Interpretable CBIR for Medical Images

Antoine Basseto* ETH Zurich
Loïc Houmard† ETH Zurich
Jieming Li‡ ETH Zurich
Michael Sommer § ETH Zurich

ABSTRACT

In the past few years, the merits of ML methods for medical image analysis have been demonstrated, often surpassing human experts in performance and accuracy. However, the decision process in models has been largely intransparent, drawing into question how safe and how applicable these models are. For this reason, we develop Skinterpret, a platform that uses a VAE to provide content-based image retrieval (CBIR) of skin lesions. We include a concise workflow that allows doctors to upload lesion images and receive matching samples with patient and disease information, as well as explore latent space representations to gain further insight. We believe Skinterpret can be developed into a great practical tool for diagnosis.

Index Terms: VAE—Medical Image Analysis—Interpretability—;

1 INTRODUCTION

Machine Learning is permeating most facets of modern life, including the medical sector. There has been substantial interest and progress in the field, with Deep Learning models even outperforming trained radiologists in tasks like diagnosing disease from x-ray images [8].There is also research in various other problems, from survival analysis to research on underlying biological patterns.

Since performance of machine learning methods is limited by availability of high-quality input data, there have also been efforts to curate medical documents and images for training. One recent example of a valuable dataset is HAM10000 [10], containing a large corpus of image data on pigmented skin lesions. While methods have been proposed that achieve high disease classification performance on HAM10000 [3], they offer less in the way of interpretability. There could be an interest in a more exploratory approach that would provide insight into the exact reasons for classification and the patterns that show up in the dataset. Further, it would be valuable if experts could find information on lesions that present similarly to a sample at hand, using a sort of content-based image retrieval (CBIR). Recently, generative models such as GANs or Variational Autoencoders (VAE) have shown success for explainable generation of images from a low-dimensional latent space and high-performing, interpretable CBIR [9]. We utilize this in developing Skinterpret, a React/FastAPI application that allows for automatic analysis of images by harnessing a VAE trained on the HAM10000 dataset. We offer a concise workflow for uploading, processing and interpretation of skin lesions. The model shows promise in supporting and enhancing practitioners in their work.

2 MACHINE LEANING

2.1 Methodology

Our framework relies on using a low-dimensional latent space encoding of an auto-encoder-like architecture to define similarity of an image on "semantic" level. This works well because a auto-encoder-like architecture is forced by its bottleneck to learn highly condensed, informative representations of images. In addition, we let the encoder produce disentangled representations, isolating the effect of a single latent dimension. This allows us, by traversing single latent dimensions, to observe and identify the effects on the decoded image throughout the whole data set.

2.2 Model Architecture

Objective Function The encoder-decoder of our framework is based on Variational Autoencoders (VAE), a generative model, which aims to approximate the data generating function $P(*)$ through a true parameterized distribution $p_\theta(*|z)$ and to be defined prior p(z). In training, the 'prior' of latent distributions is modeled by the encoder $q_\theta(z|x)$ mapping an input image x to its latent representation z. Disentanglement as defined in [6] is achieved when single latent dimensions z_i are independent from each other with regards to a generative factor, i.e. only one is affected by changes to this factor while the others stay invariant to it. Taken together the VAE loss objective can be described as

$$\mathbf{L}(x,\theta,\psi) = \mathbb{E}_{q_\psi(z|x)}[log(p_\theta(x|z))] + KL(q_\psi(z|x)||p(z)) \quad (1)$$

The first term describes the reconstruction loss, i.e. how well the decoded image corresponds to the input image, while the second term describes how well the encoder distribution approximates the prior.
When the prior $p(z)$ is defined to be an isotropic Gaussian $\mathbf{N}(\mu,\mathbf{I})$, the KL-divergence becomes an easily trainable objective that guides the model to learn disentangled representations. Parametrizing the KL-divergence term with multiplicative factor β yields the β-VAE objective ([4])

$$\mathbf{L}(x,\theta,\psi,\beta) = \mathbb{E}_{q_\psi(z|x)}[\log(p_\theta(x|z))] + \beta KL(q_\psi(z|x)||p(z)) \quad (2)$$

where the size of β parameterizes the extent of disentanglement.
The KL-penalty term has been identified in literature as too big a penalty to achieve disentanglement. To achieve better overall reconstruction we in additionally utilized the so called $\beta - TCVAE$-loss ([2]), where only the total correlation, identified in literature as the main source of disentanglement in KL-divergence, is penalized, thereby reducing the regularization induced by forcing the generating function q to be similar to the prior.

