Guided Representation Learning for the Classification of Hematopoietic Cells

Philipp Gräbel
Institute of Imaging and Computer Vision
RWTH Aachen, Germany
graebel@lfb.rwth-aachen.de

Martina Crysandt
Department of Hematology
UKA Aachen, Germany
mcrysandt@ukaachen.de

Barbara M. Klinkhammer
Institute of Pathology
UKA Aachen, Germany
bklinkhammer@ukaachen.de

Peter Boor
Institute of Pathology
UKA Aachen, Germany
pboor@ukaachen.de

Tim H. Brümmendorf
Dept. of Hematology
UKA Aachen, Germany
tbruemmendorf@ukaachen.de

Dorit Merhof
Institute of Imaging and Computer Vision
RWTH Aachen, Germany
merhof@lfb.rwth-aachen.de

Abstract

Cell classification in human bone marrow microscopy images is a challenging image analysis task due to the number and inter-connection of cell types. While machine learning techniques have vastly higher throughput and could thus be more reliable, humans are intrinsically capable of understanding relations between cell types. In this paper, we propose methods to incorporate such intrinsic model knowledge based on representation learning. To this end, we construct a manually defined, two-dimensional reference embedding, coined embedding guide*, which we use together with inverse dimensionality reduction, a distance-based loss and a growing embedding technique. Results show improved classification scores as well as a visually interpretable and clearly defined embedding space.*

1. Introduction

Automated classification of hematopoietic cells in bone marrow microscopy images is a challenging task. This is due to a large number of visually similar cell types, which also occur in varying stages of maturity. A first approach on high resolution whole slide images was presented in [3] with good results on a subset of cell types. However, a wider range of classes needs to be considered in a clinical setting. Accurate prediction of the distribution of various cell types in a given bone marrow sample would then enable a much faster, more reliable and objective diagnosis.

Since cell types across different cell lineages and maturity stages are biologically and visually related, we not only investigate classification networks but also focus on representation learning [1] techniques: the high-dimensional feature space should ideally show the relation between different classes and preferably also take their dependencies into account during training. An embedding feature space would further allow more detailed analyses, such as better interpretability of results by visualization and confidence estimation (as for example proposed in [8]).

Established representation learning techniques for extracting such embeddings often focus on selecting appropriate triplets of data (mining) and then minimizing an distance-based loss (e.g. with triplet loss as described in [9]). Recently, a class-centered triplet loss [5] was proposed that uses the means of all representation vectors of a single class as anchor points instead of a single sample representation. Apart from the separability of classes, however, these techniques provide no suitable means of control over the embedding space.

We propose to leverage expert knowledge by providing a manually defined *embedding guide* to steer the network towards a sensible embedding. An embedding guide is a simple two-dimensional representation with one 2D point per class, making it easy to define. We further propose two methods to utilize the embedding guide in higher dimensions, which are required in most representation learning scenarios. Lastly, we propose to use a growing embedding.

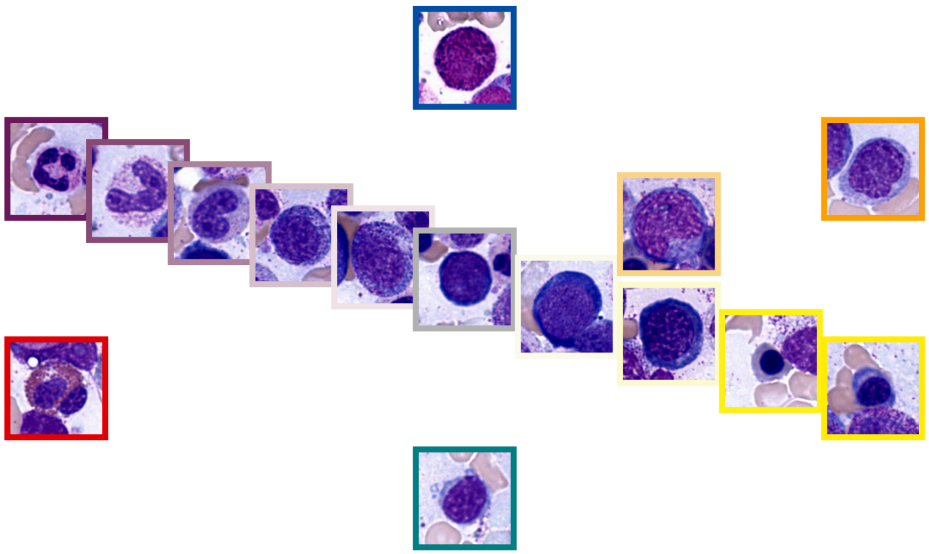


Figure 1. Image samples for each cell type (15 classes). The cells are arranged in the same way as the proposed embedding guide in Figure 2 and the predicted embeddings in Figure 4. The color-coding follows the same notation with a blast in the center (gray) and different lineages in different angles.

