A latent space exploration for microscopic skin lesion augmentations with VQ-VAE-2 and PixelSNAIL

Gallucci, Alessio, Pezzotti, Nicola, Znamenskiy, Dmitry, Petkovic, Milan

Alessio Gallucci, Nicola Pezzotti, Dmitry Znamenskiy, Milan Petkovic, "A latent space exploration for microscopic skin lesion augmentations with VQ-VAE-2 and PixelSNAIL," Proc. SPIE 11596, Medical Imaging 2021: Image Processing, 115962X (15 February 2021); doi: 10.1117/12.2580664

SPIE.

Event: SPIE Medical Imaging, 2021, Online Only

A latent space exploration for microscopic skin lesion augmentations with VQ-VAE-2 and PixelSNAIL

Alessio Gallucci[1], Nicola Pezzotti[1,2], Dmitry Znamenskiy[2], and Milan Petkovic[1,2]
1) Eindhoven University of Technology, Eindhoven, Netherlands
2) Philips Research, Eindhoven, Netherlands
{a.gallucci, n.pezzotti, m.petkovic}@tue.nl,
{nicola.pezzotti, dmitry.znamenskiy, milan.petkovic}@philips.com

ABSTRACT

Skin cancer affects more than 3 million people only in the U.S. Comprehensive microscopic databases include around 30 thousand samples, limiting the richness of patterns that can be presented to machine learning. To this end, generative models such as GANs have been proposed for creating realistic synthetic images but, despite their popularity, they are often difficult to train and control. Recently an autoregressive approach based on a quantized autoencoder showed state of the art performances while being simple to train and provide synthetic data generation opportunities. In the first part of this paper we evaluate the training of VQ-VAE-2 with different latent space configuration. In the second part, we show how to use a learned prior over the latent space with PixelSNAIL to generate and modify skin lesions. We show how this process can be used for powerful data augmentation and visualization for skin health, evaluating it on a downstream application that classifies malignant lesions.

Keywords: dermatology, autoregressive generation, skin lesions, image generation, autoencoders.

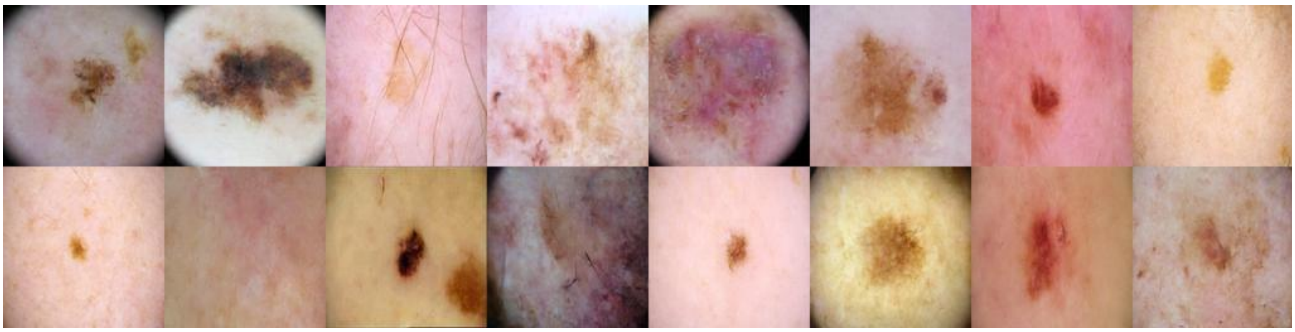

Figure 1. Synthetic skin lesions that do not exist in the real world. The lesions are generated in the latent space of a two-layer VQ-VAE-2 autoencoder. The codes are sampled from top left to bottom right using an autoregressive model. First, the top codes are generated, then the bottom ones conditioned on the top. Then, VQ-VAE-2 decoder reconstructs the image from the latent codes, which represent embedding quantizing vectors.

1. INTRODUCTION

Skin cancer is the most common cancer in the U.S. [1], and, the number of treated adults has increased over time, from 3.4 million in the 2002-2006 period to 4.9 million in the 2007-2011 period [2]. Usually, screening and diagnosis are primarily carried by clinical visual inspection and biopsy if necessary. There is an urge to automate and facilitate this procedure with image-based screening since early detection is crucial for treatment options [3]. Many state of the art Convolutional Neural Network (CNN) models performed on par, or better, compare to dermatologists: in 2017, Esteva et al. [4], trained a CNN that performed on par with 21 dermatologists and in 2019, Brinker et al. [5], [6], shows that Resnet50 [7] outperformed 136 of 157 dermatologists. However, core research is still needed to reach a fully automated diagnostic system which addresses the ethical and technical hurdles of deploying such models in the real world. A challenge is the lack of data since collecting real samples is time-consuming and difficult, for example, considering rare diseases. Another challenge is the fairness of the AI systems concerning underrepresented populations. Two common biases are the skin type—skin lesions datasets are skewed towards light skin types—and body location acquisition—most lesions come from easy to collect places such as limbs or torso and few from less conformable areas. While there are many reasons for the nature of such

Medical Imaging 2021: Image Processing, edited by Ivana Išgum, Bennett A. Landman, Proc. of SPIE Vol. 11596, 115962X · © 2021 SPIE · CCC code: 1605-7422/21/$21 · doi: 10.1117/12.2580664

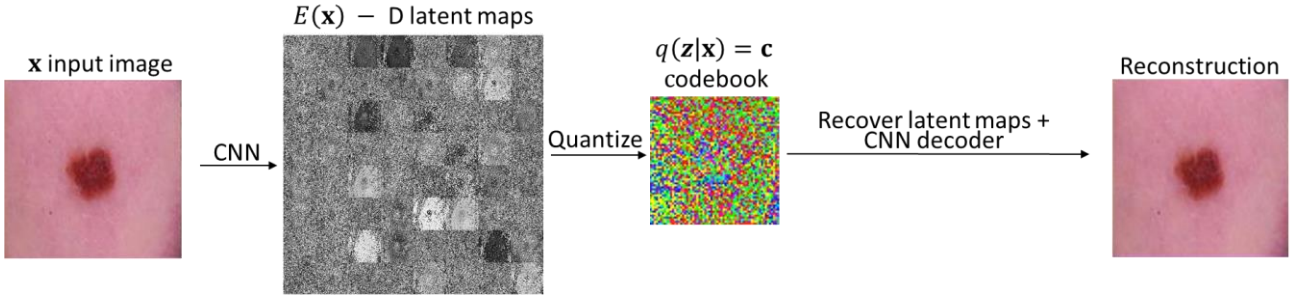

Figure 2. Encoding and quantizing a single image with a single layer vector quantizing autoencoder. The image is first encoded with a fully convolutional encoder into D latent maps and then quantized using K quantizing vectors.

biases, some of them difficult to solve by simply collecting more data, it is possible to hamper their effect by creating synthetic examples covering the underrepresented class.

In the past years, generative models have improved very fast and significantly. The growth in popularity is related to the visually appealing results and the continually increasing computational power commonly available. The synthetic samples are difficult to spot compared to the original data even from a careful human inspector [8]. The state of the art models are called Generative Adversarial Networks (GANs) [9], and they rely on two competing architectures, one generating images starting from noise, the other trying to distinguish synthetic from real samples. The mapping from a noise vector to the sample coupled with the adversarial training produces very realistic images [10], but it is often challenging to train and control afterwards.

