Global and Local Explanations for Skin Cancer Diagnosis Using Prototypes

Carlos Santiago*, Miguel Correia, Maria Rita Verdelho, Alceu Bissoto, and Catarina Barata

Institute for Systems and Robotics, Instituto Superior Técnico, Portugal

Abstract. Providing visual cues to justify the decisions of deep neural networks contributes significantly to increase their explainability. Typical strategies to provide explanations rely on saliency or attention maps that may not be easy to interpret. Moreover, the actual decision-making process is still a black-box. This paper proposes to overcome these limitations using class prototypes, both at the global (image-wide) and local (patch-based) levels. These associate images with the corresponding predictions by measuring similarity with learned image/patch descriptors. Our approach offers both global and local explanations for the decisions of the model, providing a clearer justification that resembles the human reasoning process. The proposed approach was applied to the diagnosis of skin lesions in dermoscopy images, outperforming not only black-box models, which offer no explanations, but also other state-of-the-art explainable approaches.

Keywords: Skin Cancer · Prototype Networks · CBIR · Explainable AI.

1 Introduction

In the last years, the landscape of medical image analysis has been transformed, mainly due to the adoption of deep learning (DL). The field of skin image, in particular dermoscopy, is a clear example, where recent studies have shown that DL achieves similar or even superior performance to that of clinicians [6]. While most experiments were conducted in artificial settings, it is undeniable the collaborative value of AI [18]. Another lesson to be taken from these studies is that any AI model should incorporate mechanisms to explain its decisions, increasing its safety and pedagogical value. As matter of fact, the incorporation of such mechanisms was recently recommended in a set of guidelines [5].

Explainable models can be divided into two main categories [20]: i) those that are intrinsically interpretable, being possible to understand the decision making process; and ii) those that resort to additional models to explain their output (post-hoc methods). Most works in dermoscopy fit in the latter. Methods like

* This work was supported by FCT projects LARSyS UIDB/50009/2020, 2022.07849.CEECIND, CEECIND/00326/201 and PRR projects PTSmartRetail C645440011-00000062 and Center for Responsible AI C645008882-00000055.

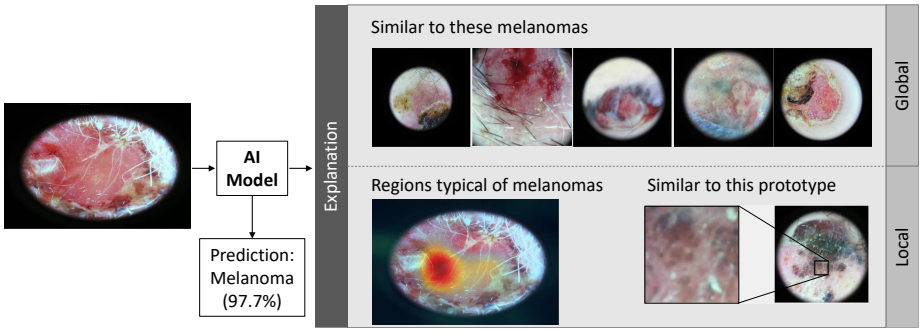

Fig. 1. The proposed method is able to justify its decisions using both image-level (global) explanations, obtained by retrieving the most similar training images for the predicted diagnosis, and patch-level (local) explanations that identify discriminative regions using heatmaps of similarity to local prototypes from training images.

[18, 7, 10] use saliency maps (*e.g.*, Grad-CAM [15]) to visualize the regions of the image that contributed to the predictions of DL models. Other methods like LIME [11] have also been used [16]. While these approaches are quite visual, the actual decision process still lacks clarity.

Example-based approaches are based on the assessment of past cases to infer a diagnosis. One of the most popular approaches is content-based image retrieval (CBIR) [17, 12]. This family of methods use the features of a DL model, usually trained for classification, to compute image distances, identifying dermoscopy images that are close in the latent space. However, there is no guarantee that the latent space is actually capturing lesion similarities. Moreover, the original classifier still comes short of being explainable. Finally, clinicians also screen the lesions for local structures that are hallmarks of each class. Adding a region-based reasoning to a diagnostic system may increase its complexity and often requires additional domain knowledge, such as annotations to identify clinically relevant structures [8]. Recent works in computer vision have overcome this issue using a prototypical part-based architecture called ProtoPNet [2], which is able to identify relevant region prototypes with minimum supervision. However, this method has been shown to underperform when compared with non-interpretable networks. Additionally, the learning process requires setting a trade-off between different loss terms. This is not trivial and leads to prototypes that lack diversity.

