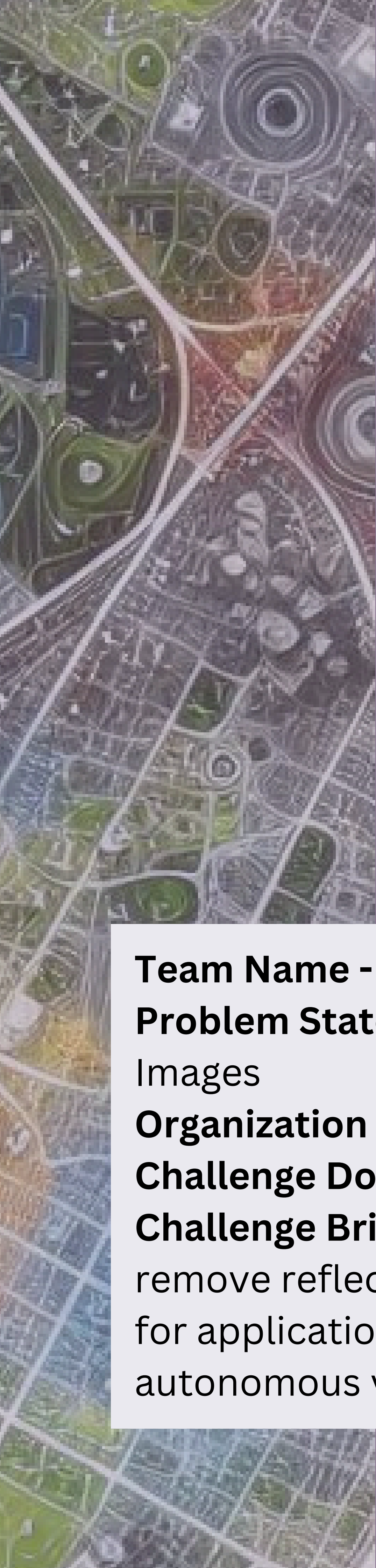
PowerPoint
PRESENTATION

Removing Reflections from Images

2024-25



Introduction

Team Name - Knot
Problem Statement - Removing Reflections from Images
Organization Name - IIT Bombay
Challenge Domain - Qualcomm VisionX
Challenge Brief - Build an AI model to detect and remove reflections from images, enhancing clarity for applications in security, photography, and autonomous vehicles.

Index

Darshan Solanki

I'm Darshan Solanki, a first-year Automotive Mechanical Engineering student at Delhi Technological University with a passion for robotics, Machine Learning, and Artificial Intelligence. Having learned the Robot Operating System (ROS), I'm eager to apply my skills to innovative robotics projects and push boundaries in the field.

Skills: C++, Python, ROS, Machine Learning, Artificial Intelligence

Contact: 9950248443

Email: solankidarshan0321@gmail.com

Saksham Kumar

I'm Saksham Kumar, an enthusiastic first-year Computer Science student at Delhi Technological University, deeply interested in the foundational technologies of robotics and AI. I love to create projects from the ground up using low level technologies

Skills - Java, Python, Game Dev, Artificial Intelligence

Contact No. - 9625908939

Email - saksham.vvip@gmail.com

Theodore Regimon

I am Theodore Regimon, a second-year Biotech Engineering student at Delhi Technological University. My passion for Machine Learning and Artificial Intelligence is a driving force in my academic journey, and I'm currently expanding my knowledge in these areas.

Skills - C++, Python, Machine Learning, Artificial Intelligence

Contact No. - 8178074978

Email ID - theoregi61@gmail.com

Aadish Saini

I am Aadish Saini, a first-year Software Engineering student at Delhi Technological University. My passion for Machine Learning and Artificial Intelligence is a driving force in my academic journey, and I'm currently expanding my knowledge in these areas. I

Skills - Python, ML, C++, Networking.

Contact No. - 9599548083

Email ID - ksainiaadish@gmail.com

Description

Reflection in images often obscures the view of underlying objects or scenes, making it difficult to analyze or interpret the actual content. This problem arises in scenarios such as taking photos through glass windows, capturing images of water surfaces, or reviewing footage from security cameras. The objective is to develop an AI model that effectively removes such reflections while preserving the details and clarity of the original scene. This solution aims to improve the quality of images for practical use cases like photography, surveillance, and autonomous systems.

