Investigating The Use of VAEs To Generate Images for a Class Imbalanced Dermatoscopic Dataset

Ritesh Ahlawat
Ryerson University
Toronto, Ontario
rahlawat@ryerson.ca

Alif Munim
Ryerson University
Toronto, Ontario
alif.munim@ryerson.ca

Jaculine Medley
Ryerson University
Toronto, Ontario
jmedley@ryerson.ca

Filipe Gorodscy
Ryerson University
Toronto, Ontario
fgorodscy@ryerson.ca

***Abstract*— We propose a method of applying a feature perceptual loss for a Variational Autoencoder (VAE) to a dataset where it generally doesn't work well in. In addition to this, we use a data augmentation approach of generating synthetic data in the scenario mitigating the class imbalance of a classifier. Two model architectures for the classifiers were used - a regular Convolutional Neural Network (CNN) and ResNet - so that a more robust conclusion can be done.**

I. Introduction & Overview

In this research project, Variational Autoencoders (VAE) are leveraged to generate synthetic data as a way to improve performance on a multi-image classification task. To do so, this project analyzes the performance of two different model architectures of classifiers with a class imbalanced dataset: regular Convolutional Neural Network (CNN) and ResNet (with different architectures) - so that a more robust conclusion could be achieved.

Initially, our focus was on generating synthetic audio data, using a Speech Commands dataset (mentioned in the "Problem Statement and Dataset" section). However, when transforming audio data into spectrograms, the images displayed a majority of blank content instead of actual data. This happened due to the variation in the length of the inputted audio files. Therefore, the VAE was not able to be trained properly with the mentioned dataset.

As a consequence of the failed training of the VAE with audio data, our team decided to change the project's focus to generating synthetic medical image data instead of audio data. Therefore, our hypothesis was kept the same while our methodology was tweaked in order to obtain successful training on our artificial neural network architecture.

A. Challenges

The main challenge our team faced in the project was undoubtfully the amount of time a VAE needs to be trained completely. In fact, VAEs are notoriously difficult to be trained in audio data, which was the initial base of the project. Another challenge was the few open-source types of research with respect to using VAEs in a sound dataset. The ones that were found had complicated architecture models which would take a long time to be implemented correctly.

Furthermore, when transforming audio images in spectrograms, several challenges were encountered. For instance, the conversion from audio to image would work, however, the images displayed a large portion of blank content instead of actual data.

To overcome the mentioned issue, we tried cropping the blank content out of the images, nonetheless, the audio files had variational length which would be problematic for our data content as well. For this reason, we decided to change focus and use raw images, with the HAM10000 dataset instead of audio files.

B. Prior Works

VAEs were introduced as a method for ensuring that the latent space has good properties that enable a generative process [2]. Many future works improve upon this model by encouraging disentangled representations [12, 13], using autoregressive models in the decoder network for generation of sharper images [14], and using discrete, rather than continuous representations in the latent space [15].

VAEs have been used in data generation, and in particular, generation of synthetic images in class imbalance scenarios which is especially common in the medical field where there is a significant cost with data acquisition and labeling. Using VAEs alongside data augmentation has been an effective approach in spine ultrasound and brain MRI classification, achieving significant improvements from a baseline model [16].

II. Problem Statement and Dataset

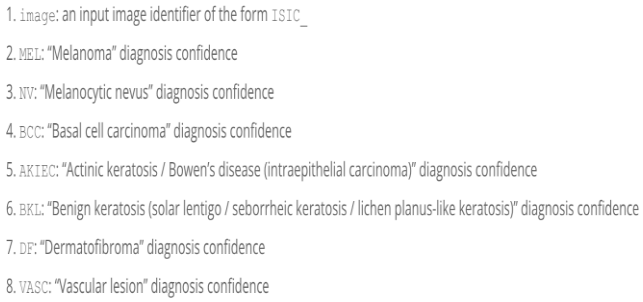
As one might know, datasets with a class imbalance are considerably hard to train. The classifiers' accuracy is highly affected by the classification degree of a particular dataset, and this is the problem our research project aims to solve.

One of the main reasons why imbalanced datasets are difficult to train is the severely skewed class distribution and the unequal misclassification costs. In addition, the difficulty is also compounded by properties such as the dataset size, label noise and data distribution. Therefore, because the class distribution is imbalanced, most machine learning algorithms will require changes and adaptations to avoid predicting the majority class in all cases and will perform poorly if these changes are not applied. In order to remediate this class imbalance problem and improve classification accuracy, we leveraged VAEs to generate enough synthetic data to be inputted in the classifier.

