KALASALINGAM
ACADEMY OF RESEARCH AND EDUCATION
(DEEMED TO BE UNIVERSITY)
Under sec. 3 of UGC Act 1956. Accredited by NAAC with "A++" Grade
Anand Nagar, Krishnankoil, Srivilliputtur (Via), Virudhunagar (Dt) - 626126, Tamil Nadu | info@kalasalingam.ac.in | www.kalasalingam.ac.in

IMAGE DENOISING USING GENERATIVE ADVERSARIAL NETWORKS

Submitted By

A.Surya Prakash Reddy
Reg.No - 99220040385

INTRODUCTION:

- Image denoising is the process of removing unwanted noise from digital images while preserving important visual information.
- It involves the use of various techniques to reduce the effects of noise, which can manifest as random variations in pixel values.
- The goal of image denoising is to enhance the overall quality and clarity of images by eliminating distracting or irrelevant noise patterns.
- This process is essential in fields such as photography, medical imaging, and computer vision, where high-quality images are crucial for accurate analysis and interpretation.
- Traditional methods for image denoising include filtering techniques, such as Gaussian or median filtering, which smooth out noise but may also blur image details.
- Overall, image denoising plays a vital role in improving the visual quality and interpretability of digital images across various applications.
- Here we are going to develop a model for Image Denoising using Generative adversarial networks which is the best model among all the existing models.

What are GAN's:

- Generative Adversarial Neural Network (GAN) is a type of machine learning model that consists of two neural networks: a generator and a discriminator. The generator network is trained to generate realistic images, while the discriminator network is trained to distinguish between real and generated images.
- GANs have been widely used in image denoising tasks. By training the generator network on a dataset of clean images and noisy images, the GAN can learn to generate denoised images that are visually similar to the clean images. The discriminator network helps ensure that the generated images are of high quality and indistinguishable from the real image
- **Generative:** To learn a generative model, which describes how data is generated in terms ofa probabilistic model.
- **Adversarial:** The training of a model is done in an adversarial setting.
- **Networks:** Use deep neural networks as the artificial intelligence (AI) algorithms for training purpose.

About Software Tool:

- **Integrated Development Environment (IDE):** You can use IDEs like PyCharm, Visual Studio Code, or Jupyter Notebook for writing, debugging, and executing your Python code efficiently.
- **Python:** As the primary programming language for most deep learning projects, Python provides a robust environment for developing and implementing GAN-based image denoising algorithms.
- **Deep Learning Frameworks:** You'll work extensively with deep learning frameworks such as **TensorFlow or PyTorch**. These frameworks offer APIs and tools for building, training, and deploying deep neural networks, including GAN architectures for image denoising.
- **Command Line Interface (CLI):** You'll likely utilize the command line interface for running training scripts, managing dependencies, and executing various tasks related to your image denoising project.
- **Version Control System (VCS):** Using version control systems like Git and platforms like GitHub or GitLab allows for collaborative development, code sharing, and tracking changes in your project codebase.

Usage of Tool:

- **Integrated Development Environment (IDE):**Writing and editing Python code efficiently with features like syntax highlighting, code completion, and debugging capabilities.Managing project files, organizing code, and navigating through the project structure seamlessly.
- **Deep Learning Frameworks:**Building and training GAN-based image denoising models using high-level APIs provided by TensorFlow or PyTorch.Leveraging pre-built neural network layers, loss functions, and optimization algorithms to streamline model development.Experimenting with different network architectures, hyperparameters, and training strategies to improve denoising performance.
- **Command Line Interface (CLI):**Running training scripts and executing various tasks related to model training and evaluation from the command line.Installing and managing dependencies using package managers like pip or conda.Monitoring training progress, logging metrics, and managing model checkpoints during training sessions.
- **Version Control System (VCS):**Tracking changes in project codebase, including modifications to model architectures, training scripts, and evaluation metrics.Collaborating with team members, sharing code, and reviewing contributions using branching, merging, and pull request workflows.Maintaining a history of project iterations, facilitating reproducibility and accountability in research and development.