(a) Input image (b) Generated image (c) Reflection layer

Project Breakdown

WE'VE BREAK THE PROJECT INTO FIVE STEPS

- DATA COLLECTION: IMAGES CONTAINING REFLECTIONS, ALONG WITH THEIR GROUND TRUTH REFLECTION-FREE COUNTERPARTS, ARE COLLECTED TO CREATE A ROBUST DATASET FOR TRAINING AND EVALUATION.

- PREPROCESSING: THE COLLECTED IMAGES ARE PADDED, NORMALIZED, AND AUGMENTED TO ALIGN WITH THE MODEL'S REQUIREMENTS, ENSURING THE DATASET IS DIVERSE AND COMPATIBLE.

- MODEL DEVELOPMENT: OUR MODEL IS DESIGNED AND TRAINED USING THE PREPROCESSED DATASET, LEVERAGING U-NET-LIKE ARCHITECTURE AND GRADIENT CONSTRAINTS TO EFFECTIVELY REMOVE REFLECTIONS.

- POSTPROCESSING: THE MODEL'S OUTPUT IS REFINED BY APPLYING CLIPPING AND CLAHE FOR CONTRAST ENHANCEMENT, FOLLOWED BY CONVERTING THE PROCESSED IMAGES BACK TO THE RGB FORMAT.

- EVALUATION: THE PERFORMANCE OF THE MODEL IS QUANTITATIVELY ASSESSED USING METRICS LIKE PSNR AND SSIM, ALONG WITH VISUAL INSPECTIONS TO VALIDATE THE RESULTS.

1. Data Collection

The first step for model training is dataset collection. Unfortunately, there is a lack of easily available large scaled datasets for reflection removal. The process of creating images with and without reflections for training process is very tedious and time consuming, hence the need to create an artificially generated dataset.

There are many possible different ways for generating an artificial dataset for reflection removal training. Most images with unwanted reflections have a primary subject on which the camera is focused, reflections are generally out of focus in pictures, and we can use that to our advantage.

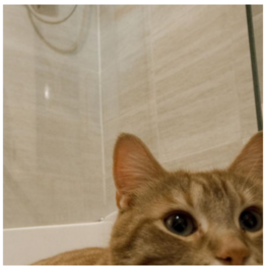
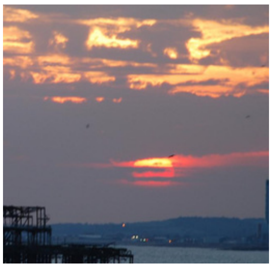
An artificial image can be assumed to be a superimposition of two images with different focal lengths.

Assume two images A(primary subject) and B(unwanted reflections).
R(synthetically generated image)can be assumed as:

$$R = [1-\alpha]A + G(\alpha B)$$

where, α = opacity constant
G = Gaussian blur function, defined as following

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$


+

=



+

=


2. Pre-processing

2.1 Padding

- Input images are padded to ensure their dimensions align with the requirements of the GCNet model. This allows the network to handle images of varying sizes without issues.

2.2 Normalization

- Pixel values of the images are normalized using predefined mean and standard deviation values. This step ensures consistency in data representation and helps the model converge more effectively during training.

2.3 Clipping

- During normalization, pixel intensity values are clipped to ensure they remain within a valid range (0 to 1). This step prevents anomalies caused by extreme pixel values.

2.4 Labelling

- Ground truth images are paired and synchronized with corresponding input data, establishing a robust foundation for supervised learning tasks.

2.5 Dataset Augmentation

- Techniques like flipping, rotation, and scaling may be applied to the input data to artificially expand the dataset, thereby improving the model's ability to generalize to unseen data.

2.6 Tiling for Large Images

- Large images may be split into smaller tiles to make them manageable for the model while ensuring no loss of critical information.

2.7 Data Loading

- The dataset is structured and loaded using a custom DataLoader, which facilitates efficient access to input images and their corresponding ground truth during training and inference.