A. *HAM10000 Dataset*

The "Human Against Machine with 10000 training images" (HAM10000) consists of a medical image dataset containing 10015 dermatoscopic images with a dimension of (256, 256) - which are released for academic machine learning research. This dataset was designed to analyze the performance of human expert diagnosis and could be used as a benchmark for comparison between the efficiency of human experts against machines when diagnosing medical images. [10]

The data represents a collection of all important diagnostic categories in the realm of pigmented lesions[1]. More than 50% of the lesions have been confirmed by pathology, while the ground truth for the rest of the cases was either follow-up, expert consensus, or confirmation by in-vivo confocal microscopy.



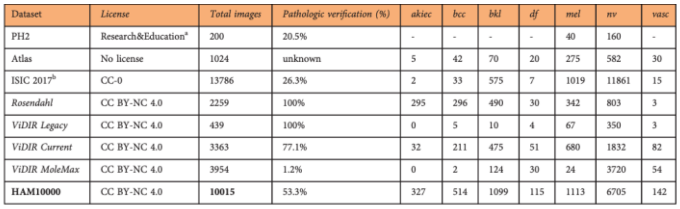
1. image: an input image identifier of the form ISIC_
2. MEL: "Melanoma" diagnosis confidence
3. NV: "Melanocytic nevus" diagnosis confidence
4. BCC: "Basal cell carcinoma" diagnosis confidence
5. AKIEC: "Actinic keratosis / Bowen's disease (intraepithelial carcinoma)" diagnosis confidence
6. BKL: "Benign keratosis (solar lentigo / seborrheic keratosis / lichen planus-like keratosis)" diagnosis confidence
7. DF: "Dermatofibroma" diagnosis confidence
8. VASC: "Vascular lesion" diagnosis confidence

Figure 1 - Diagnostic categories present in the HAM10000 dataset

The diagnosis values are expressed as floating-point numbers in the closed interval from 0.0 to 1.0, where 0.5 is used as the binary classification threshold.

In order to be used by our VAE and classification models, the dimensions of the images were modified to be (224, 224). In addition, some of the minor diagnostic categories were removed from the dataset, those being *akiec, df* and *vasc*. The modified dataset, then, contains only 4 labels - *bcc, bkl, mel* and *nv*. In order to facilitate training and generate even more synthetic data the *mel* and *nv* labels were reduced to contain only around 1000 and 200 images respectively as we use the *mel* class to train the VAE for producing synthetic *mel* data.

Other medical images datasets were considered, but the HAM10000 is the most robust and complete dataset for dermatoscopic images, as shown in the table below[2].



Dataset	*License*	*Total images*	*Pathologic verification (%)*	*akiec*	*bcc*	*bkl*	*df*	*mel*	*nv*	*vasc*
PH2	Research&Education[a]	200	20.5%	-	-	-	-	40	160	-
Atlas	No license	1024	unknown	5	42	70	20	275	582	30
ISIC 2017[b]	CC-0	13786	26.3%	2	33	575	7	1019	11861	15
Rosendahl	CC BY-NC 4.0	2259	100%	295	296	490	30	342	803	3
ViDIR Legacy	CC BY-NC 4.0	439	100%	0	5	10	4	67	350	3
ViDIR Current	CC BY-NC 4.0	3363	77.1%	32	211	475	51	680	1832	82
ViDIR MoleMax	CC BY-NC 4.0	3954	1.2%	0	2	124	30	24	3720	54
HAM10000	CC BY-NC 4.0	**10015**	53.3%	327	514	1099	115	1113	6705	142

Figure 2 - Comparison between other dermatoscopic datasets with HAM10000

It is also worth mentioning the quality of the data provided by this dataset. The images were extracted from PowerPoint slides with python.pptx library. They were, then, passed through a pipeline of quality assurance, where medical experts provided the pathological diagnosis extraction, ground truth annotation and quality review for each of the images. Then, the images were cropped in order to make the lesion centered.

Relying on this medical image dataset was a perfect match for our use case, due to its robustness of data. Our VAE - which could not be trained with an audio dataset - is trained successfully with the HAM10000 dataset. In addition, our two types of classifiers (regular Convolutional Neural Network and ResNet) were also trained successfully on this dataset.

III. Models & Methods

A. *VAE*

VAEs usually consist of two parts, an encoder that allows us to encode an image x to a latent vector $z = encoder(x) \sim q(z|x)$ and a decoder network which is used decode the obtained latent vector z back to an image $\bar{x} = decoder(z) \sim p(x|z)$. To do this, we optimize the lower bound $L\left(\theta, \phi; x^{(i)}\right)$ w.r.t. Both the variational parameters ϕ and the generative parameters θ.

$$\mathcal{L}\left(\theta, \phi; x^{(i)}\right) = \mathcal{L}_{rec} + \mathcal{L}_{KL} \quad (1)$$

$$= -\mathbb{E}_{q_\phi(z|x^{(i)})}[log p_\theta(x^{(i)}|z)] + D_{KL}(q_\phi(z|x^{(i)})||p_\theta(z)) \quad (2)$$

This, however, usually leads to blurry images when compared to natural images. This happens because the pixel-by-pixel loss does not capture the perceptual difference and spatial correlation between the two images [17]. To prevent this, Hou *et al.* [17] presented an alternative high-level perceptual loss to replace $\mathcal{L}_{rec}$. Note that we need not modify $\mathcal{L}_{KL}$ as it is required to make sure that the latent space z is a Gaussian random variable to keep the properties of generation.

