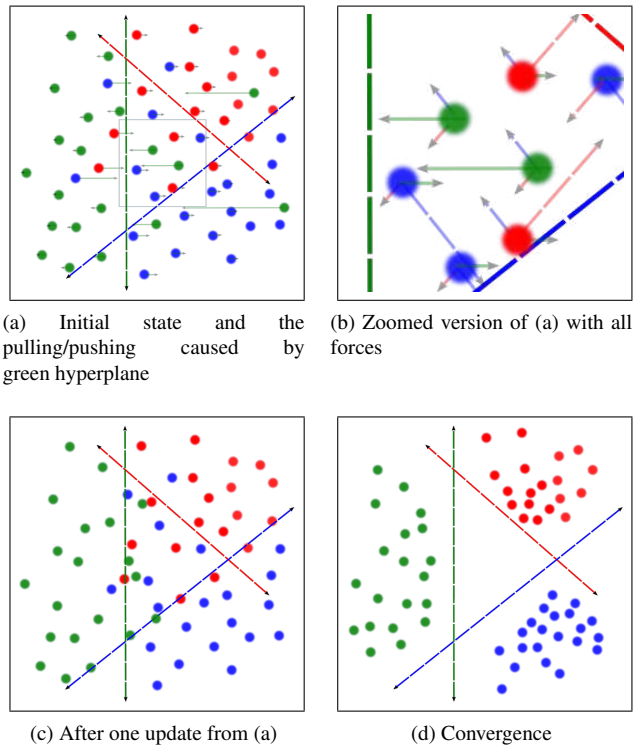
In this section, we present our framework to combine sample-based and set-based learning using our novel set-based Max-Margin Loss and two other set-based loss functions similar to the existing works. Let us begin with generalized version of the joint loss formula given by [31] as following:

$$\mathcal{L} = \sum_i \lambda_i \mathcal{L}_{i(Sample-based)} + \sum_j \lambda_j \mathcal{L}_{j(Set-based)} \quad (1)$$

Sample-based loss functions such as Softmax ($\mathcal{L}_S = -\sum_{i=1}^{m} log \frac{e^{W_{y_i}^T + by_i}}{\sum_{j=1}^{n} e^{W_{y_i}^T + by_i}}$) or Triplet are well defined and studied in the literature[20, 17, 23]. They often guide a network fed by random batch of input data and without a need of any other information while training.

Beside sample-based supervision, one can benefit from the information extracted by considering set of samples as a whole. Unlike sample-based ones, set-based terms require additional set parameters (*e.g*. linear margin parameters(ω, b), centroids(μ)) which represent statistics or characteristics of sets. Aggregation of many sample-based and set-based loss terms has a potential of leading to better representation as each may optimize different aspects of the problem.

Below we study strategies to extract set statistics and characteristic for set-based learning. Then we introduce a new set-term and review several set-based loss terms similar to the previous studies.



(a) Initial state and the pulling/pushing caused by green hyperplane

(b) Zoomed version of (a) with all forces

(c) After one update from (a)

(d) Convergence

Figure 2: Initially, Max-Margin Loss requires a good embedding as in (a) to calculate separating hyperplanes. The loss applies to all samples by green plane is indicated with arrows. (c) shows the state after one update for only green plane and (d) at convergence.

3.1. Set Parameters

Fig. 1 summarizes joint set- and sample-based learning and shows how set batches and set parameters are operated. As in traditional deep networks, a number random samples (a batch) is fed into the network to compute sample-based loss (*i.e*. Softmax). Then, set-based loss is also computed based on pre-computed set parameters and the weighted sum of their derivatives are backpropagated through the network.

Set parameters are updated periodically in two ways (online and offline) to maintain set-based terms. The best approximation to set parameters can be calculated by sampling a significant number of samples[1] from each identity. Those samples are fed into the network while the network parameters are fixed and set parameters are determined from their features. We call this operation 'offline update' which is computationally costly and therefore done in every n iterations.