3. Model Development

3.1 GCNet Model Architecture: We will be using GCNet model architecture for semantic segmentation tasks.

- GCNet is structured as two networks:

  - **Generative Network**: This part of the network produces an initial estimate of the clean background image from the input image (which has reflections).

  - **Corrective Network**: This network refines the generated output to further suppress any remaining reflection artifacts and enhance the quality of the reflection-removed image.

The combination of these networks allows GCNet to iteratively improve the output image, making it a powerful model for reflection removal.

3.2 Training Process

- Loss Functions: The model optimizes several loss functions that measure the similarity between the clean target image and the generated output. These losses may include:

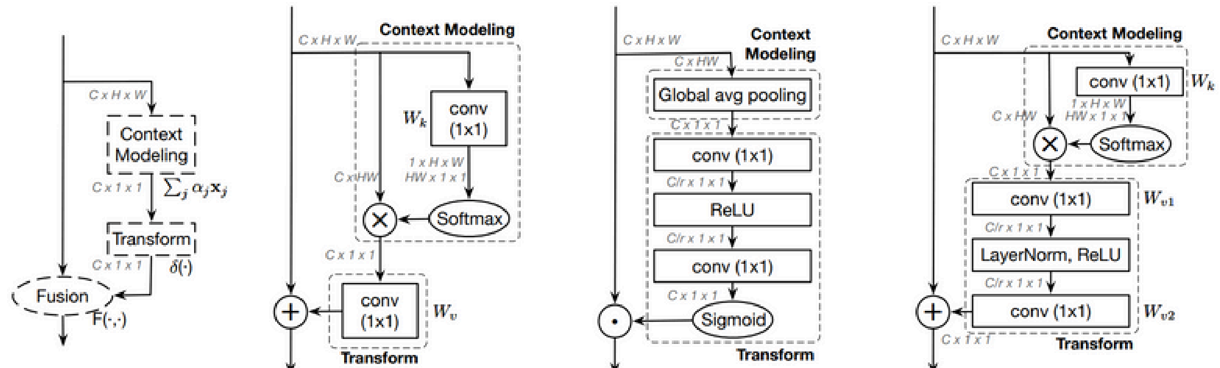
  - **Content Loss**: To ensure the output resembles the actual background content.

  - Perceptual Loss: A higher-level loss based on feature maps from a pre-trained model (like VGG) to make the output visually consistent with the clean image.

  - **Adversarial Loss**: If a GAN (Generative Adversarial Network) framework is applied, this loss helps improve the realism of the output image by training against a discriminator network.

3.3 Correction Process

- After the initial generation of a reflection-removed image, the corrective network fine-tunes it by focusing on areas where reflections might still be visible. This correction process helps GCNet produce high-quality, clear images with minimal reflection traces.



(a) Global context modeling framework (b) Simplified NL block (Eqn 3) (c) SE block (d) Global context (GC) block