Instead of a pixel-by-pixel comparison between x and $\bar{x}$, they are both passed into a pre-trained CNN Φ and the differences between hidden layer features are summed, i.e., $\mathcal{L}_{perc} = \mathcal{L}^1 + \mathcal{L}^2 + \ldots + \mathcal{L}^l$, where $\mathcal{L}^l$ represents the feature loss of the l^{th} layer. The pre-trained CNN used was VGGNet pre-trained on the ImageNet dataset [18, 19]. More formally, let $\Phi(x)^l$ represent the output of the l^{th} hidden layer when an input image x is fed. Since we use the output of convolutional layers as the hidden layers, $\Phi(x)^l$ is a 3D tensor of shape $[W^l\ x\ H^l\ x\ C^l]$, where W^l and H^l are the width and height of each feature map for the l^{th} layer, and C^l is the number of filters. The differences between the hidden layers are defined by squared Euclidean distance.

$$\mathcal{L}^l_{perc} = \frac{1}{2W^lH^lC^l} \sum_{w=1}^{W^l} \sum_{h=1}^{H^l} \sum_{c=1}^{C^l} (\Phi(x)^l_{w,h,c} - \Phi(\bar{x})^l_{w,h,c})^2 \quad (3)$$

$$\mathcal{L}_{perc} = \sum_l \mathcal{L}^l_{perc} \quad (4)$$

The VAE loss is then jointly minimized with $\mathcal{L}_{KL}$ and $\mathcal{L}_{perc}$

$$\mathcal{L}_{VAE} = \alpha \mathcal{L}_{perc} + \beta \mathcal{L}_{KL} \quad (5)$$

where α and β are weighting hyperparameters. Using purely this perceptual loss posed an issue in our case. Since the VGGNet is pre-trained on ImageNet, it stands to reason that a VAE using perceptual loss will be able to produce natural images if and only if the class(es) the VAE is trained on have a high correlation to at least one of the classes of ImageNet. DFC-VAE was trained on the CelebA Dataset [20] which is highly correlated to the *person* class in the ImageNet dataset.

Unfortunately, none of the dermatoscopy classes are highly correlated to any of the classes of ImageNet. For example, the image below is from the *melanoma* classification, when fed to the pre-trained VGGNet, outputs its top-3 predictions as *bolete*, *dough*, and *pomegranate*.

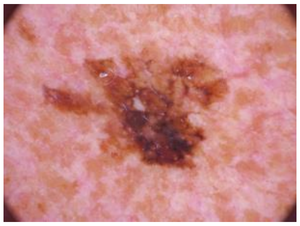
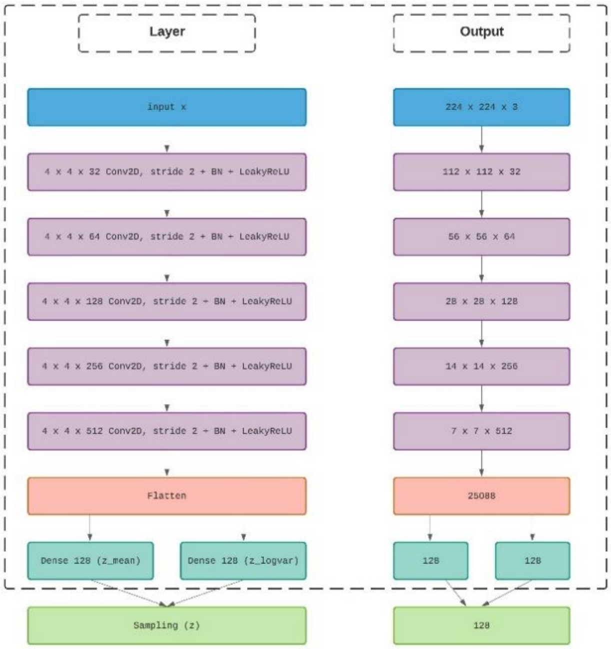


