Physics-Informed Machine Learning Models for Indoor Wi-Fi Access Point Placement

2021 IEEE AP-S Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting
December 6, 2021

Guoli Yang, Shuyang Luo, Shichen Ji,
Dongfang Cui, Aristeidis Seretis, Costas D. Sarris

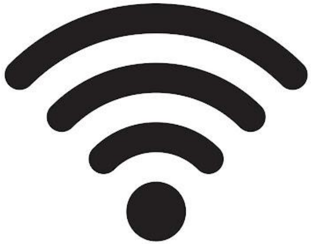
The Edward S. Rogers Sr. Department of Electrical and Computer Engineering, University of Toronto
Toronto, ON, Canada

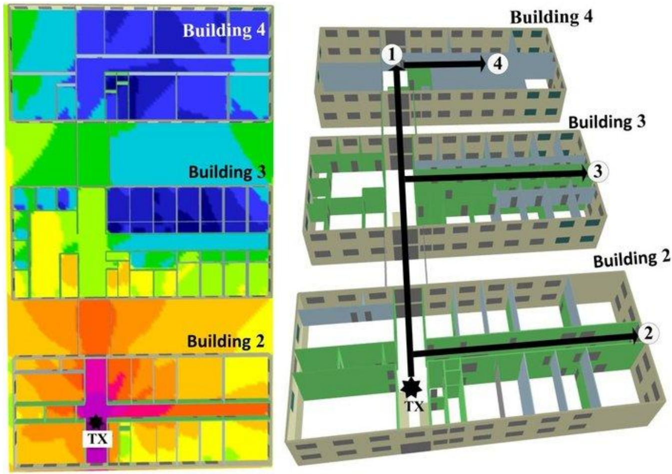
Submitted on May 14, 2021

UNIVERSITY OF TORONTO

Background

- In 2019, there were 362 million public Wi-Fi hotspots available worldwide [1]
- Optimization of Wireless Access Point (WAP) placement improves:
  - Wi-Fi coverage
  - Quality of Internet service
- Wi-Fi Signal decays during the propagation.
- An optimizer needs estimation of the Received Signal Strength (RSS) given different WAP locations
- Ray-tracing simulation is required during the optimization process





An example of power map [2]

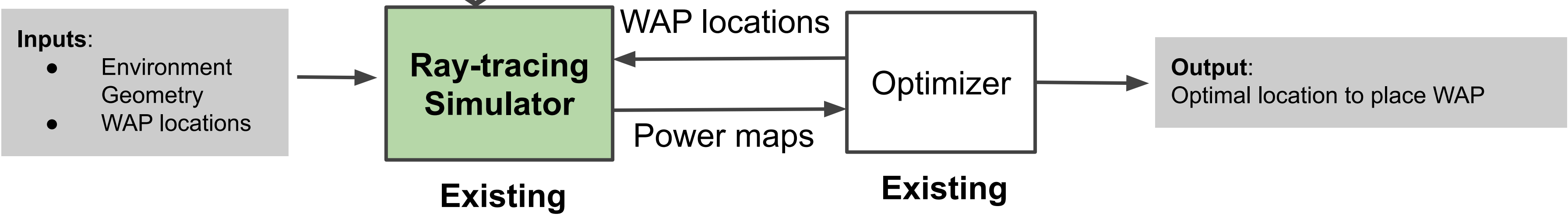
UNIVERSITY OF TORONTO

[1] L. S. Vailshery, “Global public wi-fi hotspots 2016-2022,” *Statista*, 22-Jan-2021. [Online]. Available: https://www.statista.com/statistics/677108/global-public-wi-fi-hotspots/. [Accessed: 01-Nov-2021].
[2] Fagiani, A. & Vogel, M. & Cerqueira S. Jr, Arismar. (2018). Material Characterization and Propagation Analysis of mm-Waves Indoor Networks. Journal of Microwaves, Optoelectronics and Electromagnetic Applications. 17. 628-637. 10.1590/2179-10742018v17i41548.

Problem & Motivation

How does it work:
- Uses time-consuming ray-tracing Algorithm
- Tracks each ray along its propagation

Features:
- Requires **extensive computational resources**
- **Extensive delay**, inefficient during optimization process



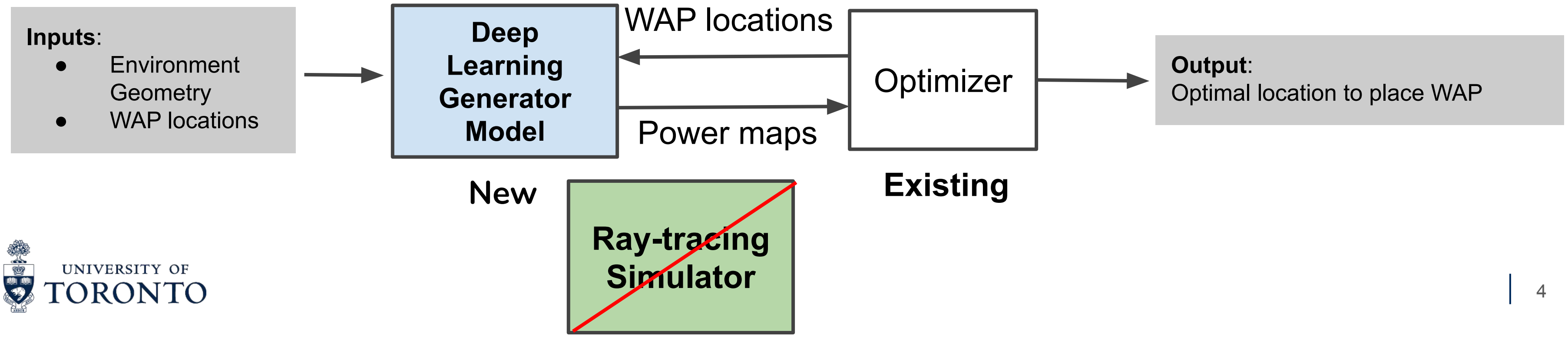
Goal

How does it work:

- Uses deep neural network (machine learning)