Literature Survey:

S .No	Title	Author	Year	Description
1	"Image Denoising Using Deep CNN with Batch Normalization"	Yang, Shuyu	2018	GAN-based image denoising approach that utilizes deep Convolutional Neural Networks (CNNs) with batch normalization. By integrating batch normalization layers into the network architecture, the method achieves significant improvements in denoising performance compared to traditional CNN-based approaches.
2	" Residual Learning of Deep CNN for Image Denoising"	Zhang, Kai	2017	GAN-based image denoising method that goes beyond traditional Gaussian denoisers by incorporating residual learning with deep CNNs. the proposed approach achieves superior denoising performance, especially in scenarios with complex noise patterns and low signal-to-noise ratios.
3	"Denoising Generative Adversarial Networks" Description	Yan, Han, et al.	2018	Denoising Generative Adversarial Networks (DnGAN), a novel approach for image denoising using GANs. DnGAN incorporates a multi-scale discriminator and a perceptual loss function to enhance denoising performance. Experimental results demonstrate its effectiveness in producing high-quality denoised images.

System Specifications:

Software Requirements

- Operating System: Windows 11
- IDE: Google Collaboratory ,Jupytyer Notebook
- Libraries: Python

Hardware Requirements

- Processor: Intel core i7
- RAM:4 GB

Objective of Project:

- The goal of the image restoration is to recover an image that has been blurred.
- In this work we try to develop deep learning based models that try to take in the low resolution images and provide a high resolution version of that particular image.
- In this work we try to retain very crucial features of the low resolution image like edges, colour textures, shadows etc.
- To enhance the image and to give better results we use Generative Adversarial Networks that help us in generating a high resolution image with rich features.

Timeline of Work Proposal

- JAN 12 – Course Start
- JAN 18 – Identification of Problem Statement
- JAN 27 – Data Collection And Preprocessing
- JAN 29 – Model Selection
- FEB 3 – Model Evaluation
- FEB 10 – Course Completion
- FEB 22 – Deployment

Work Flow:

- Collect the images which contains noise.
- Preprocess the images by resizing, normalizing, and augmenting if necessary.
- Choose a GAN architecture suitable for image denoising (e.g., DCGAN, WGAN).
- Design the generator and discriminator networks, considering factors like depth and activation functions.
- Set up the training environment, including selecting a deep learning framework and configuring hardware resources.
- Define the loss function and optimizer for training the GAN model.
- Train the GAN model on the noisy-clean image pairs .
- Evaluate the trained model.
- Measure denoising effectiveness quantitatively and qualitatively.
- Deploy the trained model with real time applications.

Existing System:

- In the existing system, we used Adobe Illustrator and Photoshop to enhance the quality of the image and there are many models with open cv and CNN based models which does not reach the user needed accuracy But the problem with these is that, it manipulates original values that doesn't produce a decent quality of the image, and there might be highly pixelated output or the crucial features might get displaced, and produces blurry or curvy edges.

Proposed System:

- In the proposed system, We are using Generative Adversarial Networks which uses two convolutional layers known as Generator and Discriminator.

