XAI Driven Image Denoising: Visualization Using Activation Maps

Md Ridwan Hasan, Tingyang Jiao, Pengcheng Yan

INTRODUCTION

Explainable artificial intelligence (XAI) is a set of processes and methods that humans use to understand and trust machine learning models. XAI is used to describe an AI's potential impacts and biases. It helps to describe the model's accuracy, fairness, transparency, and results in AI-supported decision-making. XAI is an open research field of Machine Learning that aims to increase the transparency of the existing black box AI models with relatively high accuracy levels but lack clarity and interpretability. This trade-off between interpretability and accuracy has resulted in failures to deploy some AI models with very high accuracy to critical areas like the medical sector or military, where the stakeholders cannot simply trust the accuracy of the black box models without understanding the underlying mechanism in their own terms.

Complex deep learning models are widely used for image processing. Image denoising is a key part of image processing, with many medical imaging and computer vision applications. There have been many existing effective algorithms in this part, such as Support Vector Machine, Artificial Neural Networks and Clustering. However, with the increasing demand for transparency in AI models for fairness and safety, we need to focus on XAI models in this field to increase their interpretability. For some existing transparent models like linear regression or decision trees, their interpretability is easy to be explained by simulatability, decomposability and algorithmic transparency. But besides this, Post-hoc explainability techniques are required for more complex models like CNN. Visual explanation techniques like class activation feature maps or Grad-CAM are some commonly used post-hoc techniques. Some of these post hoc approaches analyze the relevant regions in the input data. At the same time, others also look at the similarities between model predictions and ground truth training data, thus contributing to better explainability. The former approach allows for visual validation of the prediction of a network by looking at the regions of interest in an image or where the network is being activated in the image. This way, one can make sure that the network has properly learned the patterns in the images and thus explains the validity of its final output. The latter approach is necessary to identify and explain the AI model’s prediction when there is ambiguity in the training data, which often is the case with medical imaging.

Our group’s purpose in this project is to visualize and analyze the performance of an existing denoising model using XAI techniques like activation maps on different types of image datasets with different types of noises and compare the performances.

REVIEW OF EXISTING WORKS FOR IMAGE DENOISING

1. Traditional image denoising methods

Based on discovering some related algorithms about image denoising, there are mainly two categories in the classical image denoising methods: spatial filtering methods and transform domain methods. Some linear filters were used in early research in spatial filterings, like Mean Filter and Wiener Filtering. But there are some disadvantages like “fail to preserve image

textures" (Fan et al. 2). To help prevent these issues, some non-linear filters were used to denoise "without any attempts to explicitly identify it" (Motwani et al. 3), like weighted median filtering. But it also brings issues in efficiency and image blurring.

Furthermore, besides different types of filters, with the motivation of MAP (maximum a posterior) estimation, there are several models for variational denoising in image denoising. Given the image with noises and image prior, after MAP estimation, we can get the image after denoising. In the research of Linwei Fan et al., they introduced some popular methods in this field, like total variation regularization and Non-local regularization. Total variation regularization efficiently solves the problem that cannot retain sharp edges but only "accounts for the local characteristics of the image" (Fan et al. 2). For images with high noise levels, they suggest combining TV regularization with non-local means to achieve better denoising results.

There is linear and non-linear processing in transform domain methods like spatial filtering methods, but the image is mapped to another domain before applying any filters. The transform domain methods consist of data-adaptive and non-data adaptive methods. Two common algorithms in data-adaptive algorithms are independent component analysis and PCA, but they all require high computation ability in training. One popular model in non-data adaptive methods is the wavelet domain. We can use some deterministic algorithms like decision trees or statistical models like Hidden Markov Model to learn the coefficients, but they all still require large computing complexity.

2. Deep Learning denoising methods

Some deep learning methods, especially CNNs, are widely applied in image denoising tasks. In the paper of Linwei Fan et al., they emphasized the role of CNNs in this field, introduced two methods to evaluate the performance of CNN in denoising tasks (PSNR and SSIM), and compared some filtering algorithms introduced before with CNN models. They conclude that these two CNN models have better performance than some popular filtering models, such as BM3D, especially in solving the problem that "contour areas are difficult to recover" (Fan et al. 9). And the comparison between these two CNN models themselves mainly depends on the input of data.

Kai Zhang et al. deposed a CNN model based on the assumption that the noise is additive white Gaussian noise (AWGN), the DNCNN. The network basically consists of convolution + batch normalization + ReLU activation function in each level, except the first (without activation function) and the last one (only convolution). Without pooling layers, the depths of the model depend on the noise level of the input image. This model is also based on two intuitions about deep learning: residual learning and batch normalization. These two ideas in CNN are implemented by Stochastic Gradient Descent and Adam Optimization. Using average PSNR to test the performance, they concluded that the residual learning and batch normalization positively affect training performance. In their experiments, compared with some other denoising models, which required researchers to estimate the noise distribution first, DNCNN has good performance in more general cases with unknown distribution.

Since the DNCNN model is based on Gaussian noise, with the intuition of maximum a posteriori (MAP), Kai Zhang et al. also deposed another more flexible CNN model called FFDNet. The model uses images after subsampling and a noise level map as input, after several layers of CNN (Convolution, Batch normalization and ReLU), then get the output denoised image. Based on their description, the key role of the noise map is to "control the trade-off between noise reduction and detail preservation" (Zhang et al. 4), although it may increase the

complexity of training. Moreover, to increase the speed of model training without losing too many details of images the researchers proposed a method to "reshape the input image into a set of small sub-images" by reversible downsampling(Zhang et al. 4) before applying input images to CNN layers. Their experiments conclude that the FFDNet has good performance in image denoising and advantages in time/memory efficiency.