2. Image Data

Our dataset contains high resolution whole slide microscopy images of human bone marrow samples. Each whole slide is digitized with $63\times$ magnification and automated immersion oiling. The samples are stained using the standardized Pappenheim [2] staining procedure, which is commonly used by expert hematologists to make cells better distinguishable. From each whole slide image, representative and appropriate regions are selected for annotation by expert hematologist similar to clinical diagnostic. The regions and slides were selected to capture high visual variance (e.g. in terms of staining variability) between them instead of focusing on easy-to-process regions.

As this work focuses on individual cells, we automatically extract patches of size $244 \times 244\,\text{px}$ around each cell. While this yields only a single cell in the center of each patch, adjacent cells might appear at the border of the batch. Each cell patch is manually labelled by expert hematologists with the corresponding cell type. In this work, we distinguish between the following 15 different cell types: proerythroblast and three types of erythroblast (basophilic, orthochromatic, polychromatic), promonocyte and monocyte, basophilic and eosinophilic granulocytes, five types of neutrophilic granulocytes (promyelocyte, myelocyte, metamyelocyte, band and segmented granulocyte) as well as blast and lymphocyte. As the cell distribution varies between those cell types, class-imbalance can be observed in the dataset with a factor of $30 : 1$ in the most extreme case of neutrophilic segmented granulocytes compared to proerythroblasts. In total, the dataset contains 7020 patches from 13 regions of six patients. Figure 1 shows one example per cell arranged in sensible pattern, which is required for the proposed methods.

3. Methods

3.1. 2D Embedding Guide and 2D Loss

Based on expert knowledge, we construct a reference by assigning a 2D embedding to each cell type. As it needs to incorporate human expert knowledge and is therefore manually defined, higher dimensions are infeasible. We propose the biologically inspired reference presented in Figure 2, further referred to as *embedding guide* e_{2D}.

If only a two dimensional embedding is to be learnt, this guide can be employed directly in a loss function $\mathcal{L}_{\text{guide2D}}$ using the distance (e.g. L_1-distance) between predicted embedding and embedding guide for the ground truth cell

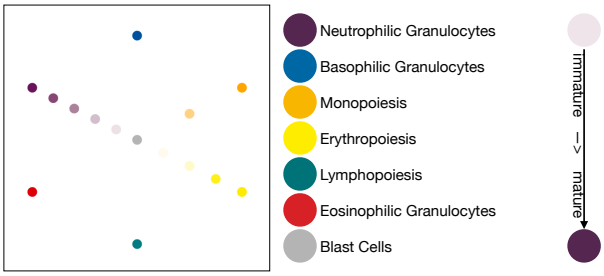


Figure 2. Our proposed embedding guide. Cell types are radially distributed with related cell types (such as basophilic, eosinophilic and neutrophilic granulocytes) being located next to each other. Along the axis defined by each cell type, immature cells (such as blasts) are in the center and the most mature cells further outward.

types. However, it is often necessary to use considerably larger embeddings to capture the required complexity. One straight-forward approach is applying $\mathcal{L}_{\text{guide2D}}$ only on the first two elements of the embedding vector, ignoring the other elements. Since training commonly uses a norm-minimization for the entire embedding vector, the additional components should be optimized towards zero while still being capable of capture some amount of information.

As the embedding guide is only defined once, prior to training, no actual interaction between human and machine-learning algorithm takes place. The approach consequently falls in between the categories of automatic and interactive machine learning [4].

3.2. nD Guide Loss using Inverse UMAP

Dimensionality reduction methods such as UMAP [6] or t-SNE [10] create sensible lower-dimensional embeddings that focus on maintaining the distance between training samples. UMAP has the additional advantage of providing an inverse transform to map from lower-dimensional embedding space to the higher-dimensional embedding space. While this is not a well-defined transform, learning from training samples makes an adequate inverse transform possible.

