CS230

Conditional GAN for Dermatology Image Generation

Andrew Cheng ac1792@stanford.edu
Jessica Frank jafrank@stanford.edu
Isabel Gallegos iogalle@stanford.edu

Abstract

Lack of accessible labeled medical images is a major limitation in medical machine learning tasks; however, generative adversarial networks (GANs) have the potential to generate realistic synthetic images that can aid in the training of machine learning models [1]. In this work, we develop a conditional GAN with gradient penalty to generate realistic benign or malignant dermatology skin lesion images and investigate how the generated images can augment training data to improve performance on a downstream classification task. This work expands on existing research on conditional GANs and proposes a novel loss function and model architecture that integrates both segmentation masks and class conditioning to aid in lesion generation. We find that using additional images generated using our GAN improves test AUC from 0.763 to 0.899 on the downstream classification task. These results are promising for work in machine learning-aided diagnosis in dermatology and support the benefits of GANs in data augmentation for biomedical image classification tasks.

1 Introduction

Despite growing interest in deep learning for medical image analysis to aid diagnosis and treatment, the lack of labeled medical imaging data remains a major limitation in the field [1, 2, 3]. In this work, we build a conditional Generated Adversarial Network (GAN) with gradient penalty, where we input a vector of Gaussian random noise and a label of malignant or benign and output a realistic skin lesion image. We explore dermatology skin lesion image generation, investigating how a GAN conditioned on malignancy can improve the quantity and quality of training data for a downstream classification task. We hypothesize that the conditional GAN improves classification performance by providing training data that reflects realistic lesion types. The focus of our project centers on generating realistic medical images that can be used in improving the performance of a classifier, particularly because there is a large data imbalance of many more benign than malignant images.

2 Background and Related Work

GANs have provided an opportunity for impressive development within the field of image generation. There are two main components: the generator, which creates a fake image, and the critic, which attempts to distinguish the real from the fake images [4]. The goal is to produce realistic fake images by maximizing the critic loss and minimizing the generator loss [4].

Previous studies use GANs to generate fake images to varying degrees of success, including within the medical image space. One review article assesses 79 papers to compare their approaches to medical image generation using GANs and outlines numerous network variations – deep convolutional GAN (DCGAN), conditional GAN (cGAN), Markovian GAN (MGAN), CycleGAN, auxiliary classifier GAN (AC-GAN), Wasserstein-GAN (WGAN), and least squares GAN (LSGAN) – which have modifications to the architecture to optimize for different types of input data [5]. Of particular note is the conditional GAN that allows for the development of images conditioned on some input, and in our case, benign or malignant image class. Another important type is the WGAN, which uses the Wasserstein distance to obtain more informative gradients [6] and stabilize GAN training. Arjovsky et al. demonstrate that as long as a 1-Lipschitz constraint on the critic holds, a WGAN will be stable

CS230: Deep Learning, Fall 2022, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

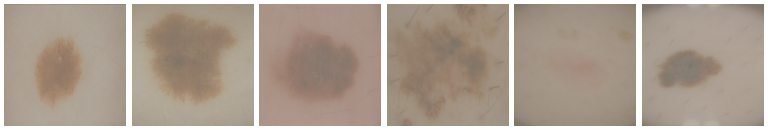

Figure 1: Sample ISIC Images

and will continue to improve as the critic improves [6]. While this WGAN paper uses a hard weight clipping to enforce the 1-Lipschitz constraint, Gulrajani et al. propose an improved WGAN-GP model, showing that a soft gradient penalty term enforces the constraint better in theory and in practice [7].

As model architecture has improved, there has been growing interest in applying GANs to generate medical images. A 2020 study reported using a GAN to generate images of skin lesions to aid in the diagnosis of skin cancer [8]. They reported an increase in classifier accuracy when augmenting the training data with their synthetically generated images [8]. A similar study investigated the use of GANs to generate images to improve the identification of skin cancer and found that the accuracy of a classifier increased by 18% with the addition of the synthetic images [9]. Another study compared various types of GANs for the purpose of medical image generation and found that some architectures perform much better than others [10]. Importantly, this paper also outlines an extensive hyperparameter search, including testing different activation functions, layers, filters, and normalization for the generator and critic, that is used as inspiration for tuning our model [10].