Recently, a different approach has proven similar results to GANs on image generation tasks. This approach, use an autoencoder Vector Quantized Variational Autoencoder (VQ-VAE-2) [11] in combination with an autoregressive model called PixelSNAIL [12], and it is able to generate high resolution realistic pictures as shown in Figure 1. The autoencoder finds a compact quantized representation of the input data using stacked VQ-VAE [13] and represents an input image with a lower resolution map of codes. Since this representation is given at lower resolution, these codes represent different image patterns and, in the case of the hierarchical implementation, patterns at different scales. VQ-VAE-2 relies on feedforward networks, thus easy to train and, according to [11], it does not suffer that much from the common pitfalls of GANs, such as the mode collapse or lacking full support over the input distribution [14]. Moreover, it easily allows for the explainability of the generative model, as each code in the latent space can be seen as a self-supervised label extracted by the model. An example of a single layer autoencoder is presented in Figure 2, where the input image is encoded and quantized into discrete codes. Once learned the quantized latent representation, the PixelSNAIL autoregressive model is used to learn the prior distribution over the discrete latent codes and to generate new realistic images. Key advantages of this approach are the explainability of the features and the potential for integrating advanced augmentation techniques. Figure 3 shows the input to the VQVAE-2 model, and the corresponding codes in the two hierarchical levels (here color-coded with a randomized palette).

In this paper, we apply the VQ-VAE-2 architecture to generate novel skin lesions, with and without the autoregressive model, to augment and increase the number of images in the lesions' datasets. While VQ-VAE-2 model has been proven to perform well in different domains like ImageNet [15], as shown in the ML community by A. D'Amour *et al.* [16] it is often a challenge to replicate ML pipelines in different domains, especially for medical images. This is one reason we present an exploration of the latent space generated and the behavior and hurdles we faced selecting the right latent space dimensions. For example, once trained with the original double-layer configuration, we report the top latent space's collapse leading to a one-layer autoencoder.

2. RELATED WORK

In this paper we present different ways to generate and manipulate skin lesions. In the related work, first we present a basis for the approach by showing the VQ-VAE and VQ-VAE-2 setups and then we introduce the autoregressive model. In the last part of this section, we present other works which employ generative models, such as GANs, to create novel skin lesions.

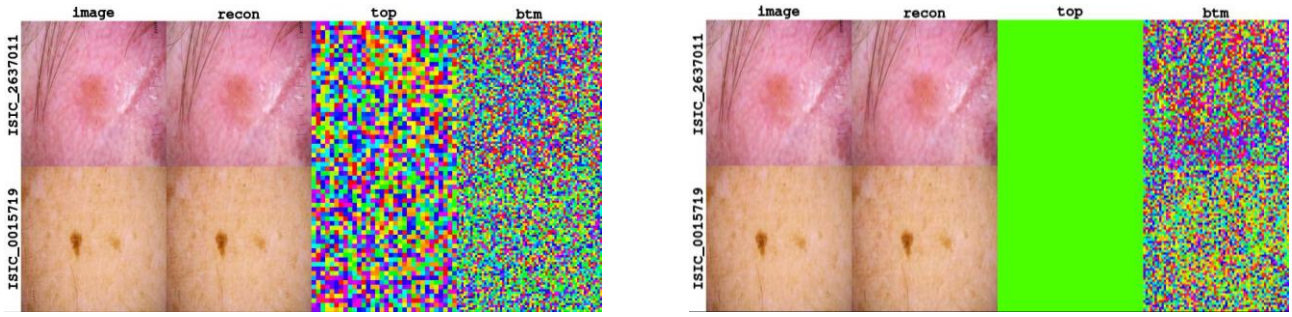

Figure 3. Left a model with latent space dimension ($K = 256, D = 8$); Right a model with latent space dimension ($K = 512, D = 64$), and the corresponding discrete embeddings, for 2 different images. Each row is a different image. The first column is the input image, the second the VQ-VAE-2 reconstruction. The third column presents the top encoding and the fourth the bottom encoding. Each colour in the third and fourth column represents a different integer code. The encoder quantizes each pixel, of $D = 8$ dimensional feature map, into one out of $K = 256$ codewords. The model on the right presents a single color in the top space since the top hierarchy is collapsed.

2.1 Vector Quantized Variational AutoEncoder

The VQ-VAE model is introduced in [13] and it builds on top of the Variational AutoEncoder (VAE) [17], [18] by generalizing ideas from classical image compression methods like jpeg. Given a dataset of observations $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(N)}\}$, the goal of a VAE is to learn, without supervision, a lower dimensional representation in terms of latent variables $\mathbf{z}$. It is composed by an encoder E, which map the input image into latent variables, and a decoder, which reconstruct the image from the compressed representation. In other words, the decoder network models the joint distribution $p(\mathbf{x}|\mathbf{z})p(\mathbf{z})$ while the encoder models the posterior distribution $q(\mathbf{z}|\mathbf{x})$.

In the VQ-VAE framework the prior distribution is based on K prototype latent vectors $\{\boldsymbol{e}^{(1)}, \boldsymbol{e}^{(2)}, \dots, \boldsymbol{e}^{(K)}\}$ of dimension D which quantize the latent maps $E(\mathbf{x})$, generated by the encoder. There are exactly K different latent vectors to choose from, so each pixel on the latent maps is represented with the nearest quantizing vector, as shown in Figure 2. In Razavi et al. [11] the VQ-VAE-2 is presented, which is the upgrade of the VQ-VAE to include multiple hierarchical layers which provide different quantize codebooks at different hierarchies. The decoder then reconstructs the image using the latent maps conditioning the higher levels, which have smaller resolution, to the bottom ones. In the original setup the input 24 bit image with resolution 256x256 was reduced to 64x64 bottom map and 32x32 top map with $K = 512 = 2^9$ different quantizing vectors of dimension $D = 64$. In [11] the authors present two layer network trained on ImageNet [19], and three layer network trained on FFHQ [20], for generating high-resolution photo-realistic facial images. In order to solve large scale dependencies, which are usually difficult to capture by the autoregressive decoder, Fauw et al. [21] successfully explored the possibility to use many layers encoder. While in another research work, Williams et al. [22] used Hierarchical Quantized Autoencoders for image compression purposes.

In our experiments we focus on a two layers hierarchy with input resolution of 256×256 and relative latent maps of dimension 64×64 and 32×32. We also call the relative quantized vector indices, or codebook, as $\boldsymbol{c}_B \in \{0, K\}^{64\times64}$ and $\boldsymbol{c}_T \in \{0, K\}^{32\times32}$ as shown in Figure 3. We first answer the question of which K, D are better in our small dataset and then we perform a data augmentation and test the impact on the classification task of predicting malignant melanomas.