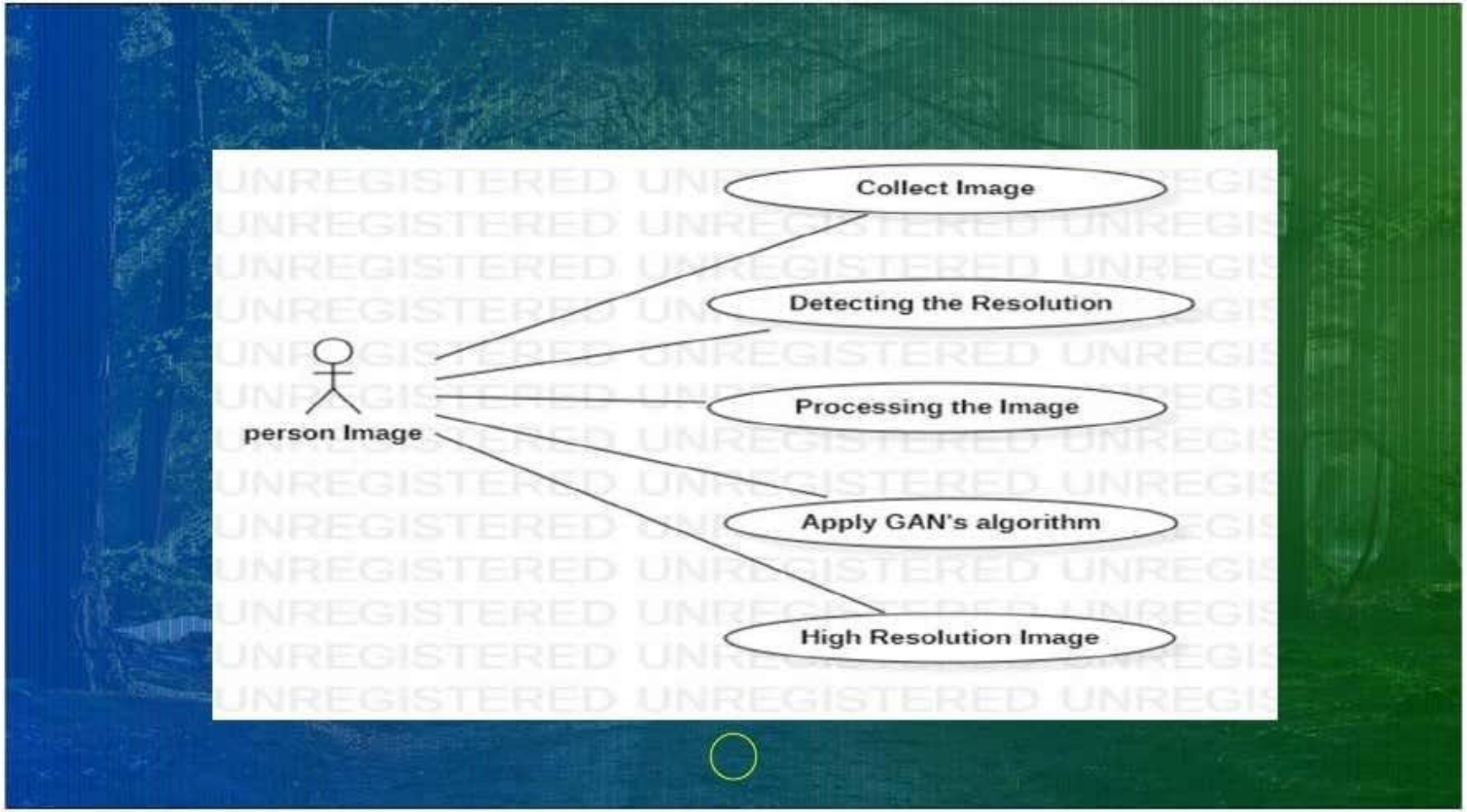
- Here the generator network is used to generate the images based on the input image, and the discriminator network takes this generated image and tells whether is true or fake.

- so here the generator tries to by-pass the discriminator and with the feedback of the discriminator the generator would try to generate a better acceptable imageBy using SRGAN, we try to retain very crucial features of the low resolution image like edges, colour textures, shadows etc.