As training continues, resulting feature space is also

[1] We found that 50 images are representative enough

changing, thus the set parameters need to be kept on track. 'Online update' intends to correct this bias in every iteration by averaging current set parameters with computed set-parameters given the current random batch at the hand. Since number of samples from each class is small, the weight of online-parameters is also small while averaging. While online update keep adapting set parameters to the changing feature space during optimization, offline update periodically correct the biased set parameters caused by mixing parameters of different feature spaces.

3.2. Set-based Loss Functions

3.2.1 Max-Margin Loss

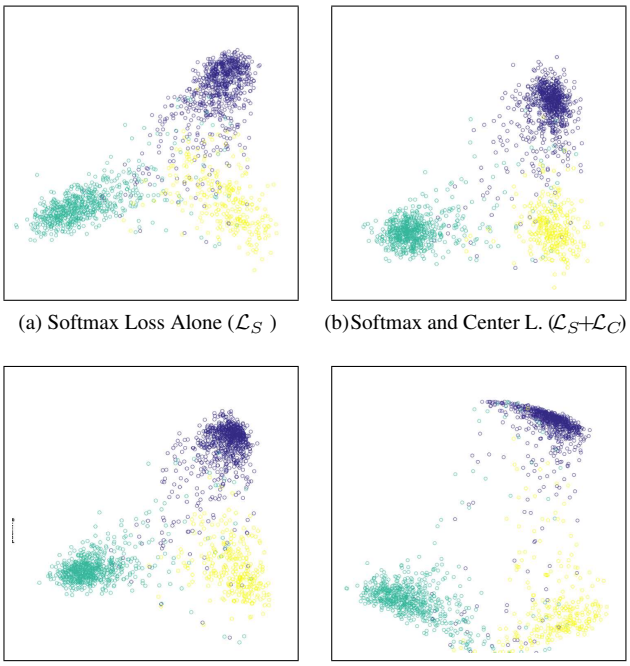
We propose a novel set-based term, Max-Margin Loss, that maximizes the maximum possible margin between classes. This objective function implicitly pushes all the samples towards correct side of the margin with a vector perpendicular to the hyperplane and a strength inversely proportional to the distance to the hyperplane. Even the samples in the correct side of the margin are kept being pushed to increase the maximum margin between the two sets without distorting the intra-class distribution. Fig. 2 illustrates a synthetic feature space over the iterations supervised by Max-Margin Loss. Given a mini-batch of random n samples uniformly sampled from m classes, and let the embeddings and the corresponding class labels denoted by (x_i, y_i), the loss and its gradient are computed by the following formulas:

$$\mathcal{L}_M = \lambda_M \sum_{i=1}^{n} \sum_{j=1}^{m} \frac{1 - \overline{\delta}(y_i = j)}{m-1} e^{-\frac{\overline{\delta}(y_i = j)(w_j^T x_i + b_j)}{||w_j||_2}} \tag{2}$$

$$\frac{\partial \mathcal{L}_M}{\partial x_i} = \frac{\lambda_M (1 - \overline{\delta}(y_i = j))}{n(m-1)} \sum_{j=1}^{m} \frac{-w_j \overline{\delta}(y_i = j)}{||w_j||_2} e^{-\frac{\overline{\delta}(y_i = j)(w_j^T x_i + b_j)}{||w_j||_2}} \tag{3}$$

where separating hyperplane for j class is defined as $w_j^T x + b = 0$ and $\overline{\delta}(condition)$ equals to 1 if the condition is satisfied and -1 otherwise.

Set parameters of Max-Margin Loss (ω_j, b_j) are determined by online and offline updates explained in the previous Section (3.1). For offline update, we pause the training in every n($= 500$) iterations and after set-sampling, features are extracted from the current state of the network. We then run a support vector machine with linear kernel for each class to obtain best separating hyperplanes in one-against-all manner and save the resulting parameters. Online update is done by running SVM for classes that are represented in the current (random) batch and averaging them with the current parameters with a small weight ($\alpha = 0.01$).