2.2 Autoregressive models

Besides the analysis of the resulting latent space, we investigated the generative capabilities of the autoregressive model, PixelCNN [23] with self-attention [24], called PixelSNAIL [12]. In this setup the autoregressive model can efficiently model the prior distribution of the latent codes, creating photo-realistic synthetic images. The idea behind the PixelCNN model is to learn the conditional distribution of the given sequence of random variables. When applied on the latent space, the latent codes of the whole image are sorted from top left to bottom right to predict the next code value, which is a discrete probability distribution over the K codes, in an autoregressive fashion. In our example, the autoregressive model learns the join distributions of the latent codes on the top layer and then the distribution of the bottom codes conditioned on the top codes. There are different options to generate new samples once the two models are trained. The main approach

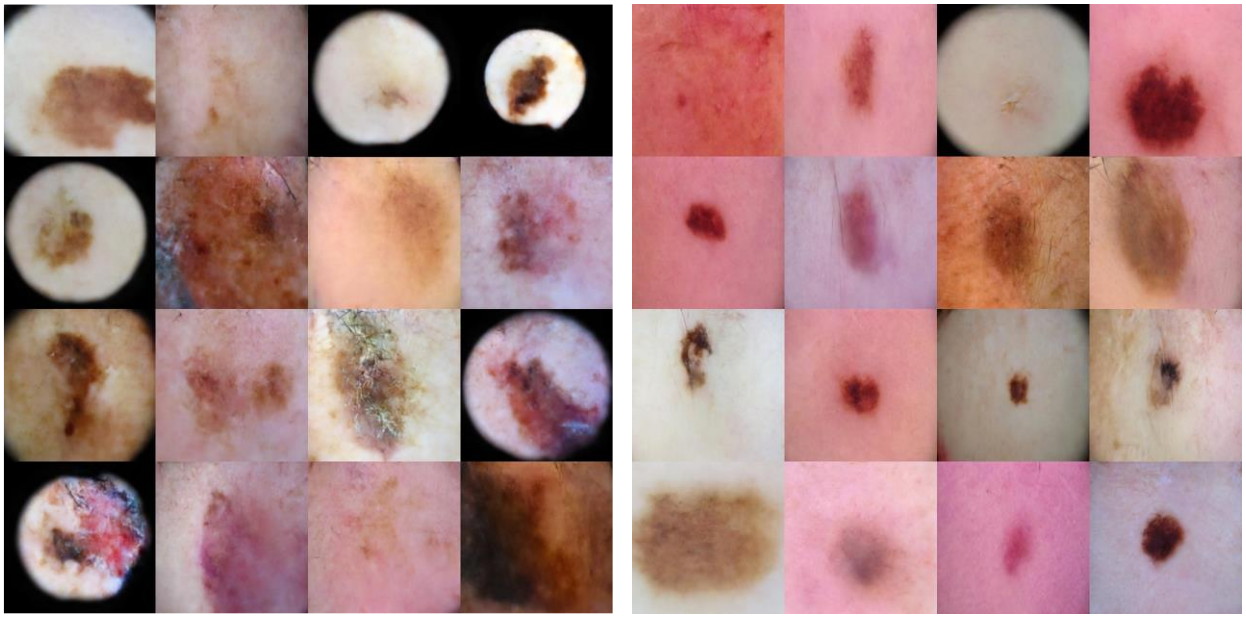

Figure 4. synthetic images generated by training the autoregressive model only on 4922 melanomas on the left and training only on 17685 nevi on the right.

is to perform a sampling of the top space $\boldsymbol{c}_T$ trained on specific image class label, and then sample the bottom space trained on the same label, while conditioning it on the sampled top codes. An example of this approach is shown in Figure 1 where the two autoregressive models are trained on K=256, D=8 with all other hyperparameters equals to the original implementation in [11].

2.3 Generative models applied on skin lesions

Several papers investigated the use of generative models in the context of skin lesions applications. In 2019, Ghorbani et al. [25] used Pix2Pix GAN to synthetize skin condition. Style transfer generative was used to enhance lesion segmentation [26] and other versions of GANs, such as LAPGAN, DDGAN, PPGAN has been tried in [27][28]. While all these examples provide visually appealing results, they suffer from the classical problems of GANs mentioned above. Another similar report of data augmentation [29] tried to hamper this using Self-Attention and PPGAN. In this paper we investigate applications of autoregressive models, in combination with quantization based autoencoders. We moreover investigate how promising is to use such methods to augment skin lesion datasets.

3. DATASET, MATERIAL AND METHODS

In this paper, we investigate the behavior of VQ-VAE-2 in combination with PixelSNAIL, applied to the skin lesion datasets 2020SIIM-ISIC [30], HAM10000 [31], BCN20000 [32] and MSK [33]. The analysis includes an extensive exploration of the impact of the model hyperparameters, as well as experiments to adopt the model as a way to augment the data for downstream tasks. For our experiments, we merge all datasets and then remove all duplicates images by using an EfficientNet pretrained model[1]. The total number of images includes 52302 benign and 4922 malign images (57224 in total). We split the dataset, stratified according to patient, into train (45718) and validation (11506) set to evaluate the reconstruction of images unseen by the network during training. We derive our PyTorch [34] implementation of the VQ-VAE-2 and PixelSNAIL from an openly available project[2]. Since we were not able to find or implement the class conditional sampling suggested in [11] we simply train one autoregressive model for nevi and another for melanoma, while keeping the "true class conditional sampling", with one PixelSNAIL model, for a future work. We present some examples of generated samples in Figure 4, where on the left we used the model trained only on melanomas while on the right it was

[1] https://www.kaggle.com/shonenkov/merge-external-data; https://www.kaggle.com/shonenkov/dbscan-clustering-check-marking
[2] https://github.com/rosinality/vq-vae-2-pytorch

trained only on nevi. The training images were obtained by cropping the center squared region and by scaling them to fit the 256x256 resolution used in our model.

3.1 Prior latent space dimension

In the original setup, the hyperparameters $K = 512$ and $D = 64$ were used for both hierarchical layers trained on ImageNet [19], and the three layer model, trained on FFHQ [20]. Our first research question is what the best configuration of (K, D) is for the latent space, given that we now consider a much smaller dataset with respect to the two mentioned above. To understand the impact of the hyperparameters on the dataset, we did not use data augmentation techniques and worked solely on training data as-is. In our simulations we experimented in reducing the number of vectors K without decreasing the quality of the reconstructed image, in particular, considering that the autoregressive approach is noticeably slow when sampling new images [35], making it hardly scalable for real world applications. This analysis is motivated by the observation that some instances of the network, trained on the natural skin lesion images, resulted in a collapse to a single layer autoencoder. This is shown in the right part of Figure 3, where the reader can see that the top quantized map degenerates to a single code when we used the original configuration $K = 512$ and $D = 64$. We believe that, when reducing K, we also reduce the risk of the code collapse at the top layer which in turn allows for a higher number of code combinations on the top and bottom layers and, therefore, a better sampling and performance of the autoregressive models. This behavior is also confirmed by Table 1, where each row is a trained model and the columns are: the number K of latent vectors, the dimension D of each vector, the number of effectively used vectors in the top hierarchy $|\text{unique}(\boldsymbol{c}_T)|$, the number of effectively used vectors in the bottom hierarchy $|\text{unique}(\boldsymbol{c}_B)|$, the number of cooccurrences of vectors in the top and bottom space $|\text{unique}(\boldsymbol{c}_T, \boldsymbol{c}_B)|$, and lastly the mean squared error for the validation set. It can be seen that, even if the top layer collapses in the model of the first row, it has a competitive MSE for the reconstructed image. We believe this is due to the tradeoff between a large latent space and the relatively small dataset compared to ImageNet. In other words, with some probability the method encodes images without taking advantage of the hierarchical model. Once this is established, we can refrain to use high values of K and D together and gain training and inference speed, which we need for the computationally demanding PixelSNAIL. A benefit for the community using this approach would be to find the optimal latent space dimension, according to their required use, in order to increase throughput speed when coming to the generation part.

Table 1. Report of three different experiments with $K = 512$. Each row is a different model trained with the same exact hyperparameters. The two rows are the latent space dimensions K and D. $|\text{unique}(\boldsymbol{c}_T)|$, $|\text{unique}(\boldsymbol{c}_B)|$ | represent the number of, top and bottom, codes used for encoding the whole dataset while $|\text{unique}(\boldsymbol{c}_T, \boldsymbol{c}_B)|$ are the cooccurrences of used codes. The metrics are the mean squared error for validation set.