Model We employ a VAE-based architecture to learn a small, yet highly efficient latent space of 12 dimensions. In the decoder we employ 5 convolutional layers of kernel size 4 with strides of 2 in order to aggressively downsample and compress the image. Two small fully connected layers are then used to create a mean μ and variance σ of each latent dimension. The such defined Gaussian $\mathcal{N}(\mu,\sigma)$ is sampled from in training using the reparametrization trick. During predictions, i.e. rollouts in our CBIR, only the mean μ is used. We invert the encoder architecture to get a mapping from the latent space to an image.

2.3 Experiments

2.3.1 Datasets

We experimented on two different medical image datasets: The NIH Chest X-rays dataset and a skin lesion dataset from the ISIC 2018

*e-mail: abasseto@student.ethz.ch
†e-mail: lhoumard@student.ethz.ch
‡e-mail: ljie@student.ethz.ch
§e-mail: sommemic@student.ethz.ch

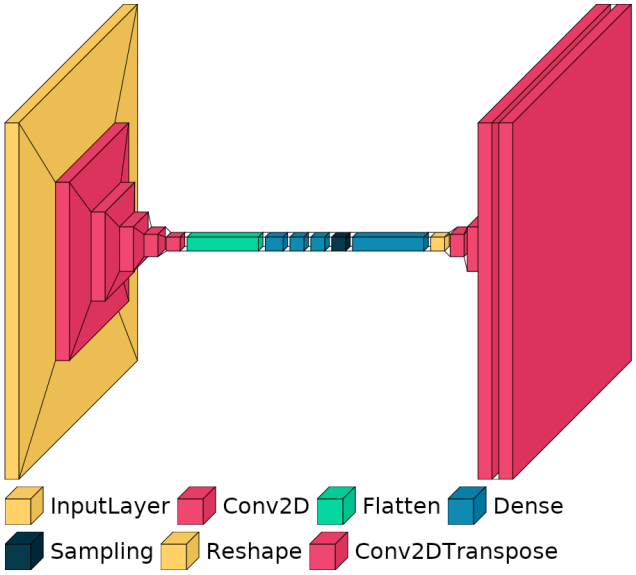


Figure 1: VAE architecture

challenge.

NIH Chest X-ray The NIH Chest X-ray dataset consists of 100,000 high resolution images of chest x-rays ([1]), with detailed labeling with regards to pathologies, patient data and even localization of diseases.

ISIC 2018 We use a subset of the ISIC 2018 challenge dataset known as HAM10000, consisting of 10015 dermatocospic images with a representative sample of pigmented lesions, including, but now limited to various carcinoma, melanoma etc.

2.3.2 Training

We experimented with various architectures and optimized hyperparameters using grid-search. Due to the lack of proper evaluation metrics, models were mainly evaluated through visual roll outs and inspection. Test loss performance and epoch-wise image samples to judge reconstruction were also utilized to aid and guide model selection.

Preprocessing Images were preprocessed to a size of 128×128 pixels, allowing for good overall reconstruction performance while preserving adequate resolution of the images.

Optimization All our models were trained for 200 epochs using a learning rate of 1e-4 and early stopping. We utilized Adam Optimizer and Cosine Annealing learning rate scheduler to adjust step size and learning rate during training.

2.4 Results

Overall, the VAEs achieved good performance for the HAM10000 dataset. Reconstruction preserved most of the high level features, only omitting finer details such as hairs, spotty colors and highly irregular boundaries, which were reconstructed to overall smoother boundaries. Roll outs allowed us to ascertain the effects of some single dimensions with regards to shape, color, size or position of the generated lesions (2).

