Final Report: Analyzing Melanoma Traits Through Diffusion and GAN Models

Howell Xia, Diegonisio D'Angelo-Cosme, Hongzun (Amy) Zhang, Ivan Garcia
{howellx,diegodc,hongzun,fayker}@bu.edu

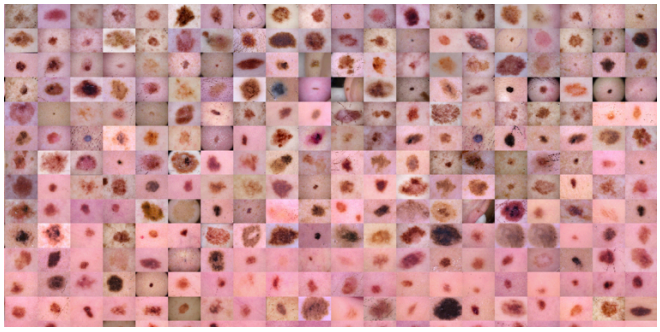


Figure 1. Portion of the ISIC Dataset

1. Task

Our task is to analyze melanoma trends in dermatology models by using a conditional diffusion model to generate counterfactual images. By altering attributes, we aim to uncover what features are exemplified by our model to highlight melanotic attributes. Additionally, we want to evaluate the efficacy of our diffusion model's conditional image generation compared to a reference generative adversarial network (GAN) model through the use of a pre-trained classifier.

2. Related Work

Qiangguo et al. (2021) focused on applying CKDNet's feature entanglement to aggregate task-specific features, directing the model's attention towards disease relevant areas. This method reduces the model's reliance on irrelevant cues, which will provide a much clearer focus on significant features thus improving results across diverse patient groups [1].

Recent work on improving skin lesion diagnosis models has utilized generative AI to reveal key decision-making features. De Grave et al. (2023) trained a GAN model, where the output of a classifier of interest was used as a condition during training. They then used the GAN to generate counterfactual images based on an input image with target labels. Professional physicians then analyzed these counterfactual images for attribute differences and enabled statistical analysis on which image features influenced the images the most [2]. De Grave et al. (2023) was the basis for our project.

3. Approach

Our reference article used generative AI to create altered versions of the input images (counterfactual images) and evaluate how the models behave. To begin, we utilized their GAN models to create a baseline for this research and initialize the working environment with the necessary functions and workflows. This provided a structured starting point for our project, ensuring that the essential components are in place before actuating our own model. In order to implement this, we utilized the SCC environment.

For our research, we followed a similar approach but instead generated the counterfactual images via a conditional diffusion model. We used a contextual UNET model [3] for our diffusion model and we will analyze the DeepDerm classifier model [4] from the paper, whose weights are publicly available.

A. The Dataset

The International Skin Imaging Collaboration (ISIC) dataset [5] is a publicly accessible archive of thousands of images with the goal of improving skin cancer diagnosis through sharing and combining dermatologic images. This is a dataset commonly used for medical and machine learning tasks and is a source for many other major datasets such as HAM10000 and BCN20000.

The ISIC dataset includes various collections of high quality, metadata annotated dermoscopic images. Some of the metadata provided includes the anatomical location of the designated area, lesion diagnosis, and whether it is benign or malignant.

In our testing, our training set consisted of over 20,000 images from the melanoma section of the ISIC dataset. Each image was given a label of 0 for benign or 1 for malignant and was saved in a ground truth file for training and testing. There was, however, an imbalance within the dataset, with the ratio of benign to malignant images being approximately 5:1. This was due to the fact that the ISIC dataset uses various, more specific labels for its images, but we decided to keep the classification simple by only focusing on benign and malignant labels. Despite this, the model was

trained normally without data augmentation. The images within the dataset contained distinct features such as mole size, color, shape, skin tone and quality, and hair presence. The diversity of the input images helped to create a more generalized model to perform well for unique inputs with these various features.