K	D	$\lvert\mathbf{c}_T\rvert$	$\lvert\mathbf{c}_B\rvert$	$\lvert(\mathbf{c}_T, \mathbf{c}_B)\rvert$	MSE
512	32	1	512	512	0.0028
512	64	8	512	4095	0.0030
512	64	1	140	140	0.0040

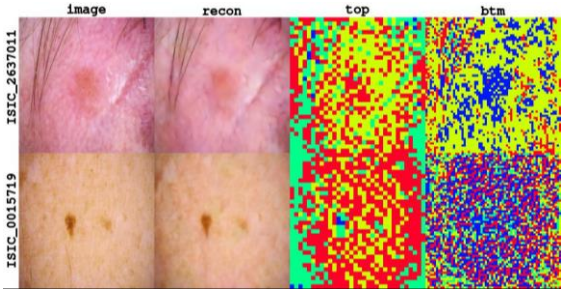

Figure 5. $(K = 4, D = 8)$ — Discrete embeddings for 2 different images. Each row is a different image. The first column is the input image, the second the autoencoder reconstruction. The third column presents the top encoding and the fourth the bottom encoding. Each colour in the third and fourth column represents a different integer code.

To understand better the models, it is relevant to visualize its latent space expression. For example, one can directly spot the richness of patterns (Figure 3 – left), or vice-versa the collapse of one hierarchy, see (Figure 3 – right). It is observed that, sometimes, the model extracts semantic regions, acting as a loosely defined segmentation model. This effect would be particularly interesting when clustering similar codes pattern after training or even during the training phase applying mask to objects of interest. In the following section, we explore the potential of code replacement in the latent representation for the generation of augmented, but realistic, training samples. While the top latent space, which undergo 8x compression of the image into D latent maps tends to encode low frequency information there is not always a clear boundary between it and the bottom space. In fact, as presented in the previous section, sometimes the information is encoded only in the bottom space and other instead use only a part of the top space. A possible way to visualize the codes is by color-coding the input space, but with 256 colors is very difficult that any pattern emerges simply looking at those. A different situation arises when considering very small K. For this reason, we trained also a very small set of VQ-VAE-2 with $K = 4$. While such models do not provide high resolution reconstruction, they are still deeply helpful to test the model behaviors. In Figure 3 and Figure 5 respectively the latent space of three models is visualized. When using a small number of quantizing vectors, it is easier to visualize directly emerging patterns in the code space. Direct manipulation of

the codes in the latent space as a way for creating and modifying lesions is possible. However, it is not the desired approach since complex relationships that arise in the pixel domain due to the interaction of the codes in the latent maps. Intuitively, the codes have an overlapping receptive field in the underlying images. Therefore, the code resulting image patches are not given solely by the corresponding latent codes, but by the interaction between the neighboring ones. In the next section, we provide evidence of this behavior, we present several examples of manipulations of the latent codes and demonstrate that, by using specific manipulations of the codes based on the autoregressive models, it can be used for data augmentations.

3.2 Data augmentation

In this section, we present various techniques to augment the dataset using the VQ-VAE-2 and manipulations of its latent space. The power of having multiple layers in the VQ-VAE-2 architecture, is that one can modify an image by manipulating the latent codes only in one layer, for example the bottom one, retaining the top layer which usually encodes global structure information. Without learning any prior over the latent space one can already compose novel images by "mixing" codes from different images at the cost of losing spatial consistencies. This idea is similar to the pseudo-labeling [36], where to generate new labels and augment the dataset multiple images part are mixed together. An example of mixing skin lesion is presented by Perez et al. [37], where after learning a segmentation model, they mix foreground and background of different lesions. Here, instead of mixing codes based on regions, as shown in [38], we mix them by hierarchy of the learned VQ-VAE-2. This procedure can be achieved in many ways even without the autoregressive model. Given one input image $(\mathbf{c}_{\mathrm{T}}, \boldsymbol{c}_B)$ we replace the bottom codes before reconstruction given another image with the same label, i.e. malignant melanomas or benign lesions. In Figure 6 an example of mixing top and bottom codes is presented without considering the actual ground truth label. In the first row, the top code image is selected to be mixed with the second row, which is the source for the bottom codes. In the third row the mix of the first two rows shows that the global structure is mostly retained from the top source for example looking at the black circular box or the lesion geometry. On the contrary, the high frequencies patterns, like skin hair, are taken from the bottom source image. We highlight such behavior in the second to last bottom row (rng_btm) of the figure where the bottom codes are resampled randomly creating a noisy pattern but maintaining the global structure of the input source top image. On the other side, when the top codes are resampled randomly, as presented in the last row (rng_top) of the figure, the global structure is partially lost while it is easy to see the hair like patterns. It is clear by looking at the bottom two rows that this approach destroys the information and makes the resulting image not useful for any realistic task, but it provides a baseline behavior to compare other augmentations and to understand how much information is lost and which is most relevant for example when classifying lesions.

Another approach is to resample the bottom space by using the autoregressive model. Given an encoded image one can use the real top $\mathbf{c}_{\mathrm{T}}$ while sampling a new bottom encoding $\boldsymbol{c}_B^{new}$ and decoding using the original decoder network. This will lead to the same global low frequency structure while updating high frequencies encoded in the bottom space. An example of such behavior is presented in Figure 7, where two images augmented by resampling multiple times the bottom latent codes without conditioning on the diagnosis. While we expected a more unstable behavior, the change in skin tone does not ruin the quality of the image from an inexperienced human observer perspective. Also, to be observed is that the skin tone is consistent over the whole image for each resampling. This means that the model can represent very long relationships between pixels as also pointed out by the original authors of the VQ-VAE-2 paper. The difference in each row lies in the temperature parameter, which flatten out the estimated probability distribution before sampling dividing the PixelSNAIL decoder's output by it before applying Softmax and sampling according to a multinomial distribution. While in Figure 7 we considered the diagnosis, and the pictures seems to have wide variety on the chrominance, when using models trained only on one particular diagnosis the pattern changes considerably. For example, when using the ABCD rule [39], a commonly used but easy to learn tool to judge lesions according to Asymmetry, Border, Colour and Diameter, multiple changing colors are usually features of Melanomas while a uniform color is characteristic of Nevi. In fact, in Figure 8 the model train only on nevi try to resample the bottom codes to match nevi-like global appearance by giving a uniform skin lesion color. On the contrary, in Figure 9 the model, trained only on Melanomas, prefers darker and changing colors as characteristic of Melanomas. While we appreciate this model behavior, we don't expect that it is what dermatologists want to see but simply observations of a particular dataset coupled with an augmentation technique. Finally, the completely synthetic images when training autoregressive model on a subset of the input data, 4922 melanoma and 17685 nevi, are shown respectively in Figure 4 left and right. We can see that the nevi seem to be sharper in details, and this can be due to the differences in the limited number of samples. Future work can increase the number of images or augment them prior to fitting the autoregressive models.

In the next section, we present preliminary results in which we use augmented data to enrich the training for a downstream task like the classification of skin lesions malignancy.

