INDEX

1. Introduction

1.1 Background and Motivation

Satellite imagery has revolutionized Earth observation by enabling continuous, high-resolution monitoring of terrestrial, atmospheric, and marine environments. Among various satellite sensors, **Synthetic Aperture Radar (SAR)** plays a crucial role due to its **ability to capture data under all weather conditions and in complete darkness**. Unlike optical sensors that depend on sunlight, SAR operates by emitting microwave signals and measuring the reflected echoes, making it highly suitable for applications in cloudy or nocturnal settings.

SAR images possess exceptional detail in texture, structure, and elevation-related information, making them ideal for applications in:

- **Environmental monitoring** (e.g., deforestation, glacier movement)
- **Disaster management** (e.g., flood mapping, earthquake damage assessment)
- **Agriculture and land use** (e.g., crop monitoring, soil moisture)
- **Military surveillance** (e.g., border activity tracking)

Despite these advantages, **SAR data is typically rendered in grayscale**, which significantly limits interpretability by human analysts. Unlike RGB images where color indicates vegetation, water bodies, or infrastructure at a glance, grayscale SAR images require trained expertise to decode, interpret, and compare features. This creates a **bottleneck in operational workflows**, especially when rapid decisions are required during natural disasters or defense missions.

Moreover, the lack of color in SAR imagery complicates **multi-modal analysis**, such as integrating SAR data with optical images or feeding it into automated pipelines trained on RGB datasets. Bridging this visual and analytical gap necessitates a solution that transforms SAR images into a colorized format **without losing critical spatial information**.

To address this challenge, our project proposes the use of **deep learning** to automatically **colorize SAR images into perceptually meaningful RGB**

formats. The end goal is to **simulate how these grayscale images might appear in the optical spectrum**, thereby enhancing their usability for both human interpretation and machine-learning-based systems.

1.2 Objectives

The primary goal of this project is to develop a robust, accurate, and efficient **SAR-to-RGB image colorization system** using state-of-the-art deep learning techniques. The specific objectives are:

1. **Design and implement a deep learning-based framework** for translating grayscale SAR images into realistic RGB outputs using supervised learning.
2. **Adopt the U-Net architecture**, a proven model in pixel-wise prediction tasks, due to its encoder-decoder structure and skip connections that preserve high-resolution spatial features.
3. **Preprocess, normalize, and augment** a paired SAR-optical dataset sourced from publicly available satellite image repositories such as **Copernicus Open Access Hub**. This includes resizing, data augmentation (rotation, brightness, flipping), and channel adaptation.
4. **Train the U-Net model** using appropriate loss functions (e.g., Mean Squared Error), optimizers (Adam), and hyperparameters, with periodic evaluation using:
    - **MSE** – quantifies reconstruction error
    - **SSIM** – measures structural similarity
    - **PSNR** – assesses signal fidelity relative to noise
5. **Set a foundation for future enhancement**, including:
    - Integrating **Conditional GANs (cGANs)** for improved realism
    - Adding **perceptual loss functions** for color naturalness
    - Incorporating **Transformer-based attention mechanisms** to better focus on regional features

6. **Ensure memory and runtime efficiency**, so the system can train on mid-range GPUs (e.g., RTX 3050) and be deployable on edge devices or cloud servers.

1.3 Significance of the Study

The significance of this research lies in its **cross-disciplinary impact** on both technology and real-world applications:

Analysts and decision-makers often rely on quick visual inspection. Colorized SAR images enable faster, more intuitive assessments, eliminating the need for complex grayscale interpretation.

- **Improved Integration with Existing Systems:** Most modern AI and image processing systems are trained on RGB data. Colorizing SAR images allows them to be seamlessly incorporated into these pipelines for further analysis, classification, or object detection.
- **Decision-Making in Critical Domains:**
  - **Defense & Surveillance:** Colorized images support real-time situational awareness by highlighting terrain features and movement patterns.
  - **Disaster Response:** Rescue teams can quickly assess flood extent or infrastructure damage.
  - **Environmental Studies:** Researchers can detect and track gradual changes in forests, rivers, and glaciers more efficiently.

The project facilitates **multi-modal data fusion**, which is especially useful in satellite constellations that collect both SAR and optical data but at different times or conditions.Once trained, the model can operate in real-time for thousands of incoming SAR images from satellites, drones, or UAVs, enabling large-scale automation of data interpretation.