Figure 3 – HAM10000 Dataset melanoma example

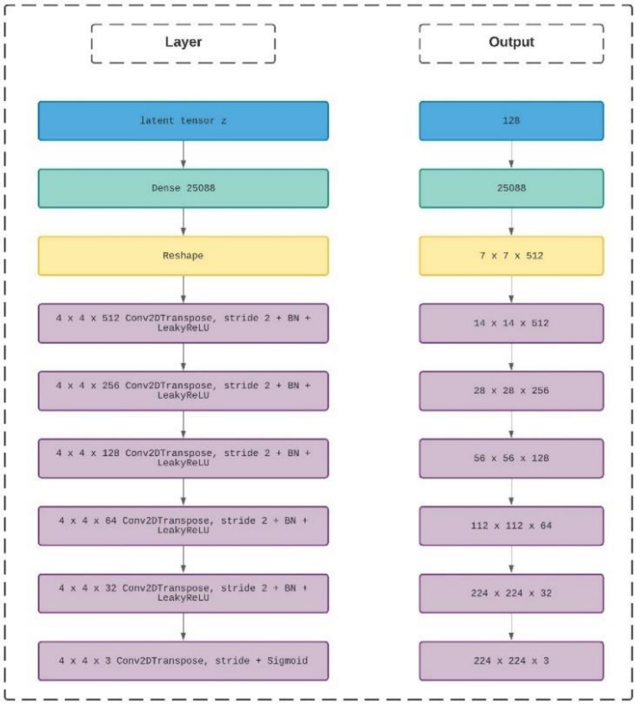
To combat this, we propose an architecture that combines all three losses

$$\mathcal{L}_{VAE} = \gamma\mathcal{L}_{rec} + \beta\mathcal{L}_{perc} + \alpha\mathcal{L}_{KL} \tag{6}$$

where γ, β and α are weighting hyperparameters. The final architecture of our VAE is shown below.


Layer
Output
input x
224 x 224 x 3
4 x 4 x 32 Conv2D, stride 2 + BN + LeakyReLU
112 x 112 x 32
4 x 4 x 64 Conv2D, stride 2 + BN + LeakyReLU
56 x 56 x 64
4 x 4 x 128 Conv2D, stride 2 + BN + LeakyReLU
28 x 28 x 128
4 x 4 x 256 Conv2D, stride 2 + BN + LeakyReLU
14 x 14 x 256
4 x 4 x 512 Conv2D, stride 2 + BN + LeakyReLU
7 x 7 x 512
Flatten
25088
Dense 128 (z_mean)
Dense 128 (z_logvar)
128
128
Sampling (z)
128

a. Encoder network


Layer
Output
latent tensor z
128
Dense 25088
25088
Reshape
7 x 7 x 512
4 x 4 x 512 Conv2DTranspose, stride 2 + BN + LeakyReLU
14 x 14 x 512
4 x 4 x 256 Conv2DTranspose, stride 2 + BN + LeakyReLU
28 x 28 x 256
4 x 4 x 128 Conv2DTranspose, stride 2 + BN + LeakyReLU
56 x 56 x 128
4 x 4 x 64 Conv2DTranspose, stride 2 + BN + LeakyReLU
112 x 112 x 64
4 x 4 x 32 Conv2DTranspose, stride 2 + BN + LeakyReLU
224 x 224 x 32
4 x 4 x 3 Conv2DTranspose, stride + Sigmoid
224 x 224 x 3

b. Decoder network

Figure 4 – VAE network architecture. Top is the encoder network and bottom is the decoder network. Note that we decided to use a transposed convolution instead of upscaling + convolution as described in [17] due to the former having filters that are learnable in the training process.

In all VAE trained, early stopping was used which monitored $\mathcal{L}_{VAE}$ with a patience of 15 epochs and was trained with the Adam optimizer [21]. No other regularization was used.

B. Classifier

Figure 4 - Convolutional Neural Network Architecture

We designed the Convolutional Neural Network classifier model using the Sequential model to accept inputs of shape (224, 224) and the following parameters: kernel = 4, strides = 2, validation split = 0.2, batch size – 48, and epoch = 50.

In the Sequential model, we have a convolutional layer that used the Conv2D keras library with a single filter and reLu as the activation. We then have a flatten layer followed by two Dense layers, one that uses reLu as the activation the other using SoftMax. We compile the model, we used categorical cross entropy as the loss function and Adam as the optimizer. This classifier performed very poorly on the imbalanced dataset.

To achieve more accurate classification results, we decided to employ a widely used CNN architecture known as a residual network, or ResNet. ResNets were introduced in 2015 in a paper titled 'Deep Residual Learning for Image Recognition,' [24] and led to a major breakthrough in the domain of computer vision through the use of skip connections between convolutional layers. These skip connections allowed ResNets to outperform VGG networks, the state of the art during that time, using far fewer parameters.

For our classification task, we leveraged ResNet50, one of the shallower variations on the ResNet architecture, along with pre-trained ImageNet weights. The ImageNet dataset is a large research dataset consisting of 1.4M images and 1000 classes, and the knowledge gained from training a large network like ResNet on this dataset served as a basis for transfer learning.