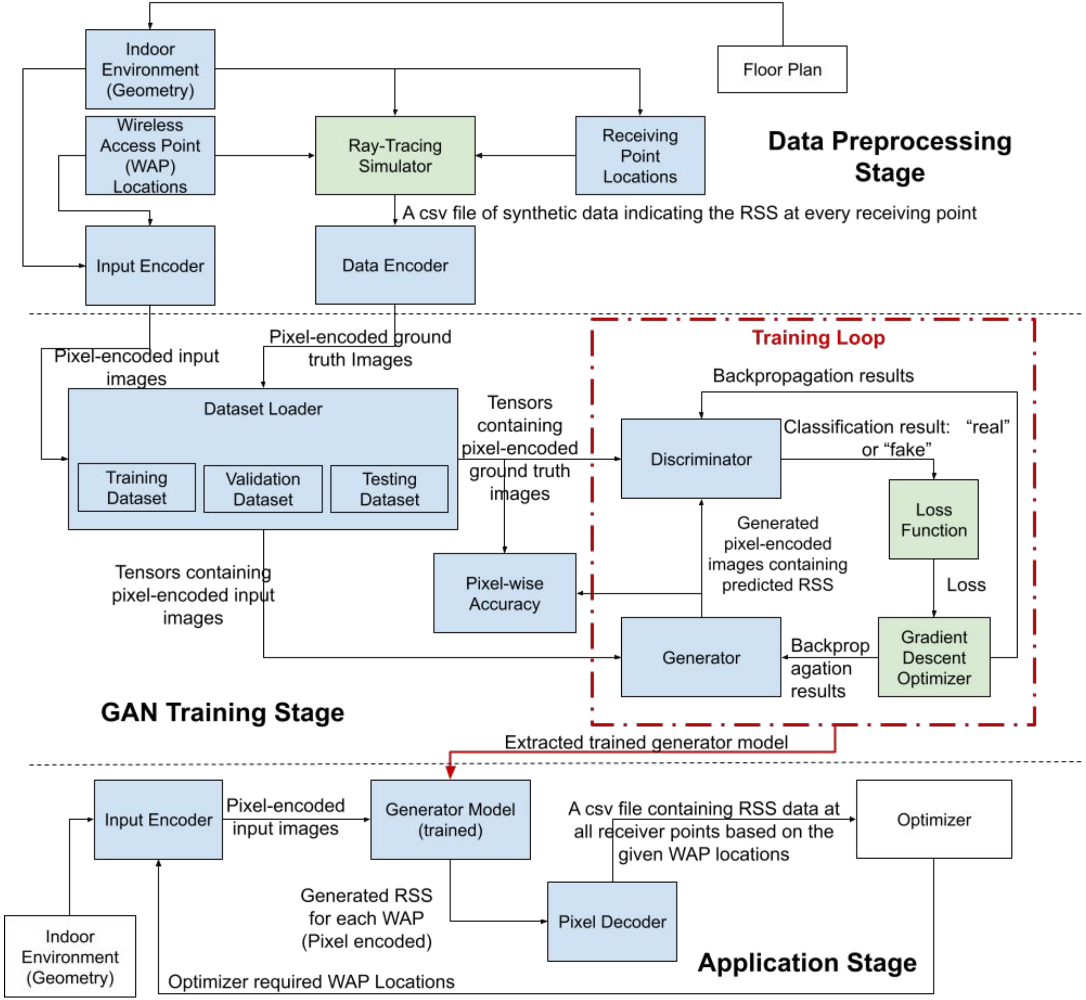
Features:

- Requires much **less computational resources**
- **Less runtime**, **more efficient** during optimization process



System Overview

- Data Preprocessing Stage
- GAN Training Stage
- Application Stage

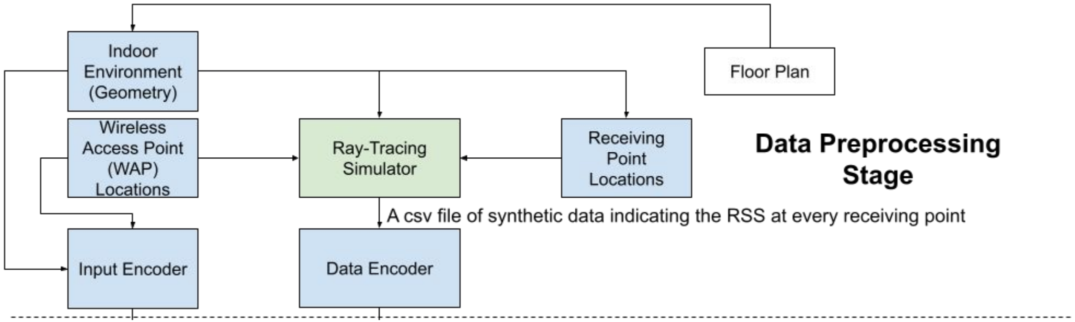


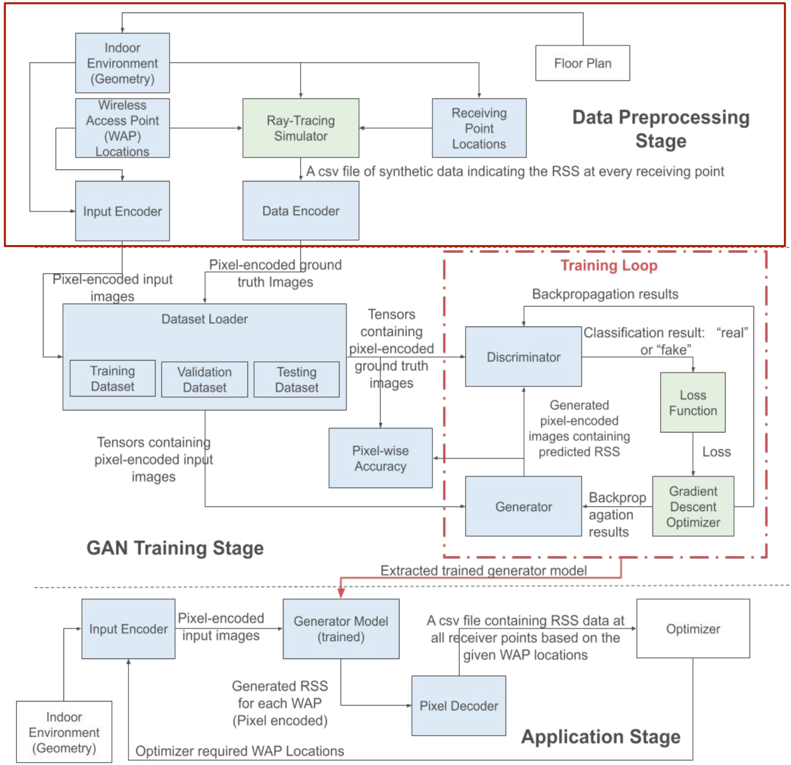
* White boxes: user inputs
* Green boxes: provided or external resources
* Blue boxes: implemented modules