3 Dataset and Features

We use the SIIM-ISIC skin lesion dataset, created by the International Skin Imaging Collaboration (ISIC) [11] (see Figure 1). The ISIC dataset contains 33,126 images of benign and malignant skin lesions, but is highly imbalanced, with only 584 malignant images. Due to the small number of malignant images, we augment the training dataset by rotating each malignant image by 90, 180, and 270 degrees and flipping each image horizontally and vertically. Finally, we randomly upsample the augmented malignant images. This augmentation and upsampling for the training set is important to ensuring that the GAN does not generate solely benign images.

We take equal numbers of benign images as malignant images for training and keep the original imbalanced data distribution for the development and test sets to mirror the data distribution on which the model will typically be evaluated. We split the augmented dataset to have 13,076 training images for each class for a total of 26,152 training images. The development and test sets each have 3,190 images. We feed these images to the model using dataloaders [12, 13]. All ISIC images are resized to 299×299, and images are normalized by the ImageNet statistics [14] before feeding them into the downstream classification model.

4 Methods

4.1 GAN Architecture

For increased training stability and robustness against mode collapse, we follow algorithm 1 from Gulrajani et al. and introduce a soft Lipschitz constraint for the critic loss in the form of gradient penalty for both lesion images and segmented images (see Section 4.1.2) [7]. Our generator is composed of a series of de-convolution blocks followed by a hyperbolic tangent (Tanh) output function and a critic composed of a series of convolution blocks followed by a linear output activation function. We utilize skip connections in both the critic and generator architectures to allow for increased model depth, and we train the critic for multiple steps for every generator step, as suggested by Gulrajani et al. and Arjovsky et al. [6, 7]. In our case, we train the model for 10 epochs and use a noise dimension of 512. We find that for our task, 3 critic updates for every generator update are sufficient to produce high-quality images without sacrificing computation. For the critic of our final model, we transfer up to but not including "layer3" of a ResNet-152 model [15] pre-trained on ImageNet [14] and freeze the weights to use as an initial feature extractor. We follow Gulrajani et al. and use layer normalization instead of batch normalization in the critic [7], but include batch normalization in the generator. For intermediate layers, we use LeakyReLU in the generator and ReLU

in the critic. We heeded advice from several sources to correctly code the GAN [7, 16, 17, 18, 19]. Our code utilizes numerous libraries, including PyTorch [20], sklearn [21], pandas [22], numpy [23], matplotlib [24], PIL [25] and pickle [26] as well as time, tqdm, sys, os, glob, shutil, google.colab, copy, and gdown. See Appendix A for a GAN architecture diagram, including the class conditioning and segmentation modifications described below.

4.1.1 Conditioning on Image Class

To generate class-specific malignant or benign images, we update the GAN to incorporate class labels similar to other conditional GANs [27, 28, 29]. For the generator, we embed the labels with a latent layer of size 512 and then concatenate the embedded layer to the input vector of random noise taken from a normal distribution of mean 0 and variance 1 before passing the resulting vector through the rest of the network. For the critic, we similarly embed the labels with a latent layer of size 512, but we concatenate the embedded labels with the input image later in the network, after the pre-trained layers. We take this strategy because the pre-trained network assumes the input image will have three channels (one for each of red, green, and blue); adding additional channels by concatenating the input image with the embedded labels is not compatible with the transferred layers. Furthermore, the frozen layers expect image features as input, so having additional labels in the channel does not make sense. In the training loop, we pass the true malignant/benign labels to the critic when we are evaluating the critic on the real input image, but we pass uniformly random malignant/benign labels to the critic when we evaluate the generated images.

