Project Title:

Conditional Generative Adversarial Network (cGAN) for Image Synthesis with Class Labels

Computer Vision (CSL7360)

By:- Ritu Singh (M23CSE023)
Bhavesh Khatri (M23CSE011)
Sireejaa Uppal (M23CSE023)
Ritobina Ghosh (M23CSA021)

Submitted To:-
Dr. Rajendra Nagar Assistant Professor Department of Electrical Engineering Indian Institute of Technology Jodhpur (IITJ)

Github Link for Code [Link]
Output Link [Link]

Problem statement:

The goal of this project was to create a Conditional Generative Adversarial Network (cGAN) capable of generating high-quality images from sketches. The objective was to develop a machine learning model that could be used in various applications such as digital art creation, game character design, and educational tools.
The core challenge lies in accurately transforming grayscale sketches, which primarily represent the basic shape and form of an object, into colorful, detailed images that reflect the intended characteristics as defined by the conditioning labels. The cGAN structure, with its generator and discriminator components, was chosen to address this challenge because it provides a framework for adversarial training. This technique has proven effective in producing realistic outputs by pitting the generator and discriminator against each other, fostering iterative improvement.

To accomplish this goal, the project aimed to:

1. Develop a generator that can take a sketch and conditioning labels as input and generate full-color images.
2. Build a discriminator that can distinguish between real images and images generated by the generator.
3. Train these two components in a manner that ensures a balanced equilibrium, leading to improved generation quality over time.

Additionally, the training process needed to maintain stability, requiring techniques like U-Net architecture with skip connections, careful loss function design, and optimization strategies to prevent issues such as mode collapse or overfitting. By addressing these elements, the project intended to demonstrate a reliable method for generating detailed images from simple sketches.

Methodology

The methodology for this project involved the development of a Conditional Generative Adversarial Network (cGAN) designed to generate high-quality images from grayscale sketches. The overall process can be broken down into the following key components: model architecture, dataset and preprocessing, training, and evaluation.

Model Architecture

The cGAN comprises two main parts: a generator and a discriminator.

1. **Generator (cGen)**
    - **Input Layer**: Accepts two inputs: a sketch (grayscale image) and a label (conditioning data). The label provides additional information on specific attributes or classes that the generated image should reflect.
    - **Processing Layers**: Combines the sketch and label, which then pass through a series of convolutional layers, increasing in depth while decreasing in spatial dimensions.
    - **U-Net Architecture**: Incorporates skip connections to connect layers of similar sizes in the encoder and decoder portions. This helps retain spatial information during the downsampling process.
    - **Upscaling**: Transposed convolutional layers in the decoder upscale the encoded features, aiming to reconstruct detailed images.

    - **Output Layer**: Ends with a tanh activation function to produce a full-color image matching the sketch input and satisfying the given conditions.
2. **Discriminator (cDiscr)**
    - **Input Layer**: Takes three inputs: a real image, a generated image, and a label. The real images serve as positive examples during training, while the generated images come from the generator.
    - **Combining Inputs**: Concatenates image data (real or generated) with label data to produce a conditioned output.
    - **Processing Layers**: The concatenated data goes through a series of convolutional layers, increasing in depth while reducing spatial dimensions, to extract relevant features for real or fake determination.
    - **PatchGAN Architecture**: Provides a matrix of outputs representing different patches of the image, allowing the discriminator to focus on authenticity at the patch level.
    - **Output Layer**: Uses a sigmoid activation function to predict whether patches of the input image are real or fake. This output is used to calculate loss during training, guiding the generator toward more realistic image synthesis.

Dataset and Preprocessing

The dataset comprised pairs of images: real RGB images and corresponding sketches. The preprocessing steps included:

- **Resizing**: The images were resized to 128x128 pixels.
- **Normalization**: The pixel values were normalized to a range between -1 and 1, stabilizing GAN training.

Training

Training involved the following steps:

1. **Generator Training**
    - The generator generates images from sketches and labels, aiming to create realistic images. These images are then passed to the discriminator for evaluation.
2. **Discriminator Training**
    - The discriminator is trained with real images and those generated by the generator, aiming to classify them as real or fake.
    - To avoid discriminator overpowering, the generator was trained twice for every discriminator training step.
3. **Adversarial Loss**

- The adversarial loss for the discriminator was calculated using Binary Cross Entropy (BCE), differentiating between real and fake images.
- The generator's adversarial loss was calculated by comparing the discriminator's predictions with an array of ones (representing real images), encouraging the generator to produce realistic outputs.
- The generator's L1 loss compared the pixel values between generated images and real images, penalizing differences.

