

Constellation Loss: Improving the efficiency of deep metric learning loss functions for optimal embedding.

Alfonso Medela
Tecnalia
Derio, Spain
alfonso.medela@tecnalia.com

Artzai Picon
Tecnalia
Derio, Spain
artzai.picon@tecnalia.com

Abstract

Metric learning has become an attractive field for research on the latest years. Loss functions like **contrastive loss**, **triplet loss** or **multi-class N-pair loss** have made possible generating models capable of tackling **complex scenarios** with the presence of many classes and scarcity on the number of images per class not only work to build classifiers, but to many other applications where measuring similarity is the key. Deep Neural Networks trained via metric learning also offer the possibility to solve **few-shot learning problems**. Currently used state of the art loss functions such as triplet and contrastive loss functions, still suffer from **slow convergence** due to the selection of effective training samples that has been partially solved by the multi-class N-pair loss by simultaneously adding additional samples from the different classes. In this work, we extend triplet and multiclass-N-pair loss function by proposing the *constellation loss* metric where the distances among all class **combinations are simultaneously learned**. We have compared our *constellation loss* for visual class embedding showing that our loss function over-performs the other methods by obtaining more compact clusters while achieving better classification results.

1 Introduction

Distance metric learning approaches [1, 2, 3] work by learning embedding representations that keep close together for **similar data points** while maintaining them far for dissimilar data points. Among distance metric learning applications we can find **face recognition**, [4], signature verification [5], authorship verification [6], few-shot learning [7, 8] and visual similarity for product design [9] among others.

With the popularization of convolutional neural networks [10, 11], deep metric learning has been deeply analysed on the last years. Deep metric learning [7, 12, 13, 14, 6, 15] have proven to be effective at learning nonlinear embeddings of the data outperforming existing classical methods. Normally, specific network architectures are trained to minimize an euclidean based loss function where a nonlinear embedding representation is learned to bond together embeddings from similar classes while taking apart embeddings of different classes.

The definition of appropriate loss functions is crucial for fast convergence and optimal global minimum search [7] and they have received lot of attention on the last years. In this sense, losses such as contrastive loss function [16] focuses same-class or different-class pairs are normally used. Triplet loss function [2, 17] extended contrastive loss by considering a query sample and two additional samples (one positive and one negative). This triplet loss simultaneously enlarges the distances between the embeddings of the query and negative sample while reducing the distance between the positive and query samples. However, these methods suffer from slow convergence and poor local optima [15] as, at each update, embeddings are only optimized against one negative class. This was

partially solved by the incorporation of the multi-class N-pair loss [15] that generalizes triplet loss by simultaneously optimizing against N-1 negative samples from different classes instead of a single negative class yielding to better performance and faster convergence.

However, multi-class N-pair loss function is still ignoring the distances among the different negative classes among them and thus, not assuring optimization among the different negative class embeddings. In this work we extend multiclass-N-pair loss with the proposed *constellation loss* metric where the distances among all class combinations are simultaneously learned. Figure 1 graphically shows the interaction among the different class samples distances during a single gradient descent step for the analyzed losses.