4.1.2 Enhancing Lesion Generation with Segmentation Masks

We found that the GAN initially generated realistic skin tone images, but had difficulty recreating the lesions in the images. To improve the ability of the GAN to capture the lesion structure, we incorporate segmentation masks into the model, which is a novel feature of our network. We use an existing segmentation model developed by Chen et al. that uses a UNet16 model [30] that was trained and evaluated on ISIC images as part of the ISIC segmentation challenge [31]. This segmentation model generates four different types of masks that segment the pigment network, negative network, streaks, milia-like cysts, and globules that we add together to form one combined mask outlining all of the important features of the image. See Figure 2 for examples of generated images after incorporation of segmentation masks and class conditioning into the GAN.

To incorporate segmentation information into the GAN, we update the loss. We add a term to the generator's loss function that maximizes the critic decision on the segmented fake images (term 2 of Equation 1). This term encourages the generator to focus on creating realistic content within the lesion segmented part of the image, and we find it is successful in promoting realistic lesions. In order to ensure that skin color generation is maintained, we keep the original GAN loss term as well. The modified generator loss given generator G, critic (i.e. discriminator) D, and segmentation model S is:

$$\mathcal{L}_{\mathcal{G}} = -\lambda_1 D(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) - \lambda_2 D(S(\tilde{\mathbf{x}}), \tilde{\mathbf{y}}) \tag{1}$$

where $\tilde{x}$ are the fake images created by our generator, $\tilde{y}$ are malignant/benign labels chosen uniformly at random, and λ_1, λ_2 are the weights associated with the original GAN loss (overall image and skin color) and the focused skin lesion loss, respectively.

We modify the critic loss used by Arjovsky et al. [6] to include segmentation information:

$$\mathcal{L}_{\mathcal{D}} = \lambda_1(D(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) - D(\mathbf{x}, \mathbf{y})) + \lambda_2(D(\mathbf{S}(\tilde{\mathbf{x}}), \tilde{\mathbf{y}}) - D(S(\mathbf{x}), \mathbf{y})) + gp(\hat{\mathbf{x}}, \tilde{\mathbf{x}}, \tilde{\mathbf{y}}) \tag{2}$$

$$gp(\hat{\mathbf{x}}, \bar{\mathbf{x}}, \tilde{\mathbf{y}}) = \alpha(\|\nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}}, \tilde{\mathbf{y}})\|_2 - 1)^2 + \beta(\|\nabla_{\bar{\mathbf{x}}} D(\bar{\mathbf{x}}, \tilde{\mathbf{y}})\|_2 - 1)^2 \tag{3}$$

where $\hat{x} = \epsilon x + (1-\epsilon)\tilde{x}$ and $\bar{x} = \gamma S(x) + (1-\gamma)S(\tilde{x})$ for some $\epsilon, \gamma \sim U[0,1]$, real images x, true class labels y, uniformly random class labels $\tilde{y}$, and hyperparameters $\alpha, \beta \in \mathbb{R}$ weighting the gradient penalty of the critic on normal images and masked images, respectively. The purpose of the second gradient penalty term is to ensure that the segmentation part does not prevent our Lipschitz constraint objective and is motivated by Gulrajani et al. [7]. Our final model uses the following hyperparameter values: $\lambda_1 = 10.0$, $\lambda_2 = 1.0$, $\alpha = 1.0$, and $\beta = 0.1$.

4.2 Experiments to GAN Architecture

To improve generated image quality, we've tried altering the GAN architecture and hyperparameters. Experiments include changing activation functions from ReLU to LeakyReLU in the generator, which

Figure 2: GAN generated images and their corresponding segmentation masks.

helps along with skip connections to prevent vanishing gradients in the final model. Switching the final activation of the GAN from sigmoid to Tanh improves generated image quality. This is counter-intuitive because images we input into the critic are scaled from 0-1, suggesting a sigmoid function would make more sense than a Tanh function. Next, we notice an improvement when we add residual convolution layers to both the generator and the critic because these help to prevent vanishing gradients, particularly when we increase the depth of our generator. The biggest improvement we've observed in generated image quality results from training for more epochs.