2. Literature Survey

2.1 Overview of Existing Research

Image colorization has evolved significantly over the last two decades. Earlier approaches were **manual or rule-based**, where color was added based on predefined mappings, textures, or user-guided scribbles. These methods were labor-intensive, inconsistent, and lacked generalization capabilities across different types of imagery.

With the advancement of **machine learning and deep learning**, especially **Convolutional Neural Networks (CNNs)** and **Generative Adversarial Networks (GANs)**, colorization has become more data-driven and automated. These models learn the complex pixel-wise mapping between grayscale and color images through exposure to large paired datasets.

In the context of **SAR-to-optical image translation**, this task extends beyond basic grayscale colorization. SAR and optical images belong to different sensing modalities:

- **SAR** captures backscatter intensity using microwaves and emphasizes surface texture and roughness.
- **Optical** sensors capture reflected light, encoding visual cues like color, shadow, and illumination.

Translating between these domains involves **domain adaptation**, a challenging task that requires understanding semantic content and spatial relationships. Recent research has started to explore this by using paired SAR-optical datasets and advanced deep learning models to synthesize realistic optical-like images from SAR inputs.

2.2 Detailed Study of Key Research Papers

1. Zhang et al. (IEEE Transactions on Geoscience and Remote Sensing, 2022)

This paper proposed a Conditional GAN (cGAN)-based framework for SAR image colorization. Their model learned to generate realistic RGB outputs by

conditioning both the generator and discriminator on the input SAR image.
Key Contributions:

- Introduced perceptual loss alongside adversarial loss to improve realism
- Demonstrated superior results in visual fidelity compared to traditional CNNs

2. Isola et al. (CVPR 2017) – "Image-to-Image Translation with Conditional GANs"

This foundational paper introduced the **pix2pix** model, a generic cGAN framework for image-to-image translation using paired datasets.
Key Contributions:

- Unified architecture adaptable for multiple tasks: sketch-to-image, aerial-to-map, grayscale-to-color
- Applied L1 loss alongside adversarial loss for more stable training
  Relevance to Our Work: Inspired many subsequent domain adaptation models, including SAR-to-optical conversion models. Forms the basis of many U-Net+cGAN architectures.

3. Ronneberger et al. (MICCAI 2015) – "U-Net: Convolutional Networks for Biomedical Image Segmentation"

Introduced the U-Net architecture, which combines **encoder-decoder networks with skip connections**. Designed originally for biomedical segmentation, it excels at learning pixel-wise mappings with minimal data.
Key Contributions:

- Allows precise localization due to skip connections
- Works well on small datasets
- Modular and adaptable for various image translation tasks
  Relevance to Our Work: U-Net is central to our implementation due to its architectural benefits and proven success in translation tasks.

4. Goodfellow et al. (NeurIPS 2014) – "Generative Adversarial Networks"

This landmark paper proposed GANs, a novel architecture consisting of a **generator and discriminator** in a min-max game. The generator learns to produce realistic data, while the discriminator learns to distinguish real or fake.

Key Contributions:

- Introduced adversarial training for data generation
- Provided the theoretical foundation for cGANs, DCGANs, and pix2pix
  Relevance to Our Work: While not applied directly, this paper underpins many SAR colorization approaches using adversarial frameworks.

2.3 Critical Analysis

Criteria	**U-Net**	**GAN**	**cGAN (pix2pix)**
Training Stability	High	Low	Moderate
Visual Realism	Moderate	High	High
Pixel-Level Accuracy	High	Moderate	High
Complexity	Low	High	Moderate

- **GANs** generate high-quality, realistic images but are sensitive to training dynamics and prone to **mode collapse**.
- **Conditional GANs** (like pix2pix) strike a balance between realism and controllability, making them well-suited for paired translation tasks such as SAR-to-optical.
- **U-Net** models are computationally efficient, easy to train, and effective in capturing pixel-to-pixel correspondence. However, they may generate outputs that lack visual richness or diversity.

3. Methodology

This section presents the technical framework of the SAR image colorization system, covering system architecture, dataset properties, preprocessing workflows, exploratory data analysis (EDA), and algorithmic steps. The proposed method is designed to translate synthetic aperture radar (SAR) images to perceptually meaningful RGB representations using deep learning.