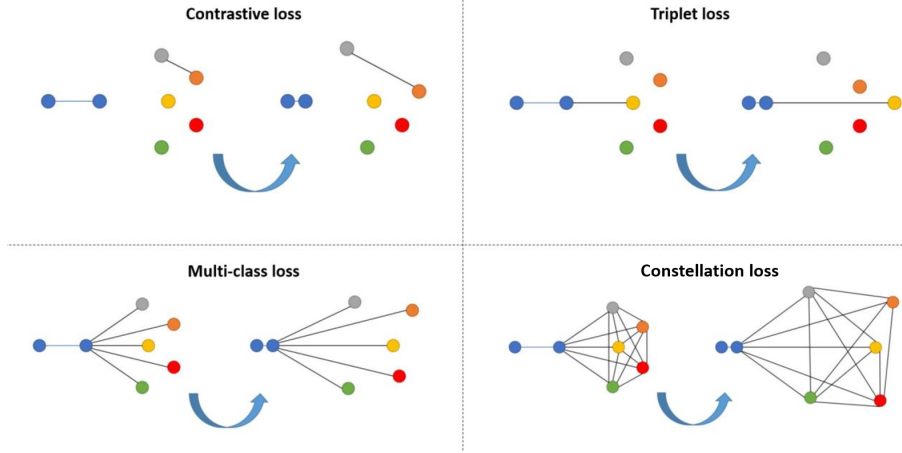


Figure 1: Visual representation of a gradient descent step for each of the compared losses: Contrastive loss [16], triplet loss [2], multi-class N-pair loss [15] and the proposed Constellation loss. Each color represent a different class

In experiment, we validate that *constellation loss* outperforms other metrics for class embedding tasks resulting into higher class classification performance and better cluster separability metrics such as Silhouette [18] and Davis-Boulding index [19]. We also propose an methodology that removes the need of using specific supporting architectures such as Siamese Neural Networks [7] for learning the embedding representations that can be learn by the use of smart batch selection dealing to the same functional cost while improving multi class scalability and reducing training memory needs.

2 Discriminative loss functions for distance metric learning

Traditionally, most of the image classification networks such as AlexNet [10], VGGNet [20], GoogleNet [21] or ResNet [17] adopted cross-entropy based softmax loss function to solve classical classification problems. However, discriminative metric learning loss functions have better generalization ability [22] and have received more attention for feature learning purposes in the latest years not only for verification problems [4, 2] but also for few-shot learning [7, 8] as they overcome learning capabilities of traditional classification approaches under small number of training images conditions.

This is achieved by learning an image embedding f_i from an image x_i . This embedding represents a class-representative vector, that is, a vector that contains the most important features associated to the corresponding class of the image. To this end, euclidean-distance-based loss functions are normally used. These functions conceptually constrains the learned embeddings to have distance 0 among elements of the same class and greater distances among elements from different classes.

To this end, euclidean-distance-based loss functions like contrastive loss [16] measure pairs of samples. This was extended by triplet loss [2] by comparing triplets with a positive and a negative samples. Multiclass-N-pair loss objective function [15] has focused on improving previous distance metric loss functions by generalizing triplet loss. First, it allows joint comparison among more

than one negative examples, concretely, N-1 negative examples and secondly, an efficient batch construction strategy was introduced to reduce computation. This loss function has demonstrated superiority over triplet loss as well as other metric learning functions. Finally, we propose the constellation loss where distances among all negative classes among them are taken into account simultaneously. The different losses are detailed below.

2.1 Contrastive loss

Contrastive loss (1) only focuses on positive or negative pairs. Positive pairs are composed by same-class images and negative ones by distinct-class pairs. Formally, the network transforms the pair of input images $\{x_{1,i}, x_{2,i}\}$ into $\{f_{1,i}, f_{2,i}\}$ embedding vectors. The labels are either $y_i = 0$ for positive pairs or $y_i = 1$ for negative pairs.

$$\mathcal{L}_c = \frac{1}{2N} \sum_{i=1}^N [(1 - y_i) \|f_{1,i} - f_{2,i}\|_2^2 + (y_i) \{ \max(0, m - \|f_{1,i} - f_{2,i}\|_2) \}^2] \quad (1)$$

where m is the margin, usually set to 1.0 and N is the batch size.

Intuitively, this loss penalizes when a positive pair is far away or a negative pair too close. Therefore, in an optimal case, positives are nearby 0.0 and negatives close to 1.0.

2.2 Triplet loss

Triplet loss (2) goes one step further by taking into account positive and negative pairs at the same time. This is done by setting an anchor, from which a distance will be calculated to a sample of the same class (positive) and a sample of a different class (negative). So, the set of input images is a triplet $\{x_i^a, x_i^p, x_i^n\}$ and their correspondent embedding vectors are $\{f_i^a, f_i^p, f_i^n\}$. No label is needed.

$$\mathcal{L}_{triplet} = \frac{1}{N} \sum_{i=1}^N \max(0, \|f_i^a - f_i^p\|_2^2 - \|f_i^a - f_i^n\|_2^2 + \alpha) \quad (2)$$

where α is a parameter to avoid convergence to trivial solutions and N is the batch size.