Given the importance of having a stronger critic than generator, we use transfer learning in the critic to transfer and freeze up to the third layer of ResNet152 [15] pre-trained on ImageNet [14] and omit the last two layers. Based on a literature search, it appears that including transferred layers in the critic is an uncommon practice, but we were interested in experimenting with transfer learning to strengthen the critic. Following the pre-trained layers, we keep a sequence of trainable convolutional layers with layer normalization and ReLU activation. Strengthening the critic by incorporating transfer learning into the first layers of the network has greatly enhanced the quality of images generated. To experiment with the transfer learning aspect in particular, we have experimented with replacing the pre-trained model's batch normalization with layer normalization because Gulrajani et al. claim batch normalization violates the gradient penalty constraint strategy [7]. We have found that replacing the pre-trained model's batch normalization layers leads to mode collapse. We have also tried unfreezing the first two layers of the pre-trained network after the first training epoch, which results in improved skin color in the final images but a loss of the lesions.

In addition to changing the GAN architecture, we have tested different values for β, which is a hyperparameter that weights the gradient penalty for the segmented images. Increasing β from 0.1 to 0.5 results in mode collapse. We've also tried only training the critic three times every time the generator is trained (as opposed to training the critic 5 times every time the generator is trained) as well as trying different learning rates for the generator and critic, using a learning rate of 0.003 for the critic and 0.001 for the generator, as suggested by Heusel et al. [32]. Most of these changes drastically decrease training time but appear to result in images that look very similar, potentially suggesting mode collapse. We have also tested the incorporation of mixed precision training and find that it speeds up training without visible detrimental changes to the final generated images.

5 Results

5.1 GAN Qualitative Evaluation

To evaluate the GAN, we first qualitatively inspect the images generated by the GAN every epoch and determine if they look like the real images. This qualitative evaluation has been particularly useful in analyzing what changes to the GAN architecture to keep during our experiments, as described above. With our final model, we can see that the GAN generates realistic skin color as well as lesions (see Figure 3). There are some images with artifacts that we have attempted to minimize through our experiments with the GAN architecture; however, the majority of images seem to appear fairly realistic.

5.2 Downstream Classification Evaluation and Baseline

We evaluate the GAN quantitatively by assessing whether generated images improve a downstream classification task. We find that the images enhance classifier performance (see Figure 4). To perform this evaluation, we fine-tune the pre-trained HAM10000 classifier described in Daneshjou et al. [33] and use the classification and evaluation code from this paper's GitHub [34], which classifies lesion images as benign or malignant. Fine-tuning code is based on PyTorch's Finetuning Torchvision Models tutorial [35]. The model is fine-tuned using an Adam optimizer, a learning rate of 0.0001, and a weight decay of 0.001.