There are several enhanced CNN-based models in denoising for denoising tasks with some very complex noisy images. Chunwei Tian et al. introduced a model named "attention-guided denoising convolutional neural network (ADNet)" in their paper. Their model mainly consists of 17 layers and four blocks: Sparse block, Feature enhancement block, Attention block and Reconstruction block. The key idea of their "attention mechanism" is in the attention block to solve the problem that "complex background from the given image or video is easier to hide the features" (Tian et al. 118). In this block, there's only one convolution layer. It uses the information we get in the current layer to adjust the previous layer to discover those noises hiding behind the complex backgrounds of images. Besides the attention-guided mechanism, they also use the sparsity (SB) and feature enhancement (FEB) before the attention block to increase the efficiency and performance of their model. After testing, Chunwei Tian et al. conclude that this ADNet model has advantages in complex image denoising with lower complexity.

2.1 Autoencoder in image denoising

Image denoising is an important preprocessing step in image analysis, and various algorithms have been published in the last few decades. Deep learning algorithm performance shows great promise. However, deep learning models require a large training sample size and high computational costs. On the other hand, Autoencoder has efficient performance with a small sample size using convolutional layers. Lovedeep et al. [10] used a convolutional denoising autoencoder with a simple network to recover highly noisy images with small sample size. Xie et al. [15] stacked sparse autoencoders for image denoising with similar efficiency and performance compared to K-SVD. Zhuoqun et al. [14] used a convolutional autoencoder (CAE) with smaller image patches of the laser stripe image, showing that the CAE outperforms other traditional algorithms in image denoising and provides high-precision range data.

REVIEW OF EXISTING ALGORITHMS CONTRIBUTING TO THE EXPLAINABILITY OF IMAGE DENOISING

Based on our literature review of existing models in image denoising, the XAI in image denoising is mainly about the post-hoc explanation in deep learning. Based on the paper of Alejandro Barredo Arrieta et al., there are three main categories in post-hoc explanation: "model simplification, feature relevance estimation and visualization techniques" (92). Feature extraction is a key part of image denoising and explaining the results of a model. The ADNet introduced above is an example in this field (attention-guided). It uses the Attention Block to send the features obtained from the present stage to the previous one for adjusting. Quanshi Zhang et al. proposed a method to modify CNN without changing activation functions to increase the interpretability. They added losses for filters in networks after ReLU activations, so each filter "must be activated by a single part of the object" (Zhang et al. 8829) in the

convolution layer. This method might slightly decrease the precision of the model but has a considerable contribution to interpretability.

METHOD (ALGORITHMS)

Denoising Convolutional Neural Networks (DNCNNs)

We chose DNCNN as the convolutional neural network denoising model. Besides the above information about DNCNN in related work, the DNCNN model uses the idea of residual learning to predict the noises in the image. The input of the DNCNN is an image with Additive Gaussian noise. The output of this model is a residual noise image after several layers of convolutions, activation functions and batch normalization. The DNCNN can work in both images with specific noisy levels (standard deviation of Gaussian distribution in noises) and images with unknown noise levels. The exact amount of layers in DNCNN can vary in different experiments. The basic structure of DNCNN is shown below:

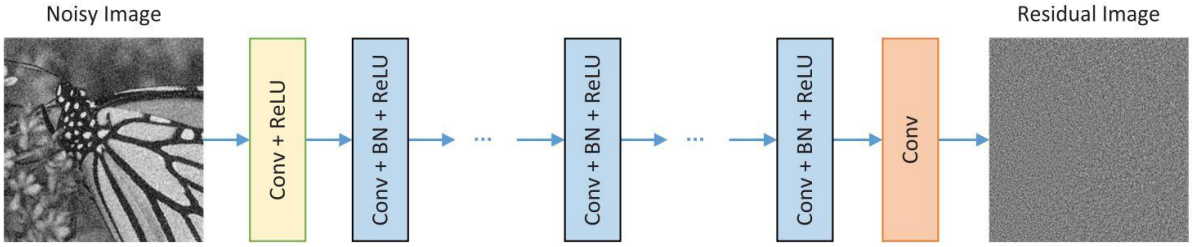


Fig.1. The basic structure of the DNCNN image denoising model, Zhang Kai et al. Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising, pp. 3145

And the error in training the DNCNN model is defined as:

$$\ell(\Theta) = \frac{1}{2N}\sum_{i=1}^{N} \|\mathcal{R}(\mathbf{y}_i;\Theta) - (\mathbf{y}_i - \mathbf{x}_i)\|_F^2$$

Fig.2. The error function in the DNCNN training process, Zhang Kai et al. Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising, pp. 3145

Autoencoder

Autoencoder is an unsupervised artificial neural network that is trained to copy its input to output. The overall structure of this neural network consists of two parts: an encoder and a decoder. An autoencoder takes input x and encodes it to a hidden representation y, also known as latent representation, through a weighted value or the learning rate W, using deterministic mapping, such as

$$y = s(Wx + b)$$

where s is any nonlinear function. Latent representation y is then decoded back into representation z, which is a similar representation to inputted x.

$$z = s(W'y + b')$$

W, W', b, b' are model parameters which are optimized to minimize reconstruction error from z, based on different loss functions.

