

Fine-tuning of explainable CNNs for skin lesion classification based on dermatologists' feedback towards increasing trust

Md Abdul Kadir¹[0000-0002-8420-2536], Fabrizio Nunnari¹[0000-0002-1596-4043],
and Daniel Sonntag^{1,2}[0000-0002-8857-8709]

¹ German Research Center for Artificial Intelligence (DFKI), Saarland Informatics
Campus, D3 2, 66113 Saarbrücken, Germany

² University of Oldenburg, Marie-Curie Str. 1, 26129 Oldenburg, Germany
{info,abka03}@dfki.de
<http://www.dfki.de>

Abstract. In this paper, we propose a CNN fine-tuning method which enables users to give simultaneous feedback on two outputs: the classification itself and the visual explanation for the classification. We present the effect of this feedback strategy in a skin lesion classification task and measure how CNNs react to the two types of user feedback. To implement this approach, we propose a novel CNN architecture that integrates the Grad-CAM technique for explaining the model's decision in the training loop. Using simulated user feedback, we found that fine-tuning our model on both classification and explanation improves visual explanation while preserving classification accuracy, thus potentially increasing the trust of users in using CNN-based skin lesion classifiers.

Keywords: Skin lesion · XAI (Explainable Artificial Intelligence) · SENN (Self Explainable NN) · XIL (Xplanatory Interactive Learning)

1 Introduction

A Convolutional Neural Network (CNN) can detect malignant skin lesions [8]; however, it cannot produce by default the explanation behind a prediction. In image classification, there are several ways to explain a prediction [13, 18, 11]. Nonetheless, in some cases, the explanations can be misleading, and by default the network does not provide the flexibility of learning from a given correction. As a result, the acceptance of such algorithms in the medical domain is quite rare. Differently, in the same situation, given the availability of correct feedback, decision-makers can perceive the reason for the mistake and take the necessary action to avoid the same kind of mistake in the future. These limitations of neural networks fall under the category of lack of interactivity.

In the domain of image classification, researchers introduced visual techniques to introduce explainability in deep learning-based image classification. They visualize the discriminatory regions of an image based on specific class identity [13]. Highlighting class discriminative regions in an image is an example of

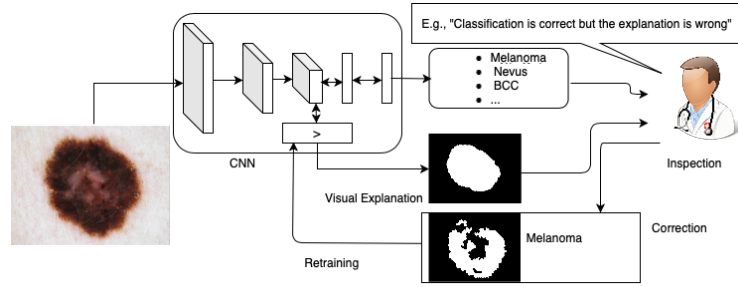


Fig. 1: Method overview: a pre-trained classification model produces classification result and visual explanation. Our interactive method allows a specialist to provide, if needed, corrections on both of them. Corrected information is then used to fine-tune the model on both output types.

explainability. This helps in spotting biased CNNs that wrongly identify the location of interest used to achieve the classification result. For example, sometimes classification performance is good, but the model classifies based on unrelated features. How can we tell the CNN that it is looking at the wrong image region and at the same time improve its classification accuracy?

We address this problem through the user-feedback approach is depicted in Fig. 1. We assume to start from a model that has been trained solely on classification data. Whenever a new sample is passed for classification, the resulting classification and visual explanation can be, independently, correct or wrong. If the correction is applied to the classification results, the solution is rather trivial. One simply needs to collect a batch of corrected labels and use them to perform transfer learning, or fine-tuning, on the starting model. Technically, this is done with the same data formats used for the original training. However, the challenge comes when the correction is applied to the visual explanation image.