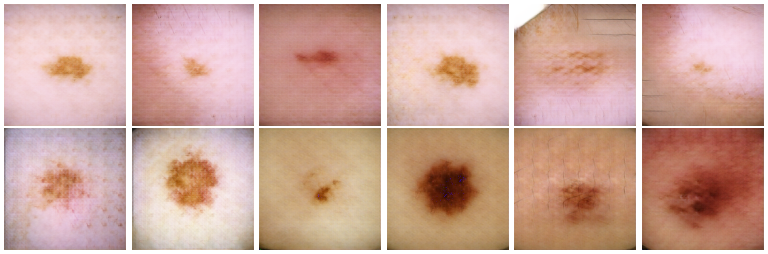

Figure 3: GAN generated benign (top) and malignant (bottom) skin lesion images

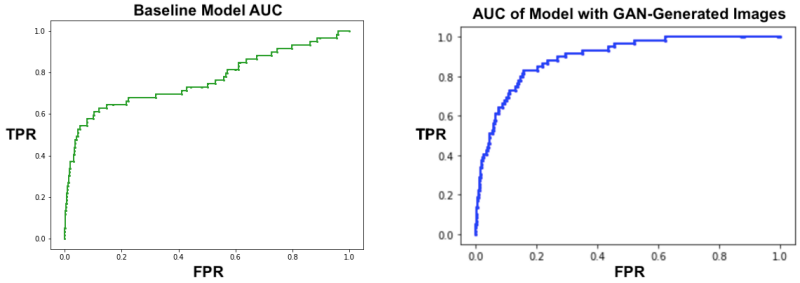

Figure 4: AUC graph for classifying malignant/benign images. Left: AUC of 0.763 for Baseline HAM10000 fine-tuned and evaluated on ISIC. Right: AUC of 0.899 for HAM10000 fine-tuned on ISIC and GAN-generated images and evaluated only on ISIC images.

For the baseline, we train the model on the training ISIC images for 50 epochs with a mini-batch size of 256 and select the checkpoint with the minimum validation loss as the baseline model. The classification model is evaluated using the area under the receiver operating characteristic curve (AUC). No additional data augmentation is applied to the baseline. To evaluate the GAN-generated images, we select the GAN checkpoint after epoch 10 and use it to generate 128 images (with equal numbers of benign and malignant images) and concatenate them with 128 real training images to form a batch of 256 images for every training step. Thus, in total, this version of the model is trained on all of the real ISIC training images in addition to an equal number of generated images. After evaluating both fine-tuned models on the same test set of real ISIC images, our baseline classification model produces a test AUC of 0.763, while the GAN-assisted classification model attains a test AUC of 0.899 – an improvement of 18% compared to baseline (see Figure 4).

6 Discussion and Future Directions

Our combination of a Wasserstein GAN with class conditioning, segmentation masks, and transfer learning provides a novel architecture for improved synthetic image generation. Basing our model off of a Wasserstein GAN aids with improved training stability. The incorporation of transfer learning strengthens the critic and helps to generate more realistic skin color in images after a shorter number of training epochs. The addition of class conditioning promotes the creation of images with distinct benign and malignant lesion morphology and structure. Furthermore, the integration of segmentation masks into the GAN encourages increased content generation within the lesion, transitioning the network from creating images of uniform skin to images of skin containing realistic lesions. If provided additional time and compute resources, we would further fine-tune the hyperparameters in our network to balance the weight placed on the segmented images within the loss function.

We find that GAN-generated images improve the performance of a classifier by 18%, which supports how the creation of realistic synthetic medical images provides opportunities for machine learning development, particularly when there exists limited labeled real data. Future extensions include generating synthetic images with diverse skin tones. Our proposed combination of machine learning techniques in developing a GAN model provides a foundation for a novel architecture that in the future can be applied to additional biomedical datasets.

7 Contributions

All group members contributed to the project and report. Andrew developed the core architecture for the GAN and the loss function modifications and fine-tuned the HAM10000 classifier with the GAN-generated images. Jessica integrated class conditioning into the GAN, set up the segmentation model to generate masks of the images, and tested optimizations to the GAN architecture. Isabel integrated the segmentation masks into the GAN, augmented the training dataset, and established and fine-tuned the HAM10000 classifier for the baseline metrics.

References

[1] August DuMont Schütte, Jürgen Hetzel, Sergios Gatidis, Tobias Hepp, Benedikt Dietz, Stefan Bauer, and Patrick Schwab. Overcoming barriers to data sharing with medical image generation: a comprehensive evaluation. *npj Digital Medicine*, 4(1):1–14, September 2021. Number: 1 Publisher: Nature Publishing Group.

[2] Johann Li, Guangming Zhu, Cong Hua, Mingtao Feng, BasheerBennamoun, Ping Li, Xiaoyuan Lu, Juan Song, Peiyi Shen, Xu Xu, Lin Mei, Liang Zhang, Syed Afaq Ali Shah, and Mohammed Bennamoun. A systematic collection of medical image datasets for deep learning. *arXiv*, 2021.