We propose a new model that easily integrates the best characteristics of CBIR and ProtoPNet, while simultaneously overcoming their limitations. The proposed approach learns: i) a set of global prototypes for each lesion class, thus achieving a more interpretable classifier that predicts a diagnosis from similarities; and ii) local prototypes to perform an interpretable part-based classification. Both the global and local feature spaces can be used to perform CBIR, in order to identify class specific images or image patches that justify the decision, as shown in Fig. 1. We conduct extensive experiments to validate our approach using the ISIC 2019 dataset and various CNN backbones. Our results demonstrate

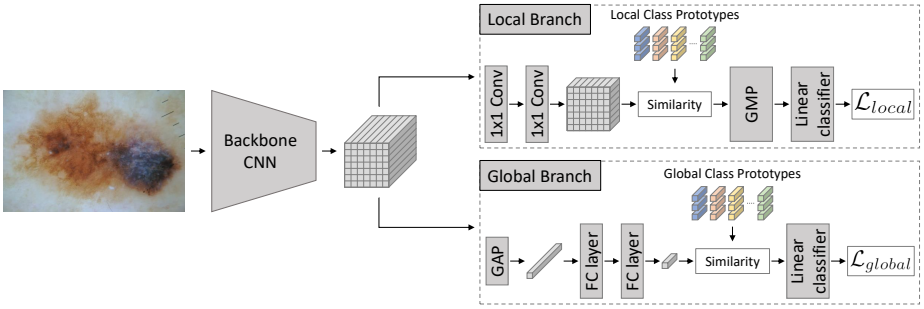

Fig. 2. Proposed approach - the method comprises two branches: i) a global branch that compares an input image with a set of class prototypes; and ii) a local branch, where regions of the input image are compared with local prototypes.

that the proposed approach achieves competitive performances when compared to the black-box models and ProtoPNet-based approaches, while providing a more transparent classification.

2 Proposed Approach

Fig. 2 shows the scheme of our proposal. A CNN backbone is used to extract a set of feature maps, F. Any CNN backbone can be used, as shown in our experimental results, where we compare several architectures. The feature maps are forwarded to the **global** and **local** branches. Each branch is responsible for estimating a probability vector $\hat{y} \in \mathbb{R}^C$, where C is the number of lesion classes. These estimates are obtained by computing a weighted average of the similarity between the latent vectors of the input image and learned class prototypes. The final classification is then the class with the highest probability obtained from averaging the two estimates.

The local branch identifies image patches that are specific of each class, while the global branch learns image-level representations of those classes. In both cases, the proposed approach is learning feature representations and simultaneously clustering them, ensuring that both local and global CBIR explanations can be provided.

To train the model, we combine the cross entropy losses for the global and local branches, $\mathcal{L}_{\text{global}}$ and $\mathcal{L}_{\text{local}}$, with a clustering loss, $\mathcal{L}_{\text{cluster}}$, that ensures the learned prototypes represent centroids of class-specific clusters. The final loss is

$$\mathcal{L} = \mathcal{L}_{\text{global}} + \mathcal{L}_{\text{local}} + \lambda \mathcal{L}_{\text{cluster}} \quad , \tag{1}$$

where λ is a hyperparameter. Each individual loss term is detailed below.

2.1 Clustering

ProtoPNet [2] combines two loss terms to force prototypes to be near patches from the corresponding class. However, these losses behave poorly when the

training set is severely imbalanced, as it will repeatedly push prototypes from minority classes away from the patch-level representations of the dominant classes, while seldomly pulling them towards the correct patches representations.

As such, we modified the learning process of all prototypes to ensure that they effectively capture clusters from their respective classes. Specifically, let $p_{c_k} \in \mathbb{R}^D$ define the prototype vector, of size D, corresponding to the k-th (global or local) prototype of class c. We adopt a mini-batch K-Means algorithm [14] to iteratively update the desired position, $\bar{f}_{c_k} \in \mathbb{R}^D$, of prototypes p_{c_k} according to

$$\bar{f}_{c_k} \longleftarrow (1 - \frac{1}{n_{c_k}})\bar{f}_{c_k} + \frac{1}{n_{c_k}} f_i \quad , \tag{2}$$