The aim of this loss is to maximize the distance between the anchor and the negative whilst minimizing the distance between the anchor and positive. Nonetheless, there is no gain when $\|f_i^a - f_i^p\|_2^2 < \|f_i^a - f_i^n\|_2^2 + \alpha$, and for that reason hard-triplet mining is commonly applied. This technique consist on taking into account only hard or semi-hard triplets, that is, using for computation only the triplets that give a positive loss. This way, it forces the network to try harder and it improves convergence.

2.3 Multiclass-N-pair loss objective

Multi-class-N-pair loss objective is a generalization of triplet loss, that incorporates at each optimization update the other negative classes that the triplet loss does not take into account. This allows joint comparison among more than one negative example at each update while reducing the computational burden of evaluating deep embedding vectors. So, when having N classes, the distances to the $N-1$ negative classes are also considered. When only one negative sample is used for calculation ($N=1$), this loss is comparable to triplet loss.

$$\mathcal{L}_{m-c} = \frac{1}{N} \sum_{i=1}^N \log(1 + \sum_{j \neq i} \exp(f_i^\top f_j^+ - f_i^\top f_i^+)) \quad (3)$$

2.4 Constellation loss

The constellation loss takes the best of both triplet and multiclass-N-pair loss. It uses the same batch construction than triplet loss and a similar loss formulation than multiclass-N-pair loss. The

hyperparameter K sets the number of triplets we want to incorporate in the formula, this way, taking into account more negative terms than the usual triplet loss. Even though increasing K parameter means a bigger computational effort, we prove for our dataset that at some point the fact of increasing K does not affect much the result. This is due to the randomness in the choice of each term, that can be composed of several distinct negative values. Therefore, there is no need for a high K value to improve triplet loss or multiclass-N-pair loss. The main difference is that multiclass-N-pair-loss subtracts dot products of same class pairs whereas constellation loss does something similar to triplet loss by subtracting a dot product of an anchor and negative embedding; and a dot product of an anchor and positive embedding.

$$\mathcal{L}_{constellation} = \frac{1}{N} \sum_{i=1}^N \log(1 + \sum_j^K \exp(f_i^{a\top} f_j^n - f_i^{a\top} f_i^p)) \quad (4)$$

3 Deep Neural Network for Embedding Learning

Deep embedding extraction can be performed by the use of a Siamese Neural Network architecture [7, 6] in a similar way as done in [8]. These network employ a twin network architecture where their weights are tied and joined by a loss function that is computed in the top. This network is normally selected from the state-of-the-art CNN architectures such as VGGNet [20], that was successfully used by [7, 8], Inception [23], ResNet[24]among others.

In our experiments we have chosen Inception V3 [23] classification architecture, where the last layer is replaced as suggested by [17] by a global average pooling layer and a 128-neuron layer as an embedding layer, f_i , that acts as a low-dimensional representation of the input images. The embedding layer has a sigmoid activation and it is L2 normalized in the case of triplet and constellation loss. L2 normalization makes the training process more stable for these loss functions. There is no need for L2 normalization when training with multiclass-N-pair objective loss as it already incorporates in the loss. After loss optimization procedure, the trained base network is able to extract a embedding f_i from an image x_i that is able to keep similar samples together and dissimilar one apart.

3.1 Smart batch

However, this siamese neural network approach causes memory and scalability problems in more advanced loss functions where triplets or K-plets are the inputs of the network, specially for large values of K . Instead of building a siamese structure we follow an *smart batch* strategy that leads to an equivalent loss formulation which is more optimal in terms of speed, network memory and scalability.

Online triplet mining loss presented in [17] where the authors take the embeddings array, computes all the dot products and euclidean distances, and then selects the hard and semi-hard triplets to compute the loss function. Hard-triplets are where the negative is closer to the anchor than the positive and semi-hard triplets are triplets where the negative is not closer to the anchor than the positive, but which still have positive loss.