In this paper, we propose a novel fine-tuning method that accepts human-corrected visual explanation images as part of the forward/backward propagation loop. On the technical level, our contribution consists of: i) the implementation a modified differentiable version of the Grad-CAM [13] explanation technique (so that corrected explanation images can be included in the back-propagation phase), and ii) the definition of a training procedure that takes into account simultaneously the classification and the visual feedback. With this approach, the difference between the original explanation image and the corrected version can be used as additional term of a loss function that includes, together with a classification error, also a visual explanation discrepancy.

Experiments are conducted in the domain of skin lesion classification, simulating how dermatologists could possibly identify wrong classifications and explanations and provide correction feedback.

2 Related work

This section presents the researches related to explainability and interactivity in the machine learning and the deep learning domains. At the beginning phase of their development, machine learning and deep learning algorithms were black-box models, except for linear and tree-based models. Black-box models only predict, but they do not present the reason behind their prediction. There are two ways for explanation: explaining the model itself, or explaining the reason for a specific prediction. They are known as global and local methods. In this research, we will only focus on the local methods because the research aims to improve the explanation of a sample-based prediction. First, we will cover the different explanation generation techniques of CNN-based and typical ML-based image classifiers. After that, we will discuss existing interactive methods where users provide feedback to NNs/AI algorithms to improve accuracy. We will also present the limitation of these methods.

2.1 Class activation mapping

A class activation technique generally works in CNNs. In this technique, the discriminatory classification features are extracted from the activation of any convolutional layer. Generating class activation map (CAM) is a technique for localizing class-specific significant features used in explaining convolutional neural networks. CAM has a remarkable localization capability. Zhou [18] describe a procedure for generating CAM using global average pooling (GAP) on convolution layers. GAP is the weighted sum of the convolutional feature map. In some CNN, a neural decision is made from the weighted sum of the GAP outputs. According to the authors, we can spot important image regions by linearly combining the weights of the output layer with the activation of the last convolution layer. Selvaraju [13] propose an explanation technique known as Grad-CAM, which is an extended version of CAM. It utilizes any target class flow gradients through the final convolution layer. Grad-CAM++ [3] is an extended version of Grad-CAM. In "Improved Visual Explanations for Deep Convolutional Networks", Chattopadhyay [3] propose this generalized method. It can produce improved visualization behavior of CNNs' predictions and performs better in visualizing multiple instances of an object during a classification. It produces an explanation similar to how Grad-CAM produces an explanation, but the only difference is that it only considers the positive gradient of the output class. Barata [2] proposed a hierarchical CNN-LSTM attention model that uses hierarchical information about classes and then produces attention mapping and hierarchical classification results. The attention map hints at how the classification algorithm looks at the objects in an image.

In this research we leverage class activation mapping to generate post-hoc explanation of our model while tuning. Any gradient based explanation technique can adapted to our method.

2.2 Interactive and explainable AI

Teso and Kersting [17] argue that interactive learning places the user into the loop, but the learner stays as a black-box for the user. They also suggest a novel explanatory interactive learning (XIL) framework that can overcome the limitation of interactive learning. Moreover, it can help the user gain trust in the learner by introducing completeness, directability, and understandability. In XIL, a user gives feedback to the learner’s output in an active learning manner when required. The proposed framework utilize LIME [12] as a local explainer and an additional component. They call the framework Caipirinhas (CAIPI). They use three functionalities, labeling the unlabeled data using user input, fitting the model on labeled and unlabeled features, and explaining a prediction using the local explainer. By introducing counter example, authors allow CAIPI to learn from the user’s feedback on the label and the explanation. Counterexamples are nothing but original input images with randomized irrelevant regions. There are three scenarios during the interaction between the learner and the user: the prediction and the explanation are correct, both are wrong, or only the label is correct. CAIPI focuses mainly on the last. It trains itself from the user’s feedback. After the evaluation, the authors see that the trust/distrust of the user increase based on the interactions. CAIPI’s performance increases due to the feedback explanation. However, the counterexample requires more disk space to store and GPU performance to retrain the model. The main difference between CAIPI and this research is how we train the network on feedback. CAIPI augments the original training data based on user feedback and then fine-tune or train a new model. There is no change in the objective function of the model. Also, storing original training data is necessary for retraining. However, the approach we follow in this research only requires post-deployment test images and feedback. Initial training data is unnecessary.