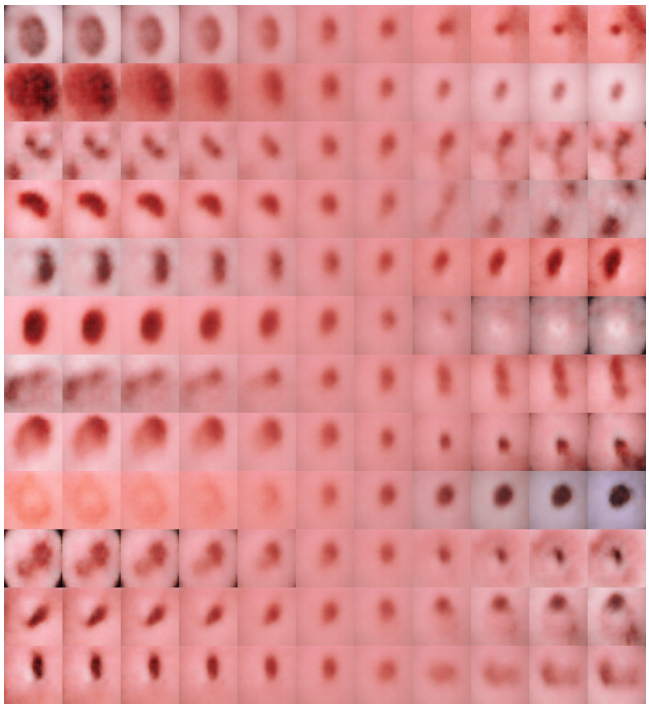
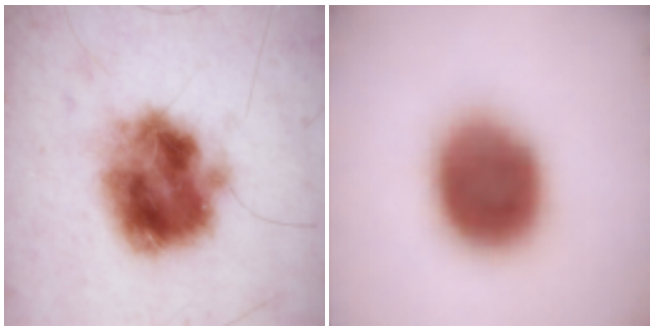
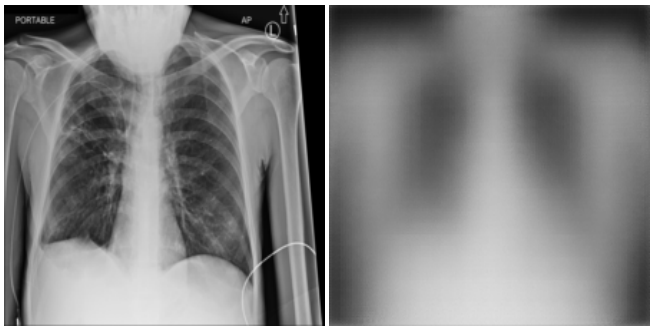


Figure 2: Example rollout: One can clearly identify the effects of some single dimension

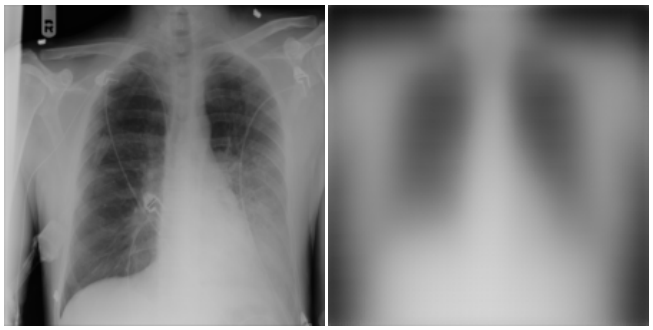


(a) Skin Lesions



(b) NIH Sample 1

Figure 3: Real vs Reconstructed



(a) NIH Sample 2, demonstrating mode collapse when compared to Sample 1

Figure 4: Real vs Reconstructed

The same performance could not be achieved for the NIH dataset, where overall reconstruction was very bad, bordering on mode collapse, resulting in an inability to visually discern the impact. None of the finer details where preserved, and mode collapse made reconstruction produce the same images even in cases where the original samples where vastly different.

2.5 Discussion

In general, our results indicate that disentangled representation learning methods based VAEs struggle on more complex data sets and good performance is limited to already widely used datasets (refer to [5]). In case of more complex images, the isotropic prior used to get disentanglement is too restrictive, leading to subpar reconstruction performance and less explainability of the latent space.

Possible improvements could be achieved by i.e. skip connection based networks in connection with autoregressive flows such as NVAE([11]). We have implemented a sample model, but have yet to actually explore it or implement the auto-regressive flows.

We have also implemented layerwise relevance computation for the network to visualize how important certain latent features are for the actual decoding. Additional clustering of the produced images based on feature importance should yield possibly better insights into the captured effects of single latent dimensions, thereby possibly making interpretations less dependent on visual exploration.