where f_i is the feature vector of sample i assigned to prototype p_{c_k}, and n_{c_k} is the current total number of samples that were assigned to p_{c_k}. Then, the clustering loss used to regularize the prototypes is given by

$$\mathcal{L}_{\text{cluster}} = \frac{1}{CK} \sum_{c=1}^{C} \sum_{k=1}^{K} \left\| \bar{f}_{c_k} - p_{c_k} \right\|_2 \quad , \tag{3}$$

where K is the number of prototypes per class. This loss term is applied to each branch, since each performs a similar tasks but either at a global (image) or local (patch) level. In the following sections, we will refer to global and local prototypes as $p^G_{c_k}$ and $p^L_{c_k}$, respectively.

2.2 Global Prototypes

The global branch aims to learn a set of K prototypes for each class, $\{p^G_{c_k}\}$, with $c = 1, \ldots, C$ and $k = 1, \ldots, K$. These prototypes are used to classify images based on the similarity of their image-level representations to the prototypes. To achieve this goal, the feature maps F computed by the CNN backbone are first combined using a global average pooling (GAP) layer, and then embedded into a smaller dimension latent space $f^G \in \mathbb{R}^{D^G}$ using two fully connected layers (see Fig. 2). The latent representation is compared to each prototype using the cosine similarity, $s(p^G_{c_k}, f^G)$. Then, we compute the probability of class c, $\hat{y}^G_c$, using a linear classifier with softmax

$$\hat{y}^G_c = \frac{e^{\sum_{k=1}^{K} w_{c_k} s(p^G_{c_k}, f^G)}}{\sum_{c'=1}^{C} e^{\sum_{k=1}^{K} w_{c'_k} s(p^G_{c'_k}, f^G)}} \quad , \tag{4}$$

where w_{c_k} is the weight given to the k-th prototype of class c. These weights are frozen and set to $w_{c_k} = \frac{1}{K}$ when training the prototypes and encoding layers, which means that each class score is given by an average of the similarities to the corresponding prototypes.

The prototypes are latent variables learned in an end-to-end fashion, together with the backbone layers. Given a batch of N samples, the global branch loss is

$$\mathcal{L}_{\text{global}} = -\sum_{i=1}^{N} \sum_{c=1}^{C} y_{i,c} \log(\hat{y}^G_{i,c}) \quad , \tag{5}$$

where $y_{i,c}$ is the one-hot encoding of the ground-truth and $\hat{y}^G_{i,c}$ is given by (4).

2.3 Local Prototypes

The local branch performs a similar analysis to the global branch, but in a patch-wise way. First, instead of finding a latent representation for the entire image, the feature maps, F, extracted by the CNN backbone are transformed into a lower dimensional latent space through two 1×1 convolutional layers, as shown in Fig. 2. This results in a new feature maps, $F^L \in \mathbb{R}^{H \times W \times D^L}$, where the j-th pixel contains the latent representation of the corresponding patch in the input image, denoted by f_j^L. Then, we compute the cosine similarity between the local prototypes, $\{p_{c_k}^L\}$, with $c = 1, \ldots, C$ and $k = 1, \ldots, K$, and the latent representation of each patch, f_j^L, $j = 1, \ldots, H \times W$.

A global max pooling (GMP) is used to obtain a single vector with the similarity of each local prototype to the image. This vector is then fed to a linear classifier to obtain the final probabilities of each class c, $\hat{y}_c^L$, following a similar approach to (4). Finally, the classification loss for this branch is given by

$$\mathcal{L}_{\text{local}} = -\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c} \log(\hat{y}_{i,c}^L) \ , \tag{6}$$

where $y_{i,c}$ is the one-hot encoding of the ground-truth and $\hat{y}_{i,c}^L$ is the predicted probability of class c for sample i.

2.4 Pruning and Final Classifier

Once the global and local prototypes have been learned, the final step of the training procedure is to tune the linear classifier, similarly to the training procedure described in [2]. For this last part, we freeze all the other parameters in the model, including the prototypes, and focus on improving the performance of the classifier by tuning the weight of each prototype. Specifically, the similarity vectors, given by the global and local branches, are concatenated into a single vector, $s \in \mathbb{R}^T$, where $T = 2CK$ is the total number of prototypes. Then, we build a weight matrix, $W \in \mathbb{R}^{C \times T}$, such that it initially computes exactly the same average used during the training of the prototypes – i.e., $w_{c,t} = \frac{1}{K}$ if the t-th prototype belongs to class c and $w_{c,t} = 0$ otherwise. The resulting matrix is used as initialization of a fully connected layer with no bias, which is then trained with the cross-entropy loss.