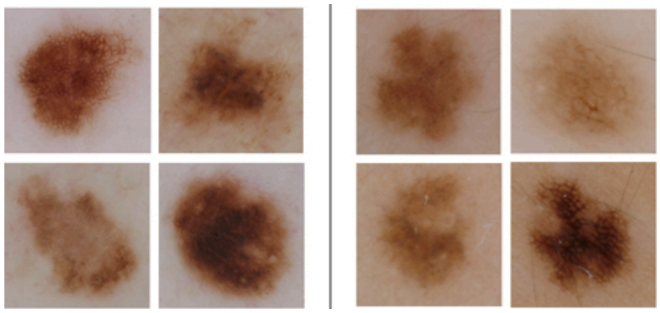


Figure 2: Sample of Images of the ISIC Dataset with Melanoma on the Left and Benign on the Right

As an input, we trained the diffusion model to be able to add noise and denoise our image dataset. The images were scaled to be 256x256 in order to be more compatible with the classifier. We attempted to generate images by first adding noise to input images, then denoising the images with the classifier guiding the denoising process. The classifier-guided model attempted to denoise the anchor images by using an adversarial label: the opposite diagnosis label as the original input image (assigning a benign label for a malignant diagnosis and vice versa). This resulted in an altered version of the input image with the opposite classifier label. This denoising process was repeated but with the original label maintained. This was done so that we could observe what features were accentuated when generating a benign or malignant image.

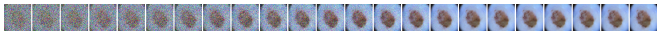


Figure 3: Denoising an Anchor Image

Over 1,000 counterfactuals were generated and were then passed through the DeepDerm classifier in order to quantify the quality of the denoised images. The DeepDerm classifier is a publicly available skin lesion classification system used to help detect skin cancer. It utilizes input images of size 299x299 and returns a prediction within the range of 0 to 1. The classifier established a threshold value of 0.687 for determining the label of an image. If a prediction is above the threshold, the classifier considers the image malignant, and the image is considered benign if it is below the threshold. The final output of each image showcased the anchor image, the benign counterfactual, and the malignant counterfactual with the DeepDerm predictions stored to a file and displayed in the title of the output image.

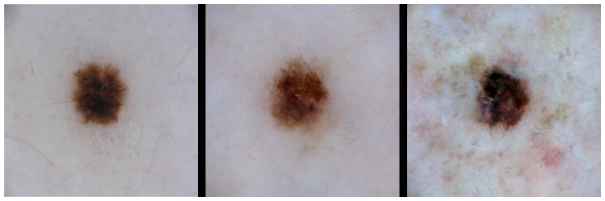


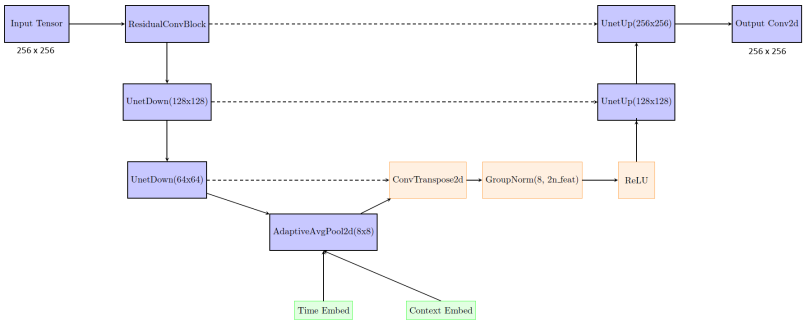
Figure 4: Example Output Image with Title: combined_00001_0.0_0.447_0.334_0.785.png

B1. The Model

The diffusion training was guided using the ground truth labels as opposed to a classifier output. This approach simplified the process by allowing the model to learn directly from the provided labels instead of relying on an intermediate classifier.

The diffusion model utilizes a Contextual UNet which is an encoder-decoder structure that uses residual blocks and conditional mechanisms in order to do class-conditional image generation.

Figure 5: Contextual UNet architecture



The UNet takes an input image resolution 256x256 and then proceeds to downscale the image's spatial resolution through convolutional layers and then restore it in the upscaling stage, but now also incorporating the conditional embeddings for the class label and timesteps.