[3] Cho Young-Won Lee Hyunna Kim Guk Bae Seo Joon Beom Kim Namkug Lee June-Goo, Jun Sanghoon. Deep learning in medical imaging: General overview. *kjr*, 18(4):570–584, 2017.

[4] Alankrita Aggarwal, Mamta Mittal, and Gopi Battineni. Generative adversarial network: An overview of theory and applications. *International Journal of Information Management Data Insights*, 1(1):100004, 2021.

[5] Salome Kazeminia, Christoph Baur, Arjan Kuijper, Bram van Ginneken, Nassir Navab, Shadi Albarqouni, and Anirban Mukhopadhyay. Gans for medical image analysis. *Artificial Intelligence in Medicine*, 109:101938, 2020.

[6] Martín Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *ArXiv*, abs/1701.07875, 2017.

[7] Ishaan Gulrajani, Faruk Ahmed, Martín Arjovsky, Vincent Dumoulin, and Aaron C. Courville. Improved training of wasserstein gans. In *NIPS*, 2017.

[8] Zhiwei Qin, Zhao Liu, Ping Zhu, and Yongbo Xue. A gan-based image synthesis method for skin lesion classification. *Computer Methods and Programs in Biomedicine*, 195:105568, 2020.

[9] Pooyan Sedigh, Rasoul Sadeghian, and Mehdi Tale Masouleh. Generating synthetic medical images by using gan to improve cnn performance in skin cancer classification. In *2019 7th International Conference on Robotics and Mechatronics (ICRoM)*, pages 497–502, 2019.

[10] Youssef Skandarani, Pierre-Marc Jodoin, and Alain Lalande. Gans for medical image synthesis: An empirical study. *arXiv*, 2021.

[11] Veronica Rotemberg, Nicholas Kurtansky, Brigid Betz-Stablein, Liam Caffery, Emmanouil Chousakos, Noel Codella, Marc Combalia, Stephen Dusza, Pascale Guitera, David Gutman, et al. A patient-centric dataset of images and metadata for identifying melanomas using clinical context. *Scientific data*, 8(1):1–8, 2021.

[12] Akshaj Verma. Pytorch [tabular]-binary classification. `https://towardsdatascience.com/pytorch-tabular-binary-classification-a0368da5bb89`, Oct 2021.

[13] Sasank Chilamkurthy. Writing custom datasets, dataloaders and transforms. `https://pytorch.org/tutorials/beginner/data_loading_tutorial.html`.

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[15] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2015.

[16] u7javed. Conditional-wgan-gp/train.py at master · u7javed/conditional-wgan-gp. https://github.com/u7javed/Conditional-WGAN-GP/blob/master/train.py.

[17] Caogang. Wgan-gp/gan_cifar10.py at master · caogang/wgan-gp. https://github.com/caogang/wgan-gp/blob/master/gan_cifar10.py.

[18] Wgan implementation from scratch (with gradient penalty). https://www.youtube.com/watch?v=pG0QZ7OddX4, Nov 2020.

[19] user118967user118967 4 and IvanIvan 30.2k77 gold badges4848 silver badges8888 bronze badges. Meaning of grad_outputs in pytorch's torch.autograd.grad. https://stackoverflow.com/questions/68778401/meaning-of-grad-outputs-in-pytorchs-torch-autograd-grad, Oct 1968.

[20] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[22] The pandas development team. pandas-dev/pandas: Pandas, December 2022.

[23] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[24] John D Hunter. Matplotlib: A 2d graphics environment. *Computing in science & engineering*, 9(3):90–95, 2007.

[25] P Umesh. Image processing in python. *CSI Communications*, 23, 2012.

[26] Guido Van Rossum. *The Python Library Reference, release 3.8.2*. Python Software Foundation, 2020.

[27] Jason Brownlee. How to develop a conditional gan (cgan) from scratch. https://machinelearningmastery.com/how-to-develop-a-conditional-generative-adversarial-network-from-scratch/, Sep 2020.