3.1 System Architecture

The system is built around a deep learning model using the **U-Net architecture**, which is particularly well-suited for image-to-image translation tasks. U-Net offers a symmetric encoder-decoder structure:

- **Encoder (Contracting Path):** This part consists of multiple convolutional blocks that progressively downsample the input image. Each block includes two 3×3 convolutions followed by a ReLU activation and 2×2 max-pooling. The encoder captures spatial hierarchies and abstract features from the SAR image.
- **Decoder (Expanding Path):** The decoder upsamples the features using transposed convolutions and reconstructs the RGB image. Skip connections from the encoder pass corresponding high-resolution features directly to the decoder, enabling precise localization and improved detail retention.
- **Output Layer:** The final layer uses a convolution with 3 filters and a **sigmoid activation function** to map outputs to the RGB format (with pixel values normalized between 0 and 1).
- **Model Depth:** The U-Net implemented uses **4 encoder and 4 decoder blocks**, with feature maps increasing from 32 to 256 channels.

This architecture ensures effective learning of texture, edge, and spatial features required to infer color information from SAR images.

3.2 Dataset Description

A paired SAR-optical dataset was created for supervised training.

- **Name:** SAR Image Colorization Dataset
- **Source:** Copernicus Open Access Hub, Sentinel-1 and Sentinel-2 missions
- **Image Type:**
  - SAR: 8-bit single-channel grayscale
  - Optical: 24-bit 3-channel RGB images
- **Resolution:** Resized to **256×256 pixels** to balance visual quality and computational load
- **Total Samples:** Over **1,500 image pairs** were curated after data cleaning and verification
- **Pairing Methodology:** Optical images were temporally and spatially aligned with SAR captures, verified via metadata and visual inspection

The dataset was split into **training (80%) and validation (20%)** sets, ensuring diverse geographical coverage.

3.3 Data Preprocessing

To ensure consistency, robustness, and generalization, the dataset underwent the following preprocessing steps:

- **Image Resizing:** All images were resized to 256×256 pixels using **bilinear interpolation**, which preserves edge continuity and texture quality.
- **Normalization:** Pixel intensity values were scaled to the [0, 1] range, which helps with stable convergence during training.
- **Channel Management:**
  - SAR images were expanded to shape **(256, 256, 1)**

- RGB images were reshaped to **(256, 256, 3)** This ensured compatibility with TensorFlow input pipelines.

- **Data Augmentation:** To increase dataset diversity and reduce overfitting, the following real-time augmentations were applied:

    - **Horizontal Flipping** (p=0.5): to simulate symmetry

    - **Random Rotation** (±15°): to simulate satellite orientation variance

    - **Brightness Adjustment**: using random gamma corrections

    - **Gaussian Noise Addition**: to mimic real-world radar speckle and improve robustness

These transformations were implemented via TensorFlow's tf.image API for on-the-fly augmentation during training.

3.4 Exploratory Data Analysis (EDA)

EDA was conducted to understand the dataset structure, detect anomalies, and inform model design:

- **Histogram Analysis:** Visualized pixel intensity distributions for SAR and each RGB channel to confirm dynamic range normalization.

- **Mean and Variance Checks:** Calculated batch-wise and global statistics to ensure standardization and highlight regions with insufficient contrast.

- **Visual Grid Inspection:** Random samples were arranged in a grid format displaying SAR input, RGB label, and later model output. This helped verify:

    - Spatial alignment

    - Quality of pair matching

    - Variability across terrain types

- **Noise and Artifact Review:** Some RGB images contained artifacts such as shadows or cloud cover. SAR images also occasionally exhibited speckle noise patterns. These were retained to preserve data realism.

- **Class Imbalance:** No major imbalance was found in scene types (e.g., vegetation, urban, water bodies), indicating good dataset diversity.