System Overview: Data Preprocessing Stage

Purpose: Collect and encode input and data (ground truth) for training network

- Main Modules:
  - Ray-Tracing Simulator
  - Data Encoder
  - Input Encoder

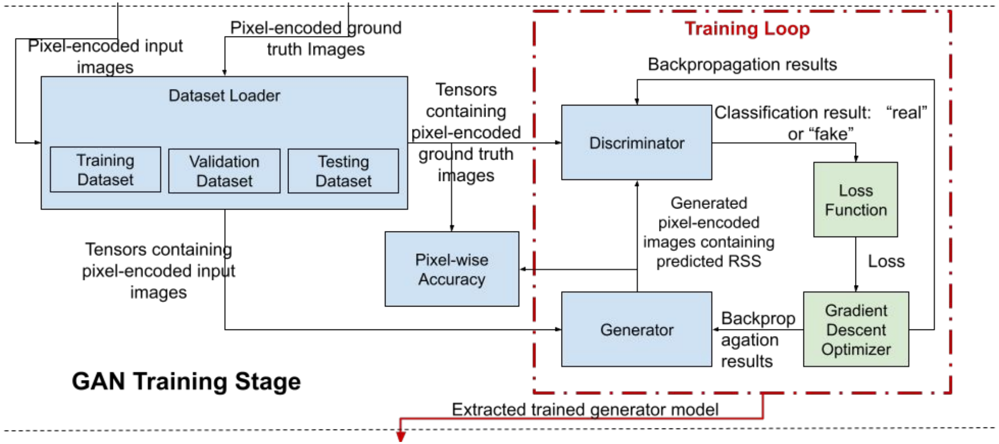


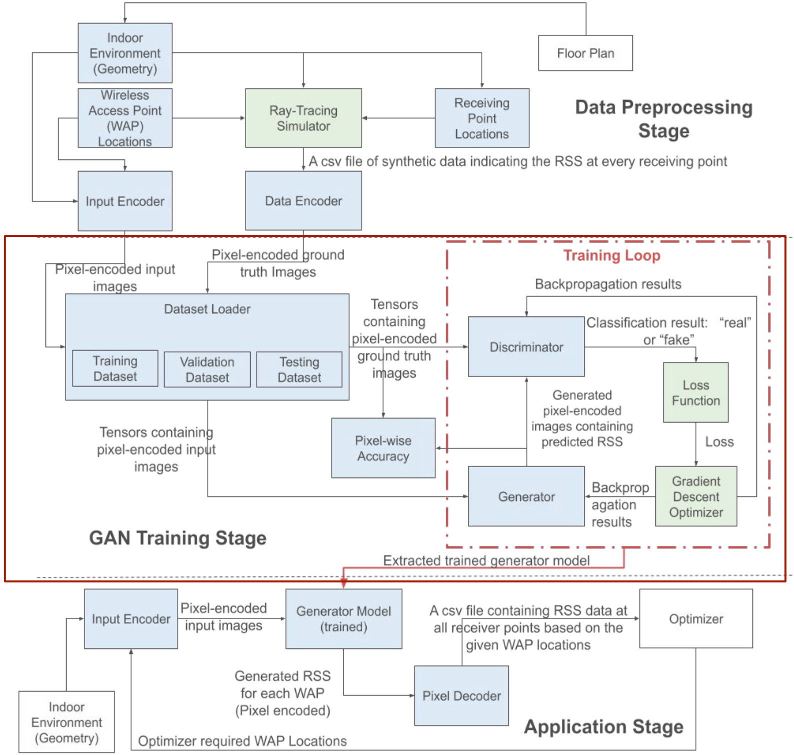


System Overview: GAN Training Stage

Purpose: Train the generator to generate high quality pixel encoded images

- Main Modules:
  - Dataset Loader
  - Training Loop

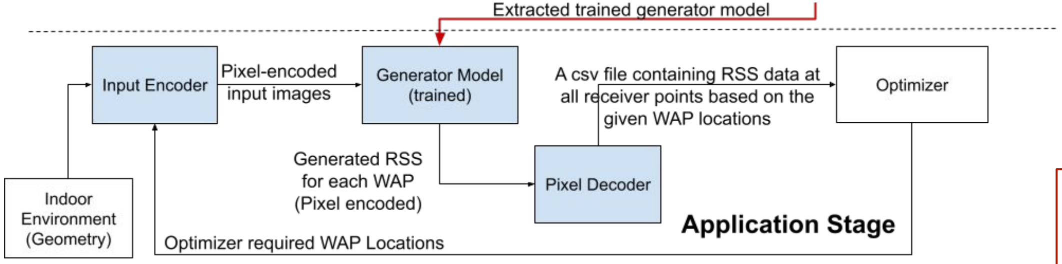


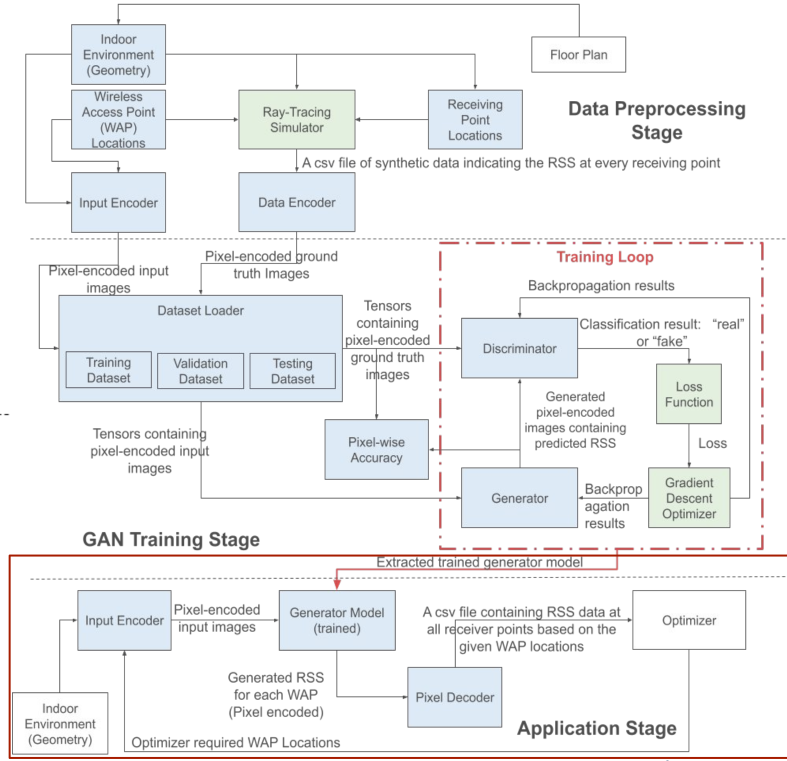


System Overview: Application Stage

Purpose: Enable our trained generator model to generate RSS data with given inputs

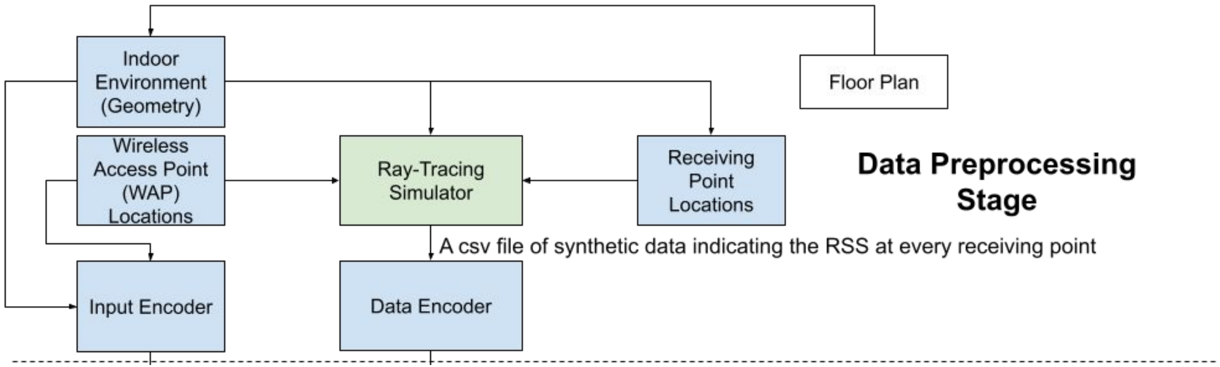
- Main Modules:
  - Generator Model (Trained)
  - Input Encoder
  - Pixel Decoder





