5-1-2022

An Explainable Artificial Intelligence Approach to Convolutional Neural Network Optimization and Understanding

Nicholas Laudermilk

AN EXPLAINABLE ARTIFICIAL INTELLIGENCE APPROACH TO CONVOLUTIONAL NEURAL NETWORK OPTIMIZATION AND UNDERSTANDING

by

NICHOLAS LAUDERMILK

Presented to the Faculty of the Honors College of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

HONORS BACHELOR OF SCIENCE IN BIOMEDICAL ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON

May 2022

ACKNOWLEDGMENTS

I would like to express my gratitude and thanks to Dr. Juhyun Lee, Tanveer Teranikar, Bohan Zang, and senior design group partners: Murtaza Khokar, Xi Tan, and Nowshin Faiza for their support in creating our neural network, and imaging setup. I would also like to thank Dr. Khosrow Behbehani for his assistance in pushing our project to a higher standard and providing guidance over the past year in my senior design course.

Next, I would like to thank my friends who have helped me throughout my time at The University of Texas at Arlington, especially Mary, James, Duncan, Caden, Jose, and Ryan. Without your support over the past 4 years, I would not be here today. I'd also like to thank my family: Mom, BJ, Chris, Aaron, and Naomie, for helping me become who I am today and always supporting me, no matter what.

Lastly and most of all, I want to thank my partner Carina for her support over the past year. Without your help, I never could have made it to this point on my own.

May 03, 2022

ABSTRACT

AN EXPLAINABLE ARTIFICIAL INTELLIGENCE
APPROACH TO CONVOLUTIONAL NEURAL
NETWORK OPTIMIZATION
AND UNDERSTANDING

Nicholas Laudermilk, B.S. Biomedical Engineering

The University of Texas at Arlington, 2022

Faculty Mentor: Juhyun Lee

Advancements in artificial intelligence (AI) show promise for the technology's use in widespread biomedical applications. As these models grow more complex, understanding how they work becomes increasingly more difficult. To use these systems in the healthcare setting, it is imperative to reduce model ambiguity and increase user trust in their decision-making. Explainable AI (XAI) techniques were used to optimize the development of a super-resolution convolutional neural network (SRCNN). Image augmentation was performed on the training data, and k-fold cross-validation was used to obtain more reliable metrics. Activation maps were used to show the output of each convolutional layer, and the final neural network (NN) weights were visualized. Using these techniques, the model was shown to focus primarily on the circular lenslet patterns of input LFM images, with the center of images being the main focus of the model. The

final trained model was able to outperform bicubic interpolation in PSNR by 27% and SSIM by 7%.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Project Background

Over recent years, the interest in usage of artificial intelligence (AI) for biological and medical research has skyrocketed, due to their ability to produce highly accurate results for almost any use case. One of the biggest issues with these NNs however, is what is known as a black box model. This occurs when there is a lack of understanding of how and why a model works, leaving the reasoning behind the results generated by said networks unclear. This project aims to address this issue, by applying Explainable Artificial Intelligence (XAI) techniques to the training and testing of a super-resolution convolutional neural network (SRCNN) for light field microscopy (LFM) image upscaling/deconvolution.

1.1.1 Project Aim

The work of this project is an addition to the aim of one of the bioengineering senior design projects: to create a NN capable of taking an input raw 2D LFM image and outputting an upscaled deconvoluted 3D image. The final design consists of two separate NNs, one for upscaling the 2D raw LFM image, and another for converting this upscaled image into a 3D deconvoluted LFM image. As the 3D deconvolution NN was a late addition to the original senior design project, analysis of its structure was not able to be completed. Due to this, this project focused on the upscaling portion of this NN and can be divided into three main sections: training, testing, and analysis of the upscaling NN portion

of the final design. To do this, XAI techniques will be implemented at all stages of its development, as well as after the completion of the design, which will be discussed in a later section in more detail.

1.1.2 Light Field Microscopy

Light Field Microscopy is a 3D imaging technique in which spatial resolution is sacrificed to capture angular resolution. To achieve this, a microlens array is inserted into the intermediate image plane of a normal optical microscope and a sensor placed at the back focal plane of the microlens array. This allows the sensor to capture multiple discrete ray angles and produces raw LFM images which consist of a lattice of small circles as shown below.