4. **Backpropagation and Weight Update**
    - After computing the losses, gradients were calculated through backpropagation, and the weights of the generator and discriminator were updated using the Adam optimizer. The learning rate was set to 0.0001 with a beta parameter of 0.5.
5. **Equilibrium Seeking**
    - The training sought to reach a balance between the generator and discriminator, targeting a discriminator loss of approximately 0.5, indicating equal uncertainty between real and fake classifications.
    - A periodic evaluation was performed to ensure the quality of generation improved over time.
    - Model checkpoints were saved at intervals to allow training to be paused and resumed, and to retain the best-performing models.
6. **Training Loop**
    - The above steps were repeated for each batch of data over multiple epochs. The training continued until convergence or a defined stopping criterion was met, such as a fixed number of epochs or minimal improvement between epochs.

These methodological steps formed the core structure of the cGAN project, ensuring a robust and reliable approach to generating high-quality images from sketches.

Results

The Conditional Generative Adversarial Network (cGAN) was designed to generate high-quality images from grayscale sketches, aiming to match real images in detail and diversity. The results of the project were evaluated through various metrics and qualitative observations to assess the quality of generated images and the effectiveness of the training process.

Loss Metrics

During training, the loss metrics for the generator and discriminator were tracked to monitor the convergence and balance between the two components. A successful GAN training is indicated by a discriminator loss of approximately 0.5, suggesting that it is equally uncertain about real and fake images. The generator's loss should ideally decrease over time, indicating an improvement in the generator's capability to produce realistic images.

- **Discriminator Loss**: The loss stabilized around 0.5, indicating a balanced equilibrium between real and fake image classification.
- **Generator Loss**: The generator's loss showed a decreasing trend, indicating improved image synthesis capabilities.

Image Generation
The quality of generated images was periodically evaluated at different stages of the training process, typically every 50 epochs, to observe the progress and refinement in image quality.

- **Epoch-by-Epoch Improvement**: The generated images demonstrated gradual improvement in detail and color accuracy with each subsequent evaluation. By the 500th epoch, the generated images closely resembled real images, exhibiting high-quality color and detail.
- **Generated Image Examples**: Images generated at different epochs (0th, 100th, 200th, 300th, and 400th) displayed a clear progression in terms of quality and realism. The 400th epoch yielded images with significant visual improvement compared to the earlier stages.
- **Final Generated Images**: The images generated at the 500th epoch achieved a level of quality that closely matched real images, suggesting that the model effectively learned from the dataset and improved over time.

Frechet Inception Distance (FID)
FID compares the distribution of generated images to that of real images using feature vectors extracted from the InceptionV3 model. A lower FID score indicates a smaller distributional distance, signifying that the generated images closely resemble the real images in terms of key characteristics.

- **FID Score**: The FID score was evaluated at different stages, showing improvement as training progressed. This metric demonstrated that the generated images became increasingly similar to real images, supporting the qualitative observations from the image generation analysis.

  FID Score : 95.78

Inception Score (IS)
IS measures the clarity and diversity of the generated images by calculating the exponential of the average KL divergence across all images. A higher IS suggests a greater variety of high-quality images.

- **IS Score**: The IS score also improved over time, indicating that the generated images displayed increased diversity and clarity as training continued.

Inception Score (IS): 13.007454872131348

The GAN training loss Plot is as shown below:

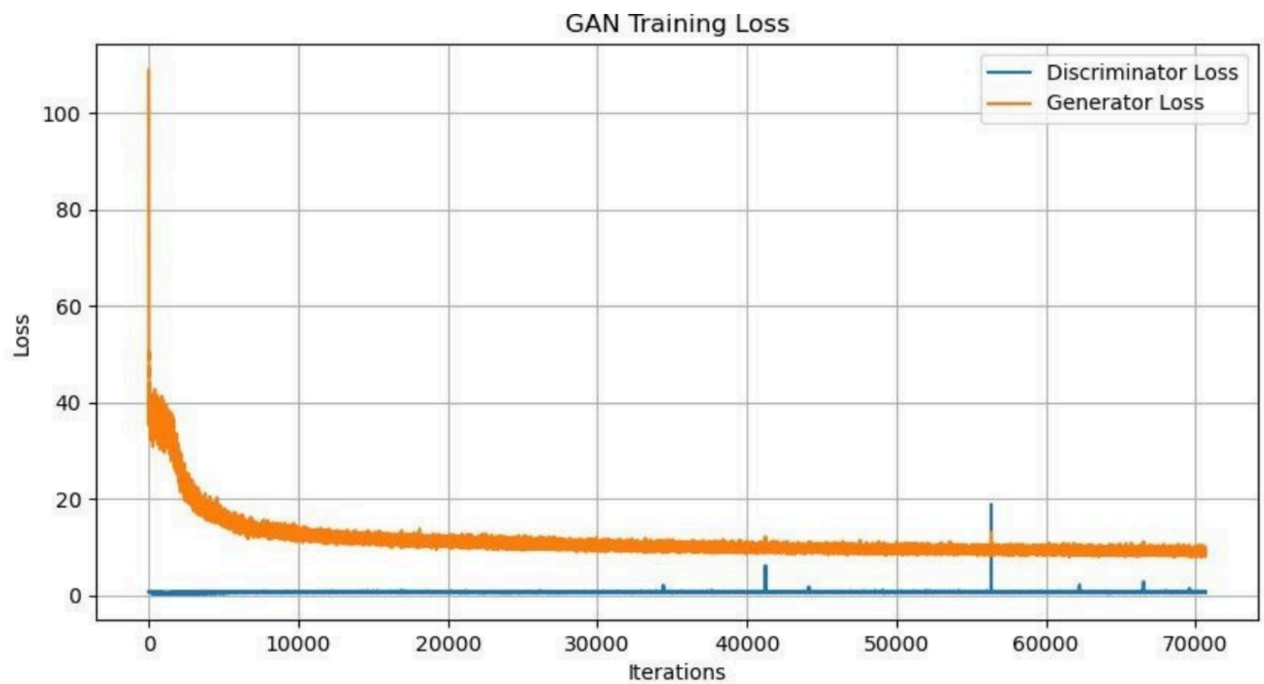
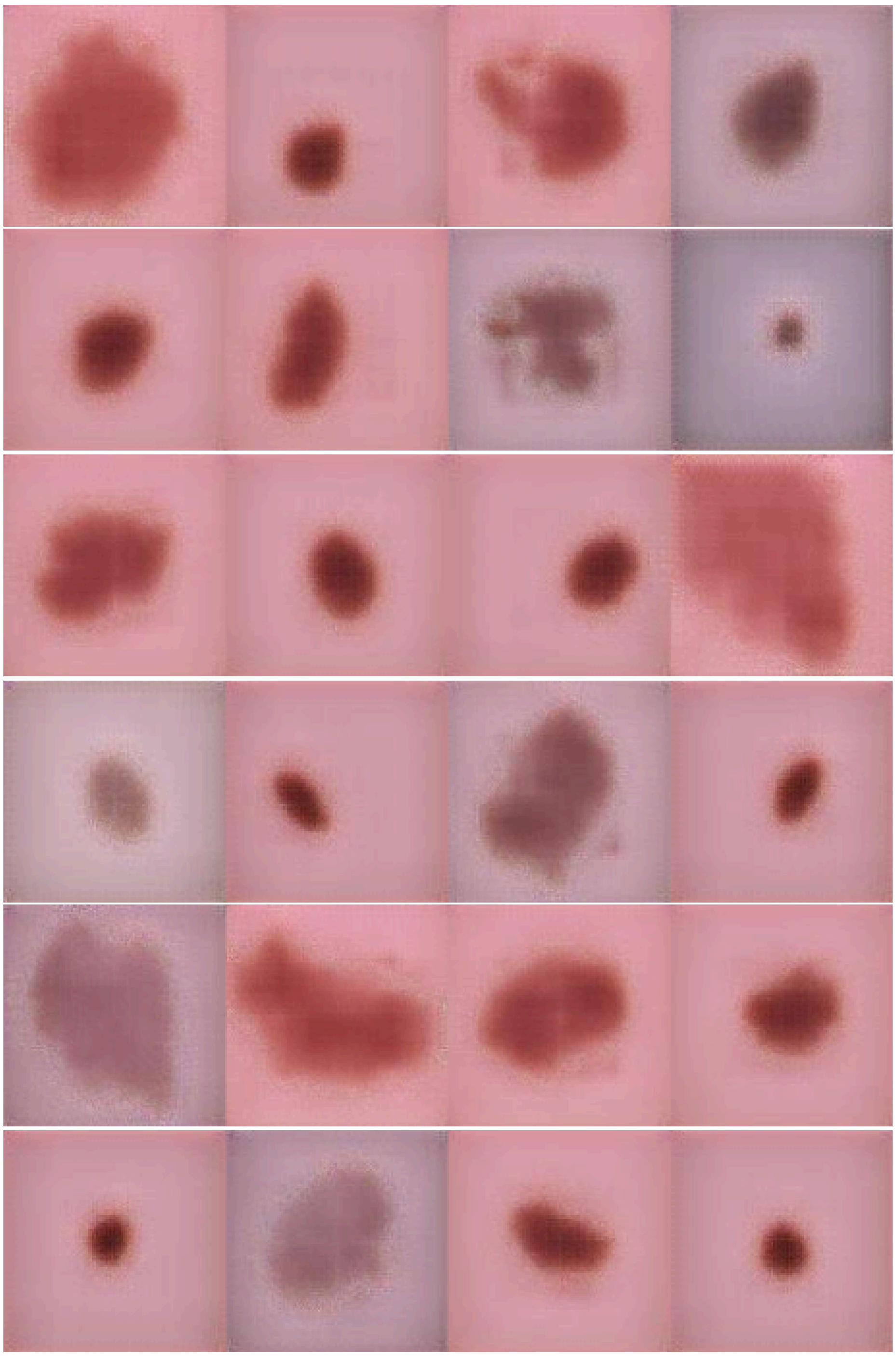


