Pressure Sensor Mapping

BHANU TEJA

**DATASET**

A pixelated image of a galaxy

Description automatically generatedThe pressure pad has1024 sensors arranged into a 32 x 32 grid. Using the pressure pad and a prop we took 900 readings representing all possible edge cases. The raw data has 1024 columns of pressure readings, which we depict as images. These images are used to train our SRGAN model. Here on the right is an example pressure map. Different colors indicate the levels of pressure applied at that point.

**OBJECTIVE**

We aim to reduce the number of sensors needed to get an equally good pressure map. To achieve this, we are using the SRGAN model which will help us generate high resolution images from low resolution images.

**SRGAN Model**

The model has a generator and a discriminator. Generator uses the low-resolution images to generate high resolution images. Discriminator compares the generated high resolution images with the original high resolution images and the loss is given as feedback to the generator to improve it’s performance. After 100 epochs the generator will improve through feedback and be efficient in generating realistic high-resolution images. This helps us draw an accurate pressure map with less sensors.

**Generator Design**

The Generator in the SRGAN is responsible for creating high-resolution images from low-resolution inputs. It achieves this by processing the image through a series of steps:

1. Initial Convolution: The generator starts with a convolutional layer that applies 64 filters, each of size 9x9, to the low-resolution RGB image (3 channels). Padding is added to preserve the original image size, and this layer extracts essential features from the input image.
2. Feature Refinement: The next layer applies smaller 3x3 filters, refining the features extracted earlier. This step focuses on improving the image's details.
3. Activation with ReLU: After each convolution, the ReLU activation function is applied. This function helps the network learn complex patterns and adds flexibility to the transformation process.
4. Final Reconstruction: A final convolutional layer reconstructs the enhanced image, producing a high-resolution RGB image with three output channels. This design ensures the generator captures both the overall structure of the image and the finer details, making it capable of producing high-quality results.

**Discriminator Design**

The Discriminator in the SRGAN evaluates the quality of the images created by the generator. It distinguishes between real high-resolution images and those generated by the generator through these steps:

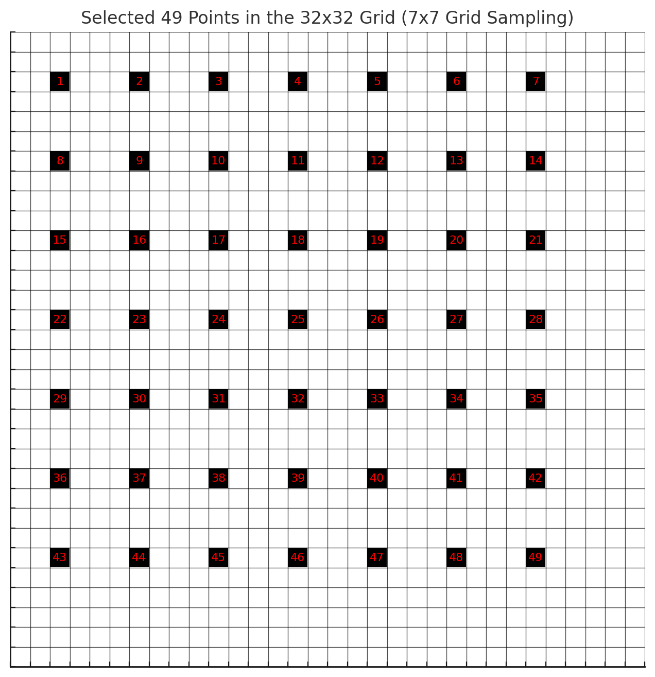
1. Basic Feature Extraction: The first layer applies 64 filters with a 3x3 kernel to capture basic features such as edges and textures. The output is then passed through a LeakyReLU activation function, which helps the network handle negative values and learn complex patterns.
2. Progressive Refinement: As the image moves through the discriminator, more filters are added (128, 256, and 512). These layers use strides of 2, progressively reducing the spatial size of the feature maps. This downsampling allows the discriminator to focus on broader patterns and relationships within the image.
3. A diagram of a block diagram