According to Teso [16], the Explanatory Active Learning (EAL) [17] algorithm depends on a post-hoc explainer, and it can generate a fragile and unfaithful explanation. He says the self-explainable active learning model is a solution to that. It is a combination of active learning and self-explainable neural networks (SENNs) [1]. Ghai [5] introduce Explainable Active Learning (XAL) in An Empirical Study of How Local Explanations Impacts Annotator Experience in 2020. Stuntebeck [15] propose a human-in-loop machine learning framework. This framework collects data from the patient using health sensors and trains a machine learning model on that data. Sometimes, due to the inefficiency of the sensors, the model prediction becomes wrong. To overcome this problem, they involve the user in the learning loop. Occasionally, the model gives the prediction, and the user gives feedback on the prediction by comparing what they are experiencing. Based on the feedback, the model tune itself. This framework is similar to this research involving humans in the learning loop. However, the feedback in this framework is only a yes or no decision. Holzinger *et al.* [6] argues that, while automatic ML suffers in performance because of insufficient training data, it is also true that interactive ML has the flexibility to allow a user to select suitable features heuristically from a vast search space. As a re-

sult, it can reduce the complexity of NP-hard using outside knowledge (Human intervention). The authors demonstrated the effectiveness of interactive machine learning and showed how to open a black-box technique to a glass-box one, enabling humans to interact with an algorithm. In the Skincare project, Sonntag *et al.* [14] describe the functionality and interface of an interactive decision support system for differential diagnosis of malignant skin lesions. The methods in the report give generic ideas and importance for interactive machine learning.

Besides lesion segmentation [7], several pieces of research are published in skin lesion classification [9]. We see that there are several approaches for skin cancer classification. Many of the methods have human-level accuracy. However, the application of these methods is constantly challenged by critique for legitimate reasons; for example, the proposed classifiers are black-box models, need training updates on new data, and lack interactivity with humans and the environment. So, in this research, we explore the interactive side of skin cancer classification.

3 Methods

Our method consists of a deep neural architecture that explains decisions to users and gives them the possibility to perform corrections that improve the model’s performance. Fig. 1 provides an overview of the system.

From the left side, we see a skin lesion fed into a convolution neural network, which predicts the class of the lesion and shows the areas that mostly contributed to the result to a dermatologist (on the right side). If the dermatologist is not convinced of the classification result or with its explanation, s/he gives a feedback. Based on the feedback, we re-train the model.

The remaining of this section explains the details of the implementation and describes how we simulated the feedback of dermatologists to validate our approach.

3.1 Integrating the explanation with the classification results

Our method starts from a pre-trained CNN which is able to perform image classification. A classifier based on convolutional networks is typically composed of two functional blocks: the convolutional layers (CL) block and the fully connected layers (FCL) block. Images (X) are fed into the CL at first, and the outputs of CL are flattened and passed through the FCL to generate the output (y). The output of the last convolutional layer is analyzed by explanation local techniques like Grad-CAM to produce a saliency map. A saliency map is a grey-scale image, with the same resolution of the convolutional layer, whose pixels with higher luminance are associated to the areas of the image that mostly contributed to the classification, while areas with dark pixels were mostly ignored.