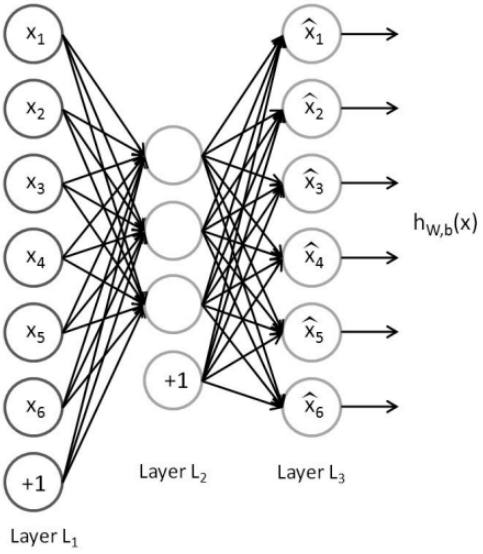


Fig 3. A basic autoencoder. Gondara, Lovedeep. "Medical image denoising using convolutional denoising autoencoders." pp. 242

The goal of this autoencoder process is to learn how to reconstruct input data. However, we do not stress on the output data, the latent space representation, which is the lowered dimension data from the encoder, is more interesting. We hope that the algorithm can extract more useful features from input images as training continues.

The original code we modified is from [11]. We used four types of noises to train the autoencoder, including Salt and Pepper Noise, Gaussian Noise, Poisson Noise and Laplace Noise. And the dataset we used is Fashion MNIST. The original encoder structure, which Rahul wrote, consists of three convolutional layers and three max-pooling layers. Max pool will downsample the image dimension and return the max data in the subregions of the original data representation. The activation function is Relu. The decoding part is similar, with three convolutional layers and three upsampling layers instead of max-pooling layers. The goal of the upsampling layer is to upsample the input data, which is the lowered dimension data from the encoder, to a higher resolution. We tried to improve the performance and accuracy of this algorithm by adding encoder and decoder layers, changing the activation function, and adjusting the learning rate. Those attempts have failed; the accuracy dropped by around 2%. We analyzed failures and believe that this is because an autoencoder is a simple algorithm. Even though it is convolutional, it is based on only encoding and decoding, downsampling and upsampling, so the algorithm we can modify or parameters we can adjust are minimal. Directly adding layers can lower the performance and cause drawbacks like overfitting.

EXPERIMENTS AND EVALUATION OF RESULTS

Autoencoder Experiment:

In this experiment, we use a simple denoising autoencoder [11] and introduce different types of noises to the dataset images and denoise them. The model consists of 7 convolution layers with RELU activation and max-pooling. The summary of the model is shown in the Fig.4 and Fig.5 below.

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 28, 28, 1)]       0

 conv2d (Conv2D)             (None, 28, 28, 64)        640

 max_pooling2d (MaxPooling2D  (None, 14, 14, 64)       0
 )

 conv2d_1 (Conv2D)           (None, 14, 14, 32)        18464

 max_pooling2d_1 (MaxPooling  (None, 7, 7, 32)         0
 2D)

 conv2d_2 (Conv2D)           (None, 7, 7, 16)          4624

 max_pooling2d_2 (MaxPooling  (None, 4, 4, 16)         0
 2D)

 conv2d_3 (Conv2D)           (None, 4, 4, 16)          2320

 up_sampling2d (UpSampling2D  (None, 8, 8, 16)         0
 )

 conv2d_4 (Conv2D)           (None, 8, 8, 32)          4640

 up_sampling2d_1 (UpSampling  (None, 16, 16, 32)       0
 2D)

 conv2d_5 (Conv2D)           (None, 14, 14, 64)        18496

 up_sampling2d_2 (UpSampling  (None, 28, 28, 64)       0
 2D)