Figure 1.1: Example of raw 2D LFM image

After they are captured, these images are processed by postprocessing software, and produce a focal stack of images, which may be displayed as a final 3D image. The drawback with this technique is that some spatial resolution must be given up in order to capture the angular views, which results in a much lower resolution final image (Broxton & Grosenick, 2013). This issue aimed to be solved by the development of the NN.

1.2 Artificial Intelligence in the Healthcare Setting

Given the increase in the complexity and amount of data in healthcare, AI is sure to see an increasing number of applications within the field. In fact, we can already see some of its use cases, in places like diagnosis and treatment recommendations (Ahmed & Zubair, 2022). Even though a number of studies have shown that these AI can perform as well as or better than humans at these key tasks, their adoption has been slow and on a small scale, but why is this the case? It is perhaps the most difficult issue to address that has put the biggest halt on their adoption, that being their transparency. Many deep learning algorithms and models, particularly those performing image analysis, are almost impossible to interpret or explain, and function as a black box system (Shrivastava & Kumar, 2022). When a serious diagnosis is given, it is understandable that a patient would likely want to know why, and these algorithms and even the physicians using them may not be able to provide one. Additionally, there is the ethical dilemma of what should be done in the case of an AI system making a mistake ("Artificial Intelligence in medical science", 2014). There is no clearly established accountability for these systems, and the liability of their decisions is not an easy thing to delegate. Additionally, these systems can be subject to algorithmic biases, possibly making predictions more likely based on factors that may not actually be causal factors, like gender or race. The question then becomes, how can we address these issues with NNs in order to increase their usage and ensure a fair final model? One such solution is by implementing XAI tools and techniques to develop and understand these systems.

1.3 Explainable Artificial Intelligence

XAI is a set of techniques and methods that allow for humans to both understand and trust the output of various machine learning models. These techniques may be applied to characterize the accuracy, fairness, and transparency of a NN. It is also one of the best tools for building trust and confidence when using these models in any setting (Barredo Arrieta, et al., 2020). As AI becomes more advanced, it becomes an increasing challenge to retrace how an algorithm may have arrived at a result. The model may become a black box, and be almost impossible to interpret, even for the data scientists or engineers who created the system in the first place. Implementing XAI techniques into the development of these models gives them a level of explainability that can help ensure they are working as intended. The broad adoption of these systems depends on the ability of humans to trust the output of these AI algorithms, which is based on the level of understanding we have of how it functions, as well as its safety and reliability. These factors make XAI crucial in achieving more robust and fair models, as well as ensuring they do not cause harm.

1.3.1 XAI Techniques

When using XAI to achieve interpretability, there exists two main techniques for creating interpretable models. The first method is creating a transparent machine learning model, that being a model which can be understood based on its structure alone without other techniques. The second method is applying post hoc analysis on a black box model to explain the complex behavior. Though the NN structure optimized in this project was created initially as a transparent model, it was optimized to a less understandable final state. As such, this project employed primarily post hoc techniques to analyze the final trained NN. Because of the structure of our NN, only some of these techniques may be employed.

Many XAI methods are created for NN structures aimed for classification. As our NN produces an upscaled image, these techniques are not always able to be applied to our model. Due to these factors, the main post hoc technique employed was feature visualization, specifically using feature maps and filter visualization.

Because convolutional neural networks (CNNs) are designed to work with image data, their structure and function are less transparent than other types of NNs. These models are composed of small linear filters as well as the result of applying said filters to an input, called a feature map. We can visualize both of these to provide insight as to how a model works. In the case of the NN filters, we may do so by retrieving and normalizing the learned weights from our final model's convolutional layers. These may then be represented visually, with higher weights indicating a higher focus on that portion of the input. As for feature maps, we may select an input image and pass it through our network. By taking the output of each convolutional layer, we can see the filters applied to our input image, giving us an indication of what features the network picks up on in each layer (Brownlee, 2019). Using these two techniques, we can produce a visualization of the inner functionality of our NN and gain a better understanding of how it produces its final output.