To adapt the ResNet and pre-trained weights to our dataset, we first added a classification head which consisted of a global average pooling layer appended to a dense layer with a softmax activation function for prediction. To further improve accuracy, we then made the top 100 layers of the model trainable to fine tune the pre-trained weights. We also added a data augmentation layer to the beginning of the network, which introduced random flip, rotation, zoom, and contrast operations to the images to combat overfitting.

IV. Results & Discussion

A. VAE

To generate synthetic data, instead of sampling $z_{sample} \sim \mathcal{N}(0,1)$, we adopt a different strategy, more closely related to data augmentation. For all images in the class that is imbalanced, we feed each image through the encoder of our trained VAE (which was also trained on the same data) to obtain z_{image}. Before passing this tensor to the decoder, we alter it as shown below

$$z_{altered} = z_{image} + \varepsilon \cdot z_{noise} \quad (7)$$

where $z_{noise} \sim \mathcal{N}(0,1)$ and ε is a parameter which regulates how much the generated image "deviates" from the original. Note that having ε too large introduces artifacts into the generated image and having it too low introduces no significant augmentation, thus we restrict $0 < \varepsilon < 0.5$.

As a demonstration to prove that $\beta\mathcal{L}_{perc} + \alpha\mathcal{L}_{KL}$ are not enough for our use case, we train our VAE without $\gamma\mathcal{L}_{rec}$ and see poor results. In all experiments we stick with the previous authors' recommendations and set $\beta = 0.5$ and $\alpha = 1$.

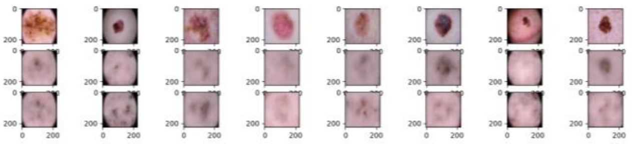


Figure 5 – VAE with latent space $\|z\| = 100$. First row are the original images from the melanoma class. Second row are the reconstructions from our VAE. Third row are the generated reconstructions with $\varepsilon = 0.3$.

Adding back $\gamma\mathcal{L}_{rec}$ yields us more natural reconstructions.

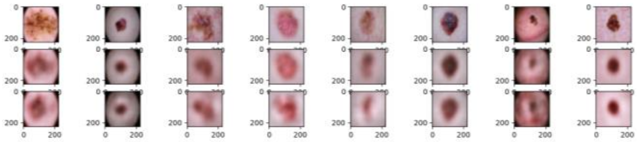


Figure 6 – VAE with latent space $\|z\| = 100$ and $\gamma = \beta = 0.5$. First row are the original images. Second row are reconstructions. Third row are generated reconstructions with $\varepsilon = 0.3$.

We also experiment with different layer combinations of VGGNet to construct the feature perceptual loss. In Figure 6 we used the activations of convolutional blocks 1-2-3's first convolutional layer – relu1_1, relu2_1, and relu3_1. However, we find that also including the last two blocks' activations, relu4_1 and relu5_1, increases the sharpness of the reconstructions.

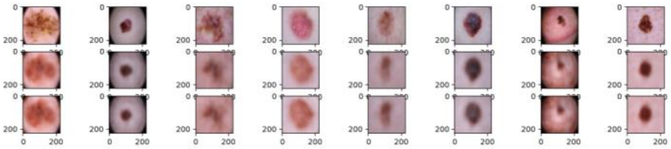


Figure 7 – VAE with latent space $\|z\| = 100$, $\gamma = \beta = 0.5$ and all convolution blocks of VGG used for calculating the perceptual loss. First row are the original images. Second row are reconstructions. Third row are generated reconstructions with $\varepsilon = 0.3$.

Next, we experiment the effects of modifying γ. We vary γ in terms of β and find that reconstruction loss is lowest when $\gamma = \beta$ and keep it as such throughout the rest of the experiments.

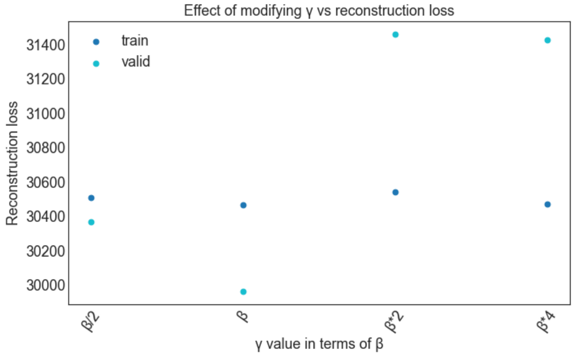
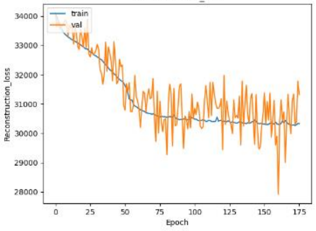
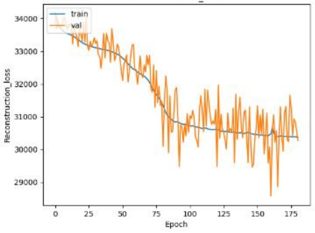


Figure 8 – Scatter plot of VAE's reconstruction loss after training with latent space $\|z\| = 100$ and varying the weight of γ w.r.t β.