4. Post-processing

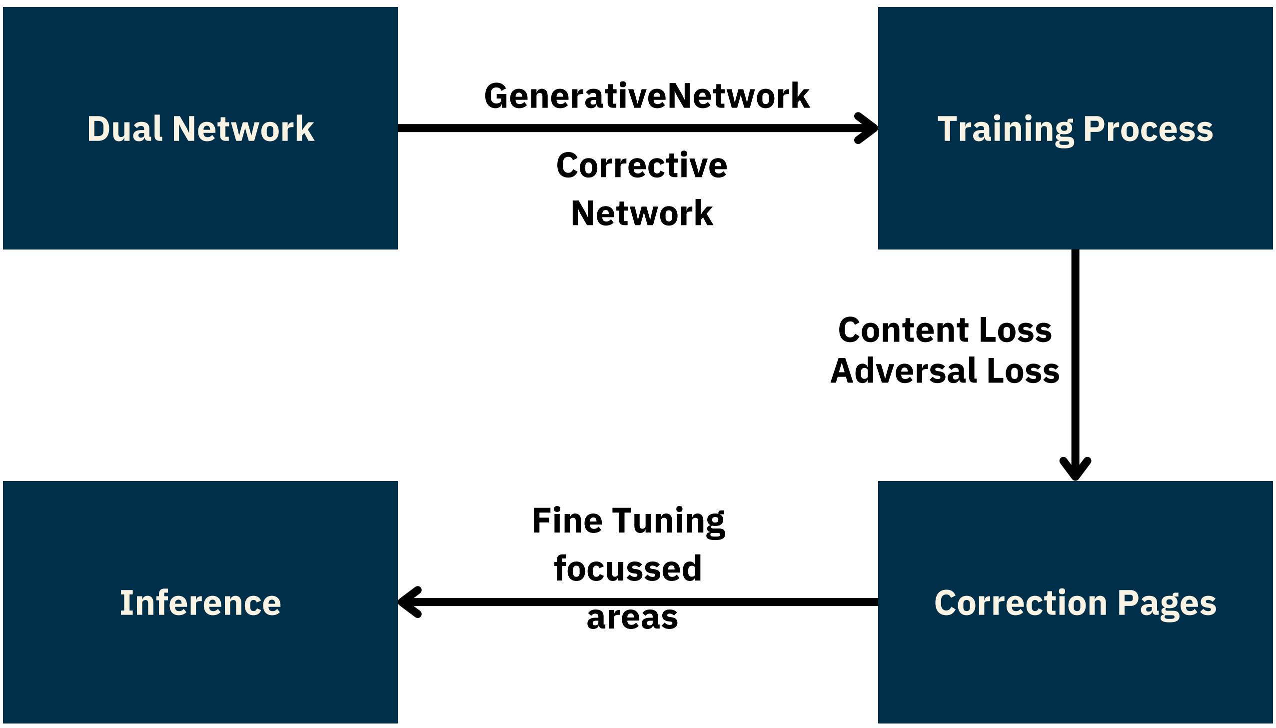
GCNet's two-stage approach (generation + correction) gives it an advantage in scenarios where standard reflection removal models might leave traces of reflection. By iterating on the correction phase, GCNet ensures cleaner and sharper results.

4.1 Thresholding: After the model predicts the mask, you may need to apply a threshold to classify pixels as either background or reflection. This helps to refine the mask and remove any small noise that may be present in the predicted output.

4.2 Morphological Operations: These operations, such as dilation or erosion, can be used to refine the mask further. For example, dilation can help close small holes in the predicted mask, while erosion can remove small artifacts.

4.3 Overlaying the Mask: Once the mask is refined, it can be overlaid on the original image to separate the reflection from the rest of the image. This allows for easy visualization of the detected reflection area.

4.4 Combining Focus and Reflection Information: In some cases, we also need to combine the focus (or depth) information with the reflection mask to enhance the result. For instance, if depth maps are available, they can help to focus on specific areas where reflections are more likely to occur.



5. Evaluation

We have used two metrics to evaluate our model to give desired, accurrate, and closest to ground truth results. The two metrics are SSIM and PSNR.

- **SSIM**

is a perceptual metric that measures the similarity between two images. It considers changes in structural information, luminance, and texture contrast, making it a better indicator of visual quality than traditional metrics like MSE (Mean Squared Error). SSIM is usually used to compare an image with a reference image.

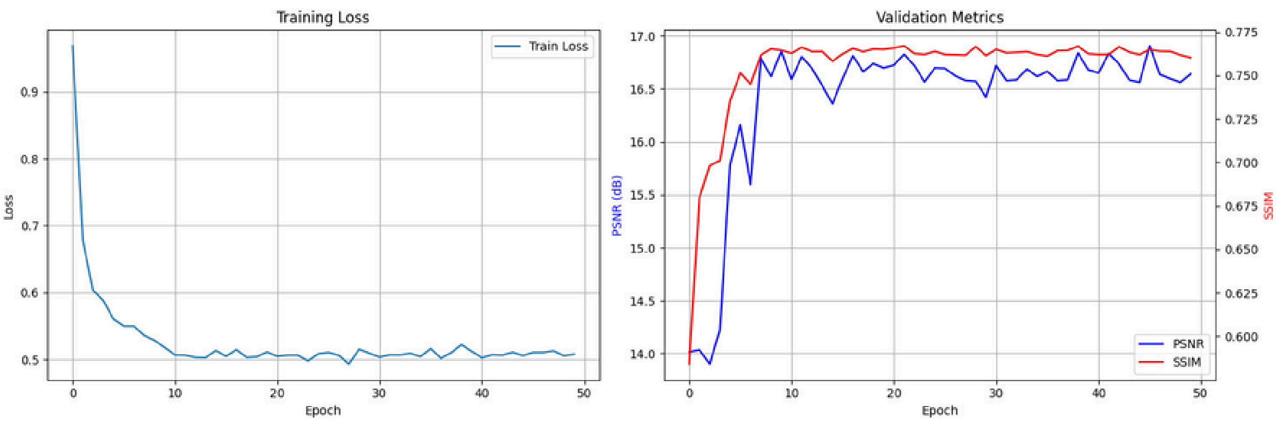
$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