Additionally, we can apply XAI techniques to generate a more balanced dataset. In this project, the two main ways this was accomplished was via image augmentation within Keras, a deep learning API for the TensorFlow library, and k-fold cross-validation during validation. Image augmentation is a method that allows for the expansion of the size of a dataset by applying transformations to the original dataset. These transformations include rotations, shifts, flips, and zoom. Applying these adds a level of variation that allows our final model to generalize better when encountering unseen data, making it more robust

overall. K-fold cross-validation is a procedure used the better measure of the metrics of a machine learning model. This procedure has a single parameter named k, referring to the number of groups a set of data will be split into. Generally, k-fold cross-validation works as follows. First, the dataset is shuffled randomly and split into a set of k equal groups. For each unique group, the group is taken out to be used as a test dataset. The remaining groups are then used to train and fit the model. After this, the testing data is used to measure the evaluation scores, which are stored. This is repeated for each unique group. After all k groups have been evaluated in this manner, the evaluated metrics are averaged to obtain a final summarized metric set for the model. By doing this, we generate metrics with low bias, and can better see the skill level of our model.

CHAPTER 2

METHODOLOGY

2.1 Neural Network Training

In this section, the methodology for the training of the SRCNN is detailed. Training was performed on multiple models then analyzed to determine the optimal structure. This was performed in Python using the Keras and TensorFlow libraries in two main steps, model optimization and final model training.

2.1.1 Model Optimization

During the model optimization process, multiple CNNs were trained with varying structures using the BSDS500 dataset, a collection of 500 images for image segmentation and benchmarking, with 300 of the images used for the training of the models. Image augmentation of the training dataset was performed using the ImageDataGenerator class within Keras, with a rotation range of 20 degrees, a width and height shift range of 10%, a shear range of 20 degrees, and a zoom range of 10%. This was used to preprocess the training data and produce 900 unique images for training from the original set. These images were saved to a separate directory to be loaded when training the model.

The process for training the models themselves consists of the following steps. First, the augmented images were loaded and converted into grayscale, then converted to a numpy array and normalized. The normalized images were then converted into two separate dataset arrays, a ground truth and low-resolution dataset. The ground truth dataset images were cropped to a 256x256 pixel region, then added to the ground truth array. For

the low-resolution dataset, the images were cropped to half that size, 128x128 pixels, and added to a low-resolution array. For compiling and training the NN model itself, the pixelwise mean square error between the images was used as a loss function, with the Adam optimizer, and a learning rate of 0.001. The low-resolution images were then passed through the NN model, producing an output image 2x in resolution, which was compared to the ground truth image. This was then run for 400 epochs with k-fold cross-validation using a k value of 10. The average PSNR and SSIM between the ground truth and upscaled images were calculated for each fold, then averaged overall to produce the final metrics for each model. These low-resolution images were then upscaled via bicubic interpolation, and the same metrics measured, then compared to those from the trained models. This process was repeated for each of the tested models, which had their structures iterated on to produce better metrics. The model with the highest metrics was then chosen as the final structure and trained according to the following section. Additionally, each model and trained weights was saved for later testing.

2.1.2 Final Model Training

Once the model's structure had been optimized using the previously mentioned technique, it was trained using simulated raw 2D LFM images from the VCD-Net dataset, containing over 2300 image. Training the final model was almost identical to the optimization procedure mentioned in the previous section. First, the raw 2D LFM images were augmented using Keras' ImageDataGenerator, with a rotation range of 20 degrees, a width and height shift range of 10%, a shear range of 10 degrees, and a zoom range of 10%. This produced around 5000 images to be used in the training of the model. These images were loaded and converted into grayscale, then converted to a numpy array and normalized.

The normalized images were then converted into two separate dataset arrays, a ground truth and low-resolution dataset. The ground truth dataset images were unaltered 176x176 pixel images and added to the ground truth array. For the low-resolution dataset, the images were cropped to half that size, 88x88 pixels, and added to a low-resolution array. Again, the pixelwise mean square error between the images was used as a loss function, with the Adam optimizer, and a learning rate of 0.0001. The low-resolution images were then passed through the NN model, producing an output image 2x in resolution, which was compared to the ground truth image. This was then run for 2500 epochs with k-fold cross-validation using a k value of 10. The average PSNR and SSIM between the ground truth and upscaled images were calculated for each fold, then averaged overall to produce the final metrics for each model. These low-resolution images were then upscaled via bicubic interpolation, and the same metrics measured, then compared to those from the trained models to have an idea of the overall performance of the model. Following this, the model and trained weights were saved and validated to determine the model's performance on unseen data.