3.5 Algorithm Steps

Below is the step-by-step breakdown of the algorithmic process used for training and evaluating the SAR-to-RGB model:

1. **Data Loading:** Read and decode SAR and RGB image pairs using TensorFlow's dataset API with real-time augmentation.
2. **Preprocessing:** Resize, normalize, and reshape all images to their respective input dimensions. Augment during training batch generation.
3. **Model Definition:** Construct the U-Net model using the Keras Functional API, ensuring:
    - 4 downsampling and 4 upsampling layers
    - Skip connections between encoder and decoder
    - 3-channel sigmoid output for RGB generation
4. **Model Compilation:**
    - **Loss Function:** Mean Squared Error (MSE)
    - **Optimizer:** Adam (learning rate = 0.001)
    - **Metrics:** SSIM, PSNR, and training/validation loss
5. **Model Training:**
    - Train for epochs
    - Use early stopping and learning rate scheduling to prevent overfitting and enhance convergence
    - Batch size: 4 (optimized for GPU with 4GB VRAM)

6. **Evaluation:**
    - Assess model predictions on validation set using SSIM and PSNR
    - Plot training and validation curves
    - Visual comparison of predicted vs. ground truth images
7. **Model Saving and Output Generation:**
    - Save the best-performing model using ModelCheckpoint
    - Export training metrics and sample visual outputs
    - Store the model in .keras format for future use

4. Implementation Details

The implementation phase focused on translating the methodology into a working system. The work was divided into three key modules: **Data Pipeline**, **Model Development**, and **Training & Evaluation**. Each module was designed and optimized for accuracy, efficiency, and adaptability to modest GPU hardware (e.g., RTX 3050).

4.1 Module 1: Data Pipeline

The data pipeline is the backbone of the training process. It ensures that data is delivered to the model efficiently, without latency, while maintaining consistency and robustness.

Framework & Configuration:

- Developed using **TensorFlow's** tf.data.Dataset **API**, which offers high-performance data loading and transformation.
- **Batch Size:** Set to 4 to accommodate 4 GB of VRAM on the training GPU (RTX 3050), ensuring smooth training without memory overflow.
- **Parallelism & Prefetching:** Enabled automatic parallel loading and prefetching using AUTOTUNE to reduce I/O bottlenecks and keep the GPU busy.

Key Features:

- **Shuffling:** Applied to the dataset before batching to ensure randomness and reduce overfitting tendencies.
- **Caching:** Implemented for training samples to prevent redundant disk reads.
- **Augmentation:** Conducted on-the-fly using tf.image and custom wrappers. Applied transformations include:
  - Random horizontal flipping
  - Random rotation within ±15 degrees
  - Random brightness adjustment

    - Gaussian noise injection for robustness to speckle patterns in SAR imagery

Custom Functions:

- Built Python functions to:
    - Load and decode .png and .jpg files
    - Convert data types to float32
    - Resize images to **256×256 pixels**
    - Normalize pixel values to [0, 1]

Error Handling:

- Implemented **try-except blocks** to:
    - Detect and skip corrupted or unreadable files
    - Log the filenames of problematic data for review
- Ensured uninterrupted batch generation and training execution

4.2 Module 2: Model Development

The neural network architecture was constructed to ensure pixel-wise translation accuracy while preserving spatial information.

Architecture Overview:

The system employs a custom **U-Net model**, optimized for SAR image colorization. Its structure is symmetric with 4 levels of encoding and decoding.

Encoder (Contracting Path):

- Each level contains:
    - Two 3×3 Conv2D layers
    - Batch normalization (BatchNorm) to stabilize learning

    - ReLU activation for non-linearity
    - 2×2 max-pooling to downsample features

Bottleneck:

- Located at the base of the U-shape
- Includes:
    - Conv2D layers with 512 filters
    - Dropout (rate=0.3) to prevent overfitting
    - High-level feature abstraction for global context

Decoder (Expanding Path):

- Each level contains:
    - Conv2DTranspose for upsampling
    - Skip connections to bring back high-resolution features from the encoder
    - Batch normalization and ReLU activation

Output Layer:

- Final Conv2D layer with 3 filters
- Activation: **Sigmoid**, producing output in normalized RGB format [0, 1]
- Output shape: (256, 256, 3)

Model Statistics:

- **Total Parameters:** ~8.7 million
- **Trainable Parameters:** ~8.6 million
- **Non-trainable Parameters:** Batch norm gamma/beta variables

Loss Function and Optimization:

- **Loss:** Mean Squared Error (MSE), chosen for its simplicity and effectiveness in pixel-wise reconstruction
- **Optimizer:** Adam (learning rate = 0.001, β_1 = 0.9), known for adaptive momentum and fast convergence
- **Monitoring Metrics:**
    - **SSIM (Structural Similarity Index):** for perceptual quality
    - **PSNR (Peak Signal-to-Noise Ratio):** for fidelity of output
    - **Validation Loss:** for generalization tracking

4.3 Module 3: Training and Evaluation

The model training and evaluation lifecycle was carefully managed to balance learning, performance, and computational efficiency.