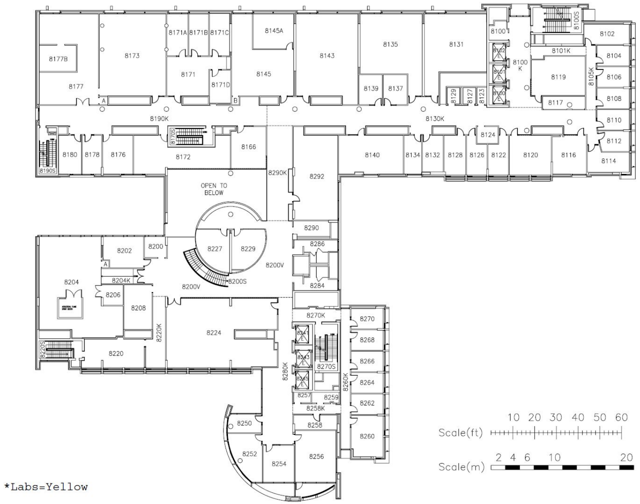
Data Preprocessing Stage

- Data Collection:
  - Ray-tracing simulator generating RSS data
- Encoding:
  - Data (Ground Truth) Encoder
    - Encoding collected RSS data into images
  - Input Encoder
    - Encoding WAP into images

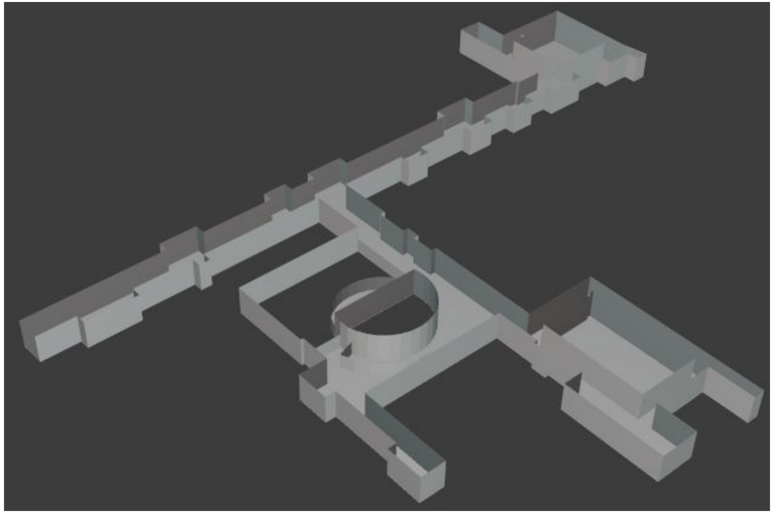


Data Collection

- Indoor environment for experiment: **the hallway of 8th floor, Bahen Centre for Information Technology, University of Toronto**
  - Generated a 3D geometry model based on its floor plan



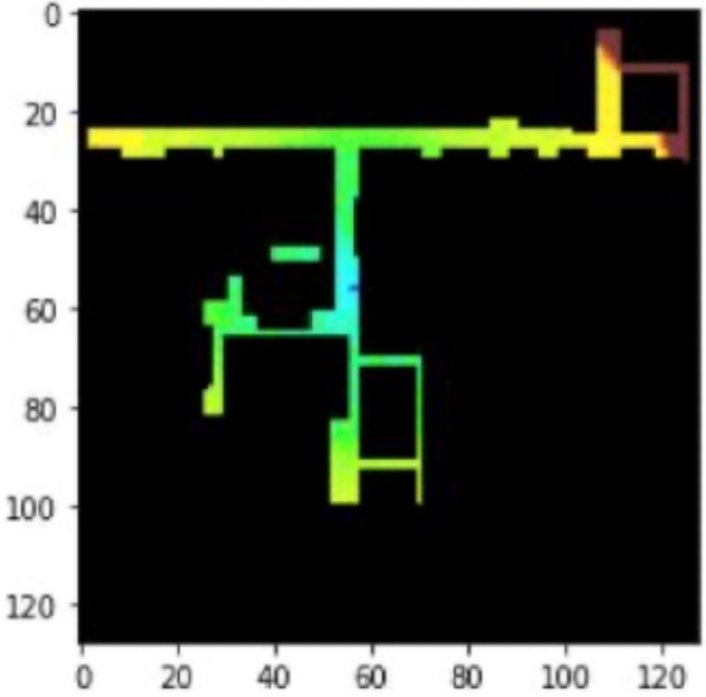
Floor Map of Indoor Environment



Generated 3D Geometry Model

Data Collection

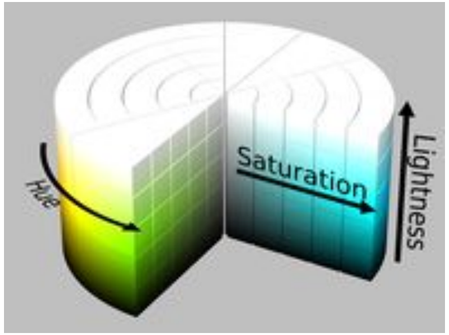
- Using a ray-tracing simulator
- Signal receiving points:
  - 0.625 m apart from each other
  - 1296 receiving points in total
- Properties for walls/boundaries:
  - Dielectric permittivity: 5 F/m
  - Dielectric conductivity: 0.1 S/m



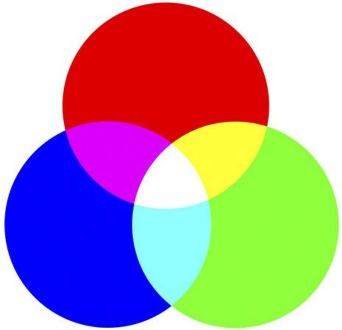
Power Map Visualization for Generated RSS Data

Encoding: Data (Ground Truth) Encoder

- **Raw Ground Truth RSS data → Pixel-Encoded Ground Truth Images**
  - Step 1: RSS-colour conversion
    - **RSS data in dBm** → hue in HSL
    - Fixed saturation and lightness level
    - HSL colour → **RGB colour pixels**



HSL Colour Scale [1]



RGB Colour Scale [2]

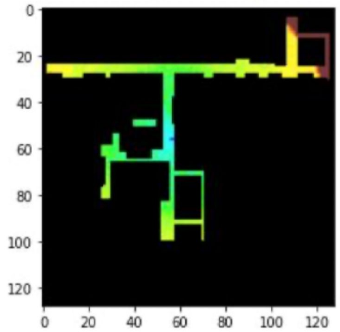
[1] J. J. Martín-Sotoca, A. Saa-Requejo, R. Moratiel, N. Dalezios, I. Faraslis, and A. M. Tarquis, “Statistical analysis for satellite-index-based insurance to define damaged pasture thresholds,” Natural Hazards and Earth System Sciences, vol. 19, no. 8, pp. 1685–1702, 2019.
[2] “RGB,” RGB (Red Green Blue) Definition. [Online]. Available: https://techterms.com/definition/rgb. [Accessed: 02-Nov-2021].