The architecture is made up of many Residual Convolution Blocks (RCB) which each contain 2 convolutional layers with GELU activations while also performing batch normalization. They also include residual connections which combine the input with the processed output when upsampling. The UnetDown consists of a RCB followed by a MaxPool2d in order to downscale the image resolution. The AdaptiveAvgPool reduces the image resolution to 8 x 8 followed by a GELU activation function.

Once the downsampling is finished, it is followed by a class that converts our binary labels into one-hot encoding. This one-hot encoding of the

classes is then passed through contextual embedding blocks which are made up of 2 fully connected linear layers with a GELU activation function in between. This converts the classes into usable information for the main network, assisting in learning more complex patterns for each class.

This is then followed by the upsampling portion of the UNet. The first upsampling block consists of a ConvTranspose2d layer which increases the spatial resolution of the input tensor back to 64 x 64. It is then followed by a GroupNorm for stabilization across all the channels and utilizes a ReLU as an activation function. The two UnetUp blocks first concatenate the residuals from the downsampling and utilize a ConvTranspose2d layer followed by 2 RCBs in order to restore the spatial resolution to 256x256. The input channels for the UnetUp blocks are also multiplied by the conditional embedding and summed with the time embedding. This is done to modulate our feature map based on class information while also adding timestep information, guiding the denoising process according to the current diffusion step.

Our output layer then generates the denoised image. This final layer is made up of a convolutional layer, followed by a GroupNorm, a ReLU activation function, and then tops it off with a final convolutional layer [6].

B2. Model Training

When training, the diffusion model continues to add noise to input images and then learns to reverse this process. The reverse diffusion portion allows for the generation of the images based on the conditioned class labels. This noise addition is handled by a noise scheduler which determines how Gaussian noise is added during the forward diffusion process and removed during the reverse process. The noise scheduler for the forward process is written as:

$$x_t = \sqrt{\bar{\alpha}_t} \cdot x_0 + \sqrt{1 - \bar{\alpha}_t} \cdot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I)$$

Equation 1: Forward noise scheduler formula

Two key properties of the scheduler is the noise variance β_t and the product of the noise schedules $\bar{\alpha}_t$. β_t is increases linearly from 10^{-4} and increases until it reaches 0.02. This gradual noise addition is to ensure that the training is stable through every timestep and results in better learning with regards to reversing the noise. The cumulative product of $\bar{\alpha}_t$ serves to determine how much of the original signal should remain at each timestep t.

The model also attempts to minimize the MSE between the actual noise ϵ and the predicted noise/ By minimizing this loss, the model is able to learn how to reverse the forward diffusion process every single time step in the reverse process.

$$L = \mathbb{E}_{t,x_t,\epsilon}\left[\|\epsilon - \epsilon_\theta(x_t, t, c)\|^2\right]$$

Equation 2: MSE loss formula

B3. Sampling Process

During the sampling process, the reverse diffusion procedure removes noise in order to generate counterfactual images. The denoising operation is written as follows:

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \cdot \epsilon_\theta(x_t, t, c)\right) + \sqrt{\beta_t} \cdot z, \quad z \sim \mathcal{N}(0, I)$$

Equation 3: Reverse diffusion formula

At each time step t, the model uses a noisy image input to compute a less noisy version of the input. It repeats this denoising operation until the model reconstructs a clean and conditioned image.

In order to form our counterfactual images, the diffusion process is performed twice on the given anchor image. It forms a benign counterfactual by conditioning the diffusion process on the benign class label (0) which guides the model to generate an image with more prominent benign features learned from the dataset. The malignant counterfactual is very similarly done, but instead conditions it with the malignant class label (1). The quality of these counterfactuals can then be assessed through classifier evaluation methods and image generative model evaluation methods in order to grade the diffusion model's performance.

4. Metrics

Our method of evaluating the quality of our model was to run various performance metrics and compare its performance to the pre-trained reference GAN (if applicable). These include, AUROC, FID, F1-score, and loss. In order to properly compare our results using diffusion with our reference article's GAN model, we ran most of these same tests on their model as well to compare the validity of our images.