(a) Softmax Loss Alone ($\mathcal{L}_S$) (b)Softmax and Center L. ($\mathcal{L}_S$+$\mathcal{L}_C$)

(c) Softmax and Pushing Loss ($\mathcal{L}_S + \mathcal{L}_P$) (d) Softmax and Max-Margin Loss ($\mathcal{L}_S + \mathcal{L}_M$)

Figure 3: 2D embeddings of images of three learned with different loss functions

3.2.2 Center Loss

Center Loss is proposed by Wen *et al*. [31], minimizes intra-class variety by penalizing distance from samples to their class centroids. This is in fact, a simplified version of the numerator term of Magnet Loss[19] with one cluster.

Beside the original definition of the Center loss which includes momentum vanilla update for centroids, we refresh centroids of classes periodically during the training. And unlike the original study [31], we introduce Center loss after pretraining the network a while with Softmax alone as having center loss from the beginning would bias deep-net as features and centroids are not yet meaningful.

Center Loss can be defined as below:

$$\mathcal{L}_C = \frac{\lambda_C}{2} \sum_{i=1}^{n} ||x_i - c_{y_i}||_2^2 \tag{4}$$

where λ_C is balancing term and centroids are computed as:

$$c_j = \frac{\sum_{i=1}^{n} \delta(y_i = j) x_i}{\sum_{i=1}^{n} \delta(y_i = j)} \tag{5}$$

where $\delta(condition)$ equals to 1 if the condition is satisfied and 0 otherwise.

3.2.3 Pushing Loss

Pushing Loss penalizes very close negative class centroids where penalty decrease exponentially with increasing distance as distant centroids should have much less influence. Here, centroid update and refreshment procedures are kept same as center loss. The formulation of pushing loss is as following:

$$\mathcal{L}_P = \frac{\lambda_P}{m} \sum_{i=1}^{n} \sum_{j \neq y_i} e^{-||x_i - c_j||_2} \quad (6)$$

This is also similar to the denominator term of Magnet loss[19]. Although magnet loss contains multiple clusters for attribute concentration, we believe its effect would be minimal for recognition/verification tasks, since deep network should be capable non-linear mapping of multiple clusters into one centroid very easily.

3.3. Toy Experiments

We modify our network by setting the dimension of the embedding layer to 2 and train it by supervision of above loss functions using samples of only three people. We plot the embeddings of the samples of individuals with different colors as shown in Figure 3 and observe the contribution of each function.

Although Fig.3(a) shows that softmax alone learn a good representation, it is improved with the contribution of set-based functions. Center Loss and Pushing Loss seems to be functioning similar and Max-Margin Loss looks like providing slightly better separation in terms of identities.

4. Experiments

In our experiments, we aim to understand contribution of different set parameter update strategies and compare Max-Margin Loss and other set-based approaches in the same settings.

4.1. Implementation Details

Training settings: We use NNS1 network from [20] for training which is a reduced version of Google's inception architecture [24]. We increase the dimension of the embedding layer from 128 to 512 and adjust Softmax layer for 2,558 identities. The network is fed with 96×96 pixel images which are augmented randomly with cropping (between $\%70 - \%100$), location, aspect ratio (between $7/8 - 8/7$), flipping (0.5 chance), blurring (0.5 chance), brightness, contrast and saturation in every iteration. Input images are linearly scaled to have zero mean and unit norm. SGD is optimized by Adam solver [12] with batch size of 1024 on Nvidia Titan X GPU and we use MatConvNet library [27] with a number of modification. We set weight decay to 0.0005 and use batch normalization to avoid overfitting. Training is started with a learning rate of 0.001 and divided by 10 at the 15th and 25th epochs and stopped at 30th epoch.

Figure 4: Fine-tuning balancing term λ

Training Data: We use non-aligned and curated version of VGG Face dataset [17] which consist of around 1M face images of 2,558 individuals who are not included in Youtube Faces (YTF) [32] and Labeled Faces in the Wild (LFW) [10] benchmark datasets. Unfortunately, the dataset is publicised by means web links, thus some samples are missing due to broken links. We end up training with reduced version of VGG dataset that consists 0.83M training samples from 2,558 identities.