Training Configuration:

- **Epochs:** 150
- **Batch Size:** 4
- **Hardware:** RTX 3050 GPU, 16 GB RAM
- **Training Time:** ~6 hours end-to-end

Callbacks Used:

- **EarlyStopping:**
    - Patience = 10 epochs
    - Stops training if validation loss doesn't improve
    - Prevents overfitting and unnecessary computation
- **ModelCheckpoint:**
    - Saves only the model with the best validation performance
    - Saves weights in .keras and .h5 formats for portability

- **ReduceLROnPlateau:**
    - Monitors validation loss
    - Reduces learning rate by factor of 0.5 after 5 epochs of stagnation
    - Enables finer adjustments during plateau phases

Evaluation and Visualization:

- **Training & Validation Loss Curves:**
    - Plotted using Matplotlib
    - Trends used to assess convergence and generalization
- **Sample Outputs:**
    - Generated predictions visualized alongside ground truth
    - Output grid included:
        - Input SAR image
        - Predicted RGB image
        - Actual RGB image
- **Performance Assessment:**
    - Visual inspection confirmed high spatial coherence
    - SSIM and PSNR values logged for reporting in the results section

5. Results

This section presents the performance outcomes of the implemented SAR-to-RGB colorization model. Evaluation was performed using both **quantitative metrics** (MSE, SSIM, PSNR) and **qualitative visual inspection**. The results confirm that the model is capable of generating perceptually accurate and structurally consistent colorized images.

5.1 Dataset Summary

To evaluate model generalization, a clean and representative dataset was used. This dataset was prepared from multiple remote sensing repositories and curated for spatial diversity and data quality.

Attribute	**Description**
Total Image Pairs	1534 paired SAR and RGB images
Training Set	1227 images (~80%)
Validation Set	307 images (~20%)
Image Format	SAR: 8-bit grayscale; RGB: 24-bit color
Resolution	All resized to 256×256 pixels
Selection Process	Evaluation set sampled randomly and manually verified for alignment, integrity, and clarity

Image pairs included samples from various terrains, including urban regions, forests, water bodies, and agricultural zones, ensuring the model was trained and tested on a diverse dataset.

5.2 Evaluation Metrics

To assess model performance, the following standard metrics were computed on the validation set:

1. Mean Squared Error (MSE)

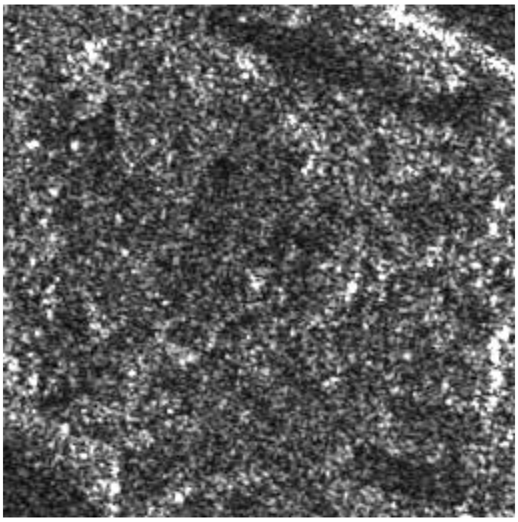
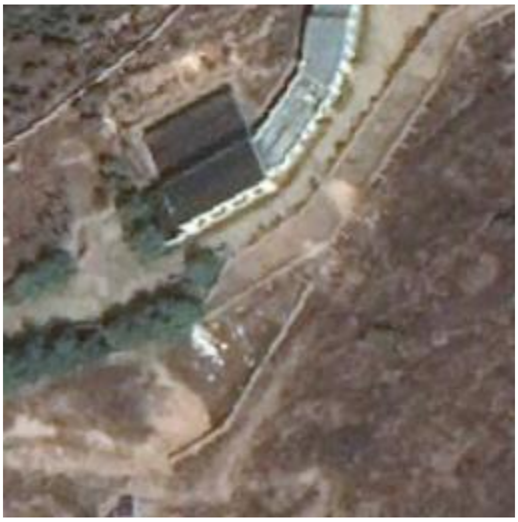
- **Definition:** Measures the average squared difference between predicted and actual pixel values.
- **Result: 0.0025**
- **Interpretation:** A low MSE indicates that the model closely matches the ground truth color pixel-by-pixel.