 conv2d_6 (Conv2D)           (None, 28, 28, 1)         577

=================================================================
Total params: 49,761
Trainable params: 49,761
Non-trainable params: 0
_________________________________________________________________
```

Fig.4. Summary of the Autoencoder Model

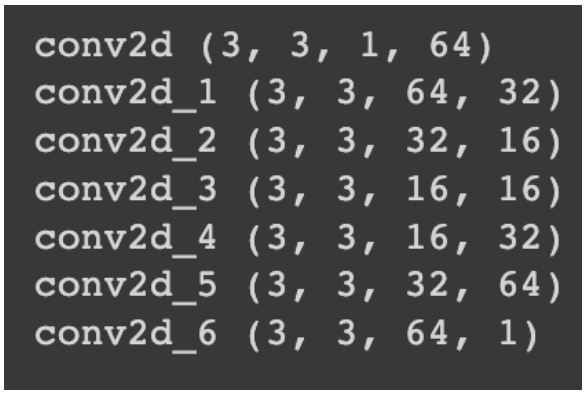


Fig.5. Convolution Layers

To train and test the network, we used synthetic images from the Fashion Mnist and Mnist dataset and introduced four different noises: i. Gaussian Noise, ii. Salt and Pepper Noise, iii. Laplace Noise, and iv. Poisson Noise. as shown in the Fig.6 below.

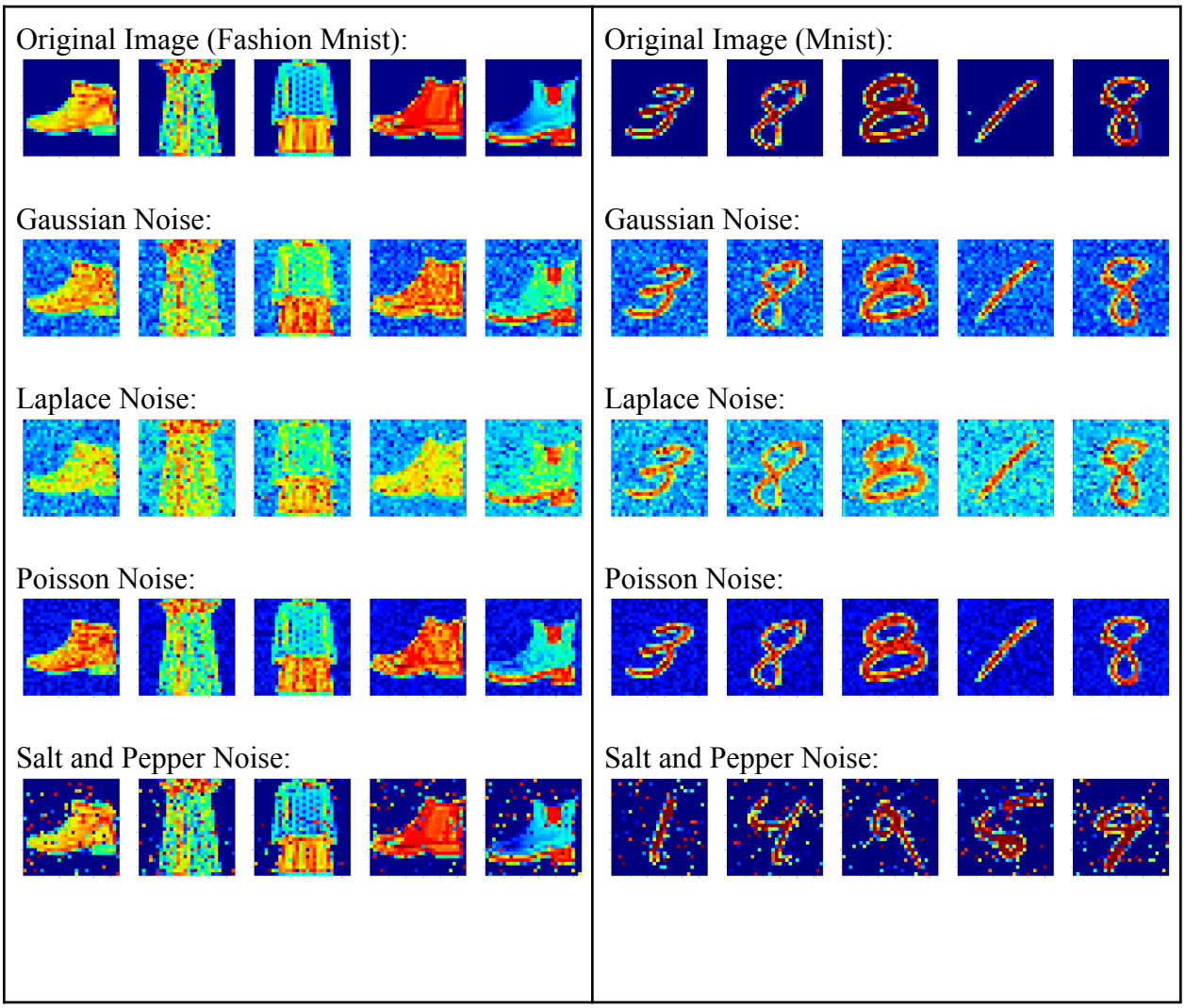


Fig.6. Original Images and Different Types of Noises

For the Mnist dataset, we trained the network up to 1000 epochs with a learning rate of 0.005 and a batch size of 48. The model's output after training can be seen in fig.7 below for different types of noises. We see that the accuracy and loss of the network prediction plateau around 200 epochs, and no significant improvement is achieved beyond that.

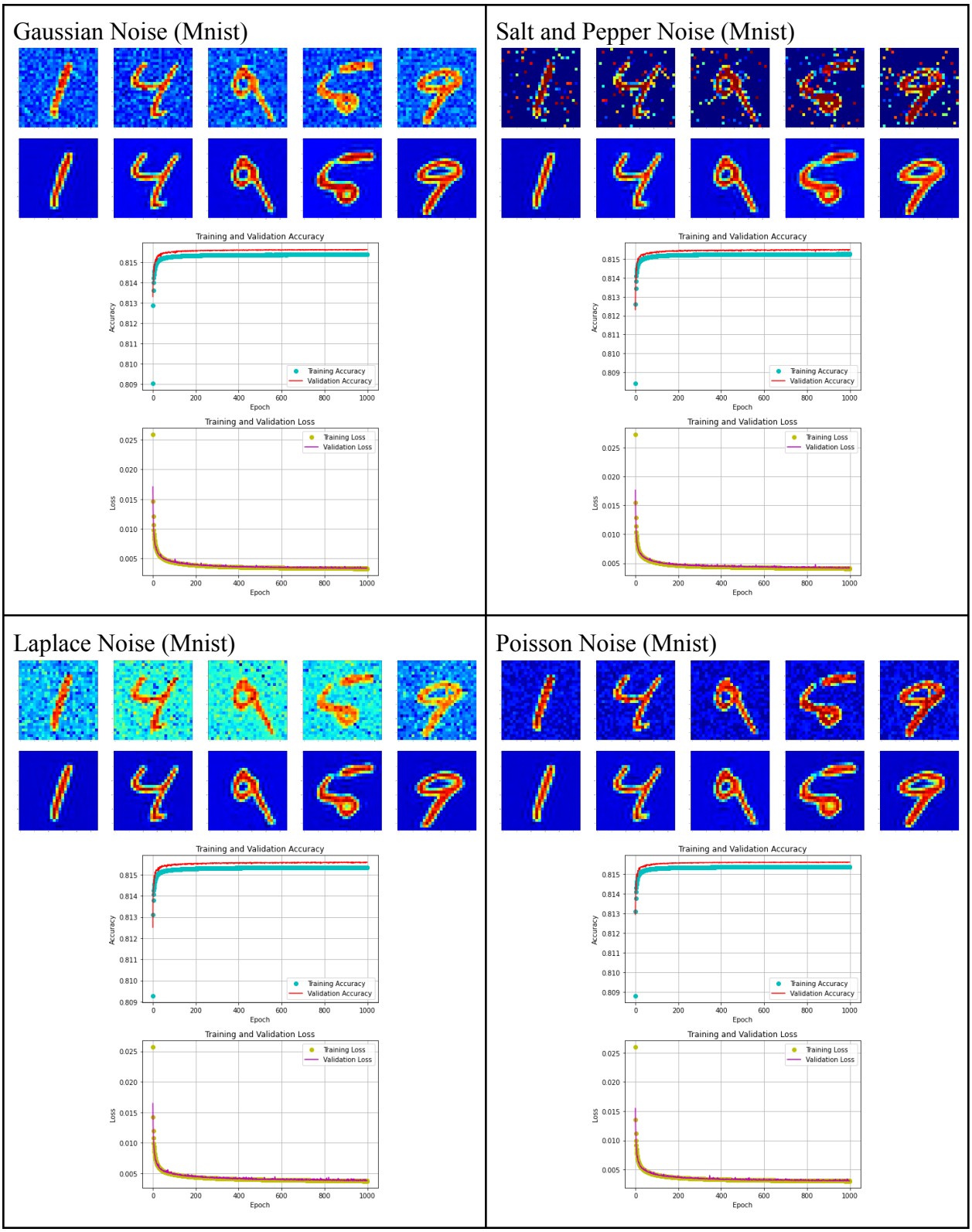


Fig.7. Model Output for Mnist Dataset

We found similar trends while training the network with the Fashion Mnist dataset. We saw that the accuracy and loss of the network do not improve significantly beyond 100 epochs, and larger batch size also does not affect the network performance. A side-by-side comparison for Gaussian noise with different epochs and batch sizes is shown in Fig.8 below.

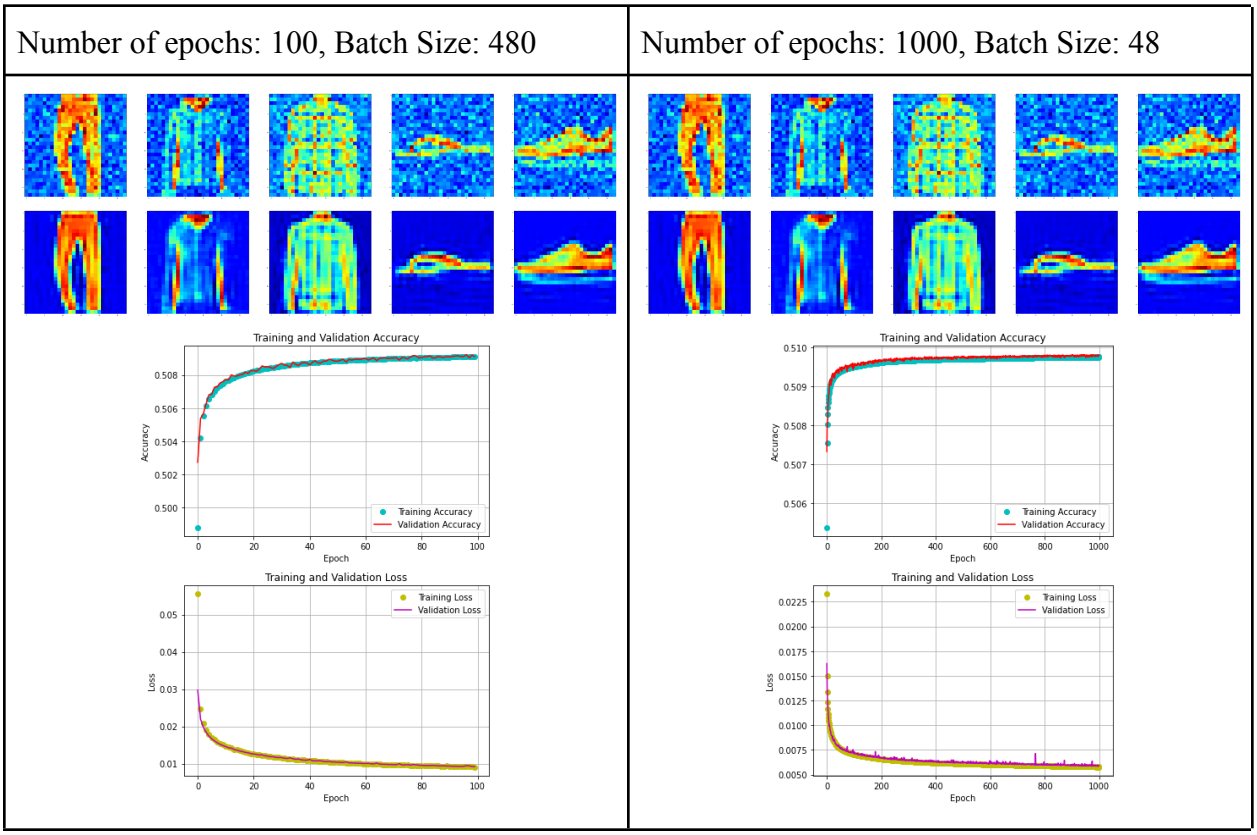