Since some of the learned prototypes may eventually be redundant, we also prune our model, discarding the less relevant prototypes. To achieve this, we rely on a binary mask, M, with the same dimensions of matrix W, that discards a prototype t by putting 0 on the t-th column of matrix M. As such, the class probabilities are obtained by first computing an element-wise multiplication of M and W, followed by the matrix product with the joint similarity vector, s. This prevents the discarded prototypes from contributing to the final prediction, similarly to a dropout strategy. As for the criteria for discarding prototypes, we chose a simple approach – if the same training sample was the nearest neighbour to multiple prototypes, we kept only the closest one.

2.5 Visual Explanations

The model's decisions are explained to clinical experts at two levels. On the global level, the representation f^G is used to perform CBIR, by comparing it with the representations of the training images associated with the closest class prototype. This process resembles the identification of past similar cases. On the local level, we show similarity heatmaps highlighting discriminative regions, along with the patch and image representing the corresponding prototype. Examples are shown in Fig. 1 and in supplementary material.

3 Experimental Setup

The proposed approach is trained and evaluated using the ISIC 2019 dermoscopy dataset [19, 3, 4][1], which contains 25,331 images for training and $C = 8$ classes, including 3 malignant ones. The dataset was normalized as proposed in [1] and split into training (80%) and validation (20%).

The proposed approach is assessed in five CNN architectures commonly used in dermoscopy image analysis: ResNet18, ResNet50, VGG16, DenseNet169, and EfficientNetB3. For each of these architectures the following models are trained: i) baseline CNN with an 8-neuron fully connected layer for diagnosis; ii) global prototypes only; iii) local prototypes only; iv) joint prototypes; v) ProtoPNet [2]; and vi) ProtoTree [9], an improved version of ProtoPNet that assumes an hierarchical organization of the prototypes. For each method, we compute the following evaluation metrics: a) the balanced accuracy (BAcc), which corresponds to the average recall; b) the average F-1 score; and c) the overall accuracy (Acc).

We optimized the training all models to convey the best results. Regarding the hyperparameters of our approach, we tested different configurations of: i) the dimension of the global prototypes $D^G \in \{128, 256\}$; ii) the local prototypes depth $D^L \in \{128, 256\}$; iii) $\lambda \in \{10^{-3}, 10^{-2}, 10^{-1}\}$ (from (1)); and iv) whether to prune the prototypes in the end. We set the initial number of global and local prototypes per class K^G and K^L to 10, as used in ProtoPNet. Nevertheless, it is important to recall that after the pruning stage, the number of prototypes will be smaller and vary across classes.

All models were trained for a maximum of 100 epochs with early stopping. We set the batch size to $N = 50$ and use online data augmentation. Additionally, we use a curriculum-learning approach to modify the importance of each training sample [13], in order to deal with the severe class imbalance. The weights of the CNN backbones are initialized using models pre-trained on ImageNet and fine-tuned with learning rate of 10^{-5}, while for the fully connected and convolutional layers in the global and local branches we used 10^{-3}, and 10^{-2} for the prototypes. The final classifier were trained for 20 epochs with the same batch size and a learning rate of 10^{-2}. For ProtoPNet [2] and ProtoTree [9], we adopted the optimal training procedures described in their corresponding papers. The models were trained on a NVIDIA Titan Xp using Pytorch[2].