- **PSNR**

is a metric for measuring the quality of an image. It compares the peak signal (the maximum pixel value, usually 255 for 8-bit images) to the noise (the difference between the original and the noisy images). A higher PSNR generally indicates better quality.

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right)$$

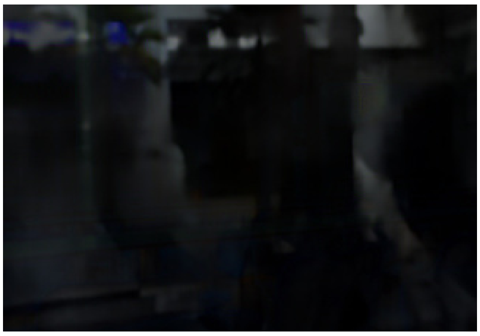
BOTH THE METRICS HAVE NOT BEEN CALCULATED MATHEMATICALLY IN CODE, BUT HAVE BEEN USED FROM THE PREBUILT LIBRARY OF scikit-image



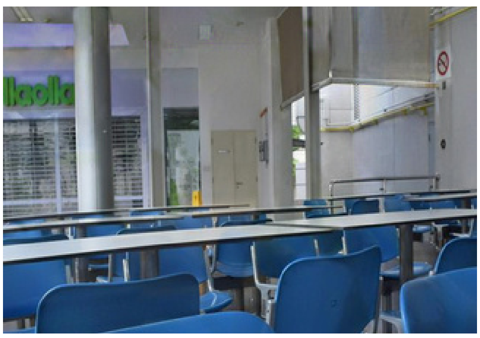
RESULT

INPUT

TRANSMISSION LAYER



OUTPUT



Future Aspects

- **Enhanced Edge Detection:** Continuous improvement of the AI/ML model to further improve edge detection, reflection removal and regeneration of image after removal

- **Faster Video Processing:** Improving model speed to generate reflection-less videos without drop in frame rate and including Multiple Image Recognition techniques in addition to Single Image Recognition to improve speed for videos.

- **Real-time Data Processing:** Explore the possibility of implementing real-time data processing, allowing the model to analyze and update images dynamically as changes occur in the environment.

- **Transfer Learning:** Implement transfer learning techniques to make the model adaptable to different lighting areas without the need for extensive retraining.

- **Improved Loss Function:** Incorporating different loss functions, like semantic analysis, edge superimposition and other techniques to improve loss function and increase model accuracy

- **Multi Model Training:** Adding other CNN's in addition to the GAN to segregate edge detection and reflection tasks and improving final result.

Applications

Reflection removal can play a huge role in our day-to-day life and various industries.

- **Personal photography and videography**

Reflection removal can be of a great use for image refinement for personal use. Glare from various light sources, unwanted reflections causing privacy invasion, camera reflections and much more.

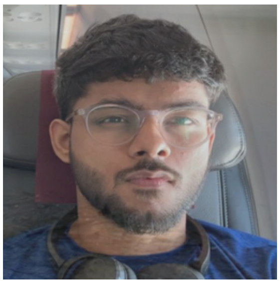
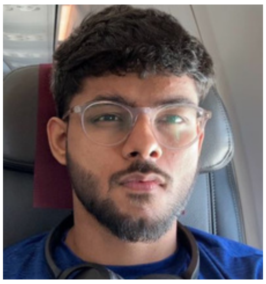


Image with unwanted reflection



Clean image

- Improving security cam footage and other crucial data perceptible to noise

- Data Pre-processing for use in model training and improving accuracies for unmanned self autonomous vehicles