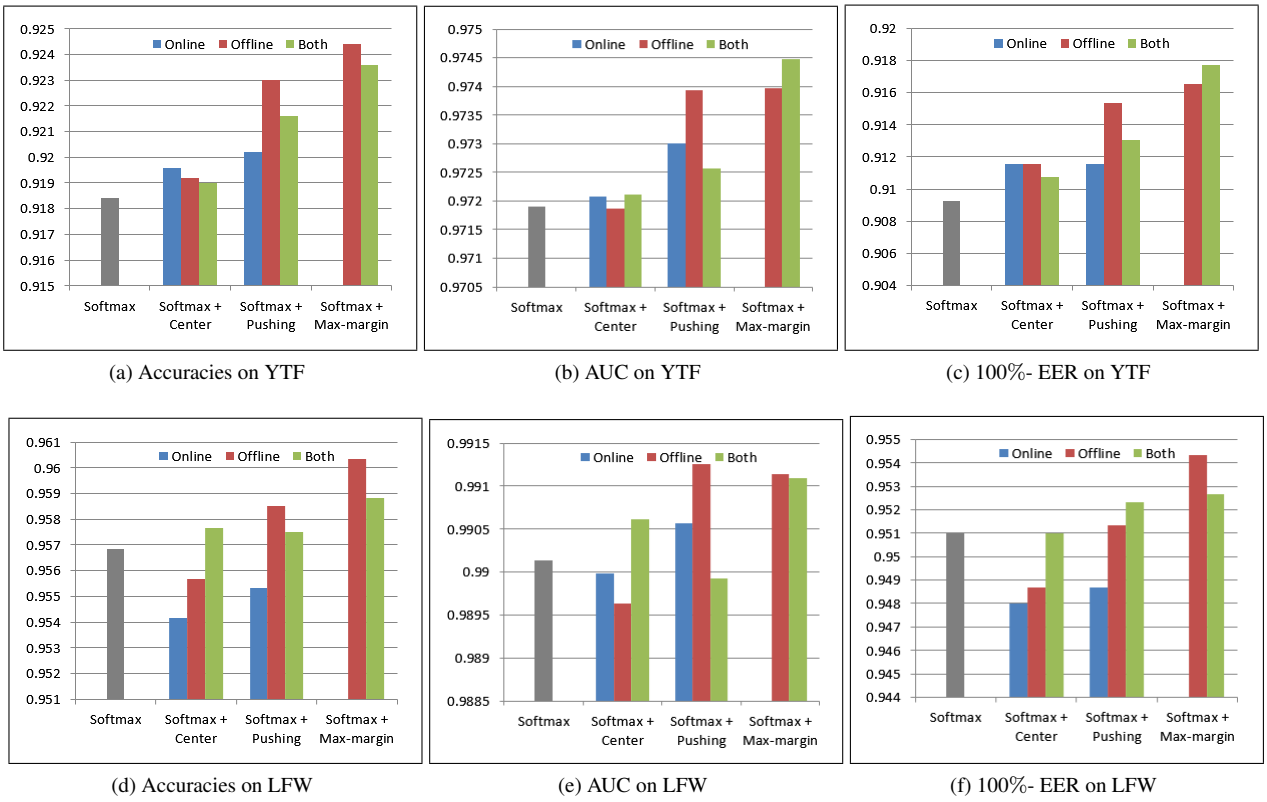
Testing: We evaluated performance of Max-Margin Loss and other functions on commonly used YTF and LFW datasets. For both, we follow the defined protocol for the restricted settings with external training. After training, we kept the models fixed and tested on those datasets without further training unlike [17]. Images are aligned as provided in YTF and for LFW, deep funneling [9] is used for alignment. We use embedding layer output of each image as representation and average features of frames from the same videos (only for YTF). Similarity between pairs of images or videos is computed by cosine distance of mean feature vectors.

4.2. Balancing Term Tuning

Despite motivations using set-based terms, sample-based terms are necessary to stabilize supervision as set characteristics may not be always fully presented in the feature space. Therefore, we train our networks to obtain good features for the first 15 epochs with only supervision of Softmax.

After keeping the pretrained model fixed, we combine set-based loss functions with a balancing term (λ) which is fine-tuned on a small subset[2]. According to the Figure 4, we fix λ parameters to $\lambda_M = 0.03, \lambda_P = 0.03, \lambda_C = 0.0001$ in the rest of the experiments.