N-pair-mc uses a different batch construction in which two embedding arrays are taken as input. Each array contains one embedding per class and are ordered same way in both arrays. Figure 2 depicts online batch construction of triplet and N-pair-mc pairs.

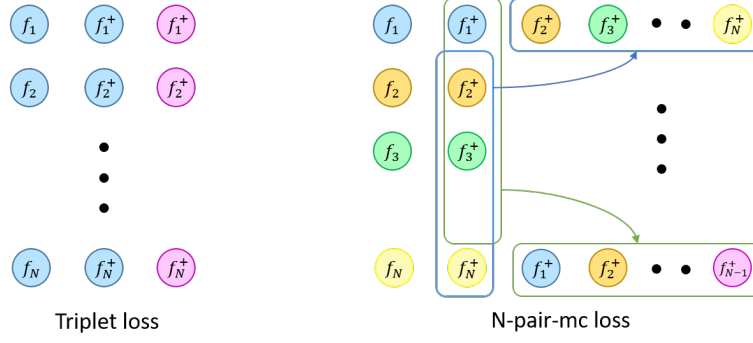


Figure 2: Batch construction of triplet loss and N-pair-mc-loss.

In the case of constellation loss, we extend the strategy as in triplet loss but dividing the batch into K groups (see Fig.3). We later show that even though we use same batch, our loss obtains better results. The nature of constellation loss that takes into account the relationship among all samples in the batch simultaneously constellation loss doesn't need to apply a mask to select hard and semi-hard triplets and neither to compute euclidean distances. It only needs to compute the dot products as triplet loss function does, and apply a mask to find the dot products between same class embeddings, anchor-positive, and the dot products between distinct class embeddings, anchor-negative.

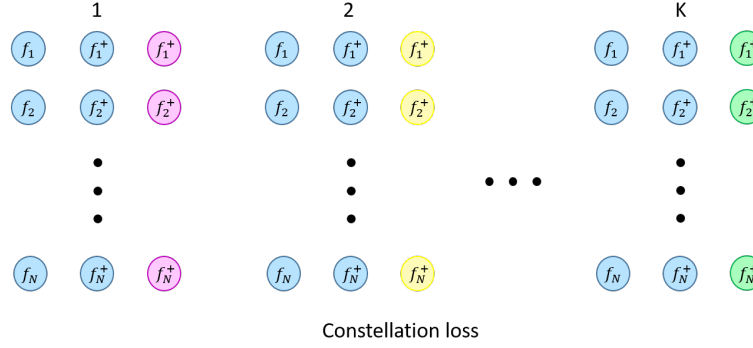


Figure 3: Batch construction of constellation loss, based on triplet loss but with K splits.

3.2 Validation

The proposed backbone network (Inception v3) is trained by means of any of previous loss functions in order to learn the image embeddings f_i . This embedding vector is a low-dimensional representation of X_i that minimizes its correspondent loss function and thus, are designed to estimate the distance among classes. In order to validate the suitability of these embeddings we analyze, the quality of the class clusters generated by the embedding vector and also the performance of a classification task that can be achieved by the learned embeddings. This validation procedure is illustrated in figure 4

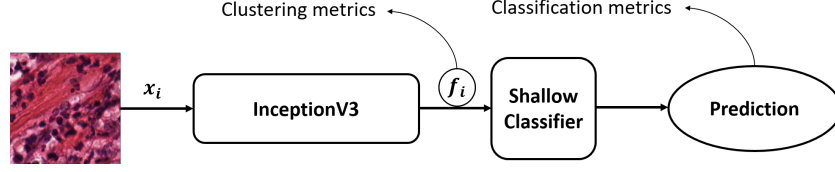


Figure 4: Illustration of the validation procedure for the extracted embeddings using a shallow classifier.