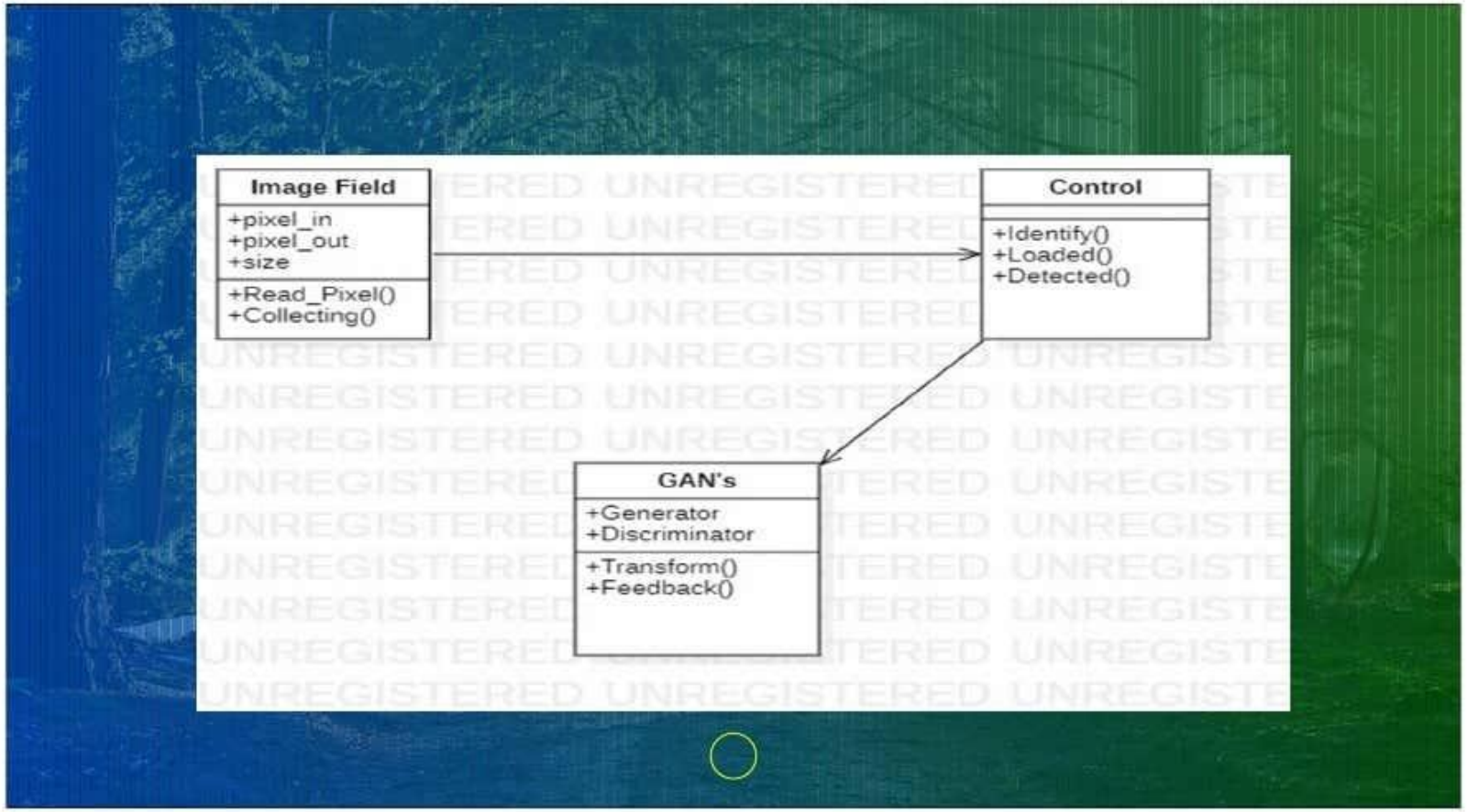
Used Algorithm:

- Generative adversarial networks (GANs) are algorithmic architectures that use two neural networks, pitting one against the other (thus the "adversarial") in order to generate new, synthetic instances of data that can pass for real data.
- They are used widely in image generation, video generation and voice generation.A generative adversarial network (GAN) has two parts:
- The generator learns to generate plausible data. The generated instances become negative training examples for the discriminator.
- The discriminator learns to distinguish the generator's fake data from real data. The discriminator penalizes the generator for producing implausible results.
- **GENERATOR**: The generator part of a GAN learns to create fake data by incorporating feedback from the discriminator. It learns to make the discriminator classify its output as real, Generator training requires tighter integration between the generator and the discriminator than discriminator training requires.
- **DISCRIMINATOR**: The discriminator in a GAN is simply a classifier. It tries to distinguish real data from the data created by the generator. It could use any network architecture appropriate to the type of data it's classifying. The discriminator's training data comes from two sources Real data instances, such as real pictures of people. The discriminator uses these instances as positive examples during training. Fake data instances created by the generator. The discriminator uses these instances as negative examples during training.