[1] Under CC BY-NC 4.0

[2] https://github.com/cajosantiago/LocalGlobalPrototypes

Model	Approach	Acc	BAcc	F1
VGG16	**Baseline**	76.2	60.3	63.2
	ProtoPNet[2]	73.4	58.9	57.7
	ProtoTree[9]	75.9	54.6	58.4
	Global	76.7	60.9	63.6
	Local	75.6	61.3	62.4
	Joint	**77.3**	**62.9**	**65.1**
ResNet18	**Baseline**	75.6	63.7	62.8
	ProtoPNet[2]	71.7	56.0	53.9
	ProtoTree[9]	**78.7**	58.9	61.9
	Global	76.0	63.2	**64.2**
	Local	73.5	61.5	61.0
	Joint	75.1	**64.8**	63.9
ResNet50	**Baseline**	76.7	64.4	65.0
	ProtoPNet [2]	71.9	49.3	50.5
	ProtoTree[9]	**81.5**	**68.3**	**71.0**
	Global	78.3	67.6	67.7
	Local	77.3	65.9	66.2
	Joint	77.7	66.2	67.5
EfficientNet	**Baseline**	82.3	73.1	74.0
	ProtoPNet[2]	64.2	46.0	44.1
	ProtoTree[9]	**84.2**	**74.1**	**76.5**
	Global	79.8	71.2	70.7
	Local	78.7	68.7	68.9
	Joint	82.8	73.1	74.7
DenseNet	**Baseline**	**83.1**	74.7	**75.5**
	ProtoPNet[2]	75.8	55.2	57.5
	ProtoTree[9]	78.6	66.0	66.0
	Global	82.7	74.3	74.1
	Local	80.9	72.1	71.9
	Joint	82.4	**75.0**	73.4

Table 1. Comparison of CNN backbones, without using pruning. Best results for each backbone in **bold**.

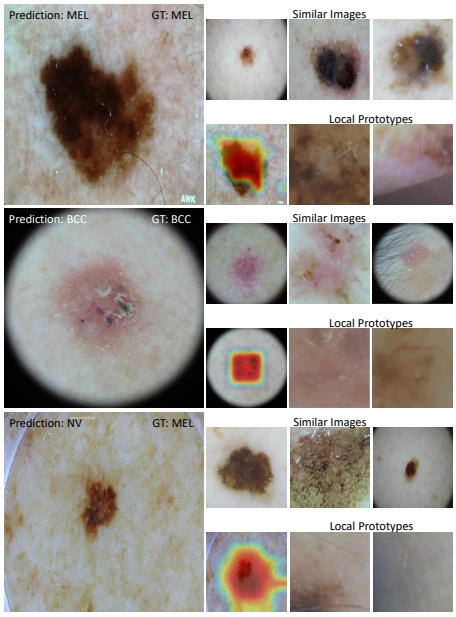

Fig. 3. Examples of predictions and corresponding CBIR explanations: global (top) and local (bottom).

4 Results

Table 1 shows the best experimental results for each CNN backbone, across all the evaluated methods (see the Supplementary Material for details on the best set of hyperparameters). Here we compare the results of our approach using a classifier without pruning, to make it more similar to the frameworks adopted in ProtoPNet and ProtoTree. In Table 2 we compare the results of our model with and without pruning. Fig. 3 shows examples of the proposed approach at inference time (additional examples can be found in supplementary material).

Global Prototypes vs Baseline: The approach based on global prototypes alone achieves competitive results across all backbones, outperforming the baseline into three out of the five architectures. This demonstrates that enforcing feature similarities between lesions of the same class does not affect the quality of the final classification. Additionally, it leads to more interpretable decisions that can be grounded in similar examples, as shown in Figs. 1 and 3.

Local Prototypes vs ProtoPNet/ProtoTree: ProtoPNet consistently exhibited lower performances when compared with all the other methods, as already reported in previous works. ProtoTree achieves better results than ProtoPNet, being the best approach for ResNet50 and EfficientNetB3. However, this method is very sensitive to the architecture, showing highly variable per-

	No Prunning			Prunning		
MODEL	GLOBAL	LOCAL	JOINT	GLOBAL	LOCAL	JOINT
VGG16	60.9	61.3	62.9	60.3	61.0	62.6
ResNet18	63.2	61.5	64.8	61.7	59.9	63.8
ResNet50	67.6	65.9	66.2	66.1	64.6	65.6
EfficientNetB3	71.2	68.7	73.1	69.7	66.8	72.4
DenseNet169	74.3	72.1	75.0	73.5	71.2	74.6

Table 2. BAcc results without and with pruning prototypes in the final classifier.

formances. The proposed local prototypes significantly outperform ProtoPNet, demonstrating the benefits of our training process. In particular, we achieve a better BAcc, since our approach handles class imbalance better than ProtoPNet. When compared with ProtoTree, our local prototypes seem to achieve more stable performances across backbones, being better in three of the five backbones. Moreover, it is interesting to observe that ProtoTree often shows a bigger gap between Acc and BAcc than our approach, suggesting that our model is also more robust to severe class imbalances than ProtoTree. Figs 1 and 3 show some examples of the local prototypes and their matching regions.