3 Interactive Dashboard

3.1 Stakeholders and Goals

Our dashboard was conceived for doctors or other domain experts and aims to provide them a fast and comprehensive retrieval of similar images. The idea behind it, is that our dashboards can help them work faster by showing them previous diagnoses made on past images which are similar to their current one and explain them why such images are considered as similar.

3.2 Workflow Summary

When we first open our dashboard, we land on a summary page where the doctor is asked to upload an image. Once done, he can explore the different dimensions of our auto-encoder, which are provided as a roll-out of images, where we show him what changes occur when one dimension only is perturbed while the others are kept untouched. Since our dimensions are partly disentangled, our hope is that some of them might be linked to a real factor which could be of medical relevance, such as the size of the pathology. Hence, the doctor can rename any dimensions he understands and finds useful. He can then go to the filter page, where he can weight differently each dimension for the similarity computation, putting more weights on the ones he thinks are relevant and less on the other ones. He can also decide to retrieve images based on other factors, such as the diseases or the age of the patients. Finally, he can let our model compute the distance and show him similar images together with their records and a visualization of distance along each dimension. We also provide him another visualization which is a 2D projection of the full latent space where the current and similar images are highlighted to help him understand the similarity.

3.3 Layout

Our layout is divided into two main parts: a main panel which shows different pages and a sidebar which let the user move between the different pages and apply different filters to retrieve similar images.

3.3.1 Uploading Page

The first page of our dashboard is made to introduce the user to our dashboard by giving him simple instructions about the workflow and a few explanations about how the machine learning works underneath. It also lets him upload an image.

3.3.2 Explore Dimension Page

The explore dimension page is made to let the user have a better understanding of the latent space of our model. It is important that he can understand it clearly, since our retrieval is entirely based on it. It is presented as a roll-out of images, which is a common method used in other papers to explain latent space of VAEs. Each image is generated by taking the mean vector of our latent space computed by the VAE's encoder, perturbing one dimension only (for each row a different dimension) with equal steps and then using our decoder to generate the corresponding reconstructed images. Since our latent space should be disentangled, the doctor can see what is the impact of one dimension on the image. Our dashboard gives him the possibility to rename any such dimension, which will also be shown on the next visualizations in other pages and which will help him to understand them without having to memorize them or move back-and-forth between different pages.

3.3.3 Filters

The filters are made to let the user choose how similar images are computed and which ones should be kept. It consists of two parts: a first one used to give different weighting for the computation of the distance to retrieve close images and a second one allowing to filter the retrieved images on some criterion. The weighting part just consists of one slider per dimension ($K = 12$ in total) where the doctor can give a weight w_j from 0 to 1. The distance used to compute the similar images is then simply the weighted Euclidean distance along each dimension between the mean of our latent space μ and the mean of every other images' latent space μ_i:

$$\text{distance}_i = \sqrt{\sum_{j=1}^{K} (w_j(\mu^{(j)} - \mu_i^{(j)}))^2} \tag{3}$$

We found interesting to let the domain experts have the freedom to choose which dimensions of our model are relevant and which are not and also to just be able to try different similarity based on different features.

The filters allow to choose to retrieve images fulfilling some criterion only. The doctor can for example decide to retrieve only images of people in a given age range or to choose images of one or more diseases present in the data set. He can also decide how many such images will be shown, to have a wider overview or in contrary focus only on a smaller subset.

3.3.4 Similar Images Page

The similar images page is used to show the results of the image retrieval. The uploaded image is displayed on the top. A list of cards containing the similar images together with information about the diagnosis, the total weighted distance and a radar-glyph which shows the difference along each dimension independently appears below it. The most interesting part to understand why our model considered these images as being most similar is the radar-glyph. It shows in a concise manner the difference along the 12 dimensions and hence make the comparison between images easily feasible. A doctor can therefore see along which dimensions the image is similar to the uploaded one and along which they are different and can refine its weighting and filtering if needed to access other images. The distance showed in the radar-glyph is re-scaled in the [0,1] range for visualization purpose so that the distance doesn't overflow in the diagram. Many methods could be used for the scaling. A first possibility would be to re-scale each image independently by dividing every values by the maximum value found over all the dimension of that image. The advantage is that each image would use the full range of values in the [0,1] interval making it visually appealing. However, it would make the comparison between two different images, which is the main goal of our dashboard, impossible. Another possibility could be to take the distance in each dimension over every images in the full data set and divide by the maximum value. Unlike the previous idea, this would lead to glyphs which can be used to compare images (since they are all scaled with the same value), but because the largest distance would be quite big, most of them would be very small and hard to visualize. Therefore, an approach which is a trade-of between the 2 aforementioned methods was chosen. It consists of scaling each dimension by the maximal value over each dimension of every similar images. It makes therefore the comparison between images easily feasible and still makes use of most of the range of values. It only has the drawback that the glyph of an image is very likely to change its scale between two different runs with different filters, which might not be very intuitive for the user.