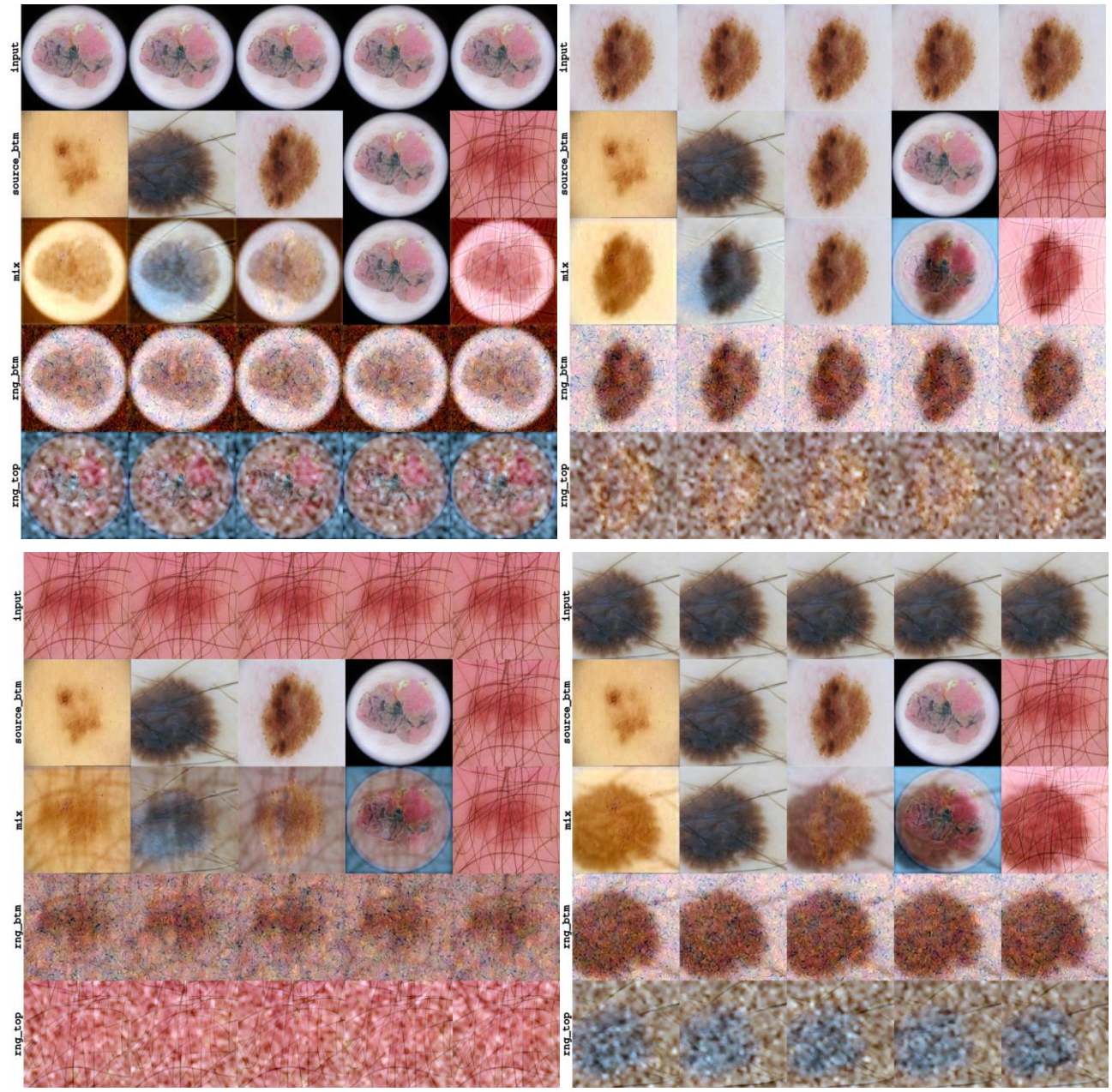

Figure 6. Examples of augmentations by mixing latent codes for the $K = 256, D = 8$ model. The top row represents the input image used to generate the top codes, while the second presents the one used for the bottom code. The third row presents the reconstruction obtained when the two codes are mixed. The bottom row is used for comparison and is created by randomly sampling the bottom codes, showing that a random replacement of the codes is not a viable solution and highlighting the relevance of the interplay between neighboring codes.

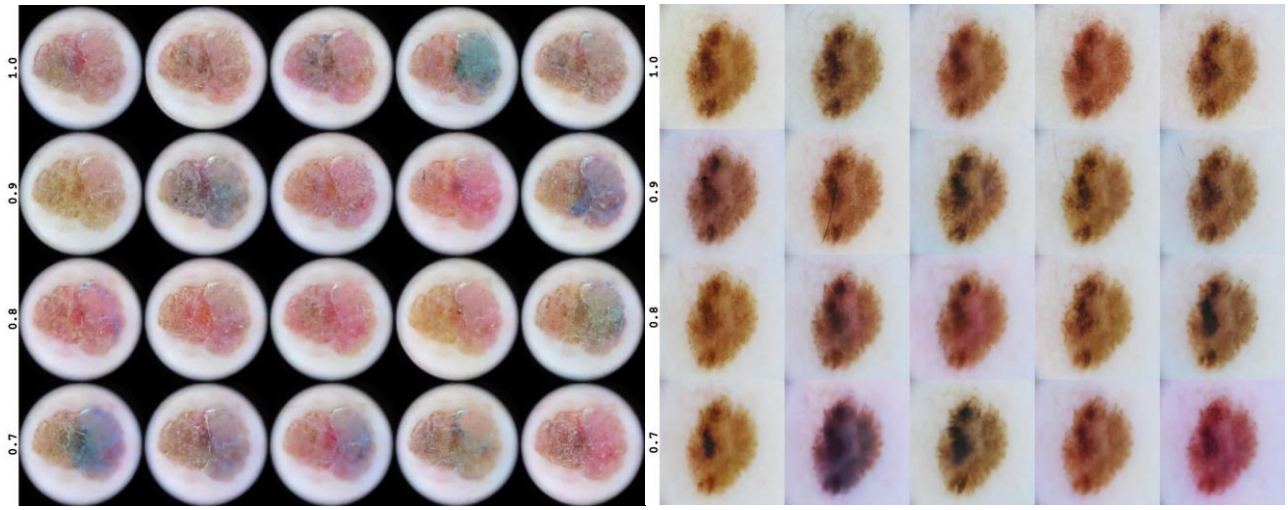

Figure 7. The same image, which original version can be seen in Figure 6, is modified by resampling the bottom codes. From the top row to the bottom row the temperature parameter goes from 1.0 to 0.7. The temperature flattens out the probability distribution before sampling according to a multinomial distribution. A lower temperature allows for more out of distribution, less realistic, samples.

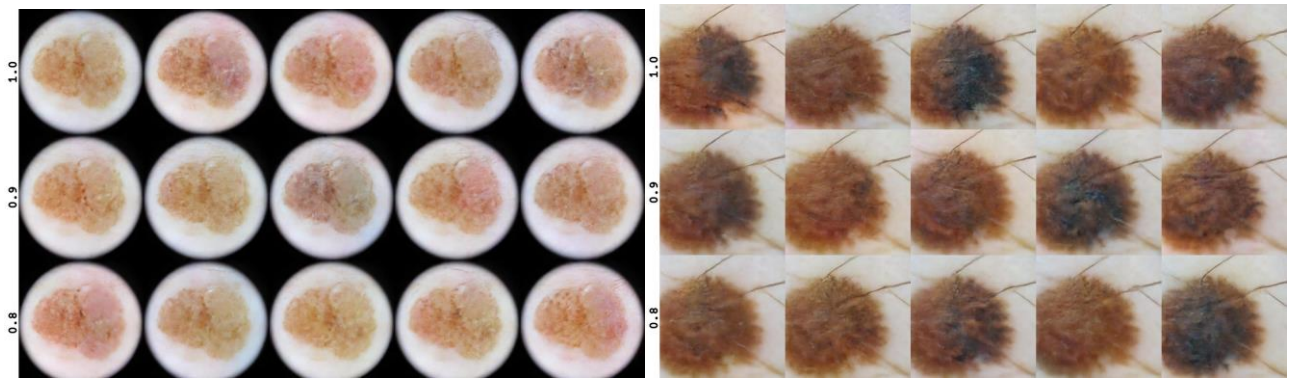

Figure 8. The images, which original version can be seen in Figure 6, are modified by resampling the bottom codes $\boldsymbol{c}_B$ using an autoregressive model trained only on Nevi.