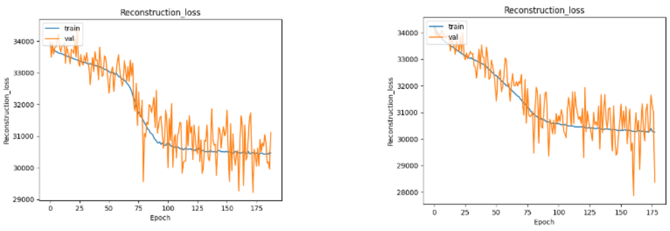
The original authors set $\|z\| = 100$, we vary the size of the latent space z to see its effect on the reconstruction loss.



a. $\|z\| = 32$



b. $\|z\| = 64$



c. $\|z\| = 100$

d. $\|z\| = 128$

Figure 9 – Reconstruction losses of VAE with varying the latent space size.

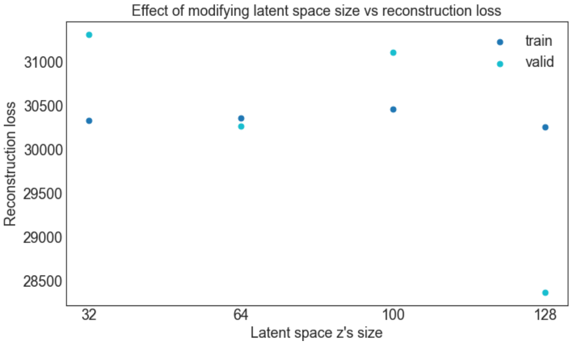


Figure 10 – Reconstruction loss of VAE after training with varying the latent space size.

We note that the validation reconstruction loss is naturally noisy which could be attributed to having a small dataset size which prevents the validation data from being $i.i.d$. However, since the loss curve of the VAE with $\|z\| = 128$ is the smoothest and reaches a lower point, we select this in our final model and present the VAE results below.

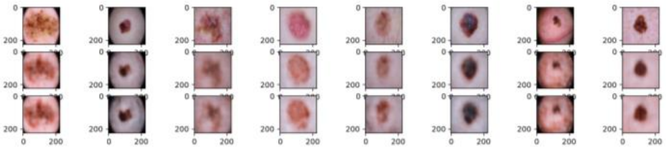


Figure 11 – VAE with latent space $\|z\| = 128, \gamma = \beta = 0.5$ *and convolution blocks of VGG used for calculating the perceptual loss. First row are the original images. Second row are reconstructions. Third row are generated reconstructions with* $\varepsilon = 0.2$.

We use $\varepsilon = 0.2$ for generating synthetic data used in the classifier.

B. *Classifier*

The results for both the learning curves of the loss and accuracy for the training and validation datasets for the Convolutional Neural Network is shown below. From the learning curves for the validation dataset, we see that the dataset is overfitting the model as indicated by how the training loss curve continues to decrease while the validation loss decreases for a period of time and then begins to increase.

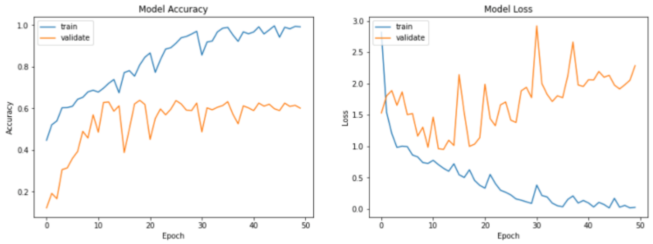


Figure 12 - Plot of the learning curves for loss and accuracy in the training and validation datasets before any tuning to the model

To overcome the overfitting, we added a l1 and l2 regularizer with a learning rate of 1e-5. After which, once the model was tuned, we were able to achieve accuracies in the range of .9 and 1.0 on the validation dataset. The final parameters of the model after tuning were: kernel = 2, strides = 2, validation split = 0.3, batch size = 48, epoch = 50. The final results for the loss and accuracy for both the training and validation datasets are plotted below.