In this work, we leverage the inverse UMAP transform to convert the 2D embedding guide e_{2D} into a higher-dimensional embedding e_{nD}. After every training epoch, we initialize the UMAP fitter with the e_{2D}, fit it to the predicted training data embeddings and ground truth labels (supervised UMAP) and apply the inverse transform to e_{2D} to obtain e_{nD}. As this can easily result in overfitting, we propose using a temporally low-pass filtered version with empirically determined weights.

$$e_{\text{nD,epoch}} = 0.6 \cdot e_{\text{nD,epoch}-1} + 0.4 \cdot \text{UMAP}^{-1}(e_{\text{2D}}) \quad (1)$$

This is initialized with a zero-padded 2D embedding guide in the first epoch. For the UMAP guide loss $\mathcal{L}_{\text{guideUMAP}}$, this n-dimensional embedding guide is compared to the predicted embeddings using the L_1 distance.

3.3. Growing Embedding

To further stabilize the training in higher dimensions, we propose to use a growing embedding, where the embedding length is $n = \min(\max(\text{epoch}, 2), n_{\text{max}}))$. In this way, the first three epochs use $\mathcal{L}_{\text{guide2D}}$ directly and then continuously grow towards the desired embedding length n_{max} with a loss working on higher dimensions. A slowly growing embedding initialized with the original 2D embedding guide can further increase the training stability of nD-guided training.

3.4. Distance Loss

As a more stable but less direct training loss for incorporating the 2D guide without altering the dimensionality of the guide, we propose to keep the distance between two points in the guide constant – independent of dimension. A higher-dimensional embedding with the same distances between all point pairs should result in an embedding with similar characteristics as the 2D guide in terms of relative positions of embedding guide points. This can be exploited to define the following distance loss, which is defined as the average difference of distances between all point pairs of the prediction and the corresponding embeddings from the 2D guide.

$$\mathcal{L}_{\text{dist}} = \frac{2}{b(b-1)} \sum_{i=0}^{b-1} \sum_{j=0}^{i-1} L_1\Big(L_1\big(p_i, p_j\big), L_1\big(e_{\text{2D,i}}, e_{\text{2D,j}}\big)\Big), \quad (2)$$

where $e_{\text{2D,i}}$ denotes the embedding from the 2D guide corresponding to the ground truth cell type of the i-th sample in the batch, b refers to the batch size and p_i denotes the i-th prediction.

4. Evaluation

4.1. Experimental Setup

We utilize a DenseNet-121 from PyTorch [7] as feature extractor and train using the Adam optimizer with default parameters. All methods that create triplets employ triplet-based hard sample mining.

We perform evaluation in five-fold cross-validation. Folds are generated such that cells from a similar region are in the same fold and that the general distribution of cell types is similar between folds. $12.5\,\%$ of the training data are used for validation after each epoch. The epoch with the highest F1-score on the validation set is kept for the final evaluation. We further apply early stopping based on the validation score (stop if there was no improvement in the last 256 epochs).

Classification results for validation and evaluation are obtained using either a linear SVM, a kNN or a multi-layer perceptron (MLP). The type of classifier hyper-parameter values is chosen by grid search on the training data. The hyper-parameters are the regularization parameter C for SVM, the number of neighbors and the type of weights for kNN as well as the number of neurons and hidden layers for MLP.

Each experiment is carried out with embedding lengths 64, 256, and 1024. If sensible, we further train a 2D embedding. In the case of embedding length 1024 with growing embedding, the embedding grows with a factor of 4 (instead of 1) per epoch.