3.3 Clustering metrics

We analyse the quality of the class clusters that are generated by the embedding vectors sets to measure the quality of the generated clusters. As the main goal of the network is to create better representations these metrics show how well are the test embeddings grouped in clusters. Two specific metrics were selected:

Davis-Boulding index[19] is a metric that evaluates clustering quality. The smaller the value, the better the result. The index uses quantities and features inherent to the dataset and its drawback is that a good value does not imply the best information retrieval.

Silhouette score[18] is a measure of how similar an object is to its own cluster compared to other clusters. The value of this metric ranges from -1 to 1 and the closest to 1 the better the result.

3.4 Classification metrics

We also evaluate the classification accuracy that machine learning models can obtain with the learned embeddings. To this end, We select k -nearest neighbours as the simpler shallow classifier to predict the class associated to each embedding. First, images go through the network and it outputs the embeddings of all the images, both training and test sets. Then, a k -nearest neighbours classifier is used to predict the classes of the test embeddings allowing to measure the obtained accuracy and balanced accuracy metrics.

4 Experimental Results

We assess the performance of the proposed *constellation loss* for visual class embedding extraction. We validate the capability of the tested loss functions to extract appropriate embeddings for the entrusted visual task of histology image classification.

4.1 Dataset

Public dataset [25] from the University Medical Center Mannheim (Germany). Contains tissue samples obtained from low-grade and high-grade primary tumours of digitalized *colorectal cancer* tissue slides. The database is divided into eight different types of textures that are present on the tumours samples: 1. tumour epithelium, 2. simple stroma, 3. complex stroma, 4. immune cells, 5. debris and mucus, 6. mucosal glands, 7. adipose tissue and 8. background, as depicted in Fig. 5. There are 625 image samples per class, producing a total dataset of 5000 image tiles of dimension 150 px x 150 px ($74 \mu\text{m} \times 74 \mu\text{m}$).

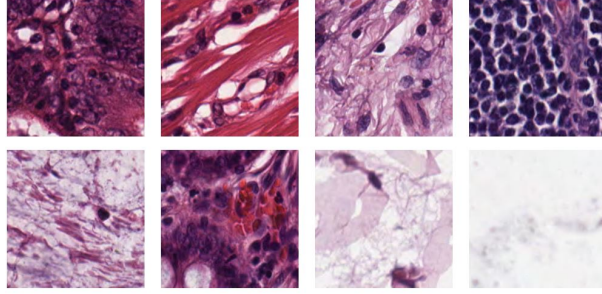


Figure 5: Sample images from the dataset. First row: Tumour epithelium, stroma, complex, immune cells. Second row: debris, mucosa, adipose and empty tile samples are depicted.

4.2 Visual classes embedding extraction

Our main goal is to evaluate the capabilities of the different loss functions to embed image description. To validate this, we train the deep metric learning architecture detailed in section 3 for the different losses to obtain a network capable of extracting the embedding vector, f_i , from an input signal.

4.2.1 Training

All the experiments were run for 10 epochs and optimized with Adam with its default parameters. A very simple generator was used in order to feed the images to the network. It makes simple spatial transformations such as inversion in both axes and rotations of multiples of 90 degrees to slightly augment the data. The size of the image is not modified during this process. No more augmentation was considered as the aim of the loss function we are working with is to train on datasets that have not enough images per class.

As a baseline we trained a classical classification Inception v3 network approach by learning with a softmax loss function that obtained an accuracy of $92.01 \pm 0.99\%$.

A Inception v3 architecture is trained over the different loss functions following the online smart batch selection as detailed in section refsec::smartbatch.

We ran each network 10 times, each time training and testing in different data splits. We calculated the metrics for each split and gathered them in Table.1 by averaging over the splits. This way, we show the mean value and the standard deviation. We adopted a K-fold cross-validation strategy to validate the constellation loss against the triplet and multiclass-N-pair loss. We divided the data in 10 folds and obtained the metrics for each fold. Then we averaged the results and gathered them in Table.1. The folds are exactly the same for every loss function, with this we mean that they have seen the same train/test split.