The main idea of the method proposed in this paper, is to provide the results of a classification in terms both of classification result *and* saliency map to an expert, ask him to perform corrections, and use the corrected result (or a batch of corrections) for further fine tuning of the model. The fine tuning will take into

account both the class *and* the saliency map into a new composite loss function. This is done by attaching to the original model a differentiable branch able to extract the saliency map during a forward pass.

However, from a human perspective, correcting a grey-scale image might be too difficult and time consuming. In fact, in the context of skin cancer detection, many datasets available for the masking tasks provide *binary* masking images. Hence, we accommodate users' usability by converting greyscale saliency maps into binary *explanation maps*, where the users corrections consists only in switching pixels status between black and white, or viceversa. The conversion from saliency into explanation map (y_{exp}) can be performed by simple thresholding [10].

The new loss function of the extended model has two components, classification loss and explanation loss. The classification loss is the loss of a pre-trained network. However, the explanation loss is a newly introduced function, and it punishes the overall cost based on the difference between generated explanation and user feedback on explanation. The loss function can be written as

$$L = (1 - \lambda)L_{cls}(y, \hat{y}) + \lambda L_{exp}(y_{exp}, \hat{y}_{exp}) \quad (1)$$

Here L_{cls} and L_{exp} are the classification and explanation loss respectively. The hyper-parameter λ modulates the contribution of the two terms, and can be set during the model tuning. The two losses are defined as:

$$L_{cls}(y, \hat{y}) = - \sum_i \hat{y}_i \log(y_i) \quad (2)$$

$$L_{exp}(y_{exp}, \hat{y}_{exp}) = \sum_i J(y_{exp_i}, \hat{y}_{exp_i}) \quad (3)$$

Here, L_{cls} is the well-know categorical cross-entropy, while L_{exp} is the result of the Jaccard index between the output explanation map and the corrected one. An explanation map y_{exp} :

$$y_{exp} = T(A_y) \quad (4)$$

is the thresholded version of the result of the Grad-CAM saliency map A_y , which is defined as:

$$A_y = ReLU(\sum a_k^c A^k) \quad (5)$$

where A^k is the activation in a convolution layer k . Given Z as the total number of pixel in A^k (with resolution $i \times j$), for a given class c , a_k^c is computed as:

$$a_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\Delta y^c}{\Delta A_{ij}^k} \quad (6)$$

3.2 Implementation details

Algorithm 1 reports a formalized description of the approach. The main aspect that is worth an explanation, is that the Grad-CAM algorithm needs an alteration of the last layer in order to compute its saliency map. For this reason, each iteration of the fine-tuning algorithm requires two forward passes. The first, for the classification, is performed on the original model, while the second is performed on a temporary copy of the model, which is modified to get the saliency map and destroyed and the end of the batch iteration.

Algorithm 1 Training the self-explainable model

Require: e ▷ e : number of epoch
Require: $m = F(\theta)$ ▷ m : pre-trained model
Require: X, Y, Z ▷ X : input sample, Y : label, Z : binary mask (i.e. explanation ground truth)
Require: N ▷ N : total samples
Require: L_{cls} ▷ L_{cls} : classification loss function
Require: L_{exp} ▷ L_{exp} : explanation loss function
Require: γ ▷ γ : learning rate

$i = 1$
while $i \leq e$ **do**
 $n = 0$
while $n \leq N$ **do**
 $m_g \leftarrow \text{deepcopy}(m)$
 $a_k^c \leftarrow \frac{1}{R} \sum_p \sum_q \frac{\delta m_g(x)[c]}{\delta A_{ij}^k}$ ▷ $\frac{\delta m_g(x)[c]}{\delta A_{ij}^k}$: class-specific gradient on layer k for image x
 $A^k, \hat{y} \leftarrow m(x)$ ▷ A^k : receptive field of layer k , \hat{y} : predicted class
 $S \leftarrow \text{ReLU}(A^k \times a_k^c)$ ▷ S : saliency map
 $y_{exp} \leftarrow \text{th}(S, t)$ ▷ t : threshold value
 $L(\theta) \leftarrow (1 - \lambda)L_{cls}(y, \hat{y}) + \lambda L_{exp}(z, y_{exp})$ ▷ λ : hyper-parameter for loss balance
 $\theta \leftarrow \theta - \gamma \nabla L(\theta)$ ▷ $\nabla L(\theta)$: gradient with respect to loss
end while
end while