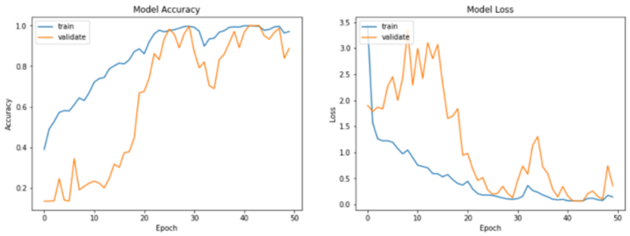


Figure 13 - Plot of the learning curves for loss and accuracy in the training and validation datasets after tuning the model

However, when tested on the imbalanced dataset, this CNN continued to overfit the majority class. In contrast, using the pre-trained ResNet model fine-tuned to the dataset produced far more promising results, even with the imbalanced data. The initial confusion matrix is shown below (fig. 14).

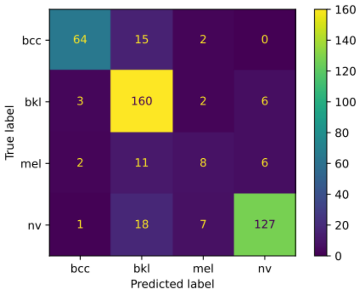
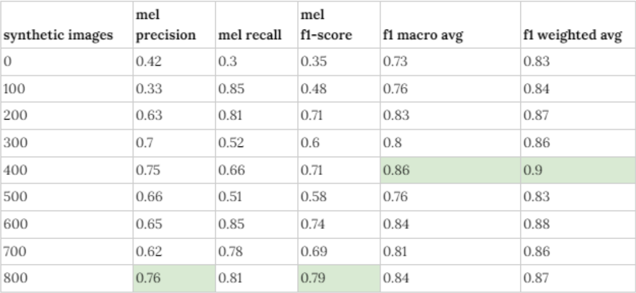


Figure 14 - A confusion matrix of the true and predicted labels for the four test classes. The minority class, melanoma (mel), had the poorest prediction accuracy. However, the remaining three classes were classified quite accurately.

Before recording our results with the ResNet classifier, we first had to decide on the correct performance metric. In scenarios where a class imbalance exists, classifiers will often achieve a high-performance accuracy merely by predicting the majority class and ignoring the minority class altogether. As such, accuracy would not be the ideal metric for our experiment.

For the purposes of our investigation in medical image processing, false positives and false negatives are of significant interest. Both of these should be minimized, as misclassifications can result in unnecessary strain on resources, or in the worst case put a patient's health at risk. Therefore, we focus primarily on the precision, recall, and their harmonic mean, the f1-score, to take these factors into account.

To test our hypothesis, we added synthetically generated images for the minority class in increments of 100 to the fine-tuned ResNet classifier and recorded our results using the aforementioned metrics.



synthetic images	mel precision	mel recall	mel f1-score	f1 macro avg	f1 weighted avg
0	0.42	0.3	0.35	0.73	0.83
100	0.33	0.85	0.48	0.76	0.84
200	0.63	0.81	0.71	0.83	0.87
300	0.7	0.52	0.6	0.8	0.86
400	0.75	0.66	0.71	0.86	0.9
500	0.66	0.51	0.58	0.76	0.83
600	0.65	0.85	0.74	0.84	0.88
700	0.62	0.78	0.69	0.81	0.86
800	0.76	0.81	0.79	0.84	0.87

Figure 15 – Results of the experiment with 200 real images and the number of synthetic images increased by increments of 100. The metrics of interest are the precision, recall, and f1-score of the minority class mel, as well as the f1 macro and weighted average of all the predictions.

The final results of our experiment closely aligned with our hypothesis that synthetically generated data from the VAE would improve the classifier's performance on the classification task. Indeed, the initial precision, recall, and f1-score for the melanoma (mel) class is very low, and continue to improve significantly as synthetic examples are added. Consequently, both the macro average and weighted average for the f1-scores improve as well. Though the results for these metrics are much improved with all 800 synthetically generated images, they peak at 400. It may be the case that this is the ideal distribution for the data, as the other minority class, bcc, has around the same number of images. However, this is an area for future investigation.

V. Implementation and Code

In our implementation, we used several different techniques and methods as explained in detail in the Methods section above. For our reference and baseline, we relied on open sources research using well-developed python packages such as TensorFlow.

One of the main source codes that we based the regular Convolutional Neural Network classifier implementation on was the Jason Brownlee article [7] where the author develops a Convolutional Neural Network from scratch using the CIFAR-10 dataset and Keras python library.

For our ResNet 50 classifier, we used the built-in ResNet model from Tensorflow and adapted the layers and weights based on our dataset dimensions and to improve classification accuracy.

Furthermore, for our VAE, [17] was used as a baseline model from which we improved upon. All papers are cited throughout the paper - where necessary - and in the references section below. All code for this project is available at **https://github.com/alif-munim/dermatoscopic-vae**.

VI. Conclusion

Through our experiments, we were successfully able to demonstrate that with sufficient data, the generative capabilities of variational autoencoders could be leveraged to mitigate class imbalance in a dataset and consequently improve performance on a multi-class image classification task.