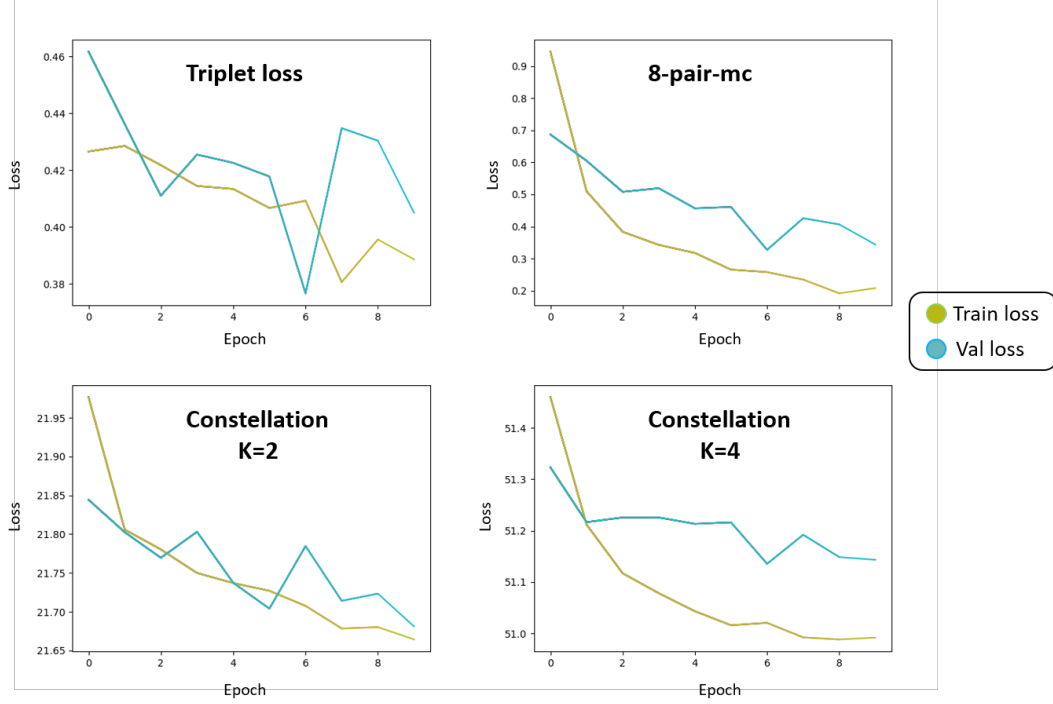


Figure 6: Train and validation loss during training. The values of the loss functions are not comparable, the only purpose of the plots is to analyze training loss and validation loss stability and convergence.

We observed that triplet loss showed a very unstable evolution during training, whereas the logarithmic loss functions such as multiclass and constellation’s convergence was more optimal. They both showed a clear downward direction in train and validation loss alike. Regarding K value, we see that $K = 4$ shows signs of overfitting in contrast to $K = 2$. However, $K = 4$ gives better results in clustering metrics and classification accuracy.

4.2.2 Results

The proposed constellation loss beats both triplet and multiclass-N-pair loss in all the metrics. Triplet and 8-pair-mc give Davis-Boulding index values over 0.6, on the other hand, constellation loss gets close to 0.4 for $K = 4$. The same happens when we evaluate the loss functions with Silhouettes score, obtaining very similar results for triplet and 8-pair-mc and significantly better for constellation loss with ant K . This can also be see in Fig.7, which shows a 2 dimensional representation of the embedding vectors of the test set. Each color represents a single class. The embedding vector dimension is reduced by the popular t-SNE [26] technique, a dimensionality reduction technique that is particularly well suited for the visualization of high-dimensional data. Even if the plots are just for visualization purposes and real data corresponds to a 128-dimensional space, we can notice that the most compact clusters are obtained with constellation loss.