3.3 Simulation of user feedback

Getting user feedback for an experiment is very costly and time-consuming especially in the medical domain. Hence, instead of performing tests with real users, we exploited an existing dataset of skin lesion images associated to both classes and explanation maps, and used it to simulate users’ feedback. The ISIC 2018 [4] dataset of skin lesion attributes contains images of skin lesions and masks of five different attributes, pigment network, negative network, streaks, milia like cyst, and globules (Fig. 3). These attribute maps are binary masks locating the different attributes. The union of all of these attribute maps provides a comprehensive indication of what are the pixel areas that would lead a human practitioner towards his/her decision.

The ISIC2018 dataset (Fig. 2) is imbalanced. Preliminary experiments showed us that such imbalance affects the results of the simulation. Hence, we equalize the data per class by upsampling in the simulation set. We know that the maximum sample belongs to the nevus class (1951 samples), but there are 437

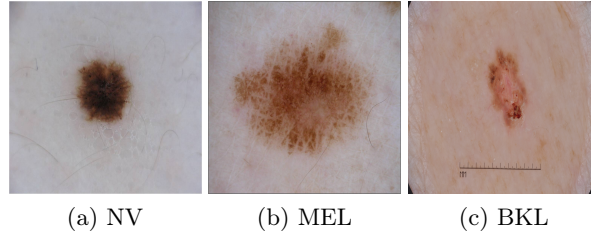


Fig. 2: Sample images from ISIC2018 task 2 dataset

examples for the MEL class and only 172 samples for the BKL class. We increase the number of the MEL and BKL samples to 1951 samples by coping them randomly.

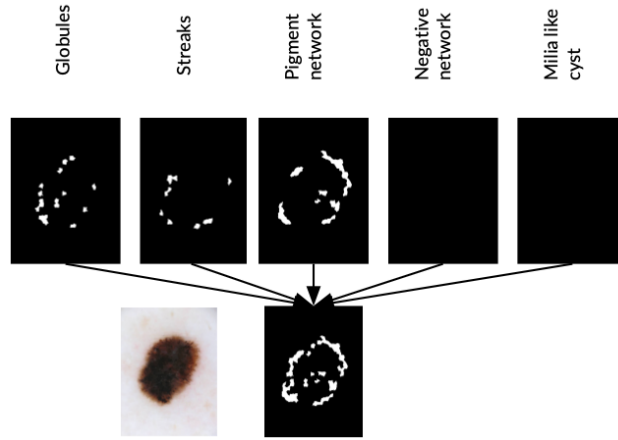


Fig. 3: The union of the feature maps.

4 Experiments and results

This sections described the two experiments conducted with our approach: the first aiming at understanding the contribution of the explanation loss function on a batch of data feedback, and the second aiming to analysing the behavior of the model when data feedback are provided in smaller batches.

4.1 Training on different loss functions

We use three different loss functions in the experiment: classification loss, explanation loss, and combined loss. The purpose of training the model on classification loss is to see how models perform when feedback is given only to classification data. During the training on classification, the model gets feedback only on classification output. So, the loss function only contains the classification part. We observe training and validation loss and average Jaccard index during the tuning process. Differently, the objective of using explanation loss is to see how the model performs when there is only explanation feedback. The explanation loss is the Jaccard loss which compares the predicted and feedback explanations. Finally, the combined loss combines both losses by a regularization parameter λ , which balances the two losses. The purpose is to see how classification and explanation feedback improve model performance.