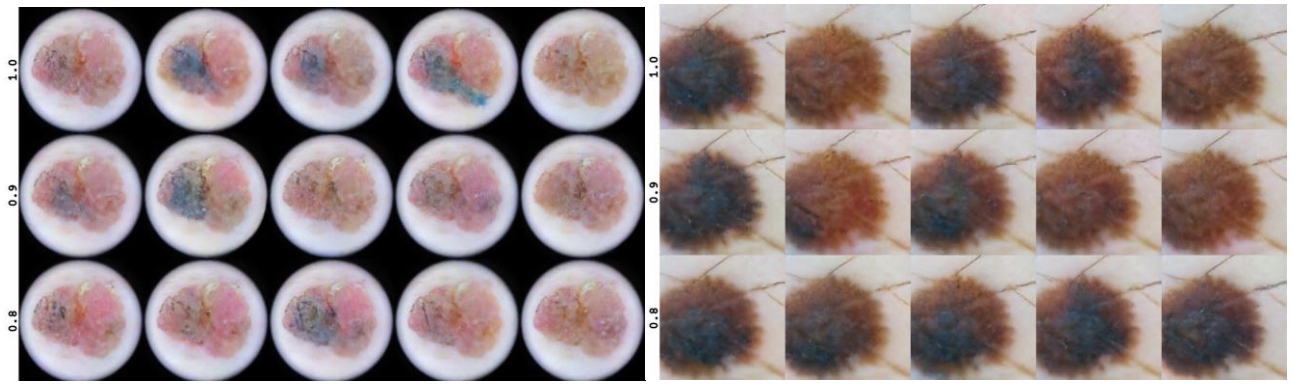

Figure 9. The images, which original version can be seen in Figure 6, are modified by resampling the bottom codes $\boldsymbol{c}_B$ using an autoregressive model trained only on Melanomas.

4. EXPERIMENTS AND QUANTITATIVE EVALUATION

In this section we report quantitative experiments for the respective methods presented above. We first present several results obtained by training various VQ-VAE-2 by changing the number of quantizing vectors and the number of latent codes. We begin by showing that K cannot drop much lower than 256 without impacting the MSE metric for the reconstruction. At the same time, it is difficult to spot big differences by human eyes even when considering only 64 latent quantizing vectors. We then show the results when the augmentations proposed in the previous section are used to train a downstream classification task. The experiments show that the performance of the model is impacted by the use of the synthetic images, hinting at the limitations of the approach. However, by showing that the impact on the performance is limited, we demonstrate the potential in the limited-data regime, where synthetically labeled data can be beneficial.

4.1 Prior latent space dimension

First, we explore the training of VQ-VAE-2. A first relevant question is which configuration of the latent space to select according to target dataset and its use. It is not clear apriori if one wants to use a single layer, two or even three hierarchical layers and what are the pros and cons of such a choice. Moreover, one can increase and decrease the latent dimensions by fixing the number of layers or the number of filters. The objective is to find the optimal values of K and D such that the model is not yet collapsing and giving a visible improvement in MSE. A similar exploration with smaller K, 2 or 4 codework, but with different objectives is carried on in [22] where the authors benchmark different lossy compression proposing a new scheme of Hierarchical Quantized Autoencoders.

The results of this training phase are presented in Table 2. Note that we trained twice the same configuration with 64 latent maps and in one model there was full collapse of the top space while in the other full use. Another relevant result is that, while there is no clear order between D, it seems easier for the model to learn with a small D $(= 2, 8)$. The model trained with $K = 128$ were performing worst in terms of metrics and visual inspection compared to $K = 256$, reaching the maximum valid score of 0.0026. We report here also very small configurations since we will display such models for visualization purposes and, therefore, understanding the behavior of the model in this extreme case. To compute the co-occurrences in column $|\text{unique}(\boldsymbol{c}_T, \boldsymbol{c}_B)|$ we simply up-sampled $\boldsymbol{c}_T$ to have the same resolution as $\boldsymbol{c}_B$, using a nearest neighbor interpolation.

Table 2. Report of different experiments K and D. Each row is a different model trained with the same exact hyperparameters but K and D. $|\text{unique}(\boldsymbol{c}_T)|$, $|\text{unique}(\boldsymbol{c}_B)|$ | represent the number of, top and bottom, codes used for encoding the whole dataset while $|\text{unique}(\boldsymbol{c}_T, \boldsymbol{c}_B)|$ are the cooccurrences of used codes. The metrics are the MSE for validation set.

K	D	$\lvert\mathbf{c}_T\rvert$	$\lvert\mathbf{c}_B\rvert$	$\lvert(\mathbf{c}_T, \mathbf{c}_B)\rvert$	MSE
256	8	256	256	64605	0.0024
512	32	1	512	512	0.0028
256	64	256	256	64687	0.0029
512	8	1	512	512	0.0029
128	8	128	128	16370	0.0030
512	64	8	512	4095	0.0030
256	32	47	256	10664	0.0031
256	64	1	256	256	0.0034
512	64	1	140	140	0.0040
128	64	1	128	128	0.0042
128	32	13	96	1248	0.0047
128	64	9	128	1152	0.0055
4	8	4	4	16	0.0137
4	32	2	4	8	0.0174
4	64	4	4	16	0.0175
4	64	2	4	8	0.0198

By inspecting the reconstruction in Figure 10, we observe that also when using small number of quantizing vectors, e.g. $K = 16$, it is difficult to observe big artifacts and only when reducing markedly the number of quantizing vectors, e.g. $K = 4$, the quality is distorted for this 256×256 resolution. On the contrary, the model is able to find the relevant features even when using few latent maps. Both the visual inspection and the results support this hypothesis; hence, we resolve to select $D = 8$ latent maps and $K = 256$ quantizing vectors as a latent space to fit the autoregressive model.

4.2 Data Augmentations

For the following section we only consider the model with K=256, D=8, we investigate how the model can be used for augmenting and manipulating the data. The model was chosen as a good tradeoff between training stability and richness of representation.