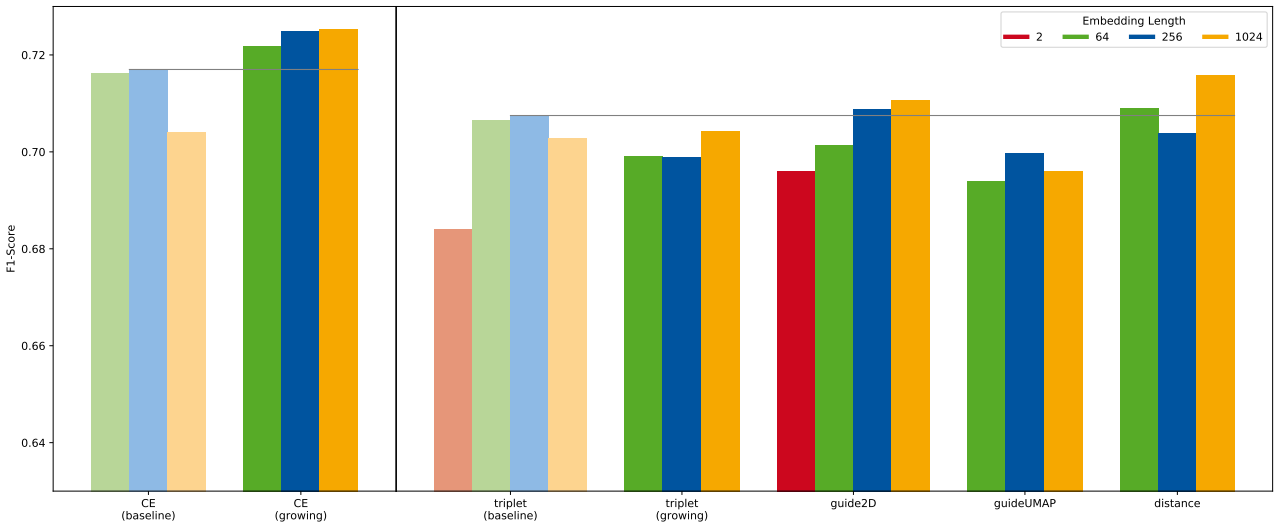


Figure 3. Classification results as bar plot for the classification comparison (left) and the representation learning comparison (right). Colors denote embedding length (red = 2, green = 64, blue = 256, orange = 1024). The gray lines indicate the best results from the corresponding baseline experiment.

4.1.1 Classification

To evaluate the classification performance, we compare a network trained with cross-entropy (CE) loss (baseline) against the same network with growing embeddings (ours). The latter network utilizes the two-dimensional embedding guide in the first three epochs. In both cases, the cross-entropy loss is applied on an additional fully connected layer after the embedding layer.

4.1.2 Representation Learning

To evaluate the representation learning techniques, we analyze the embedding space in addition to the classification scores. As a baseline, we train a network using triplet loss and compare it against our proposed methods: Triplet loss with growing embeddings, $\mathcal{L}_{\text{guide2D}}$, $\mathcal{L}_{\text{guideUMAP}}$ with growing embeddings and $\mathcal{L}_{\text{dist}}$ with growing embeddings. Each loss additionally minimizes the norm of the embedding vector with a factor of 0.001, which is particularly important in the guide2D case.

4.2. Results

Table 1 and Figure 3 show the results of each experiment for different embedding lengths as macro-averaged F1 scores.

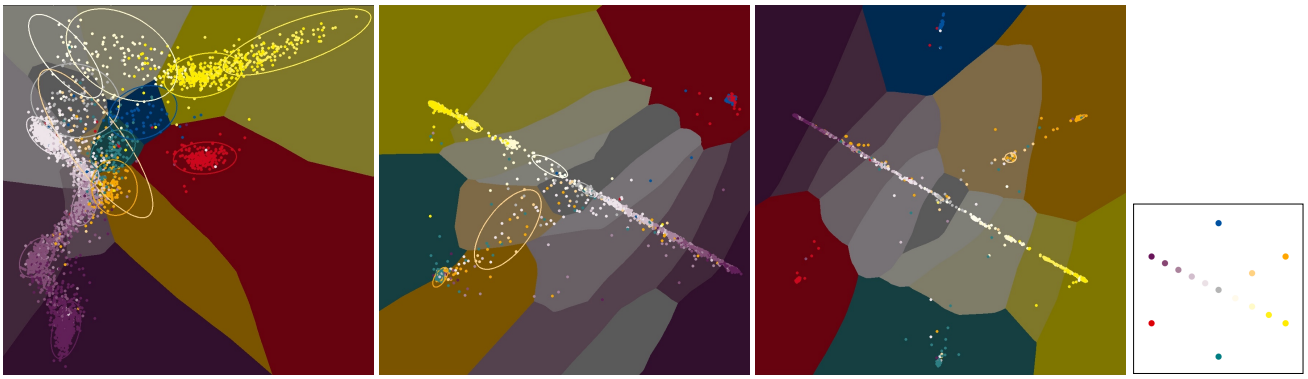
4.2.1 Classification

In terms of classification performance, the best performing network trained with the cross-entropy loss (embedding length 256) achieves a macro-average F1-Score of 0.717. There is not much difference between different embedding lengths, though both extreme cases of embedding lengths 2 and 1024 yield worse results. Particularly using a two-dimensional embedding results in much worse results (cf. Table 1). With growing embeddings, which initialize the network in the first three epochs with the embedding guide, results are improved by up to 0.021.