[2]One fifth of YTF data set



(a) Accuracies on YTF (b) AUC on YTF (c) 100%- EER on YTF

(d) Accuracies on LFW (e) AUC on LFW (f) 100%- EER on LFW

Figure 5: Comparison of the three set-based loss function with online, offline update strategies for set parameters.

λ_C	Update	AUC	Acc.	100%- EER
0.003	Online	98.98	95.45	94.80
0.0001	Online	99.00	95.41	94.80
0.003	Both	98.91	95.43	94.80
0.0001	Both	99.06	95.77	95.10

Table 1: Performance of Center Loss ($\mathcal{L}_S + \mathcal{L}_C$) on LFW dataset under different settings. The settings used in the original Center Loss paper [31] (first line) is gradually improved in our experiments.

4.3. Effectiveness of Online-Offline Updates

In order to justify the small changes we made with Center Loss, we have done some controlled experiments to observe effect of using offline update and new finetuned λ_C parameter. As can be seen in Table 1 Each seems to be contributing slightly ending up with around $\%0.3$ improvement on LFW dataset.

Figure 5 show comparison of Max-Margin Loss with other loss functions with different update rules. Although we see close performances, Max-Margin Loss show slightly better performance over other set-based terms. Effect of different update strategies show that offline update alone show consistently better performances which is not expected. The reason could be online set parameter update is done with too big α parameter. We conclude that further investigation of α parameter is needed

4.4. Benchmark Performances

Verification performance results of the proposed Max-Margin Loss is compared with the other state-of-the-art methods in Table 2 and other set-based functions in Figure 5 for LFW and YTF datasets. Among set-based terms, we obtain the best performance with Max-Margin Loss where we improve $\%0.35 - 0.6$ over Softmax. The improvement seems to be small as we add set terms after pre-training with only softmax for 15 epochs which we have the 10 and 100 times lower learning rate. Yet, we are not interested achieving state-of-the-art performance as long as we can compare different set-terms in a meaningful experimental settings.

While comparing with other methods, one should also notice the differences in number of training images, identities, input size and network specifications. Most of those studies have such advantages and not directly comparable

Method	#Training Images	#Ids	Input Size	Network (#Params.)	FT on YTF or LFW	Accuracy on YTF (%)	Accuracy on LFW (%)
DeepFace [25]	4.4M	4,030	152×152	AlexNet(120M)	No	91.4	97.35
VGG Face [17]	2.62M	2,622	224×224	VGG(138M)	Yes	97.3	98.95
VGG Face [17]	2.62M	2,622	224×224	VGG(138M)	No	91.6	-
$\mathcal{L}_S + \mathcal{L}_M$	0.83M	2,558	96×96	NNS1(26M)	No	**92.44**	**96.03**

Table 2: Verification performance comparison of different loss functions and methods on YTF dataset. Note that our method uses fairly less training samples with lower input size on a shallower network. Further we do not fine-tune our network on the target datasets like [17]

with our results. For example, VGG Face[17] yields a significant improvement in the results when they further train (FT) their network on the test set (YTF and LFW) with cross validation. Our baseline Softmax Loss achieve similar accuracy with VGG Face[17] without FT, although VGG Face is trained with around 3 times larger training set, 5 times bigger input size and 5 times deeper network. While our baseline score is around such state-of-the-art study, we can argue that including such set-based terms would improve their results as well as our framework is compatible with their designs.

5. Conclusion

This paper studies joint sample-based and set-based embedding learning for face recognition. We review different set terms in the literature and propose nove Max-Margin Loss. We also explain strategies to maintain set-based learning during training.

Our results show the contribution of different terms and validity of the proposed set-based function which yields slight improvement over softmax baseline. Further experiments give us better insight about set-based learning methods. Without aiming is still an ongoing work.

Acknowledgements

Baris Gecer is funded by the Turkish Ministry of National Education.