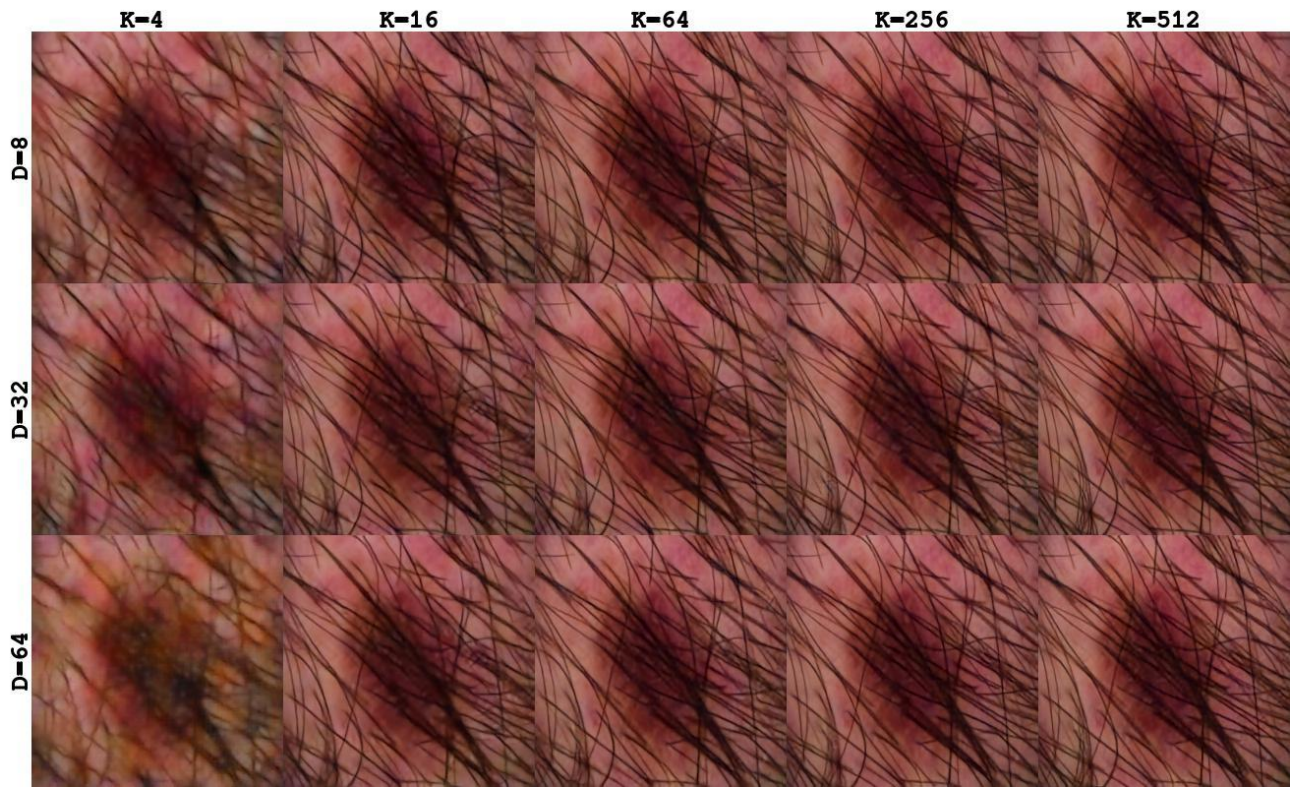

Figure 10. Comparison of VQVAE trained with different of latent codes K. All the rows show the reconstructed image ISIC_0068279.jpg. The visual quality of the reconstruction degrades very slowly, and big artefacts are clear only when using very small K.

In order to evaluate the results, we trained a fast, yet powerful, EfficientNet [40] model pretrained on ImageNet to use it as a scoring function for the augmentations. Similarly to Classification Accuracy Score (CAS) [41] we train the classifier replacing the real dataset with the generated samples and compare the results with the reconstructed version, passing through the autoencoder. The difference between real and reconstructed is what is lost in the lossy compression. Using such approach for scoring proves the quality of the samples directly in a relevant task avoiding the perplexities generated by other metrics. To facilitate and speed up the analysis, particularly when considering the cost of training PixelSNAIL for the generating of new images, we consider only Nevi (17685) and Melanoma (4922). We train the model for 50 epochs with ADAM [42] and learning rate 0.001. Prior to the training phase we split the dataset into train (80%) and test (20%) stratified according to patient. We use, as validation metric, the Area Under the Curve instead of the accuracy as it is also used as the primary metric in the ISIC2020 challenge.

The results, presented in Table 3, shows that there is a loss in performance when no real data is used. More specifically, we observe a small difference of 0.14 points between "real" data and "reconstruction", same images passed through the autoencoder. This shows that, although some reduction in performance is unavoidable due to the introduced domain shift, this is minimal and can open up new applications where data is synthetically generated.

Table 3. Metrics when replacing input dataset with Reconstruction of autoencoder, mixing images with same diagnosis, random bottom, synthetic novel images with autoregressive decoder matching diagnosis, resampled bottom codes according to diagnosis, random top codes.

training set	AUC
real	0.934
reconstruction	0.920
mixing	0.893
synthetic	0.798
resample($\mathbf{c}_B$)	0.759
rand($\mathbf{c}_B$)	0.752
rand($\mathbf{c}_T$)	0.698

When the top and bottom codes are mixed, we see a further drop in performance compared to the reconstructed and real images. The performances of the autoregressive models are lower, and completely "synthetic" images reaches 0.798 AUC, while resampling only low frequencies "resample($\boldsymbol{c}_B$)" achieves 0.759 AUC. The latter, while for human eye seems to produce realistic samples, it performs poorly since it is similar to randomly replacing the bottom codes. The worst result occurs when replacing top codes with random ones hinting that the model takes decisions more on high-frequency details rather than global structures.

5. DISCUSSION AND CONCLUSIONS

There are still many hurdles in generating and controlling high resolution medical images. Overcoming these issues can provide several benefits for training machine learning models, especially when limited labeled training data is available. In this paper we presented VQ-VAE-2 as an alternative to GANs in the context of skin lesion analysis. We provided an exploration of the hyperparameter settings, as well as novel ways to augment the data based on the VQ-VAE-2 model and the manipulation of the latent space, including the use of an autoregressive model. We also showed that the generated images are not competitive on a downstream task, hinting a limitation of the methods driven by the inability to capture fine grained structures in the images and an introduced domain shift.

Several further research directions can be investigated in the future. We believe one interesting direction is to exploit the hierarchical structure of the autoencoder directly to separate patterns. By this, we suggest constraining a quantization only on a particular region of the image, for example, using a segmentation network. In this way the augmentation process will be streamlined and facilitated by selecting and sampling only relevant features for each layer. Moreover, an interesting direction is to modify the loss function used for training, for example by encoding the downstream task directly into the autoencoder training. Finally, we would like the integration of semantic information directly into the training of the model, for example by using unsupervised segmentation models.

To conclude, we explored the use of VQ-VAE-2 to generate skin lesions, performing a detailed analysis of the resulting latent space and performing an extensive hyperparameter analysis. Then, we investigated how an autoregressive model, called PixelSNAIL, can be used to generate synthetics lesions. We presented several possibilities to create these synthetic skin lesions and we evaluated them by training a classifier on real data. The qualitative results prove the methods effective for microscopic skin lesions generation while being relatively easy to train and control. The quantitative results prove the work is promising but cannot outperform, or perform similarly, to classifiers training on real data. However, we believe that our investigation provides relevant information to devise methods to train machine learning model for skin lesions in the low-data regime.

REFERENCES

[1] R. S. Stern, "Prevalence of a history of skin cancer in 2007: results of an incidence-based model," *Arch. Dermatol.*, vol. 146, no. 3, pp. 279–282, 2010.

[2] G. P. Guy Jr, S. R. Machlin, D. U. Ekwueme, and K. R. Yabroff, "Prevalence and Costs of Skin Cancer Treatment in the US, 2002- 2006 and 2007- 2011," *Am. J. Prev. Med.*, vol. 48, no. 2, pp. 183–187, 2015.