[28] Aditya Sharma. Conditional gan (cgan) in pytorch and tensorflow. https://learnopencv.com/conditional-gan-cgan-in-pytorch-and-tensorflow/, Dec 2022.

[29] Neeraj Varshney. Step by step implementation of conditional generative adversarial networks. https://medium.com/analytics-vidhya/step-by-step-implementation-of-conditional-generative-adversarial-networks-54e4b47497d6, Dec 2020.

[30] BloodAxe. Segmentation-networks-benchmark/unet16.py at master · bloodaxe/segmentation-networks-benchmark. https://github.com/BloodAxe/segmentation-networks-benchmark/blob/master/lib/models/unet16.py, Jun 2018.

[31] Eric Z. Chen, Xu Dong, Junyan Wu, Hongda Jiang, Xiaoxiao Li, and Ruichen Rong. Lesion attributes segmentation for melanoma detection with deep learning. *bioRxiv*, 2018.

[32] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Günter Klambauer, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a nash equilibrium. *CoRR*, abs/1706.08500, 2017.

[33] Roxana Daneshjou, Kailas Vodrahalli, Roberto A. Novoa, Melissa Jenkins, Weixin Liang, Veronica Rotemberg, Justin Ko, Susan M. Swetter, Elizabeth E. Bailey, Olivier Gevaert, Pritam Mukherjee, Michelle Phung, Kiana Yekrang, Bradley Fong, Rachna Sahasrabudhe, Johan A. C. Allerup, Utako Okata-Karigane, James Zou, and Albert S. Chiou. Disparities in dermatology ai performance on a diverse, curated clinical image set. *Science Advances*, 8(32):eabq6147, 2022.

[34] DDI-Dataset. Ddi-dataset/ddi-code. `https://github.com/DDI-Dataset/DDI-Code`.

[35] Nathan Inkawhich. Finetuning torchvision models. `https://pytorch.org/tutorials/beginner/finetuning_torchvision_models_tutorial.html`, 2017.

Appendix

A GAN Architecture Supplemental

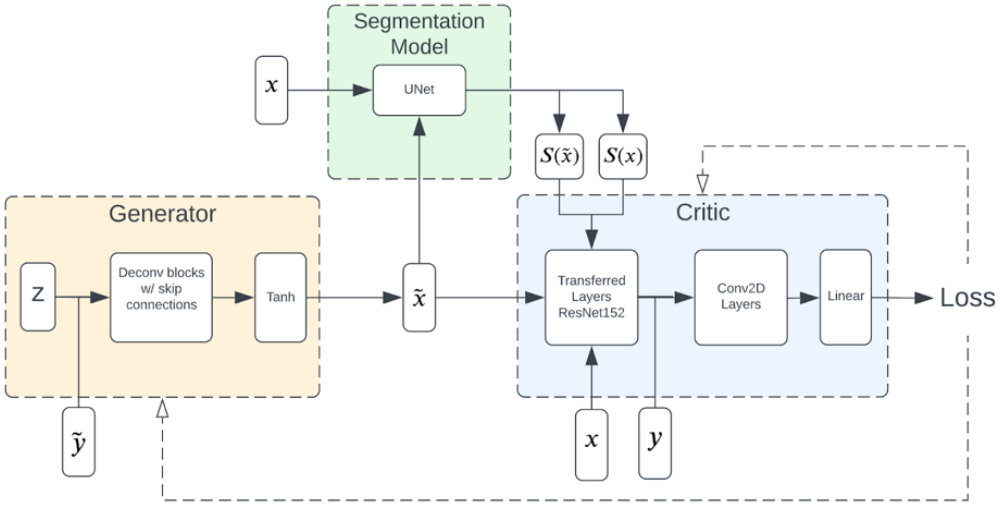

Figure 5: Diagram of the GAN architecture, including the modifications to include class conditioning and segmentation masks.