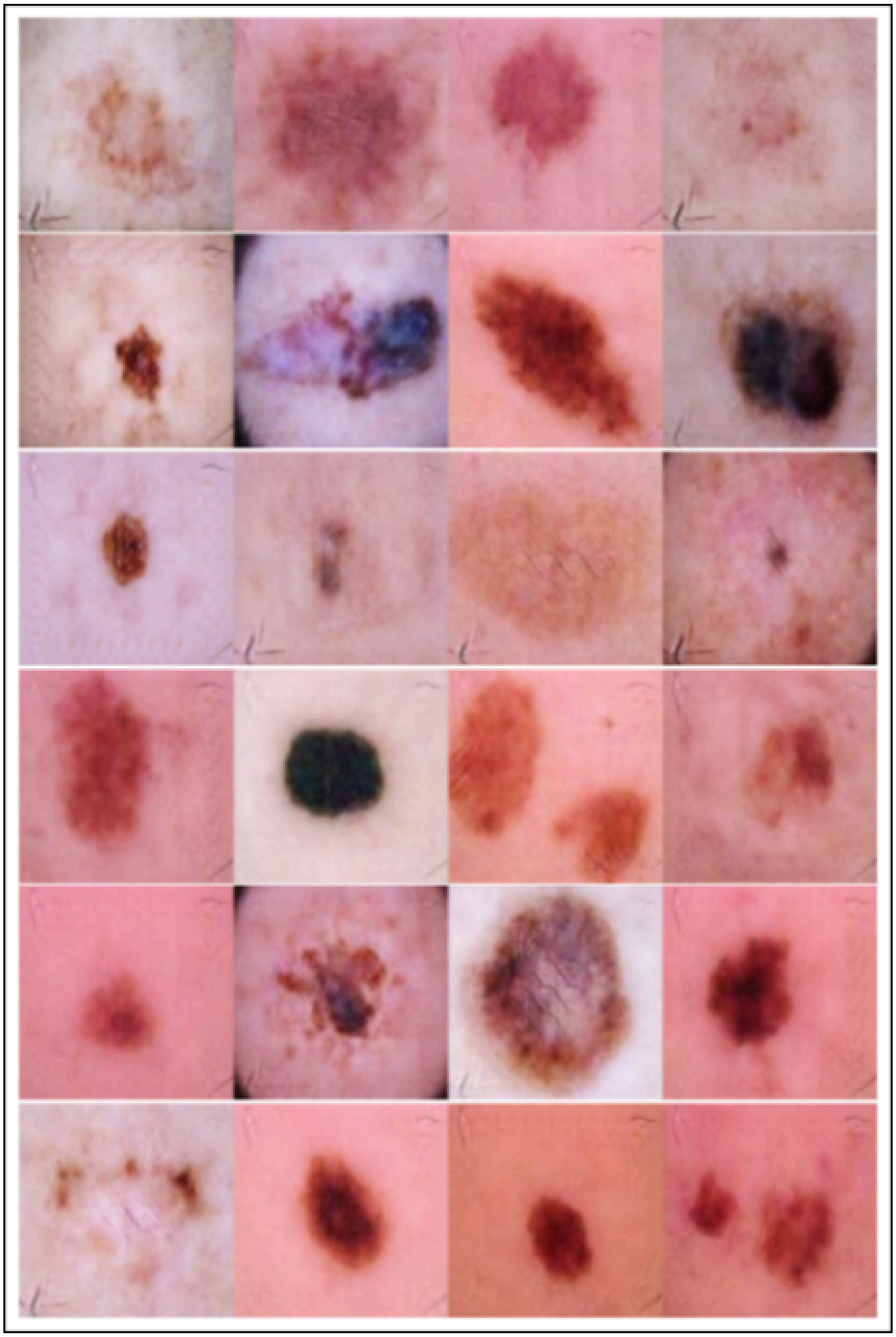
Image Generation:

The generated images are shown as below:

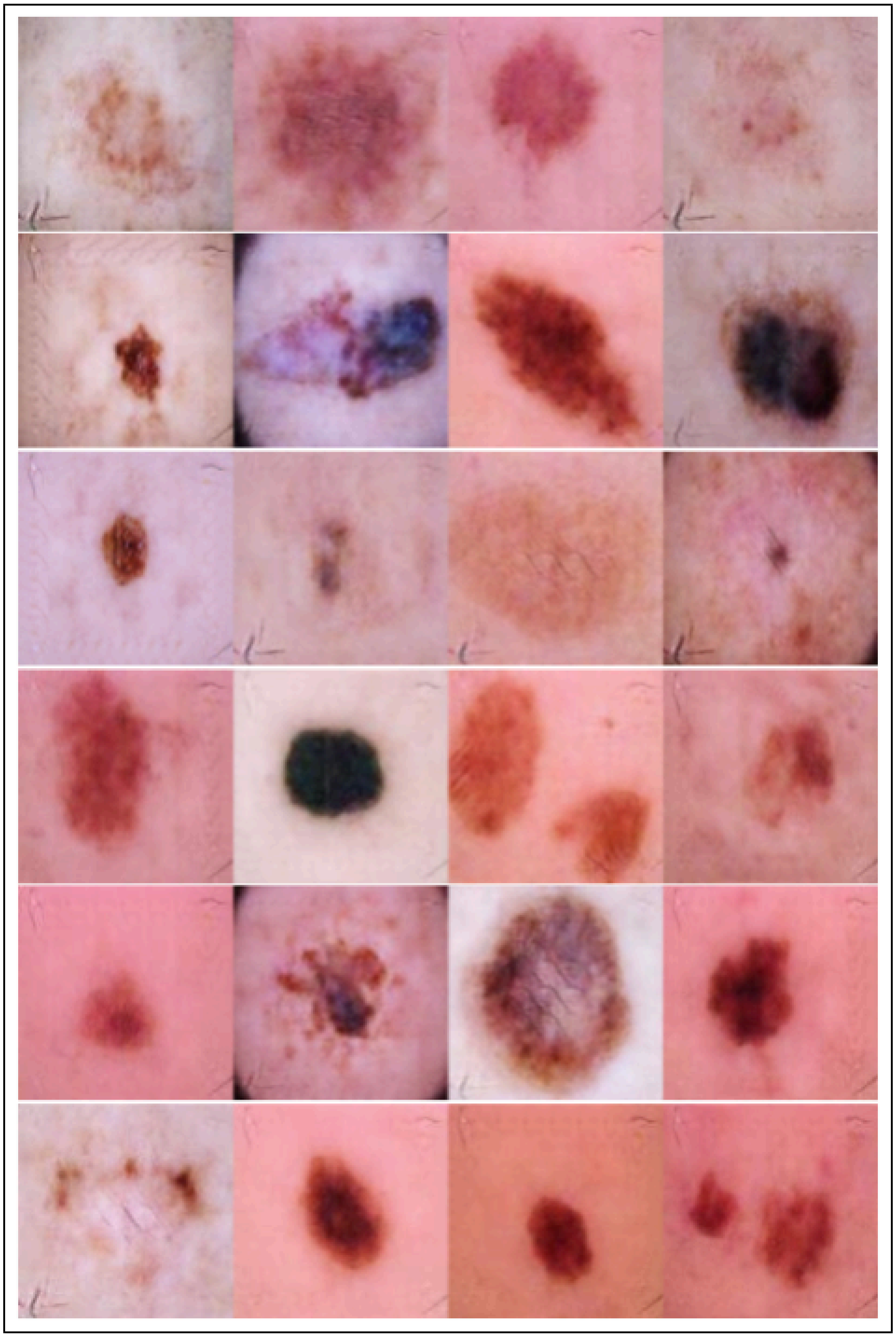
For 0^{th} Epoch:



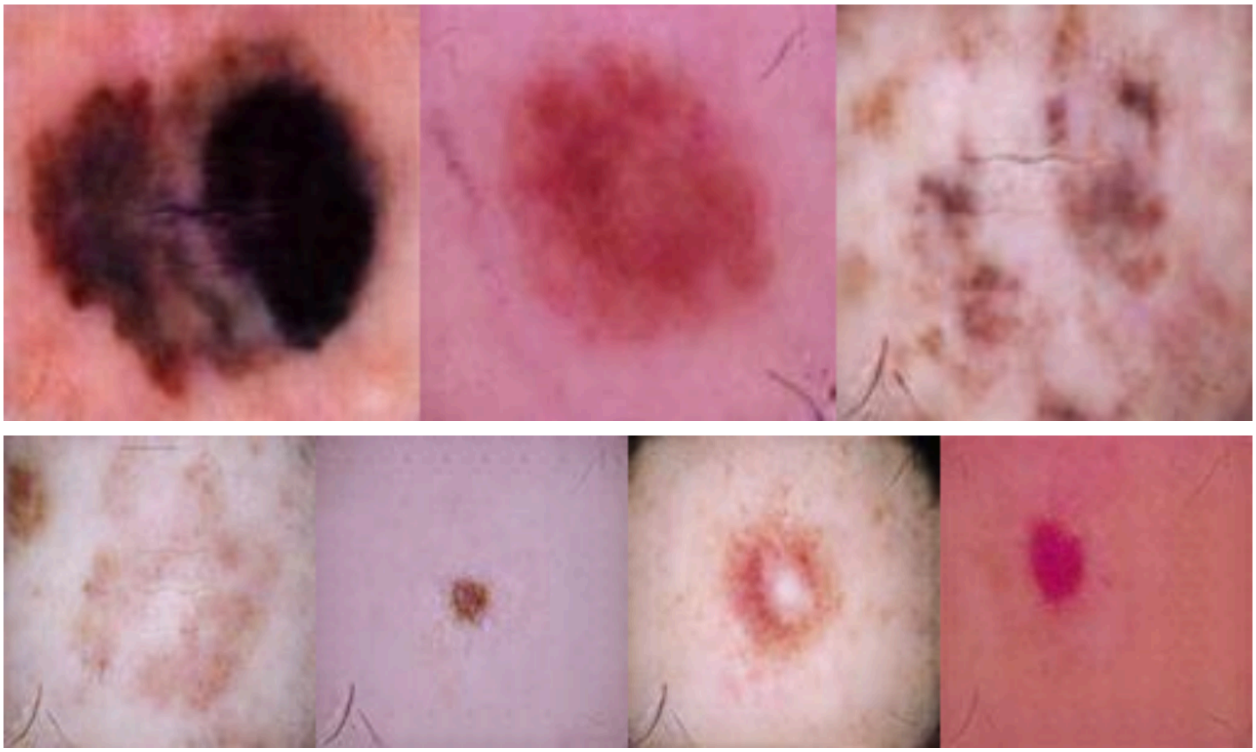
For 200th Epoch:



For 300th Epoch:



For 400th Epoch:



Observations

The Conditional Generative Adversarial Network (cGAN) project demonstrated a clear progression in generating realistic images from grayscale sketches. Throughout the training process, several key observations emerged, shedding light on the effectiveness of the model and areas for further refinement:

1. **Image Quality Improvement**:
    - The generated images showed noticeable improvement with each successive epoch. The quality, detail, and color accuracy of the images increased steadily, indicating that the generator learned to interpret the sketch inputs more effectively as training progressed.