A. Loss Curve

Our loss curve shows the training loss for our diffusion model. Every training epoch, we keep track of the noise prediction loss, calculated as the MSE between the predicted noise and the actual noise added. We note that since the labels are embedded into context layers in our diffusion architecture, it is a parameter in our loss function and is weighted in noise

prediction. This is ultimately how the model learns and characterizes the difference between a melanoma and benign image. We did not compare this metric with the GAN as the GAN was pre-trained and would have different loss functions specific to the model architecture.

B. Accuracy and Ground Truth

The first metric we use to compare classifier performance on the two models is accuracy. Naturally, since the models use the same input images, there is no difference in classification performance on the original image set. What we want to compare is how accurate the counterfactuals are to their target label. We will use the target label as the ground truth and compare the classifier prediction with the target label for all metrics that require so. As our model returns a score between 0-1, the threshold we choose to differentiate between a benign label and melanoma is important. We will choose this threshold based on the F1-score, which is explained below.

C. Precision, Recall, and F1-score

In addition to accuracy, we used F1-score to evaluate DeepDerm classifier performance on both models. F1-score is a classification performance metric and is calculated via the harmonic mean of precision and recall:

$$F_1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Equation 4: F1-score equation

Precision is the proportion of predicted positives that are true positives; in essence it is how precise the positive predictions are. Recall is the proportion of all true positives that the classifier correctly predicts as positive: in essence it is the volume of positives that the classifier was able to capture. The two metrics are inversely related, so we use F1-score as a balancing metric between the two.

With regards to a threshold, we simply just tested all viable thresholds and report all the metrics (including accuracy) using the threshold that results in the best F1-score.

D. Area Under ROC Curve

Also, we used the area under the receiver operating characteristic curve (AUROC) to measure the ability of the classifier to distinguish between negative and positive classes across different thresholds. By plotting the True Positive Rate (TPR) against the False Positive Rate (FPR) at different threshold levels, we generated the ROC curve.

$$True\ Positive\ Rate(TPR) = \frac{True\ Positive}{(False\ Negative + True\ Positive)}$$

Equation 5: TPR equation

$$False\ Positive\ Rate(FPR) = \frac{False\ Positive}{(True\ Negative + False\ Positive)}$$

Equation 6: FPR equation

The area under this curve shows the probability that the classifier ranks a random positive sample chosen higher than a negative one. An AUROC of 1 indicates a perfect classifier, while 0.5 indicates a totally random classifier.

E. FID Score

Unlike F1-score and AUROC, which are primarily used to measure the performance of the classifier, We used Fréchet Inception Distance (FID) Score to evaluate the quality of our generated counterfactual images. It calculated the distance between the feature distributions of real and generated images.

$$d_F(\mathcal{N}(\mu,\Sigma),\mathcal{N}(\mu',\Sigma'))^2 = \|\mu-\mu'\|_2^2 + \mathrm{tr}\left(\Sigma+\Sigma'-2(\Sigma\Sigma')^{\frac{1}{2}}\right)$$

Equation 7: Fréchet Inception Distance

A lower FID score for the diffusion model indicates that it can generate counterfactuals that align well with the distribution of real images. The score itself has some limitations, including sensitivity to the pre-trained network used and its tendency to image sharpness over semantic correctness.

5. Quantitative Results

Here we report and analyze our quantitative results for our performance metrics. We do note that our current model is more of a proof-of-concept rather than a polished product. Notably, our diffusion model outputs had some dimension mismatches with the DeepDerm outputs. We were not able to fully vet this issue and it affected our performance evaluation. There are many other additional optimizations that can be done on our model as well.

A. Diffusion Training Loss

Figure 6 shows the MSE loss over training epochs. Our loss curve exhibits standard exponential decay, converging at 0.009 around the 10th epoch and beyond. We can validate that our training was following the behavior as expected.