References

[1] N. Crosswhite, J. Byrne, O. M. Parkhi, C. Stauffer, Q. Cao, and A. Zisserman. Template adaptation for face verification and identification. *arXiv preprint arXiv:1603.03958*, 2016. 3

[2] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *Icml*, volume 32, pages 647–655, 2014. 1

[3] J. Feng, S. Karaman, I. Jhuo, S.-F. Chang, et al. Deep image set hashing. *arXiv preprint arXiv:1606.05381*, 2016. 2

[4] B. Gecer. Detection and classification of breast cancer in whole slide histopathology images using deep convolutional networks. *Diss. Bilkent University*, 2016. 1

[5] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov. Neighbourhood components analysis. In *Proceedings of the 17th International Conference on Neural Information Processing Systems*, NIPS'04, pages 513–520, Cambridge, MA, USA, 2004. MIT Press. 2

[6] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, volume 2, pages 1735–1742. IEEE, 2006. 2

[7] Y. Hu, A. S. Mian, and R. Owens. Face recognition using sparse approximated nearest points between image sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(10):1992–2004, 2012. 2

[8] C. Huang, C. C. Loy, and X. Tang. Local similarity-aware deep feature embedding. In *Advances in Neural Information Processing Systems*, pages 1262–1270, 2016. 2

[9] G. B. Huang, M. Mattar, H. Lee, and E. Learned-Miller. Learning to align from scratch. In *NIPS*, 2012. 5

[10] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, Technical Report 07-49, University of Massachusetts, Amherst, 2007. 5

[11] T.-K. Kim, J. Kittler, and R. Cipolla. Discriminative learning and recognition of image set classes using canonical correlations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6), 2007. 2

[12] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[13] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015. 1

[14] J. Lu, G. Wang, W. Deng, and P. Moulin. Simultaneous feature and dictionary learning for image set based face recognition. In *European Conference on Computer Vision*, pages 265–280. Springer, 2014. 2

[15] T. Mensink, J. Verbeek, F. Perronnin, and G. Csurka. Distance-based image classification: Generalizing to new classes at near-zero cost. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2624–2637, 2013. 2

[16] H. Oh Song, Y. Xiang, S. Jegelka, and S. Savarese. Deep metric learning via lifted structured feature embedding. In

Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4004–4012, 2016. 2

[17] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *BMVC*, volume 1, page 6, 2015. 1, 3, 5, 7

[18] Q. Qian, R. Jin, S. Zhu, and Y. Lin. Fine-grained visual categorization via multi-stage metric learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3716–3724, 2015. 1

[19] O. Rippel, M. Paluri, P. Dollar, and L. Bourdev. Metric learning with adaptive density discrimination. *arXiv preprint arXiv:1511.05939*, 2015. 2, 4, 5

[20] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015. 3, 5

[21] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 806–813, 2014. 1

[22] J. Snoek, O. Rippel, K. Swersky, R. Kiros, N. Satish, N. Sundaram, M. M. A. Patwary, M. Prabhat, and R. P. Adams. Scalable bayesian optimization using deep neural networks. In *ICML*, pages 2171–2180, 2015. 1

[23] Y. Sun, X. Wang, and X. Tang. Deep learning face representation from predicting 10,000 classes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1891–1898, 2014. 3

[24] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 5

[25] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701–1708, 2014. 7

[26] Y. Tang. Deep learning using linear support vector machines. *arXiv preprint arXiv:1306.0239*, 2013. 3

[27] A. Vedaldi and K. Lenc. Matconvnet: Convolutional neural networks for matlab. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 689–692. ACM, 2015. 5

[28] R. Wang, S. Shan, X. Chen, and W. Gao. Manifold-manifold distance with application to face recognition based on image set. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. 2

[29] W. Wang, R. Wang, Z. Huang, S. Shan, and X. Chen. Discriminant analysis on riemannian manifold of gaussian distributions for face recognition with image sets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2048–2057, 2015. 2

[30] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10(Feb):207–244, 2009. 2

[31] Y. Wen, K. Zhang, Z. Li, and Y. Qiao. A discriminative feature learning approach for deep face recognition. In *European Conference on Computer Vision*, pages 499–515. Springer, 2016. 1, 2, 3, 4, 6

[32] L. Wolf, T. Hassner, and I. Maoz. Face recognition in unconstrained videos with matched background similarity. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 529–534. IEEE, 2011. 5