Encoding: Data (Ground Truth) Encoder

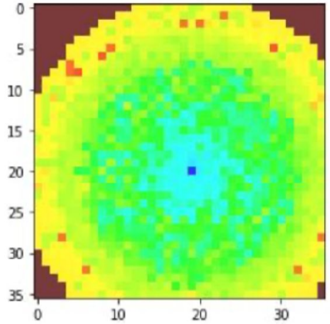
- **Raw Ground Truth RSS data → Pixel-Encoded Ground Truth Images**
  - Step 2: Physical-to-pixel coordinate mapping
    - WAP locations have unique pixel coordinates
    - For receiving points, calculate F_{recv}:

$$F_{recv} = \frac{RSS_{recv} - \min RSS}{\max RSS - \min RSS} - \frac{\|P_{recv} - P_{WAP}\|_2}{\max \|P - P_{WAP}\|_2}$$

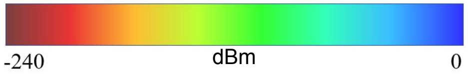
    - Higher F_{recv} value, pixel coordinate closer to WAP
  - Advantages:
    - All pixels in encoded images are useful
    - Observable patterns in encoded ground truth image



Power Map Visualization for Generated RSS Data

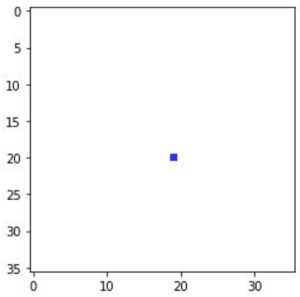


Encoded Ground Truth Image

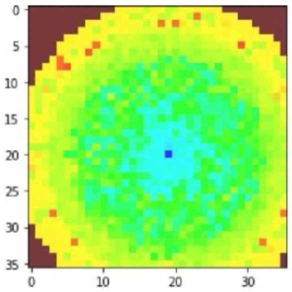


Encoding: Input Encoder

- **WAP Location → Pixel Encoded Input Images**
  - Each WAP location has a unique pixel coordinate
  - Same pixel coordinate as on the encoded ground truth image

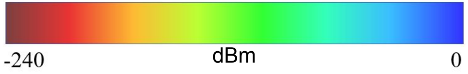


Encoded Input Image



Encoded Ground Truth Image

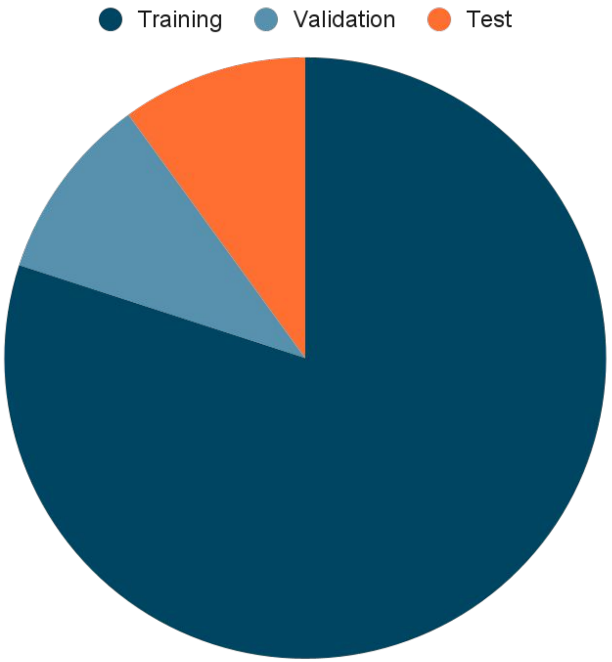
WAP Coordinate: (0.9375, -9.6875)



Dataset

Size of dataset: 1296

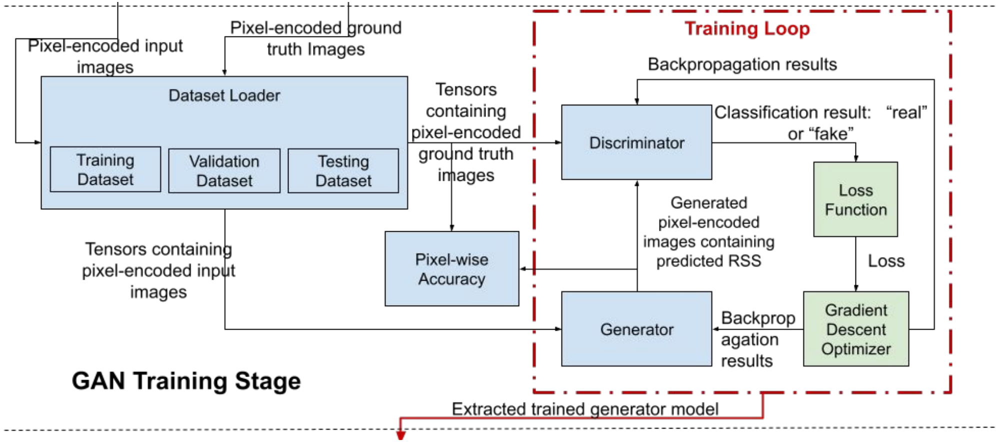
- Training dataset: 80%
- Validation dataset: 10%
- Testing dataset: 10%



GAN Training Stage: Deep Convolutional Generative Adversarial Network

- **Adversarial Training Strategy**
    - Competitions between models
- **Benefits**
    - Learn high resolution features from data
    - Generate high quality power maps

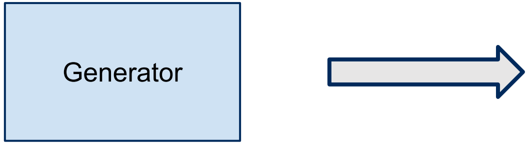
- **General Architecture**
    - Generator: Generate power maps
        - 6-Level U-Net [1]
    - Discriminator: Detect fake power maps
        - 4-Layer CNN
    - Loss Function: Binary cross entropy
    - Gradient Descent Optimizer: ADAM [2]