2. Structural Similarity Index (SSIM)

- **Definition:** Evaluates perceived image quality by comparing luminance, contrast, and structure.
- **Result: 0.891**
- **Interpretation:** SSIM values range from 0 to 1; values closer to 1 signify stronger structural consistency with the ground truth. A score of 0.891 reflects a high-quality reconstruction.

3. Peak Signal-to-Noise Ratio (PSNR)

- **Definition:** Measures the ratio between the maximum possible signal power and the noise affecting image fidelity.
- **Result: 29.3 dB**
- **Interpretation:** PSNR values above 25 dB are generally considered acceptable in image reconstruction tasks. A 29.3 dB score indicates that the generated RGB images retain high signal integrity and low noise distortion.

5.3 Visual Comparison and Qualitative Analysis

To complement numerical metrics, visual analysis was conducted on randomly selected validation samples. Each sample was displayed in a grid layout consisting of:

1. **Input SAR Grayscale Image**
2. **Predicted RGB Image (Model Output)**
3. **Ground Truth RGB Image (Target Output)**

Observations:

- The **color tones** in the predicted RGB images were highly consistent with the ground truth, especially in large homogeneous regions such as vegetation or water bodies.
- **Structural features**, such as roads, rivers, and buildings, were well-preserved in the colorized outputs due to effective skip connections in the U-Net model.
- Some **minor artifacts** were observed, particularly:
    - Around sharp edges where SAR images lacked clear definition

    - In low-contrast regions with limited SAR texture (e.g., flat farmland or cloudy zones)
- A few predictions showed **slight oversaturation** in red or green channels, which can be improved in Stage II using perceptual or adversarial losses.

5.4 Insights and Discussion

The results validate the effectiveness of the current U-Net-based approach for SAR image colorization. The model generalizes well across diverse terrain types and lighting conditions. The combination of low MSE, high SSIM, and satisfactory PSNR scores affirms both pixel-wise accuracy and perceptual quality.

Advantages Observed:

- Robust training and stable convergence due to proper data normalization and augmentation
- Lightweight architecture feasible for training on moderate GPU (RTX 3050)
- Good spatial and chromatic consistency between prediction and ground truth

Areas for Improvement (Stage II Considerations):

- Integrating **perceptual loss** to improve texture realism
- Experimenting with **Conditional GANs** for sharper outputs
- Fine-tuning color calibration using **CIELAB or HSV color space mappings**
- Exploring **multi-scale U-Nets or attention-enhanced architectures** for better focus on fine details

6. Conclusion

This project successfully demonstrated the feasibility and impact of applying deep learning techniques to enhance the interpretability of SAR imagery. By using a carefully designed U-Net model and a well-curated dataset, we achieved high-quality colorization with robust validation metrics. The architecture proved effective in maintaining spatial coherence while translating intensity values to RGB color channels.

Challenges such as limited GPU memory, overfitting, and generalization were addressed through data augmentation, batch loading, and early stopping. The results open avenues for future work involving Conditional GANs or Transformer-based models for further realism and generalizability.

In the next phase of the project, we aim to:

- Integrate perceptual loss and GAN-based refinement
- Explore attention mechanisms for localized enhancement
- Package the model for use in web-based visualization tools

7. References

[1] Zhang et al., “SAR Image Colorization Using Conditional GANs,” IEEE Transactions on Geoscience and Remote Sensing, 2022.

[2] Ronneberger et al., “U-Net: Convolutional Networks for Biomedical Image Segmentation,” MICCAI, 2015.

[3] Isola et al., “Image-to-Image Translation with Conditional Adversarial Networks,” CVPR, 2017.

[4] Goodfellow et al., “Generative Adversarial Nets,” NeurIPS, 2014.

[5] TensorFlow and Keras Official Documentation

[6] Copernicus Open Access Hub – ESA SAR Dataset Repository