We keep 10% of the data untouched to check the model performance and to see if the model learns from feedback, another 10% for validation and the remaining 80% for fine tuning the model through user simulation.

Table 1 presents the final result of the simulation. The baseline model performs 0.71 average sensitivity on the original test set (from ISIC 2019), and 0.73 when tested on the full simulation set (ISIC 2018). When performing a test on the 10% of the simulation set, using only the classification loss leads to 0.70 sensitivity and a Jaccard index as low as 0.10. Combining the classification and the explanation loss keeps the 0.70 sensitivity and increases the Jaccard index to 0.127, hence increasing the explanation power of the model. When testing using only the explanation loss, the Jaccard index boosts to 0.18, but the sensitivity drops to 0.66.

Hence, it seems that including an explanation loss term doesn't affect the classification capabilities of the model but it is able to improve its explanation power.

4.2 Sliced simulation

In this section, we present how a gradual provision of tuning data improves our explainable model's performance. Fig. 4(a) shows how model accuracy increases over an increasing amount of data. The x-axis represents data slices. Each slice has 238 samples. We have a total of 20 slices, and the model is iteratively fine-tuned on all the slices. We plotted the accuracy evolution for the three kinds of loss functions: classification only, explanation only, and combined. For the first two cases, we see that the accuracy has no significant improvement over the slices, while we see more accuracy fluctuation when combining classification and explanation loss functions. However, overall, accuracy remains stable at the end of the slices provision.

Fig. 4(b) presents the change in the average Jaccard index over the slices. Looking at the classification loss function's graph, we see that the average Jaccard index is not increasing. There is a slight improvement when we use combined loss. However, we see notable improvement of the average Jaccard index while

Model	Train data	Loss	Test set	λ	Test acc	Sensitivity			Avg sensitivity	Avg J-mean
Baseline	ISIC 2019	L_{cls}	ISIC2019	n/a	0.71	MEL	NV	BKL	0.71	n/a
Baseline	ISIC 2019	L_{cls}	2.5K test 100% Sim Set	n/a	0.73	0.53	0.77	0.88	0.73	0.10
Sim-model	80% of Sim Set	L_{cls}	10% Sim Set	0	0.74	0.74	0.75	0.61	0.70	0.106 (0.08)
Sim-model	80% of Sim Set	L_{exp}	10% Sim Set	1	0.76	0.49	0.85	0.64	0.66	0.18
Sim-model	80% of Sim Set	L_{cls} & L_{exp}	10% Sim Set	0.3	0.64	0.78	0.59	0.72	0.70	0.127 (0.10)

Table 1: Complete result of simulation on upsampled samples. We see that there is improvement of average Jaccard index when the explanation loss is included in the loss computation.

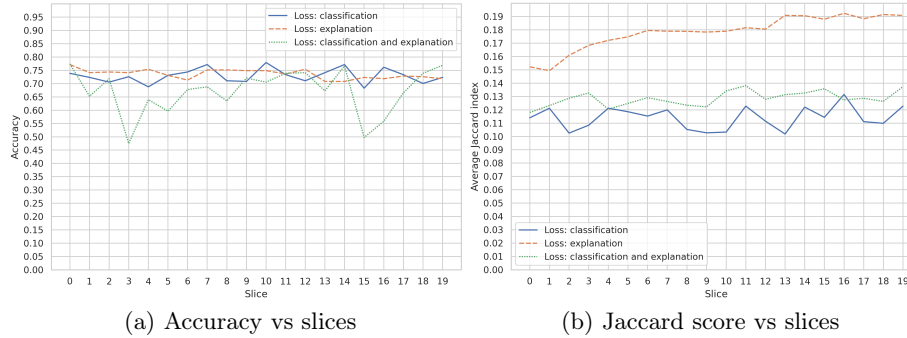


Fig. 4: An overview of performance in each epoch

using the explanation loss function. Table 2 shows the summary of the sliced simulation. After tuning the model on the first slice, we see that the average Jaccard index is 0.15. However, after the 20th slice, we see that the Jaccard index increases to 0.19. On the other hand, the accuracy reduces to 0.72 from 0.77 by keeping the average sensitivity the same when using the explanation loss function. These results conclude that model explanation performance increases without reducing classification accuracy when the model gets correction feedback in smaller chunks.