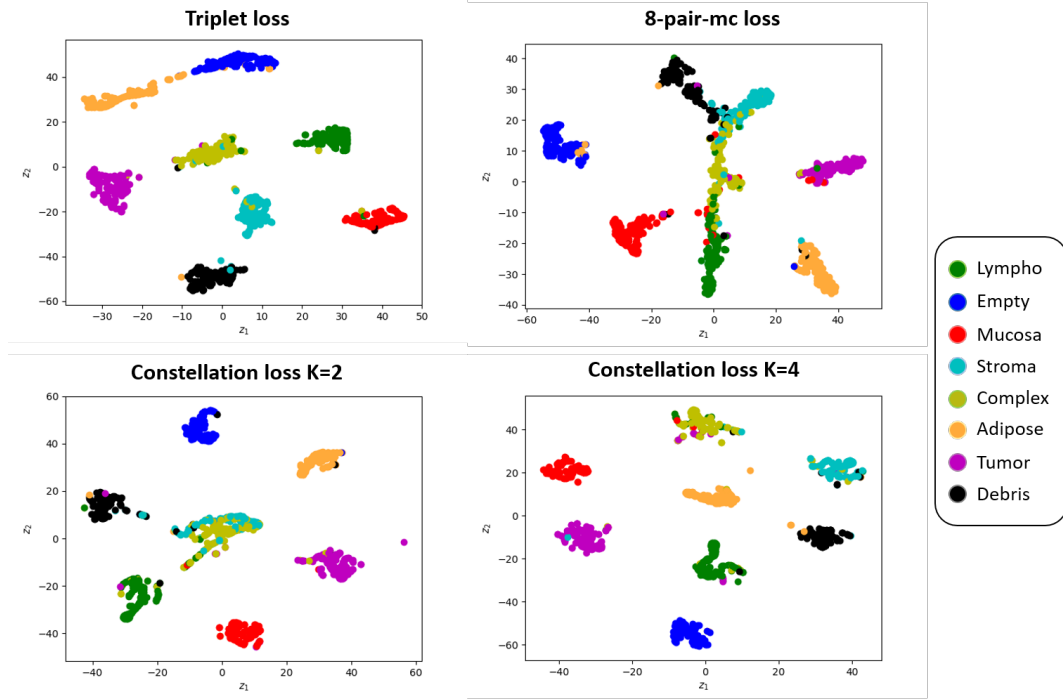


Figure 7: Comparison of the clustering capabilities. The figure shows a 2D t-SNE visualization of the embedding vectors in the test set.

So we can say that the best clusters are obtained with constellation loss for any K , based on the clustering metrics. Furthermore, constellation loss shows higher accuracy than triplet and multiclass-N-pair loss for any K . In addition, the result we obtain is better than with a classical softmax approach.

Table 1: Mean and standard deviation of the accuracy, BAC, Davis-Bouldin and Silhouette metrics for each experiment. Mean and standard deviation are calculated over the results in each fold.

Loss	Metric			
	Accuracy	BAC	Davis-Bouldin	Silhouette
Triplet	91.74 ± 0.68	92.86 ± 0.72	0.6384 ± 0.0773	0.6265 ± 0.0245
8-pair-mc	91.48 ± 0.92	91.53 ± 0.90	0.6037 ± 0.0386	0.5996 ± 0.0185
Constellation k=2	92.20 ± 0.65	92.28 ± 0.55	0.5535 ± 0.1419	0.7327 ± 0.0478
Constellation k=3	92.83 ± 0.73	92.86 ± 0.79	0.4558 ± 0.0518	0.7709 ± 0.0205
Constellation k=4	92.75 ± 0.39	92.81 ± 0.44	0.4218 ± 0.0589	0.7864 ± 0.0212
Constellation k=5	92.78 ± 0.68	92.86 ± 0.69	0.4605 ± 0.0778	0.7817 ± 0.0178
Constellation k=6	92.96 ± 0.51	93.01 ± 0.56	0.4373 ± 0.0511	0.7905 ± 0.0134
Constellation k=7	92.79 ± 0.68	92.86 ± 0.65	0.4334 ± 0.0546	0.7820 ± 0.0161

4.3 Computing infrastructure

The experiments were ran on a Gigabyte GeForce GTX Titan X 12GB GDDR5 GPU. The GPU is installed in a local server that we access by an SSH client. We used an Anaconda distribution and the main libraries we used are Keras, Tensorflow and Scikit-Learn. Keras was used for the main architecture and Tensorflow for the loss functions. Scikit-Learn was very helpful for machine learning models like k -nearest neighbours, and also for metrics BAC, David-Bouldin index and Silhouette.