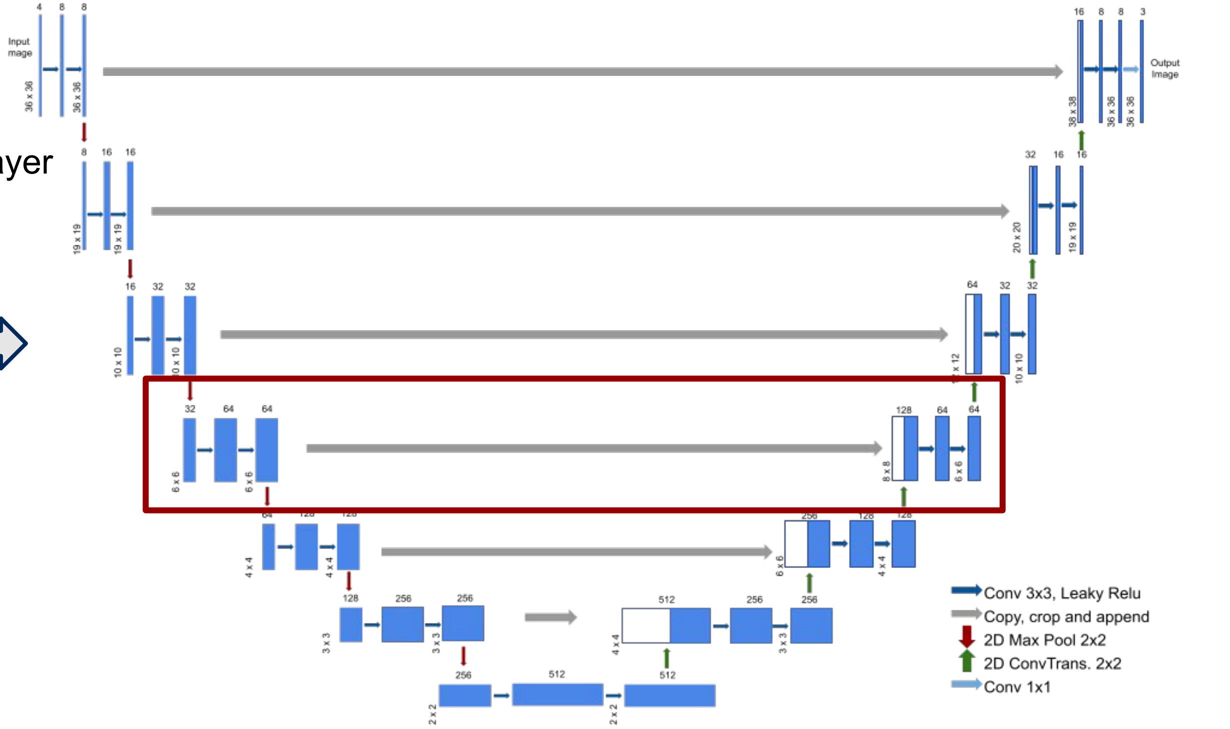


[1] O. Ronneberger, P. Fischer and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation", Computer Science Department and BIOSS Centre for Biological Signalling Studies, University of Freiburg, Germany, 2015.
[2] D. Kingma and J. Ba, "ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION", 2017

Generator: 6-Level U-Net

- **Down Sampling: 6 Levels**
  - Max-pooling layer
- **Up Sampling: 6 Levels**
  - Transpose convolutional layer

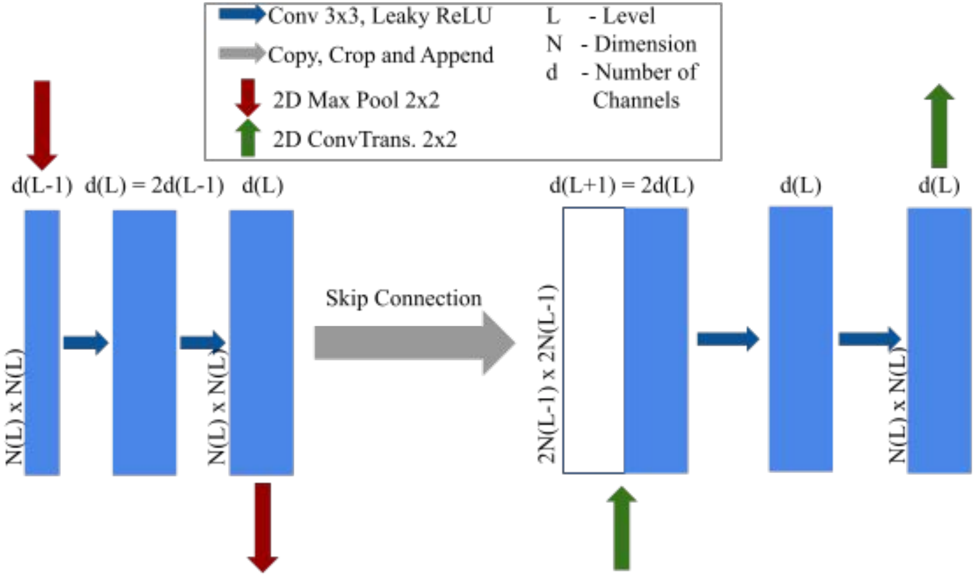




Architecture of 6-Level U-Net

Generator: 6-Level U-Net

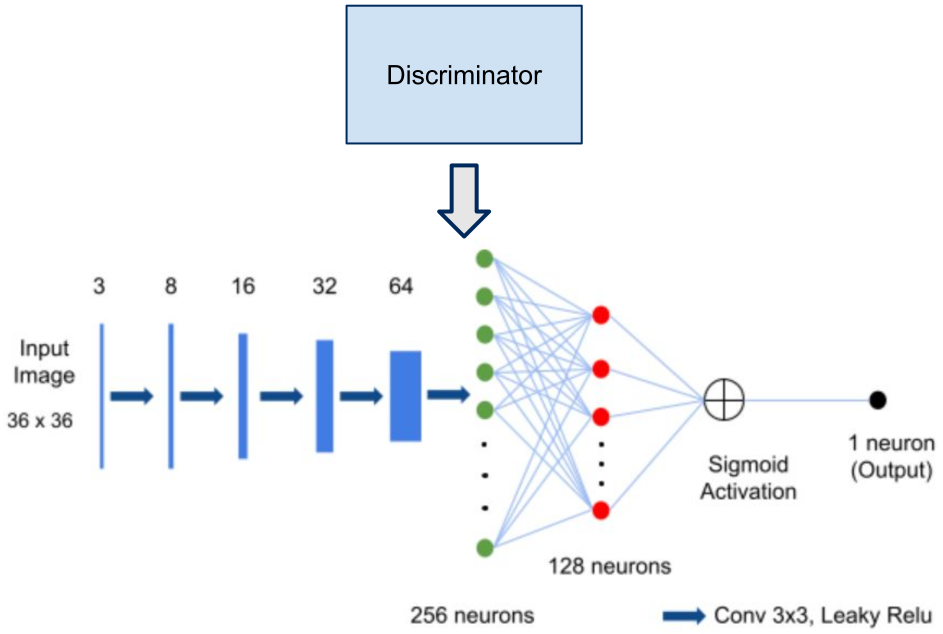
- **Double Convolutional Layers**
  - Extracts important features
  - Batch Normalization
    - Stabilizes the learning process
  - Leaky ReLU
    - Avoids vanishing gradient
- **Down Sampling Layer**
  - 2D Max Pool
    - Extracts sharp features
- **Up Sampling Layer**
  - 2D Conv Trans
    - Increase dimension
- **Skip Connection**
  - Preserve high-resolution features



Per-level architecture of U-net

Discriminator: 4-Layer Convolutional Neural Network

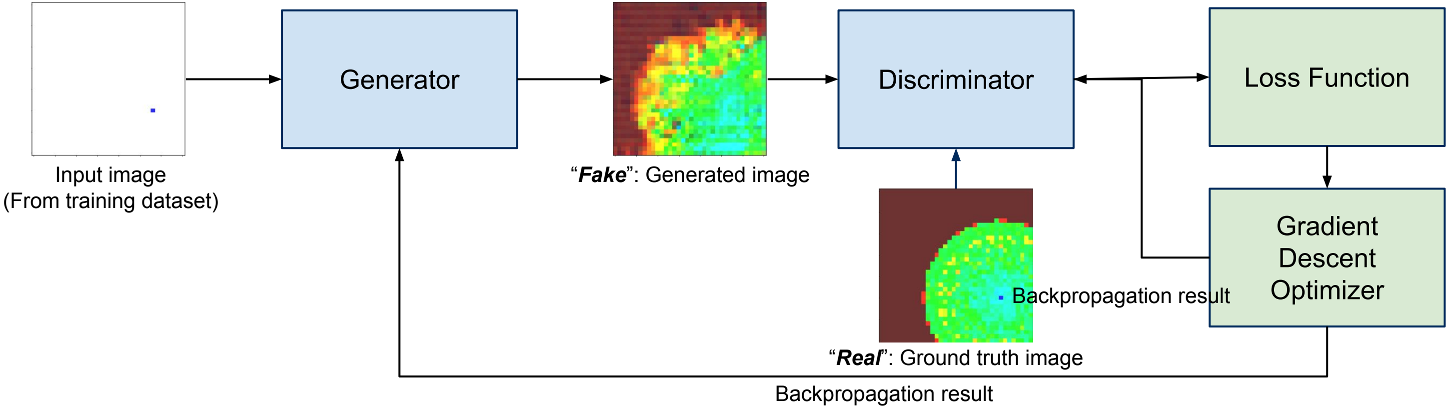
- **2D Convolutional Layers**
  - Feature learning
  - Batch Normalization
    - Stabilizes the learning process
  - Leaky ReLU
    - Avoids vanishing gradient
- **Linear Layers**
  - Classification process
- **Sigmoid Activation Function**
  - Return value in the range from 0 to 1
  - $$S(x) = \frac{1}{1 + e^{-x}}$$