Joint vs Single Models: The proposed framework allows the training of a single branch (global or local) as well as their integration into a joint model. When comparing the individual branches, it is clear that the global prototypes always outperforms the local ones. This is somewhat expect, as by resorting to a local analysis alone, we might be missing relevant context cues about the lesions. When the two branches are combined, we observe that this usually improves the performance, suggesting that both of them contain relevant and complementary information. The results in Figs1 and 3 were obtained using the joint model. These visualizations give us a better understanding of the model's behavior, including its incorrect decision (last example in Fig. 3).

Prototype pruning: Table 2 shows the results before and after pruning. Interestingly, while there is a small decrease in the performance of both branches and the combined model, the scores obtained are still competitive with other methods. Overall, these results suggest that there is some redundancy on the learned prototypes, with an average of 35% of prototypes being pruned.

5 Conclusions

This paper proposed a new approach for skin cancer diagnosis that simultaneously provides global and local explanations to support the decision. Our model integrates two interpretable classifiers based on global and local prototypes. An experimental evaluation using various CNN backbones demonstrates the potential of our approach and opens a new direction in the development of XAI in medical image analysis. In the future we plan to integrate a few annotations to regularize the training of the local prototypes, as well as incorporate this model into a user-experiment to assess the clinical value of the prototypes and incorporate medical knowledge in the system.

References

1. Barata, C., *et al.*: Improving dermoscopy image classification using color constancy. IEEE JBHI **19**, 1146–1152 (2015)
2. Chen, C., et al.: This looks like that: deep learning for interpretable image recognition. Advances in neural information processing systems **32** (2019)
3. Codella, N.C.F., et al.: Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic). In: 2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018). pp. 168–172 (2018)
4. Combalia, M., et al.: Bcn20000: Dermoscopic lesions in the wild. arXiv preprint arXiv:1908.02288 (2019)
5. Daneshjou, R., et al.: Checklist for evaluation of image-based artificial intelligence reports in dermatology: Clear derm consensus guidelines from the international skin imaging collaboration artificial intelligence working group. JAMA dermatology **158**(1), 90–96 (2022)
6. Haggenmüller, S., et al.: Skin cancer classification via convolutional neural networks: systematic review of studies involving human experts. European Journal of Cancer **156**, 202–216 (2021)
7. Jaworek-Korjakowska, J., et al.: Interpretability of a deep learning based approach for the classification of skin lesions into main anatomic body sites. Cancers **13**(23), 6048 (2021)
8. Kawahara, J.e.a.: Seven-point checklist and skin lesion classification using multitask multimodal neural nets. IEEE journal of biomedical and health informatics **23**(2), 538–546 (2018)
9. Nauta, M., et al.: Neural prototype trees for interpretable fine-grained image recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14933–14943 (2021)
10. Nunnari, F., et al.: On the overlap between grad-cam saliency maps and explainable visual features in skin cancer images. In: International Cross-Domain Conference for Machine Learning and Knowledge Extraction. pp. 241–253 (2021)
11. Ribeiro, M.T., et al.: " why should i trust you?" explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. pp. 1135–1144 (2016)
12. Sadeghi, M., et al.: Using content-based image retrieval of dermoscopic images for interpretation and education: A pilot study. Skin Research and Technology **26**(4), 503–512 (2020)
13. Santiago, C., et al.: Low: Training deep neural networks by learning optimal sample weights. Pattern Recognition **110**, 107585 (2021)
14. Sculley, D.: Web-scale k-means clustering. In: Proceedings of the 19th international conference on World wide web. pp. 1177–1178 (2010)
15. Selvaraju, R.R., et al.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE international conference on computer vision. pp. 618–626 (2017)
16. Stieler, F., et al.: Towards domain-specific explainable ai: model interpretation of a skin image classifier using a human approach. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1802–1809 (2021)
17. Tschandl, P., et al.: Diagnostic accuracy of content-based dermatoscopic image retrieval with deep classification features. British Journal of Dermatology **181**(1), 155–165 (2019)

18. Tschandl, P., et al.: Human–computer collaboration for skin cancer recognition. Nature Medicine **26**(8), 1229–1234 (2020)
19. Tschandl, P., Rosendahl, C., Kittler, H.: The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. Scientific data **5**(1), 1–9 (2018)
20. Van der Velden, B.H.M., et al.: Explainable artificial intelligence (XAI) in deep learning-based medical image analysis. Medical Image Analysis p. 102470 (2022)