5 Conclusions

In this work we have compared the performance of different metric learning losses for extracting discriminative embeddings under deep metric learning approach. Our proposed the *constellation loss* metric takes into account the distances among all class combinations are simultaneously learned. The extracted embeddings have been validated for image classification task over the selected dataset, showing that our loss function over-performs the other methods by obtaining more optimal clusters. This better representation of the classes allows machine learning models such as k -nearest neighbours achieve better results in classification.

Moreover, we showed that constellation loss its more stable than triplet during training and at the same time, surpasses multiclass-N-pair loss in classification accuracy by using a similar mathematical formulation.

Source code

Source code is available at: https://git.code.tecnalia.com/comvis_public/piccolo/constellation_loss/

Acknowledgments

This study has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No. 732111 (PICCOLO project <https://www.piccolo-project.eu/>)

References

- [1] Eric P Xing, Michael I Jordan, Stuart J Russell, and Andrew Y Ng. Distance metric learning with application to clustering with side-information. In *Advances in neural information processing systems*, pages 521–528, 2003.
- [2] Kilian Q Weinberger, John Blitzer, and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in neural information processing systems*, pages 1473–1480, 2006.
- [3] Jason V Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S Dhillon. Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning*, pages 209–216. ACM, 2007.
- [4] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 1, 2017.
- [5] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a "siamese" time delay neural network. In *Advances in neural information processing systems*, pages 737–744, 1994.
- [6] William Du, Michael Fang, and Margaret Shen. Siamese convolutional neural networks for authorship verification. Technical report, Tech. Rep., 2017.[Online]. Available: <http://cs231n.stanford.edu/reports...>, 2017.
- [7] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, volume 2, 2015.
- [8] Alfonso Medela, Artzai Picon, Cristina L. Saratxaga, Oihana Belar, Virginia Cabezon, Riccardo Cicchi, Roberto Bilbao, and Glover Ben. Few shot learning in histopathological images: Reducing the need of labeled data on biological datasets. In *IEEE International Symposium on Biomedical Imaging*, 2019.
- [9] Sean Bell and Kavita Bala. Learning visual similarity for product design with convolutional neural networks. *ACM Transactions on Graphics (TOG)*, 34(4):98, 2015.
- [10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [11] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

- [12] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition*, pages 84–92. Springer, 2015.
- [13] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li. Deep metric learning for person re-identification. In *2014 22nd International Conference on Pattern Recognition*, pages 34–39. IEEE, 2014.
- [14] Junlin Hu, Jiwen Lu, and Yap-Peng Tan. Discriminative deep metric learning for face verification in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1875–1882, 2014.
- [15] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in Neural Information Processing Systems*, pages 1857–1865, 2016.
- [16] Sumit Chopra, Raia Hadsell, Yann LeCun, et al. Learning a similarity metric discriminatively, with application to face verification. In *CVPR (1)*, pages 539–546, 2005.
- [17] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. *CoRR*, abs/1503.03832, 2015.
- [18] Peter Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. 1987.
- [19] David L Davies and Donald W Bouldin. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, (2):224–227, 1979.
- [20] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.
- [21] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.
- [22] Gong Cheng, Ceyuan Yang, Xiwen Yao, Lei Guo, and Junwei Han. When deep learning meets metric learning: Remote sensing image scene classification via learning discriminative cnns. *IEEE transactions on geoscience and remote sensing*, 56(5):2811–2821, 2018.
- [23] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015.
- [24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [25] Jakob Nikolas Kather, Cleo-Aron Weis, Francesco Bianconi, Susanne M Melchers, Lothar R Schad, Timo Gaiser, Alexander Marx, and Frank Gerrit Zöllner. Multi-class texture analysis in colorectal cancer histology. *Scientific reports*, 6:27988, 2016.
- [26] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.