4.2.2 Representation Learning

Figure 4 shows the resulting embedding for the most important representation learning methods in the two-dimensional case. All representation learning techniques utilize the embedding space more efficiently compared to training with cross-entropy loss (not shown). For some classes (particularly the neutrophilic granulocytes) the biological progression can be observed in embedding space. In the case of triplet loss, this progression follows a zig-zag pattern, while the distance loss and the guide loss enforce a straight line. The distance loss and guided embedding follow the desired guide in principle, although the distance loss only encodes relative positions, not absolute positions as the embedding guide loss. This results in differences between repeated experiments: while the guided loss enforces the same encoding every time, the distance loss might yield a rotated version. Embeddings from triplet loss (and even more so with cross-entropy) show much larger differences. Furthermore, the variance between sample embeddings of the same class (indicated by the ellipsis) is drastically smaller with embedding guides. This is particularly noticeable for classes with a lower amount of data (e.g., cells of the erythropoiesis, monopoiesis, and lymphopoiesis).

In terms of classification accuracy, the results vary



(a) triplet (b) dist (c) guide2D (d) guide

Figure 4. Sample embeddings from one test set of three different 2D, non-growing methods. Each figure shows decision boundaries (background color), predicted embeddings per sample(points) and corresponding class confidence ellipses. The colors correspond to the classes (see (e) and Figure 2). Note the learnt relation between different maturity stages of neutrophilic granulocytes (white to purple).

per method. Simply using growing embeddings together with the triplet loss, actually reduces the classification result compared to just using triplet loss (baseline). The guided learning technique using UMAP also yields less suitable results in all embedding lengths. Applying the two-dimensional embedding guide loss directly yields better results and is, for larger embeddings (lengths 256 and 1024), slightly more accurate than the baseline. Surprisingly, even an embedding length of only 2 yields acceptable results with guided learning and has much higher scores than two-dimensional embeddings with triplet or cross-entropy loss. For two out of three embedding lengths (64 and 1024), the distance loss yields better results than the baseline with an improvement of up to 0.013.

5. Discussion

The results support two findings regarding guided learning. Firstly, using a growing embedding with an embedding guide result in an improved classification performance using the cross-entropy loss. Secondly, embedding guides yield more defined embedding spaces that are easily interpretable, are stable across repeated training and have a lower inter-class variance.

The success of growing embeddings, even when not applied to representation learning, can be explained by a more suitable initialization with the embedding guide. This can also be interpreted as manually providing domain knowledge to the initial training phase. For the presented guided learning techniques, the growing embedding is an essential part to stabilize the learning process, particularly for the UMAP method. However, it did not improve the triplet loss.

In general, representation learning techniques which do not simultaneously train classifier and feature (embedding) extractor network, yield slightly worse classification scores compared to cross-entropy loss on an additional, simultaneously trained fully connected layer. Both guided learning without UMAP and distance loss surpass the triplet loss, which serves as a baseline, for most embedding lengths. Guided learning with UMAP has slightly worse results, which might be caused by overfitting due to the learnt dimensionality reduction. The superiority of the simple guide2D loss might lie in its simplicity: by focusing directly on the embedding guide without any transformation, no overfitting can happen. The minimization of the embedding vector norm simultaneously keeps the other vector elements small. This approach also yields the most advantageous embedding, with clearly defined positions for cell clusters and lower intra-class variance. This is most noticeable with under-represented classes, which have a very high intra-class variance with overlapping clusters even for non-related cell types. When using an embedding guide, this variance is strongly reduced leading to a better separability for under-represented classes. It further should be noted that for guided learning methods, a shorter training (earlier early stop) could be observed (the best models was found after 149–222 epochs for triplet loss, 60–222 for guide2D, 60–142 for guideUMAP and 131–207 for the distance loss).

Depending on the application, using just two-dimensional embeddings might be beneficial: they are capable of acceptable score while being directly visually interpretable. Longer embeddings require dimensionality reduction, preferably with UMAP initialized in the same way as described in Section 3.2 to obtain an embedding space as defined by the 2D embedding guide. In the case of $\mathcal{L}_{\text{guide2D}}$, embeddings can simply be truncated.