3.3.5 Projection Page

This page allows the user to see a 2D projection of the latent space of our data set where the newly uploaded image together with the similar images are highlighted. It has two main goals, the first one is to show again a sort of distance measure in a 2D projection (which can be easily visualized because of its low dimension) and the second one is to give an overview over the full data set which the doctor can examine. Note however that the closest images shown in the similar images page are often not the ones which are closest in the 2D projection but are often still quite close in this representation.

To project the data, UMAP [7], which is a general purpose manifold learning and non-linear dimension reduction algorithm, was used. The hyper-parameters we used were number of neighbors = 15 and min distance = 0. The first controls how UMAP balances local versus global structure in the data and the second one how tightly UMAP is allowed to pack points together.

For the visualization, we decided to give each disease a round shape with different colors to make them easily recognizable. Once an image is uploaded, we display it as a bigger black square and the similar images are shown as bigger crosses with a black border to be more visible. We also reduce the opacity of the other points to accentuate the effect. We decided to stack different channels (size, shape and opacity) to make them really pop out so that the doctor can concentrate more easily on them. We also used tool-tips on top of each point which give a useful summary of the image when the user puts his mouse over it. Hence the doctor can very easily have an overview over the full data set by only using this feature.

REFERENCES

[1] IEEE, jul 2017. doi: 10.1109/cvpr.2017.369

[2] R. T. Q. Chen, X. Li, R. Grosse, and D. Duvenaud. Isolating sources of disentanglement in variational autoencoders, 2018. doi: 10.48550/ARXIV.1802.04942

[3] S. K. Datta, M. A. Shaikh, S. N. Srihari, and M. Gao. Soft-attention improves skin cancer classification performance. 2021. doi: 10.48550/ARXIV.2105.03358

[4] I. Higgins, L. Matthey, A. Pal, C. P. Burgess, X. Glorot, M. M. Botvinick, S. Mohamed, and A. Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017.

[5] L. Klein, J. B. S. Carvalho, M. El-Assady, P. Penna, J. M. Buhmann, and P. F. Jaeger. Improving explainability of disentangled representations using multipath-attribution mappings. In *Medical Imaging with Deep Learning*, 2022.

[6] F. Locatello, S. Bauer, M. Lucic, G. Rätsch, S. Gelly, B. Schölkopf, and O. Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. 2018. doi: 10.48550/ARXIV.1811.12359

[7] L. McInnes, J. Healy, and J. Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[8] S. McKinney, M. Sieniek, V. Godbole, J. Godwin, N. Antropova, H. Ashrafian, T. Back, M. Chesus, G. Corrado, A. Darzi, M. Etemadi, F. Garcia-Vicente, F. Gilbert, M. Halling-Brown, D. Hassabis, S. Jansen, A. Karthikesalingam, C. Kelly, D. King, and S. Shetty. International evaluation of an ai system for breast cancer screening. *Nature*, 577:89–94, 01 2020. doi: 10.1038/s41586-019-1799-6

[9] N. Norouzi, M. Zarvani, S. Moghadam, and R. Azmi. Cbir-gan: A triplet generative adversarial network for content-based image retrieval. *SSRN Electronic Journal*, 01 2022. doi: 10.2139/ssrn.4057354

[10] P. Tschandl. The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions, 2018. doi: 10.7910/DVN/DBW86T

[11] A. Vahdat and J. Kautz. Nvae: A deep hierarchical variational autoencoder, 2020. doi: 10.48550/ARXIV.2007.03898

CONTRIBUTION STATEMENT

We all worked on the overall aspect and design of the dashboard, and every decision was made as a team thanks to regular meetings in which we all took part.

Antoine Basseto

I worked on the front-end and back-end necessary for the projection, changed some of the overall design, made the poster and updated the git repo for final submission.

Michael Sommer

I worked some on the very first VAE implementation, which ended up being completely reworked. I also did the backend (querying and filters) and radar charts for the Similar images page, and wrote abstract and introduction of the paper.