Fig.8. Output of the model for Fashion Mnist Dataset with Gaussian Noise

Since the number of epoch and batch size does not seem to affect the performance of the network, for faster training, we trained the network only up to 100 epoch for the Fashion Mnist dataset with a batch size of 480 and the same learning rate of 0.005. The results are shown in Fig.9 below.

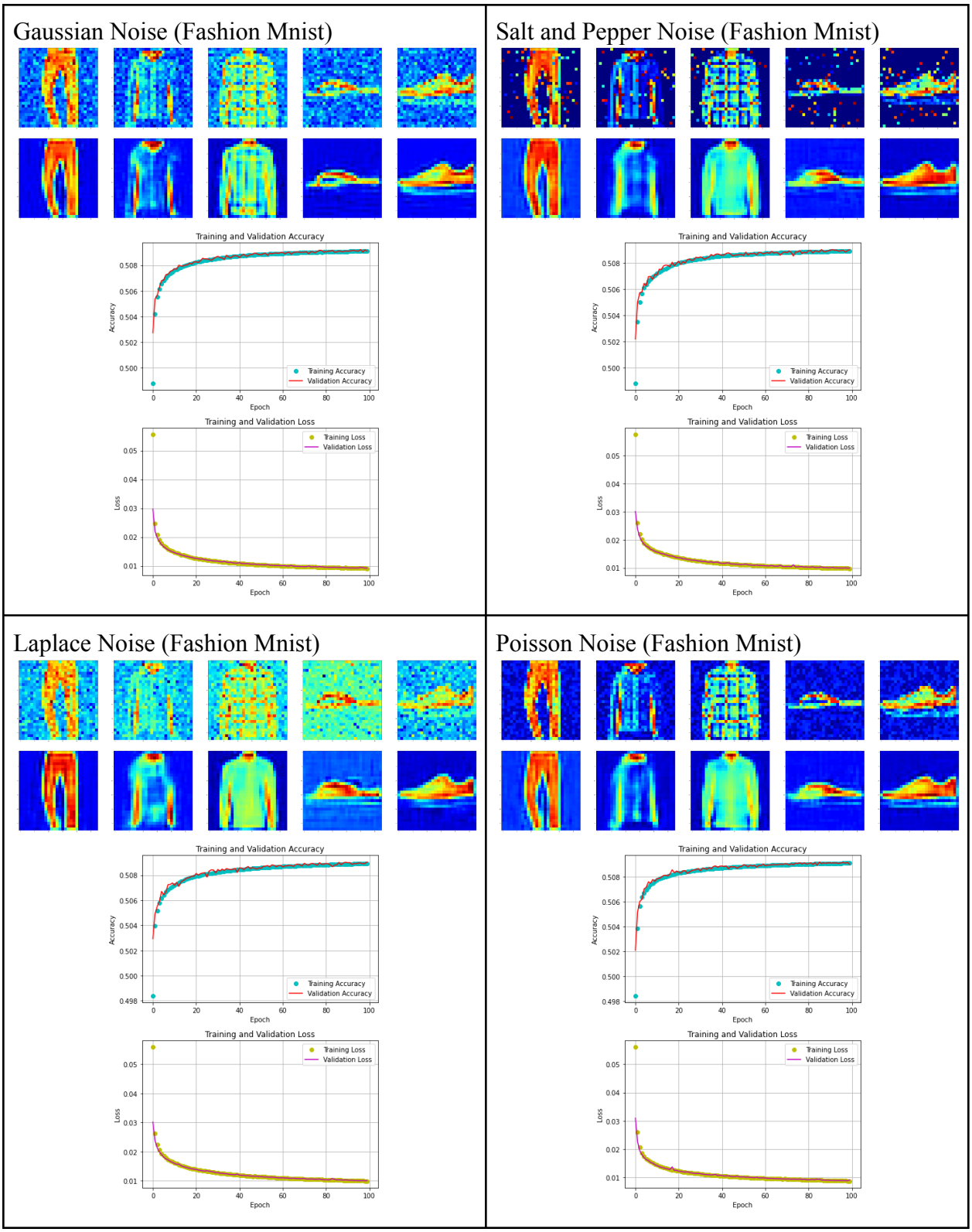


Fig.9. Model Output for Fashion Mnist Dataset

From the above two experiments, we see that the model achieves an accuracy of ~81% for the Mnist dataset, which is a set of much simpler images compared to an accuracy of ~50% for the Fashion Mnist dataset that consists of more complex images. So there is room for improvement of the model to deal with more complex images, which will be our future direction. Now we will discuss the visualization of different layers of the model and see how the model identifies the noises and produces the denoised images.

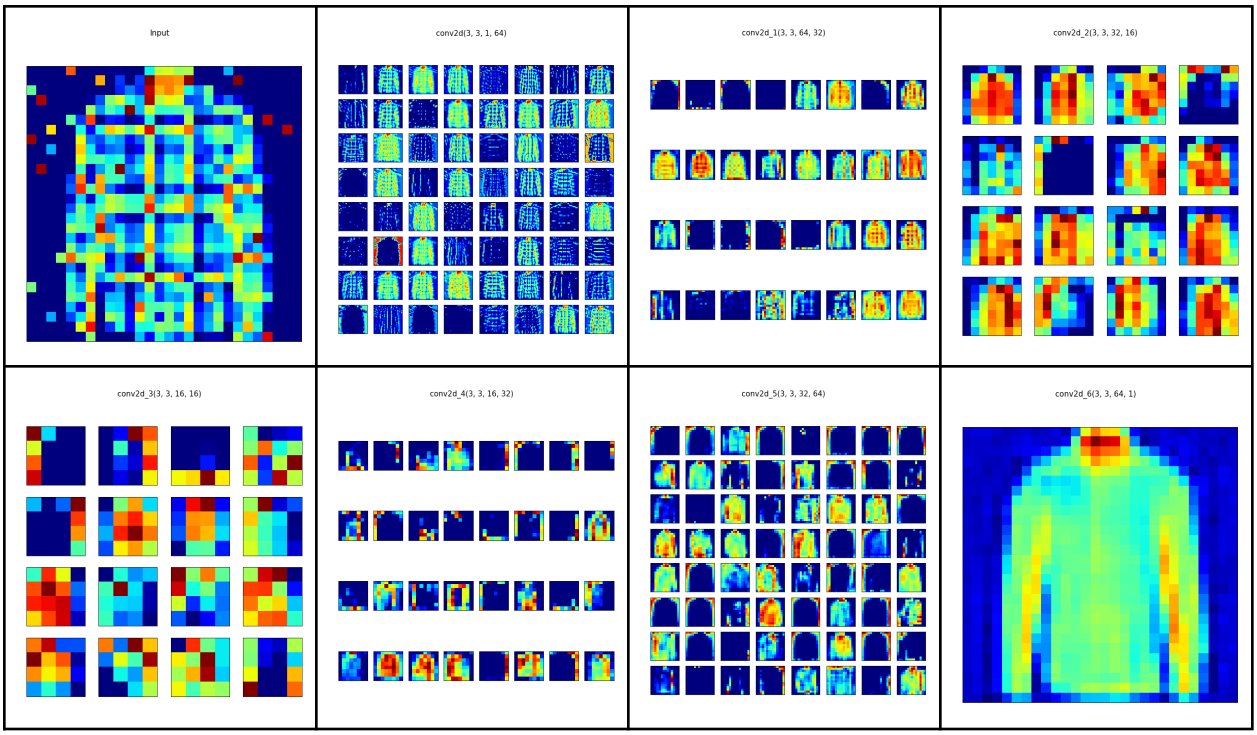
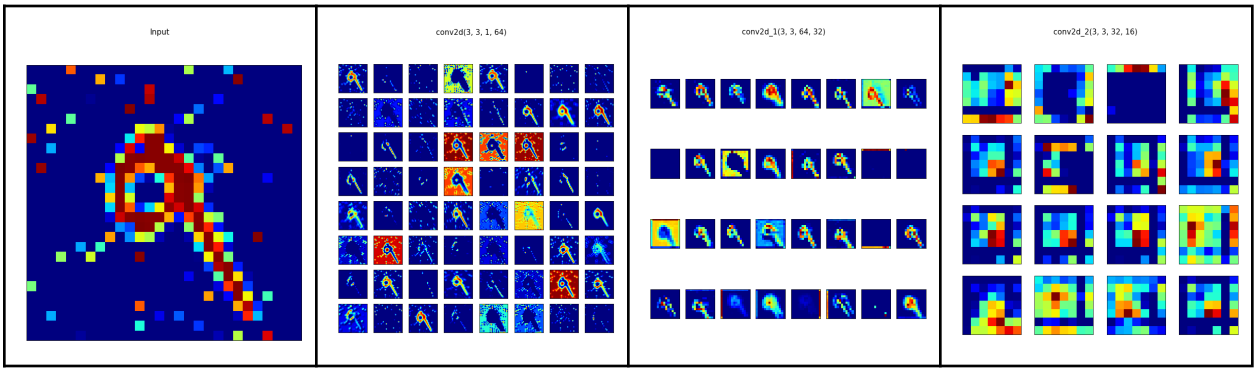


Fig.10. Denoise Visualization: Dataset- Fashion Mnist, Noise: Salt and Pepper



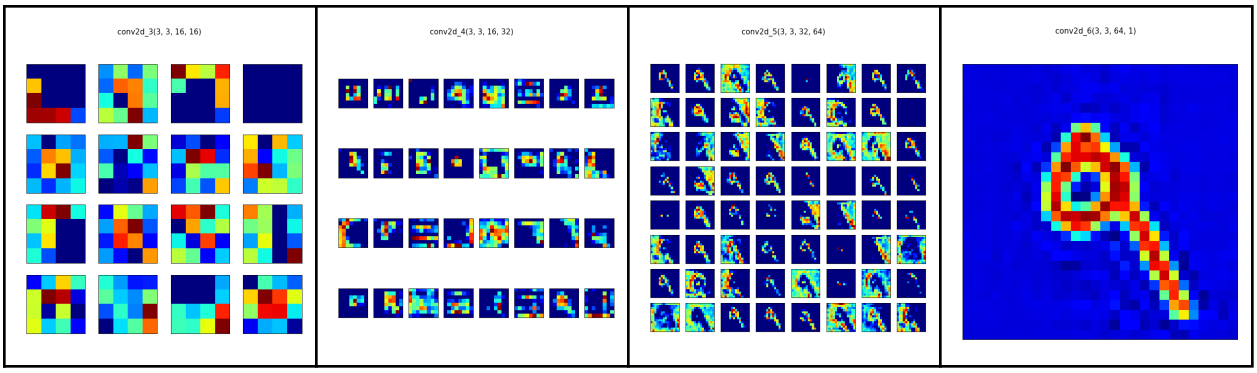


Fig.11. Denoise Visualization: Dataset - Mnist, Noise - Salt and Pepper

One of the most common visualization techniques for CNN models is to show the network's activation in each layer around the target image. In the model of our experiment, the activation function used is RELU. We can verify if the network activates around the input image's relevant parts by visualizing