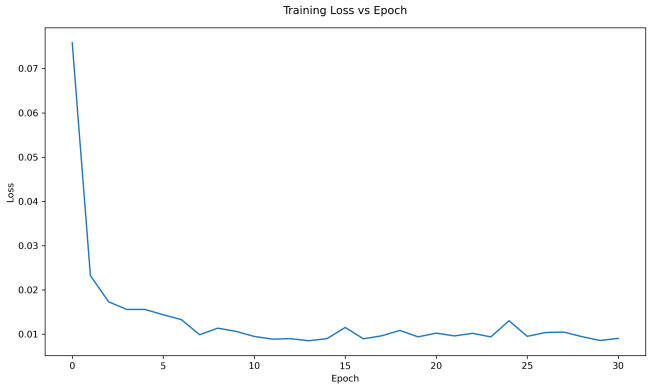


Figure 6: Diffusion Training Loss vs Epochs

B. Accuracy, Precision, Recall, and F1-score

As shown in Table 1, we report our accuracy, precision, recall, and F1-score using the best label threshold for F1-score. This was done on counterfactuals generated via our diffusion model and the pretrained GAN, using the DeepDerm Classifier.

	Original Dataset	GAN Counter-factuals	Diffusion Counter-factuals
Best Label Threshold for F1-score	0.879	0.463	0.154
Accuracy	0.781	0.991	0.863
Precision	0.595	0.981	0.593
Recall	0.453	0.974	0.951
F1-score	0.514	0.978	0.730

Table 1: Original vs GAN Counterfactual vs Diffusion Counterfactual Images

The GAN performed significantly better than our diffusion model, getting near perfect results (~1) for all the metrics. The optimal threshold is approximately 0.5, which logically makes sense and means the scores are distributed in an even manner. Our diffusion model does not perform poorly, with a decent F1-score of 0.73 and accuracy of 0.863. However, it is evident that the distribution of scores is less predictable due to the poor precision and extreme optimal threshold. The diffusion counterfactuals still do better than the original dataset though, with its F1-score of 0.514 and 0.781 accuracy. This is promising as it demonstrates that overall DeepDerm can recognize our diffusion model's counterfactuals target label, just not to the same extent as the GAN.

We did expect the GAN to outperform our diffusion with DeepDerm, as the GAN uses DeepDerm outputs during training while we only use the ISIC ground truth labels to guide training. However, the difference is very extreme, mainly due to our diffusion model not performing as well as we expected. This is due to the fact that the DeepDerm model used was trained on a separate dataset (DDI Dataset) and naturally, does not perform as well on the ISIC dataset.

We show the precision-recall curve in Figure 7 and Figure 8. The GAN curve shows near perfect performance as expected, while the diffusion model has relatively poor precision throughout. DeepDerm is classifying a significant number of benign samples as melanoma, suggesting that there is a bias.

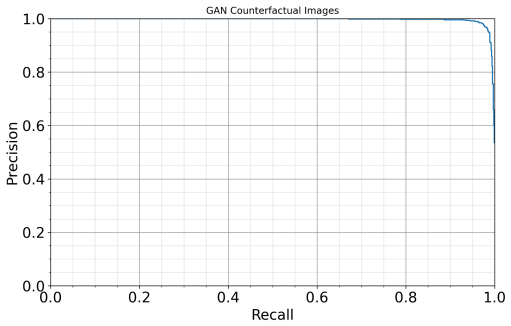


Figure 7: Precision vs Recall for GAN Counterfactuals

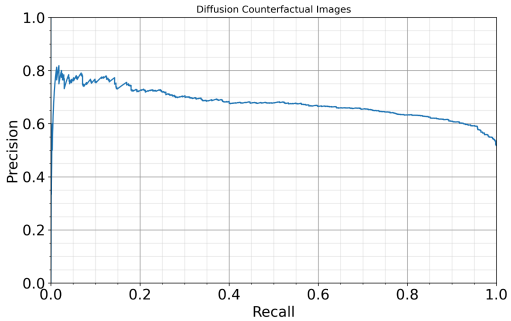


Figure 8: Precision vs Recall for Diffusion Counterfactuals

C. AUROC

Also, we plotted the ROC curves for both the GAN and the diffusion model. The slight difference of accuracy on the original dataset in both graphs is due to the random sample selected from our dataset.