From those experiments, it seems that there is a slight better ending performance in the model after training over 20 smaller feedback slices rather than on the full 80% of the simulation set. However, the margins are small and this observed behaviour needs to be further analysed on bigger datasets to be confirmed.

5 Conclusion

We presented an approach for increasing the performances of CNN-based skin cancer classification by including users feedback in a post-train, fine-tuning stage. This approach transforms a black-box VGG16 model into a self-explainable neural network (SENN), which classifies (categories) and explains (saliency maps) at the same time. The implementation consists of augmenting a pre-trained VGG16 architecture with a differentiable implementation of the Grad-CAM algorithm.

We tested this approach by simulating practitioner’s feedback using the ISIC 2018 dataset. Our experiments show that our SENN is able, thanks to users feedback, to significantly increase its explanation power without compromising its classification accuracy, thus potentially increasing the trust of practitioners into computer-assisted diagnosis systems.

The main limitation of this approach is that it is applicable only when saliency maps offer a resolution that is high enough to provide users with a significantly fine-grained explanatory picture. Unfortunately, this is not the case in very deep neural networks, where the last convolutional stage (usually the preferred one to extract saliency maps) is composed by a high number of filters, but very few convoluted areas. Hence, at the moment, the trade-off is to use less classification accurate networks with higher resolution explanation maps, in order to include humans in the correction feedback loop.

Future work will focus on testing this approach with the involvement of real practitioners and on different datasets, to verify how well these results generalize to other domains.

Loss function	Slice #	Test set	Test acc	Test avg. sensitivity	Avg. Jaccard index(sd)
L_{cls}	0	20% of simSet	0.74	0.70	0.11 (0.09)
L_{cls}	19	20% of simSet	0.72	0.69	0.12 (0.10)
L_{exp}	0	20% of simSet	0.77	0.66	0.15 (0.12)
L_{exp}	19	20% of simSet	0.72	0.67	0.19 (0.15)
$0.70 \times L_{cls} + 0.3 \times L_{exp}$	0	20% of simSet	0.77	0.66	0.12 (0.10)
$0.70 \times L_{cls} + 0.3 \times L_{exp}$	19	20% of simSet	0.76	0.70	0.14 (0.11)

Table 2: Complete result of sliced simulation. As model gets more explanation and classification feedback, there is an improvement of average Jaccard index as well as of classification accuracy (last line).