Architecture of Discriminator

Neural Network Training: Adversarial Training Strategy

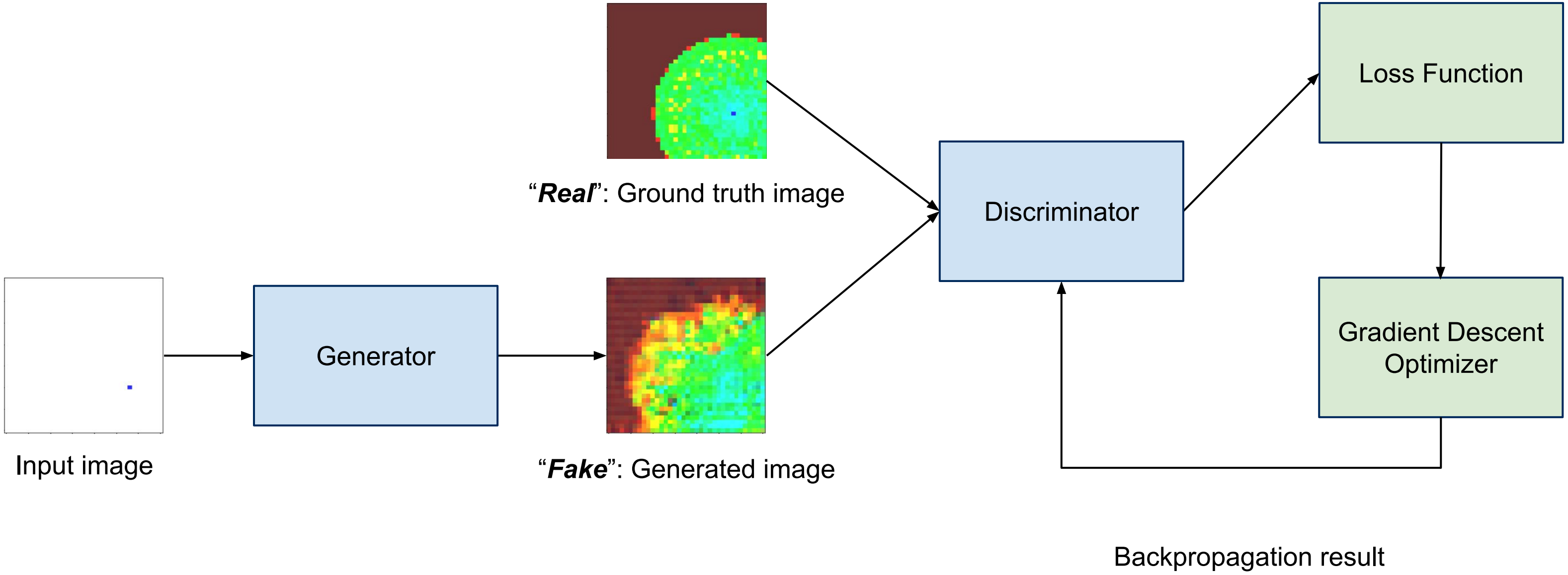
- Two players play a **Minimax** game:
  - **Discriminator**
    - maximizes its probability of classifying “real” and “fake” images correctly
  - **Generator**
    - maximizes the probability that discriminator labels generated images as “real”
  - **Minimax Loss Function** [1]
    - $$\mathcal{L}_{minimax} = \mathbb{E}_x[\log(D(x))] + \mathbb{E}_z[\log(1 - D(G(z)))]$$



[1] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y., 2014. *Generative Adversarial Networks*. [online] arXiv.org. Available at: <https://arxiv.org/abs/1406.2661?source=post_page-----b665bbae3317--------------------> [Accessed 1 November 2021].

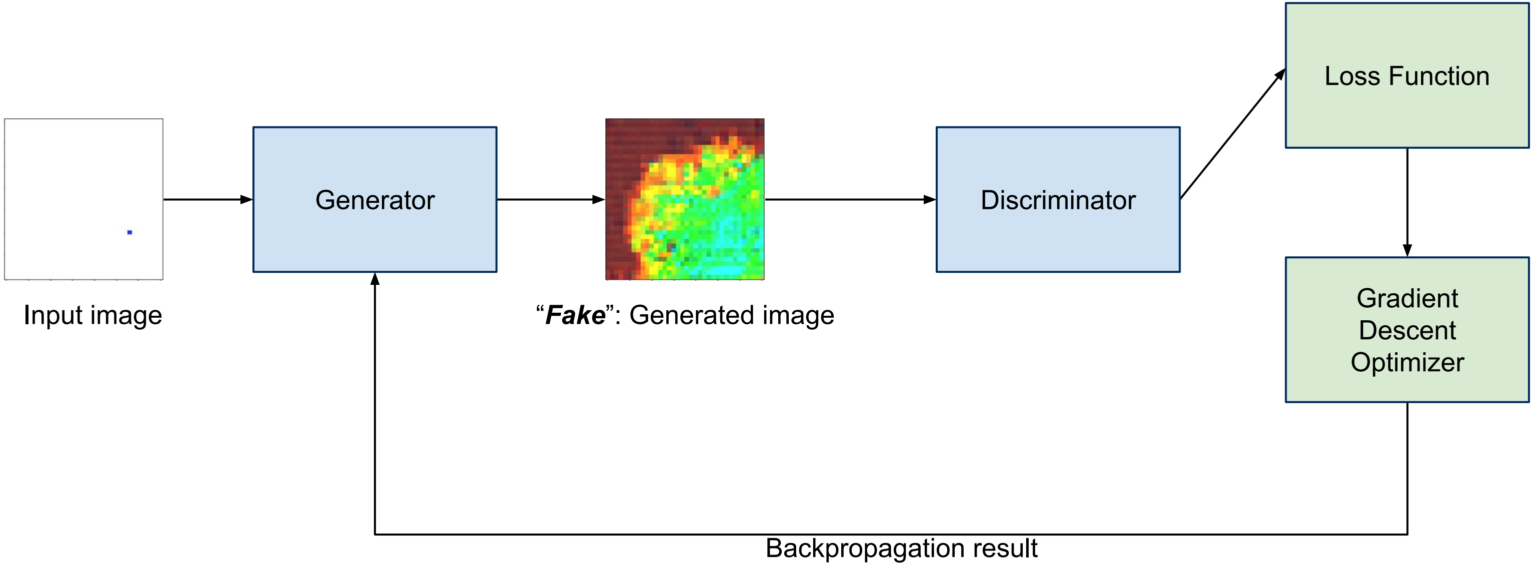
Training Procedures:

1. Discriminator Training

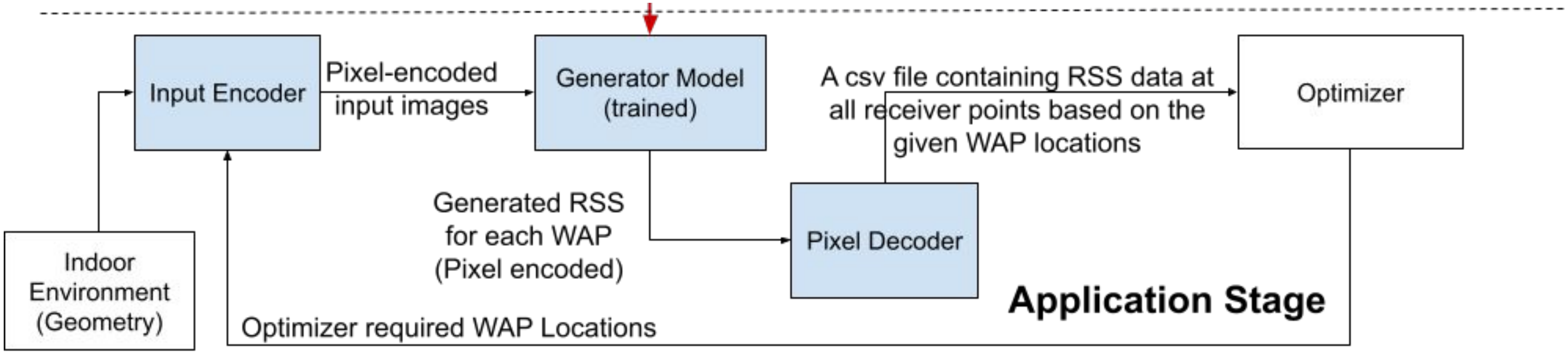


Training Procedures:

2. Generator Training



Application Stage:



- **Pixel Decoder: Pixel-Encoded Output Images → Output RSS data**
  - Reverse process of the Data Encoder
  - The decoded data can then be used for further applications
- **Simulation Process**

Results: Accuracy and Runtime

- Use pixel-wise accuracy as the quantitative approach to evaluate:

$$A_{pixelwise} = 1 - \frac{\sum_{i=1}^{N_i} \sum_{p=1}^{N_{sp}} |p_t - p_g|}{N_i \times N_{sp} \times 255}$$

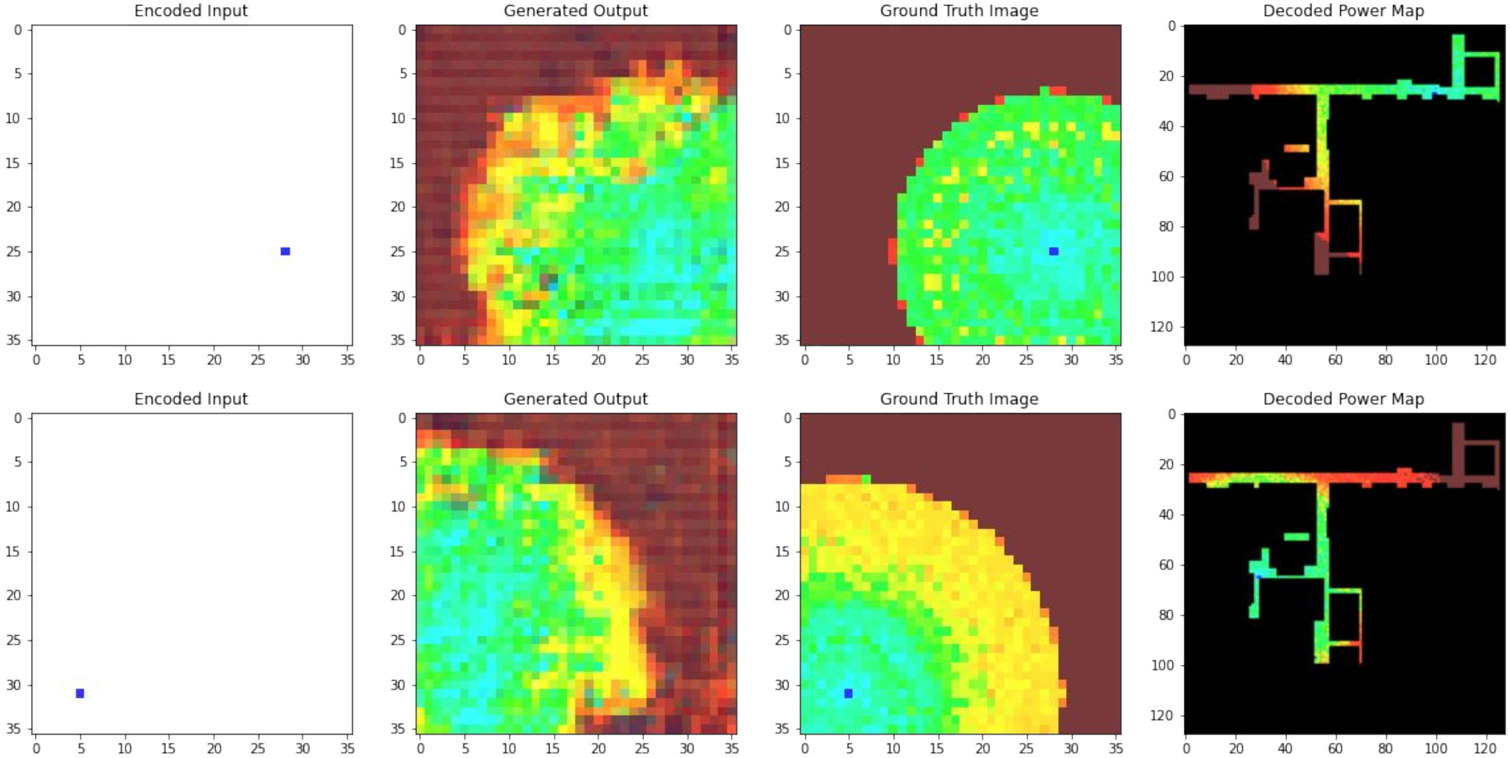
Datasets	Training	Validation	Testing
$A_{pixelwise}$ (%)	79.00	78.84	**78.02**

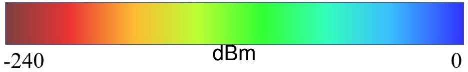
	Ray-Tracing Simulator	DCGAN Model
Average Runtime (s)	43.03	**0.42**

Results: Sample Test Cases

WAP:
(-17.2, 17.2)

WAP:
(6.5,-26.5)





Summary

- **Highlights**
  - Applied machine learning methods to predict RSS in an indoor environment
  - Unique data processing approach to encode data for machine learning operations and training
  - Reduce runtime by 99.02%, while maintaining a high level of accuracy

- **Future work**
  - Continue training with larger datasets
  - Improve generalization with different indoor environments