the activation function. Thus we can have greater confidence in the final output of the network. From Fig.10 and Fig.11 above, we can see that for the Fashion Mnist image, the network can identify the noisy pixels very well that fall outside of the shape of the original image, but it fails to distinguish between noisy pixels if they lie on top of the feature pixels of the original image. While denoising the Fashion Mnist images, the network removes some original image features by identifying them as noises since the noise and feature pixels lie on top of each other. This explains why the network's performance is lower for the Fashion Mnist dataset compared to the Mnist dataset. In Fig.11 above, we can see that for much simpler Mnist images, the model identifies the noisy pixels much better and preserves the original feature of the images relatively well, which is why the prediction accuracy is also higher in this case. Our primary focus in this project is not to improve the accuracy of the model for denoising since there are a lot of denoising models that achieve very high accuracy but to show how we can use visualization maps or class activation feature maps to better understand and interpret the output of the model. To that end, we have used a straightforward class activation feature map in a very simple denoising autoencoder. In future, we plan to implement and analyze the visualization of more complex denoising algorithms and find how the visualization can help identify bugs and improve the model’s performance.

CONCLUSION

In this project, we demonstrated how visualization techniques could be used to understand better the underlying mechanisms and decision process of a denoising algorithm like an autoencoder. Using the visualization, we can determine how different images and noises affect the model’s performance. In future, we plan to use similar visualization techniques on more advanced denoising algorithms and come up with some way to use this information to enhance the model’s performance. Another future extension of our current work can be visualizing a deep neural network for image segmentation and classification.

REFERENCE

1. Barredo Arrieta, Alejandro, et al. “Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI” *Information Fusion,* Vol.58, June 2020, pp. 82-115, https://doi.org/10.1016/j.inffus.2019.12.012.

2. Zhang, Quanshi, et al. “Interpretable Convolutional Neural Networks” *Computer Vision and Pattern Recognition (CVPR),* 2018, pp. 8827-8836, https://openaccess.thecvf.com/content_cvpr_2018/html/Zhang_Interpretable_Convolutional_Neural_CVPR_2018_paper.html

3. Fan, Linwei, et al. “Brief review of image denoising techniques” *Visual Computing for Industry, Biomedicine, and Art,* 2019, https://doi.org/10.1186/s42492-019-0016-7

4. Motwani, Mukesh C, et al. “Survey of Image Denoising Techniques” *Proceedings of GSPX*. Vol. 27. 2004

5. Tian, Chunwei, et al. "Attention-guided CNN for image denoising." *Neural Networks*, Vol. 124, 2020, pp. 117-129, https://doi.org/10.1016/j.neunet.2019.12.024