For the GAN model, we can observe that it achieves a perfect AUC value of 1.00, with its curve tightly hugging the top-left corner. The reason why this happens is because we set the target label as the ground truth instead of the criteria included in the original dataset. This adjustment ensures consistency with the ground truth we established throughout our entire research process.

For the Diffusion model, the generated counterfactuals achieves a lower AUC value of 0.72 than the model's performance on original dataset but still good enough to show its' ability to distinguish melanoma characteristics. Additionally, the result may be influenced by the bias in the original dataset, where only 20% of them are labeled as melanoma.

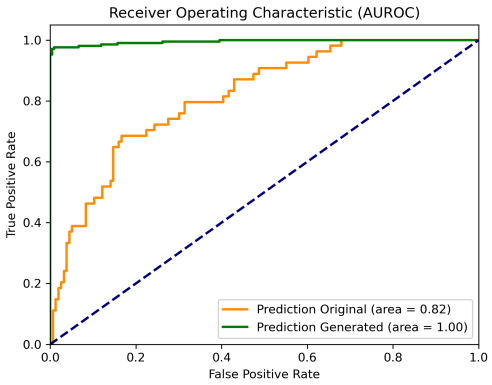


Figure 9: AUROC for GAN Counterfactuals

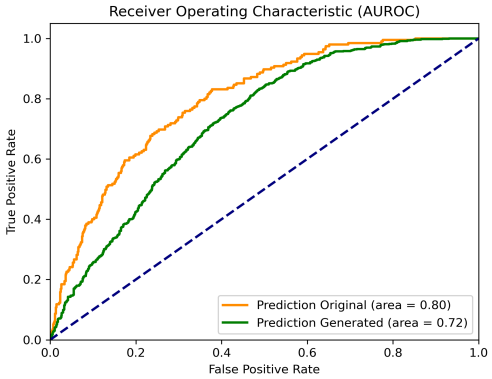


Figure 10: AUROC for Diffusion Counterfactuals

D. FID

We used a pretrained InceptionV3 model for calculating the FID score to evaluate the quality of counterfactuals generated by both GAN and Diffusion models. The GAN model performed better than the Diffusion model with a score of 46.2 after 3000 iterations. However, counterfactuals generated by the diffusion model ended at a pretty higher value of 209.1. This can be explained by the fact that GAN is a pretrained model, while the Diffusion model is trained by ourselves. Due to time constraints, we had a smaller dataset and epochs to train our Diffusion model, which may have an impact on the quality of counterfactuals.

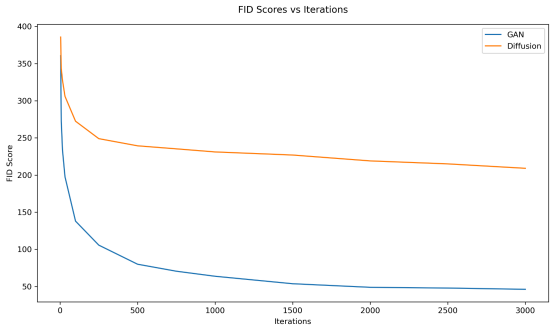


Figure 11: FID for GAN and Diffusion

6. Qualitative Results

Although the reference article utilized board-certified dermatologists to assess the results of the GAN's generated counterfactual images, we could only assess the results within our own team. Despite this, it can be seen that the diffusion model was able to generate and remove melanotic attributes on these images. For example, in figure 4, it can be seen that the malignant counterfactual adopts characteristics of some malignant images such as more discoloration on the mole and the surrounding skin. A trend of generating the benign counterfactuals was to smooth out the coloration of the mole and clear the surrounding skin. In addition to this, the model would maintain non-melanoma characteristics such as the presence of hair despite affecting the neighboring skin.