Flow Chart:


Collect Image
Detecting the Resolution
person Image
Processing the Image
Apply GAN's algorithm
High Resolution Image

Class Diagram:



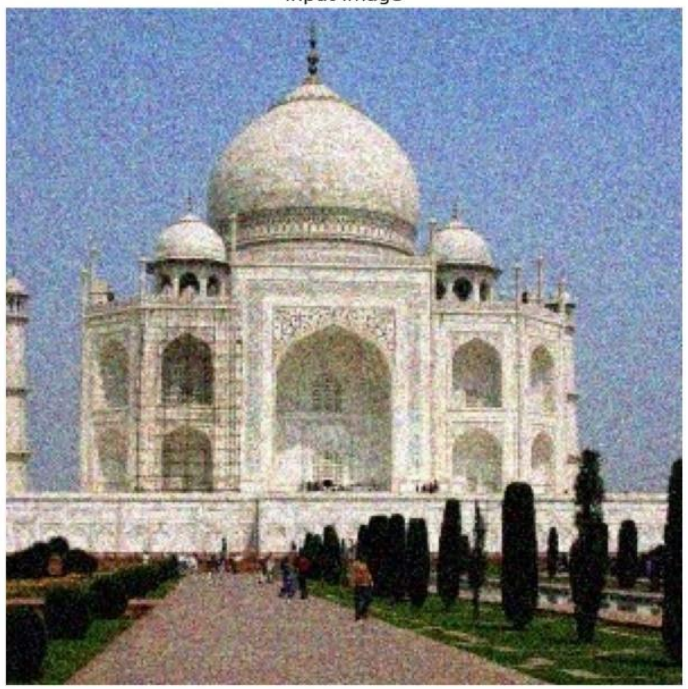
Results

Input image

GFPGAN output

Results

Input image

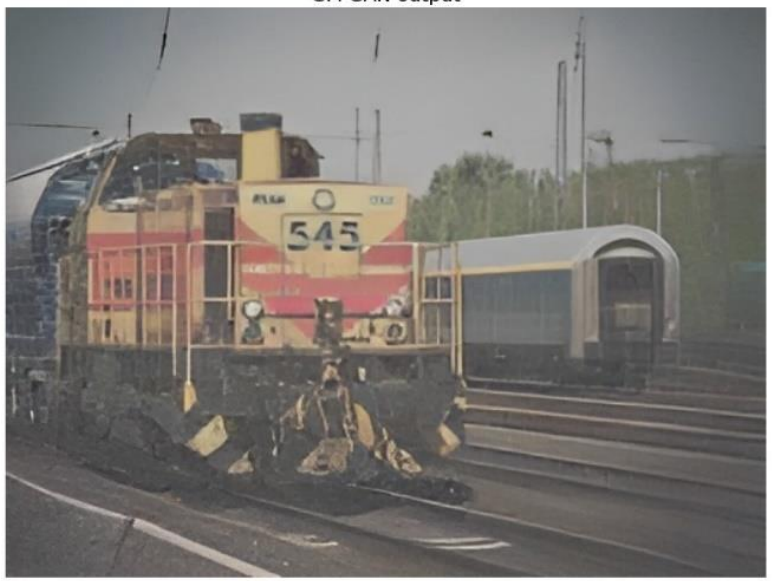


GFPGAN output

Results

Input image

GFPGAN output



Conclusion

Image denoising with Generative Adversarial Neural Networks (GANs) is a promising technique that has the potential to significantly improve image quality. By training a generator and a discriminator network together, GANs can effectively remove noise from images and produce visually appealing results. This technique has wide-ranging applications in various industries, including photography, medical imaging, and computer vision. With further advancements and research in GAN-based image denoising, we can expect even more impressive results in the future.

References

- 1.https://www.researchgate.net/publication/357828365_Image_Denoising_with_GAN_Based_Model
- 2. Yan, H.; Chen, X.; Tan, V.Y.F.; Yang, W.; Wu, J.; Feng, J. Unsupervised image noise modeling with Self-Consistent GAN. arXiv 2019, arXiv:1906.05762v1. [Google Scholar]
- 3. Yu, S.; Park, B.; Jeong, J. Deep iterative Down-Up CNN for image denoising. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Long Beach, CA, USA, 16–17 June 2017; Volume 6, pp. 2095–2103. [Google Scholar]
- 4. A. Radford, L. Metz and S. J. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," arXiv preprint arXiv:1511.06434, 2015.
- 5. Q. Yan and W. Wang, "DCGANs for image super-resolution, denoising and deblurring," Advances in Neural Information Processing Systems, pp. 487–495, 2017.

Thank You