- The results from the 500th epoch reflected a significant advancement in quality, with images resembling real photos, suggesting that the cGAN was successfully trained to produce realistic outputs.

2. **Equilibrium between Generator and Discriminator**:
   - Achieving a balance between the generator and discriminator was crucial for stable training. The desired equilibrium was generally observed, with the discriminator loss stabilizing around 0.5, indicating that it was equally uncertain about distinguishing real and fake images.
   - However, maintaining this equilibrium was challenging, with occasional fluctuations in the loss metrics, suggesting that the discriminator could become too strong, leading to an imbalance in the adversarial relationship.
3. **Effectiveness of the U-Net Architecture**:
   - The use of skip connections in the generator helped retain spatial information during downsampling and upsampling processes. This contributed to better image quality and reconstruction of detailed features.
   - The U-Net architecture played a significant role in the generator's ability to produce visually appealing and detailed outputs, making it a valuable design choice for this project.
4. **Training Challenges and Instability**:
   - Despite the general success of the cGAN model, training was not always smooth. The generator needed to be trained twice as often as the discriminator to prevent the latter from overpowering the former.
   - Instability during training was observed, highlighting the importance of techniques like separate optimizers, Adam with appropriate learning rates, and periodic evaluations to ensure the training process stayed on track.
5. **Evaluation Metrics and Model Checkpoints**:
   - The use of metrics like Frechet Inception Distance (FID) and Inception Score (IS) provided a quantitative measure of the model's performance. These metrics showed improvement, supporting the qualitative observations of image quality enhancement.
   - Model checkpoints and periodic evaluations allowed for training to be paused and resumed, providing flexibility and the ability to retain the best-performing models.

Conclusions

Based on these observations, the following conclusions can be drawn:

1. **cGAN's Potential for Image Generation**:
    - The cGAN model demonstrated a strong potential for generating high-quality images from grayscale sketches. The project achieved its objective of creating a model that could produce detailed, realistic images for various applications like digital art generation, game character design, and educational tools.
2. **Effectiveness of Adversarial Training**:
    - The adversarial relationship between the generator and discriminator proved effective in fostering improvement over time. The PatchGAN architecture, with its patch-based discrimination, was a key factor in achieving this result, allowing the discriminator to focus on different parts of the image.
3. **Challenges in GAN Training**:
    - The project highlighted the inherent challenges of GAN training, particularly the need to maintain a balanced equilibrium between the generator and discriminator. Techniques like training the generator more frequently and using separate optimizers with specific hyperparameters were crucial to stabilize the training process.
4. **Scope for Further Research and Improvements**:
    - While the project achieved success in generating realistic images, there is scope for further research and improvements. Future work could explore deeper architectures, alternative loss functions (such as Wasserstein loss), and more diverse datasets to increase the complexity and diversity of generated images.
    - Additional stability measures and strategies to prevent mode collapse could also be investigated to ensure consistent training results.

Overall, the cGAN project achieved its goal of generating images from sketches with promising results. These conclusions suggest that the approach is viable for creative applications and has the potential for continued refinement and optimization in future work.

Accuracy of the Classifier on the test images:- 62.5%

Accuracy on generated images:- 30.57%

References:

- https://www.geeksforgeeks.org/conditional-generative-adversarial-network/
- https://machinelearningmastery.com/how-to-develop-a-conditional-generative-adversarial-network-from-scratch/
- https://colab.research.google.com/github/tensorflow/gan/blob/master/tensorflow_gan/ examples/colab_notebooks/tfgan_tutorial.ipynb
- https://developers.google.com/machine-learning/gan/training
- https://pytorch.org/tutorials/beginner/dcgan_faces_tutorial.html
- https://link.springer.com/article/10.1007/s00521-023-09100-z
- https://www.futuremedicine.com/doi/10.2217/fmai-2023-0004
- https://www.tandfonline.com/doi/full/10.1080/21681163.2024.2330524?src=exp-la
- https://albuspeter.github.io/projects/JPI/JPI_2022_000502_Zhihang-Ren.pdf
- https://pyimagesearch.com/2022/07/27/image-translation-with-pix2pix/ https://docs.wandb.ai/guides/track/tracking-faq