Bibliography

- [1] Alvarez-Melis, D., Jaakkola, T.S.: Towards robust interpretability with self-explaining neural networks. CoRR **abs/1806.07538** (2018), <http://arxiv.org/abs/1806.07538>
- [2] Barata, C., Marques, J.S., Emre Celebi, M.: Deep attention model for the hierarchical diagnosis of skin lesions. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (June 2019)
- [3] Chattopadhyay, A., Sarkar, A., Howlader, P., Balasubramanian, V.N.: Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. 2018 IEEE Winter Conference on Applications of Computer Vision (WACV) (Mar 2018). <https://doi.org/10.1109/wacv.2018.00097>, <http://dx.doi.org/10.1109/WACV.2018.00097>
- [4] Codella, N.C.F., Rotemberg, V., Tschandl, P., Celebi, M.E., Dusza, S.W., Gutman, D., Helba, B., Kalloo, A., Liopyris, K., Marchetti, M.A., Kittler, H., Halpern, A.: Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (ISIC). CoRR **abs/1902.03368** (2019), <http://arxiv.org/abs/1902.03368>
- [5] Ghai, B., Liao, Q.V., Zhang, Y., Bellamy, R., Mueller, K.: Explainable Active Learning (XAL): An Empirical Study of How Local Explanations Impact Annotator Experience. arXiv e-prints arXiv:2001.09219 (Jan 2020)
- [6] Holzinger, A., Plass, M., Holzinger, K., Crisan, G.C., Pintea, C.M., Palade, V.: A glass-box interactive machine learning approach for solving np-hard problems with the human-in-the-loop (2017)
- [7] Jafari, M., Karimi, N., Nasr-Esfahani, E., Samavi, S., Soroushmehr, S., Ward, K., Najarian, K.: Skin lesion segmentation in clinical images using deep learning. In: 2016 23rd International Conference on Pattern Recognition (ICPR). pp. 337–342 (2016). <https://doi.org/10.1109/ICPR.2016.7899656>
- [8] Jain, S., jagtap, V., Pise, N.: Computer aided melanoma skin cancer detection using image processing. *Procedia Computer Science* **48**, 735–740 (2015). <https://doi.org/https://doi.org/10.1016/j.procs.2015.04.209>, <https://www.sciencedirect.com/science/article/pii/S1877050915007188>, international Conference on Computer, Communication and Convergence (ICCC 2015)
- [9] Lopez, A.R., Giro-i Nieto, X., Burdick, J., Marques, O.: Skin lesion classification from dermoscopic images using deep learning techniques. In: 2017 13th IASTED international conference on biomedical engineering (BioMed). pp. 49–54. IEEE (2017)
- [10] Nunnari, F., Kadir, M.A., Sonntag, D.: On the overlap between grad-cam saliency maps and explainable visual features in skin cancer images. In: Holzinger, A., Kieseberg, P., Tjoa, A.M., Weippl, E. (eds.) *Machine Learning and Knowledge Extraction*. pp. 241–253. Springer International Publishing, Cham (2021)

- [11] Petsiuk, V., Das, A., Saenko, K.: RISE: randomized input sampling for explanation of black-box models. CoRR **abs/1806.07421** (2018), <http://arxiv.org/abs/1806.07421>
- [12] Raghavan, H., Madani, O., Jones, R.: Active learning with feedback on features and instances. *J. Mach. Learn. Res.* **7**, 1655–1686 (Dec 2006)
- [13] Selvaraju, R.R., Das, A., Vedantam, R., Cogswell, M., Parikh, D., Batra, D.: Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. CoRR **abs/1610.02391** (2016), <http://arxiv.org/abs/1610.02391>
- [14] Sonntag, D., Nunnari, F., Profitlich, H.J.: The Skincare project, an interactive deep learning system for differential diagnosis of malignant skin lesions. Technical Report. arXiv:2005.09448 [cs, eess] (May 2020), <http://arxiv.org/abs/2005.09448>, arXiv: 2005.09448
- [15] Stuntebeck, E.P., Davis, J.S., Abowd, G.D., Blount, M.: Healthsense: Classification of health-related sensor data through user-assisted machine learning. In: Proceedings of the 9th Workshop on Mobile Computing Systems and Applications. p. 1–5. HotMobile '08, Association for Computing Machinery, New York, NY, USA (2008). <https://doi.org/10.1145/1411759.1411761>, <https://doi.org/10.1145/1411759.1411761>
- [16] Teso, S.: Toward faithful explanatory active learning with self-explainable neural nets. *Interactive Adaptive Learning* **2444**, 13 (2019)
- [17] Teso, S., Kersting, K.: Explanatory interactive machine learning. In: Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society. p. 239–245. AIES '19, Association for Computing Machinery, New York, NY, USA (2019). <https://doi.org/10.1145/3306618.3314293>, <https://doi.org/10.1145/3306618.3314293>
- [18] Zhou, B., Khosla, A., A., L., Oliva, A., Torralba, A.: Learning Deep Features for Discriminative Localization. CVPR (2016)