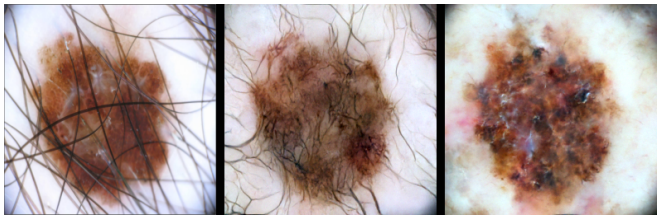


Figure 12: Example Output With Maintained Hair Presence

When comparing our results to that of the GAN model, there are some clear differences in how the two models generate the counterfactual images of the same anchor image.

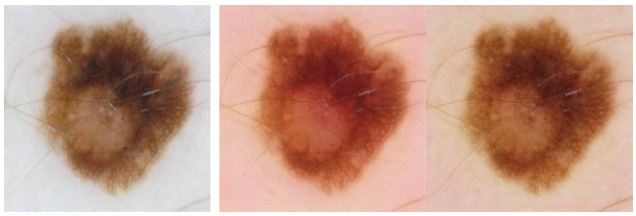


Figure 13a: GAN Generated Counterfactual Images

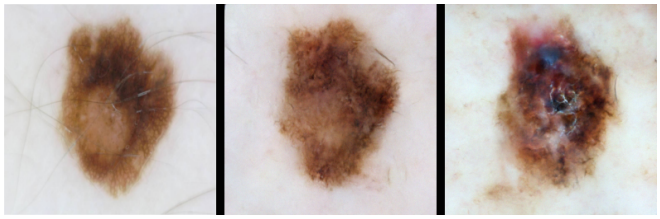


Figure 13b: Diffusion Generated Counterfactual Images

As seen between Figure 13a and 13b, the GAN model tends to adjust the overall coloration to generate the appropriate label, whereas the diffusion model mostly retains the color of the surrounding skin and mole, but will add or remove characteristics. For example, sharp reds and blue spots tend to be added in many diffusion-generated malignant counterfactuals. Additionally, it can be seen that the GAN outputs do not change or affect the skin surrounding the mole.

Based on the performances between the GAN and Diffusion models, it can be concluded that the GAN was an overall better performing model for this task. Despite this, the diffusion model still performed better than the original images on the DeepDerm classifier, and this model showcased an alternative method of creating counterfactual images by adding and removing melanoma-related features to the image directly.

7. Detailed Roles and Tasks

Task	Who
Follow GAN paper framework with DeepDerm classifier and ISIC dataset	Howell
Decide skin tone dataset	Amy & Diego
Research diffusion models and classifiers	Diego & Ivan
Set-up SSC environment	Howell
Follow GAN paper framework with ISIC	Ivan & Amy
Implement and train diffusion model with DeepDerm	Ivan
Generate counterfactual images	Howell & Ivan
Connect DeepDerm classifier to diffusion model outputs	Ivan & Diego
Perform evaluation metrics on diffusion model	All
Evaluate how diffusion counterfactuals compare to GAN	Howell, Diego, & Amy

8. Github Code

GITHUB: https://github.com/howellx/derm_audit

References

1) Qiangguo Jin, Hui Cui, Changming Sun, Zhaopeng Meng, Ran Su, Cascade knowledge diffusion network for skin lesion diagnosis and segmentation, Applied Soft Computing, Volume 99 (2021).
2) DeGrave, A.J., Cai, Z.R., Janizek, J.D. et al. Auditing the inference processes of medical-image classifiers by leveraging generative AI and the expertise of physicians. Nat. Biomed. Eng (2023).
3) Conditional Diffusion model article: [1907.01277] Conditioned-U-Net: Introducing a Control Mechanism in the U-Net for Multiple Source Separations
4) Vodrahalli, K. (2022). Diverse Dermatology Images (Code) Zenodo. https://doi.org/10.5281/zenodo.6784279
5) International Skin Imaging Collaboration. (n.d.). ISIC Archive: Melanoma and Skin Lesion Image Dataset. November 7, 2024, https://gallery.isic-archive.com
6) Contextual UNet Inspiration: https://github.com/TeaPearce/Conditional_Diffusion_MNIST