Jieming Li

I worked a bit on the backend (image querying and model integration), and some initial linking of front-end and backend. Apart from this I mainly did the ML training on X-Ray and Skin Lesion datasets and the Pytorch Lightning implementation. For the report I wrote the ML section.

Loïc Houmard

I mainly worked on the front-end of the sidebar, filters, uploading page, Similar images page and Explore dimension page. I linked some parts with the backend and did some updates there. I worked at the beginning on the very first steps of the ML model with the chest X-ray dataset (load the data, explore it and create a very basic auto-encoder). I also wrote part 3 of the paper.

APPENDIX I. DASHBOARDS IMAGES

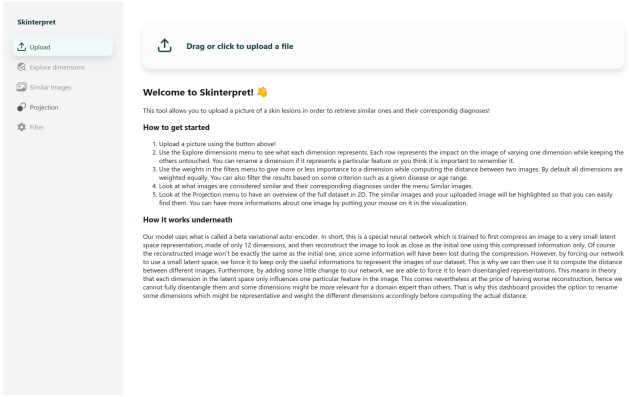


Figure 5: Uploading page

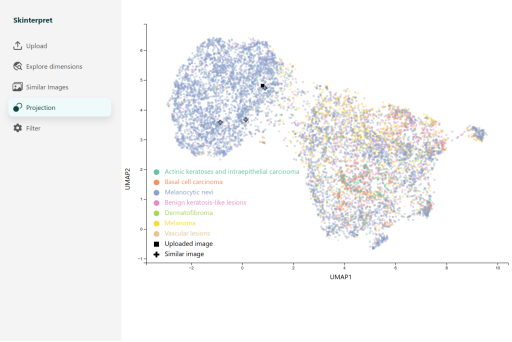


Figure 8: Projection page

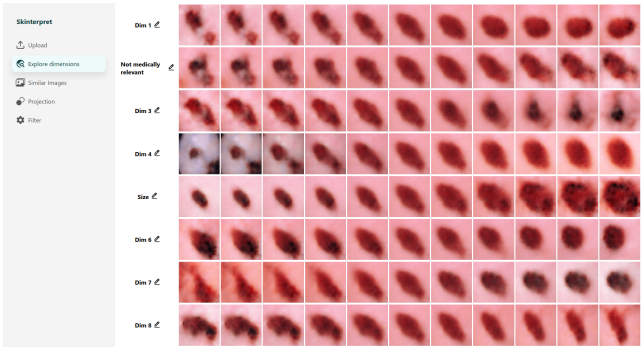


Figure 6: Explore dimension page

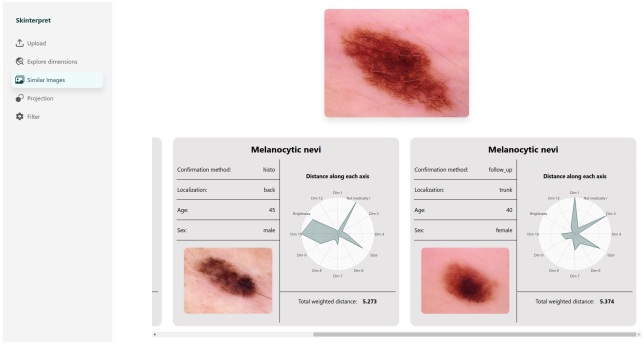


Figure 7: Similar images page

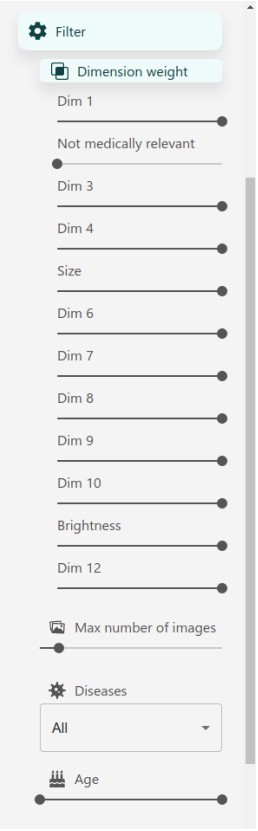


Figure 9: Filters