6. Zhang, Kai, Wangmeng Zuo, and Lei Zhang. "FFDNet: Toward a fast and flexible solution for CNN-based image denoising." *IEEE Transactions on Image Processing,* Vol. 27, No. 9, 2018, pp.4608-4622.

7. Zhang, Kai, et al. "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising." *IEEE transactions on image processing,* Vol.26 No.7, 2017, pp.3142-3155.

8. Srinivas, Suraj and Francois Fleuret. “Full-Gradient Representation for Neural Network Visualization” *Advances in neural information processing systems 32*, 2019. https://doi.org/10.48550/arXiv.1905.00780

9. Selvaraju, Ramprasaath R., et al. "Grad-cam: Visual explanations from deep networks via gradient-based localization." *Proceedings of the IEEE international conference on computer vision*. 2017. https://doi.org/10.48550/arXiv.1610.02391

10. Gondara, Lovedeep. "Medical image denoising using convolutional denoising autoencoders." *2016 IEEE 16th international conference on data mining workshops (ICDMW)*. IEEE, 2016.

11. Autoencoder_MNIST-Fashion, https://github.com/Rahulraj31/Autoencoder_MNIST-Fashion/blob/main/MNIST_Autoencoders.ipynb

12. DNCNN-PyTorch, https://github.com/SaoYan/DnCNN-PyTorch

13. Fullgrad-saliency, https://github.com/idiap/fullgrad-saliency/tree/master/saliency

14. Fang, Zhuoqun, et al. "Laser stripe image denoising using convolutional autoencoder." *Results in Physics 11*, 2018, pp.96-104. https://doi.org/10.1016/j.rinp.2018.08.023

15. Xie, Junyuan, Linli Xu, and Enhong Chen. "Image denoising and inpainting with deep neural networks." *Advances in neural information processing systems 25*, 2012