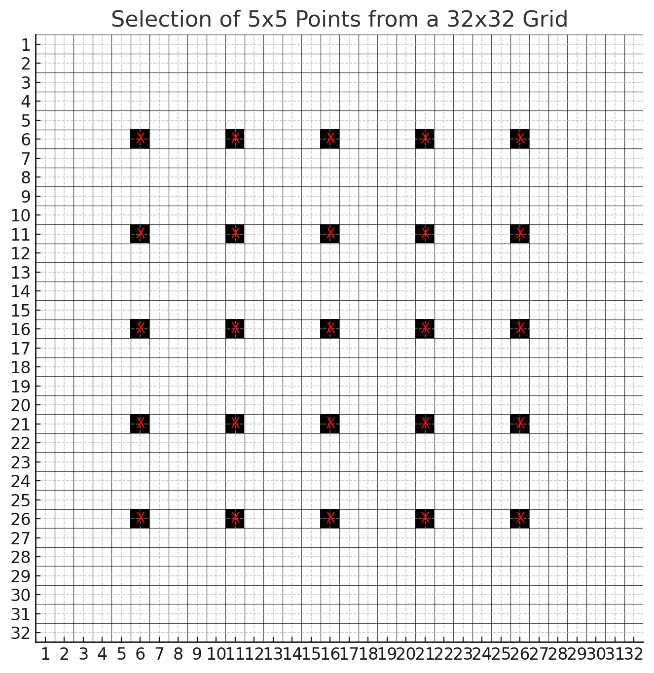
   Description automatically generatedFinal Classification: A final convolutional layer with a Sigmoid activation function outputs a probability score. This score indicates whether the input image is real or generated.

**How the Generator and Discriminator Work Together**

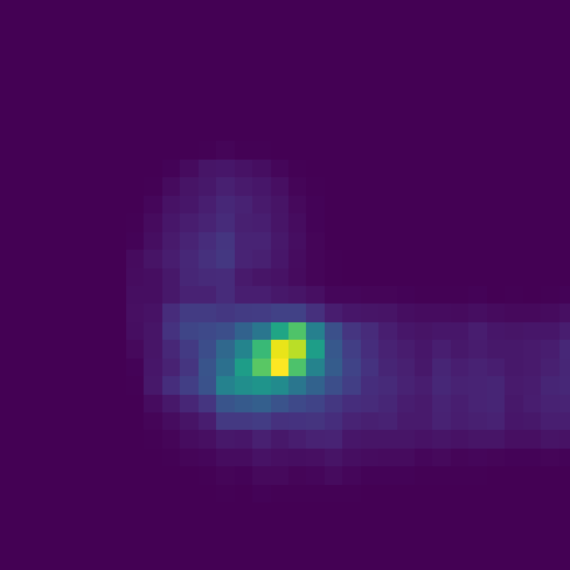
In SRGAN, the generator and discriminator are trained together in an adversarial way. The generator aims to create high-resolution images that look real enough to fool the discriminator. At the same time, the discriminator gets better at telling apart real images from the generated ones. This back-and-forth training improves both networks: the generator learns to create more realistic images with fine textures, while the discriminator ensures these images are convincing by spotting flaws. This adversarial setup pushes the generator to focus not just on reconstructing the image accurately but also on adding realistic details and textures, leading to high-quality, visually pleasing results.

**APPROACH**As we want to reduce the number of sensors we will try this with 49 and 25 sensors. We do this by selecting a 7x7 grid and 5x5 grid from the original dataset. Here are the images showing how the sensors are chosen

  
**7x7 5x5**



This arrangement helps us capture the pressure distribution throughout the grid.

****

5x5 Image

7x7 image

Original Image

All the records in the dataset are converted to 7x7 and 5x5 and stored for training the SRGAN Model.

Initially we performed kriging on both high-resolution and low-resolution images, converting them to 256x256. Then we used the SRGAN to generate high resolution images. This is the result of the initial SRGAN model.

A blurry image of a blue and green gradient

Description automatically generated

In our new approach we skip the kriging and apply SRGAN to the 32x32 and 7x7(or 5x5) directly. This will save us some computational power and increase the efficiency of the model.

**7x7**

PSNR for image 155: 32.94

RMSE for image 155: 0.0225

A close-up of a blue and green background

Description automatically generated

**5x5**

PSNR for image 155: 32.11

A blurry image of a green light

Description automatically generatedRMSE for image 155: 0.0248

A blurry image of a blue and yellow light

Description automatically generatedThis method gives us a very efficient generator that also works for edge cases.

**A graph showing a red line

Description automatically generated**We obtain RMSE and PSNR by comparing the original and generated images. The below graphs represent PSNR and RMSE plotted against the image number.

RMSE vs Image Index

**A graph showing a graph

Description automatically generated**The values of RMSE are low for all the images showing the efficiency of the model.

PSNR vs Image Index

PSNR values close to 30 are considered as a good quality images. All the images generated with our model have PSNR close to 30 which displays the performance of the model.

**CONCULSION**With this method we are able to achieve almost equally accurate pressure maps with just 49 or 25 sensors instead of 1024 sensors, reducing the cost of sensors significantly. This approach gives us good quality pressure maps with minimal sensors, the authenticity of this method can be seen the PSNR and RMSE values.