While not shown in the experiments, the performance of the UMAP-based guided learning method without growing methods was also evaluated but yielded expectedly insufficient results due to overfitting of the dimensionality re-

	2	64	256	1024
CE (baseline)	0.597 ± 0.012	0.716 ± 0.016	0.717 ± 0.026	0.704 ± 0.013
CE (growing)		0.722 ± 0.017	0.725 ± 0.017	0.725 ± 0.018
triplet (baseline)	0.684 ± 0.017	0.706 ± 0.013	0.707 ± 0.013	0.703 ± 0.012
triplet (growing)		0.699 ± 0.018	0.699 ± 0.028	0.704 ± 0.012
guide2D	0.696 ± 0.013	0.701 ± 0.014	0.709 ± 0.017	0.711 ± 0.013
guideUMAP		0.694 ± 0.030	0.700 ± 0.018	0.696 ± 0.016
distance		0.709 ± 0.014	0.704 ± 0.016	0.716 ± 0.007

Table 1. Classification performance in terms of F1-Score over all cross-validation test sets and the corresponding standard deviation. Each row denotes a different method, the columns correspond to different embedding lengths.

duction method. We further evaluated the Centered-class Triplet Loss (CTL), which yielded slightly worse results than normal triplet loss.

An advantage of most of the presented methods is a clearly defined embedding space, which opens new possibilities with respect to visual interpretation and confidence estimation. This, of course, needs to be investigated in more detail. Furthermore, this embedding allows more detailed predictions, for example of maturity stages in between the ordinal classes. Additional research should also cover different distance measures (e.g., MSE instead of L1), more complex training curricula for growing embeddings and different embedding guides. Finally, the capability of this method for solving other problems should be investigated. While it should be applicable to any classification problem with a complex set of classes, it needs to be evaluated for each use-case with a dedicated embedding guide.

6. Conclusion

In this work, we presented a novel method to incorporate an *embedding guide* into the training loss for cell classification training. This guided learning procedure outperforms classical representation learning algorithms. Additionally, the learnt representation in the embedding space is visually interpretable. Furthermore, even higher scores can be achieved by initializing standard classification networks with a growing embedding learnt from the two-dimensional embedding guide. These advantages and the straight-forward application, make *Guided Learning* techniques useful for tasks with complex class structures, as is the case in hematopoietic cell classification.

6.0.1 Acknowledgements

This work is a purely retrospective, pseudonymized analysis of bone marrow samples under the Helsinki Declaration of 1975/2000 with written informed consent of all patients. This work was supported by the German Research Foundation (DFG) through the grants SFB/TRR57, SFB/TRR219, BO3755/6-1. The authors would like to thank Reinhild Herwartz and Melanie Baumann for their efforts in sample preparation and annotation.

References

[1] Yoshua Bengio, Aaron C. Courville, and Pascal Vincent. Unsupervised feature learning and deep learning: A review and new perspectives. *CoRR*, abs/1206.5538, 2012.

[2] Thomas Binder, Heinz Diem, Roland Fuchs, Kai Gutensohn, and Thomas Nebe. Pappenheim stain: Description of a hematological standard stain–history, chemistry, procedure, artifacts and problem solutions. *Journal of Laboratory Medicine*, 36(5):293–309, 2012.

[3] Philipp Gräbel, Martina Crysandt, Reinhild Herwartz, Melanie Hoffmann, Barbara M. Klinkhammer, Peter Boor, Tim H. Brümmendorf, and Dorit Merhof. Evaluating out-of-the-box methods for the classification of hematopoietic cells in images of stained bone marrow. 2018.

[4] Andreas Holzinger. Interactive machine learning for health informatics: when do we need the human-in-the-loop? *Brain Informatics*, 3(2):119–131, 2016.

[5] Weixian Lei, Rong Zhang, Yang Yang, Ruixuan Wang, and Wei-Shi Zheng. Class-center involved triplet loss for skin disease classification on imbalanced data. In *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, pages 1–5. IEEE, 2020.

[6] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction, 2018.

[7] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[8] Tingying Peng, Melanie Boxberg, Wilko Weichert, Nassir Navab, and Carsten Marr. Multi-task learning of a deep k-nearest neighbour network for histopathological image classification and retrieval. *bioRxiv*, page 661454, 2019.

[9] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clus-

tering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[10] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.