For future research along the lines of VAE data generation for class imbalance scenarios, it may be a good idea to investigate the use of more sophisticated and modern classifiers for extracting the perceptual loss or even investigating the implementation of self-attention mechanism to produce a more diverse set of generated images. It may also be worthwhile to investigate the ideal data distribution to be achieved using the synthetically generated examples to reach optimal performance on the overall f1-score metric for the entire dataset.

References

[1] M. Elkaddoury, A. Mahmoudi, and M. M. Himmi, "Deep generative models: Practical comparison between variational autoencoders and generative adversarial networks," Github.io. [Online]. Available: https://indabaxmorocco.github.io/materials/posters/El-Kaddoury1.pdf. [Accessed: 03-Oct-2021].

[2] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," arXiv [stat.ML], 2013.

[3] D. Kunin, J. M. Bloom, A. Goeva, and C. Seed, "Loss landscapes of regularized linear autoencoders," arXiv [cs.LG], 2019.

[4] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," arXiv [cs.CL], 2018.

[5] W.-N. Hsu, Y. Zhang, and J. Glass, "Learning latent representations for speech generation and transformation," arXiv [cs.CL], 2017.

[6] P. Narayanan and Ford Motor Company. n.d. "Deep Generative Modelling for Speech Synthesis and Sensor data Augmentation" Retrieved September 27, 2021, from https://on-demand.gputechconf.com/gtc/2018/presentation/s8617-deep-generative-modelling-for-speech-synthesis-and-sensor-data-augmentation.pdf

[7] J. Brownlee, "How to Develop a CNN from Scratch for CIFAR-10 Photo Classification." Machine Learning Mastery, 27 Aug. 2020, machinelearningmastery.com/how-to-develop-a-cnn-from-scratch-for-cifar-10-photo-classification/

[8] J. Brownlee, “Why Is Imbalanced Classification Difficult?” Machine Learning Mastery, 14 Jan. 2020, machinelearningmastery.com/imbalanced-classification-is-hard/.

[9] “Task 3: Lesion Diagnosis.” ISIC 2018, challenge2018.isic-archive.com/task3/

[10] Mrgrhn. “Resnet with TensorFlow (Transfer Learning).” Medium, The Startup, 3 Feb. 2021, medium.com/swlh/resnet-with-tensorflow-transfer-learning-13ff0773cf0c.

[11] “ResNet 50.” TensorFlow, www.tensorflow.org/api_docs/python/tf/keras/applications/resnet50/ResNet50.

[12] Higgins, I., Matthey, L., Pal, A., Burgess, C.P., Glorot, X., Botvinick, M.M., Mohamed, S., & Lerchner, A. (2017). beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. ICLR.

[13] Burgess, C. P., Higgins, I., Pal, A., Matthey, L., Watters, N., Desjardins, G., & Lerchner, A. (2018). Understanding disentangling in $\beta $-VAE. arXiv preprint arXiv:1804.03599.

[14] Gulrajani, I., Kumar, K., Ahmed, F., Taiga, A. A., Visin, F., Vazquez, D., & Courville, A. (2016). Pixelvae: A latent variable model for natural images. arXiv preprint arXiv:1611.05013.

[15] Oord, A. V. D., Vinyals, O., & Kavukcuoglu, K. (2017). Neural discrete representation learning. arXiv preprint arXiv:1711.00937.

[16] Pesteie, M., Abolmaesumi, P., & Rohling, R. N. (2019). Adaptive augmentation of medical data using independently conditional variational auto-encoders. IEEE transactions on medical imaging, 38(12), 2807-2820.

[17] Hou, X., Shen, L., Sun, K., & Qiu, G. (2017, March). Deep feature consistent variational autoencoder. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)* (pp. 1133-1141). IEEE

[18] J. Deng, W. Dong, R. Socher, L. -J. Li, Kai Li and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248-255, doi: 10.1109/CVPR.2009.5206848.

[19] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[20] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In Proceedings of the IEEE International Conference on Computer Vision, pages 3730–3738, 2015.

[21] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[22] Brownlee, J., 27 Feb, 2019. "How to use Learning Curves to Diagnose Machine Learning Model Performance". Machine Learning Mastery. Retrieved December 8, 2021 from 2019, https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/

[23] Analytics Vidhya. 7 Sept. 2020., "How to Treat Overfitting in Convolutional Neural Networks". Retrieved December 8, 2021, from https://www.analyticsvidhya.com/blog/2020/09/overfitting-in-cnn-show-to-treat-overfitting-in-convolutional-neural-networks/

[24] He, K., Zhang, X., Ren, S., & Sun, J. (2015, December 10). *Deep residual learning for image recognition*. arXiv.org. Retrieved December 13, 2021, from https://arxiv.org/abs/1512.03385.