[3] R. A. Smith *et al.*, "Cancer screening in the United States, 2018: a review of current American Cancer Society guidelines and current issues in cancer screening," *CA. Cancer J. Clin.*, vol. 68, no. 4, pp. 297–316, 2018.

[4] A. Esteva *et al.*, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, no. 7639, p. 115, 2017.

[5] T. J. Brinker *et al.*, "A convolutional neural network trained with dermoscopic images performed on par with 145 dermatologists in a clinical melanoma image classification task," *Eur. J. Cancer*, vol. 111, pp. 148–154, 2019.

[6] T. J. Brinker *et al.*, "Deep learning outperformed 136 of 157 dermatologists in a head-to-head dermoscopic melanoma image classification task," *Eur. J. Cancer*, vol. 113, pp. 47–54, 2019.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[8] S. Zhou, M. Gordon, R. Krishna, A. Narcomey, L. F. Fei-Fei, and M. Bernstein, "Hype: A benchmark for human eye perceptual evaluation of generative models," in *Advances in Neural Information Processing Systems*, 2019, pp. 3449–3461.

[9] I. Goodfellow, J. Pouget-Abadie, and M. Mirza, "Generative Adversarial Networks," *arXiv Prepr. arXiv ...*, pp. 1–9, 2014, doi: 10.1017/CBO9781139058452.

[10] A. Brock, J. Donahue, and K. Simonyan, "Large scale gan training for high fidelity natural image synthesis," *arXiv Prepr. arXiv1809.11096*, 2018.

[11] A. Razavi, A. van den Oord, and O. Vinyals, “Generating diverse high-fidelity images with vq-vae-2,” in *Advances in Neural Information Processing Systems*, 2019, pp. 14837–14847.

[12] X. Chen, N. Mishra, M. Rohaninejad, and P. Abbeel, “PixelSNAIL: An improved autoregressive generative model,” *arXiv Prepr. arXiv1712.09763*, 2017.

[13] A. van den Oord, O. Vinyals, and others, “Neural discrete representation learning,” in *Advances in Neural Information Processing Systems*, 2017, pp. 6306–6315.

[14] A. Grover, M. Dhar, and S. Ermon, “Flow-gan: Combining maximum likelihood and adversarial learning in generative models,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018, vol. 32, no. 1.

[15] Jia Deng, Wei Dong, R. Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, “ImageNet: A large-scale hierarchical image database,” *2009 IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 248–255, 2009, doi: 10.1109/CVPRW.2009.5206848.

[16] A. D’Amour *et al.*, “Underspecification Presents Challenges for Credibility in Modern Machine Learning,” *arXiv Prepr. arXiv2011.03395*, 2020.

[17] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” *2nd Int. Conf. Learn. Represent. ICLR 2014 - Conf. Track Proc.*, no. Ml, pp. 1–14, 2014.

[18] D. P. Kingma and M. Welling, “An introduction to variational autoencoders,” *arXiv Prepr. arXiv1906.02691*, 2019.

[19] O. Russakovsky *et al.*, “ImageNet Large Scale Visual Recognition Challenge,” *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015, doi: 10.1007/s11263-015-0816-y.

[20] T. Karras, S. Laine, and T. Aila, “A style-based generator architecture for generative adversarial networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 4401–4410.

[21] J. De Fauw, S. Dieleman, and K. Simonyan, “Hierarchical autoregressive image models with auxiliary decoders,” *arXiv Prepr. arXiv1903.04933*, 2019.

[22] W. Williams, S. Ringer, T. Ash, D. MacLeod, J. Dougherty, and J. Hughes, “Hierarchical Quantized Autoencoders,” *Adv. Neural Inf. Process. Syst.*, vol. 33, 2020.

[23] A. den Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, A. Graves, and others, “Conditional image generation with pixelcnn decoders,” in *Advances in neural information processing systems*, 2016, pp. 4790–4798.

[24] A. Vaswani *et al.*, “Attention is all you need,” in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[25] A. Ghorbani, V. Natarajan, D. Coz, and Y. Liu, “DermGAN: Synthetic Generation of Clinical Skin Images with Pathology,” *arXiv Prepr. arXiv1911.08716*, 2019.

[26] Y. Chi, L. Bi, J. Kim, D. Feng, and A. Kumar, “Controlled synthesis of dermoscopic images via a new color labeled generative style transfer network to enhance melanoma segmentation,” in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2018, pp. 2591–2594.

[27] C. Baur, S. Albarqouni, and N. Navab, “MelanoGANs: high resolution skin lesion synthesis with GANs,” *arXiv Prepr. arXiv1804.04338*, 2018.

[28] A. Bissoto, F. Perez, E. Valle, and S. Avila, “Skin lesion synthesis with generative adversarial networks,” in *OR 2.0 Context-Aware Operating Theaters, Computer Assisted Robotic Endoscopy, Clinical Image-Based Procedures, and Skin Image Analysis*, Springer, 2018, pp. 294–302.

[29] I. S. A. Abdelhalim, M. F. Mohamed, and Y. B. Mahdy, “Data augmentation for skin lesion using self-attention based progressive generative adversarial network,” *Expert Syst. Appl.*, vol. 165, p. 113922, 2021.

[30] V. Rotemberg *et al.*, “A Patient-Centric Dataset of Images and Metadata for Identifying Melanomas Using Clinical Context,” *arXiv Prepr. arXiv2008.07360*, 2020.

[31] P. Tschandl, C. Rosendahl, and H. Kittler, “The HAM10000 dataset, a large collection of multi-source

dermatoscopic images of common pigmented skin lesions," *Sci. data*, vol. 5, p. 180161, 2018.

[32] M. Combalia *et al.*, "BCN20000: Dermoscopic lesions in the wild," *arXiv Prepr. arXiv1908.02288*, 2019.

[33] N. Codella *et al.*, "Skin Lesion Analysis Toward Melanoma Detection 2018: A Challenge Hosted by the International Skin Imaging Collaboration (ISIC)," pp. 1–12, 2019, [Online]. Available: http://arxiv.org/abs/1902.03368.

[34] A. Paszke *et al.*, "Automatic differentiation in PyTorch," 2017.

[35] P. Ramachandran *et al.*, "Fast generation for convolutional autoregressive models," *arXiv Prepr. arXiv1704.06001*, 2017.

[36] D.-H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Workshop on challenges in representation learning, ICML*, 2013, vol. 3, no. 2.

[37] F. Perez, C. Vasconcelos, S. Avila, and E. Valle, "Data augmentation for skin lesion analysis," in *OR 2.0 Context-Aware Operating Theaters, Computer Assisted Robotic Endoscopy, Clinical Image-Based Procedures, and Skin Image Analysis*, Springer, 2018, pp. 303–311.

[38] V. Olsson, W. Tranheden, J. Pinto, and L. Svensson, "Classmix: Segmentation-based data augmentation for semi-supervised learning," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 1369–1378.

[39] F. Nachbar *et al.*, "The ABCD rule of dermatoscopy: high prospective value in the diagnosis of doubtful melanocytic skin lesions," *J. Am. Acad. Dermatol.*, vol. 30, no. 4, pp. 551–559, 1994.

[40] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning*, 2019, pp. 6105–6114.

[41] S. Ravuri and O. Vinyals, "Classification accuracy score for conditional generative models," in *Advances in Neural Information Processing Systems*, 2019, pp. 12247–12258.

[42] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv Prepr. arXiv1412.6980*, 2014.