2.2 Neural Network Testing

In this section, the methodology for the testing/validation of our SRCNN is detailed. Though performance metrics were measured during training, in order to ensure the model was robust, they were remeasured using a novel set of images not used during training. The following metrics were measured to characterize the performance of each model: PSNR, SSIM, pixelwise mean square error (loss), training time per epoch, and computational time. The PSNR, SSIM, and pixelwise mean square error were measured during the training and testing of the models, while the training time per epoch was measured during training and the computational time during testing.

2.2.1 Testing Methodology

The testing of each model was done similarly to the method used during training, with the main difference being the usage of novel data unseen during training. While initially optimizing the model structure, 200 unaltered images from the BSDS500 dataset were used for validation, while the training of the final model used 200 unaltered images from the VCD-Net dataset. These images were loaded and converted to grayscale and then into a numpy array and normalized. After this, they were separated into a ground truth and low-resolution dataset. In the optimization phase, the ground truth array represented a 256x256 cropped region of the image, while in the final model, the ground truth array consisted of unaltered raw 2D LFM images. In both phases, the low-resolution array was a copy of the same images at half the resolution. The low-resolution images were then upscaled via the NN and bicubic interpolation, and the PSNR, SSIM, pixelwise mean square error, and computational time measured for each. Additionally, the training time per epoch was measured while training the models and added to these results. These metrics were then compared between bicubic interpolation and each NN structure and used to optimize the model structure as shown in the next section.

2.2.2 Model Structure Optimization

To optimize the initial structure of the SRCNN, the measured metrics were used to determine which structure performed best. As the goal of the final model was to produce a high resolution upscaled image, visual quality of the output image was the main consideration in determining this. Comparing PSNR, SSIM, and pixelwise mean square error between each structure helped in initially determining the best models, however these metrics do not provide a full representation of visual quality of an image. Because of this,

additional tests were performed to determine the model which produced images with the highest visual quality. To assess this, a poll was conducted with 10 participants, each ranking the visual quality of 6 images upscaled via 4 different model structures and bicubic interpolation. Participants were asked to rank the 5 images within each set from highest to lowest visual quality. This data was additionally used in consideration with the measured metrics to determine the best final model structure. The same poll was conducted on the final trained model as well, where the participants were asked to rank 4 images in terms of visual quality, the ground truth image, low-resolution input image, NN upscaled image, and bicubic interpolation upscaled image.

CHAPTER 3

RESULTS AND ANALYSIS

This section contains the results from the training, testing, and analysis of the SRCNN for raw 2D LFM image upscaling.

3.1 Model Optimization

When optimizing the NN, 4 separate structures were tested and compared to bicubic interpolation to determine the overall best model. Visual quality was also assessed via a user poll. Though average computational time was measured, the results were negligible between each model, and as such are not included in the tables shown below.

Table 3.1: Average metrics of BSDS500 dataset trained models

Model Name	10 Layer CNN	5 Layer CNN	3 Layer CNN	U-Net	Bicubic
PSNR (dB)	33.40	32.31	32.38	28.94	31.70
SSIM	0.937	0.926	0.931	0.921	0.921
Avg Epoch Training Time (s)	20.96	15.88	5.93	18.98	NA

Table 3.2: Results of model optimization visual quality poll

Model Name	10 Layer CNN	5 Layer CNN	3 Layer CNN	U-Net	Bicubic
% 1st Place Rankings	56.00%	12.67%	9.67%	21.67%	0.00%
% 2nd Place Rankings	27.67%	25.00%	12.00%	34.00%	1.33%
% 3rd Place Rankings	11.67%	34.33%	27.67%	23.00%	3.33%
% 4th Place Rankings	4.67%	24.00%	50.67%	14.33%	6.33%
% 5th Place Rankings	0.00%	4.00%	0.00%	7.00%	89.00%

From the previous tables, we can see that all of our models outperformed bicubic interpolation except for our U-Net structure in PSNR and SSIM. For visual quality, all of our models outperformed bicubic interpolation, with it consistently being ranked last. From the tested structures, the 10-layer CNN gave the best results, so this structure was chosen to be used for the final model. Additionally, the visual quality tests show that increased model complexity showed an improvement in perceived visual quality, though at the cost of increased training time, and a less transparent model, as discussed in earlier sections.

3.2 Final Trained Model

3.2.1 Final Model Structure

After optimizing the best model from the previous section, the final SRCNN model structure was tweaked to the following structure.

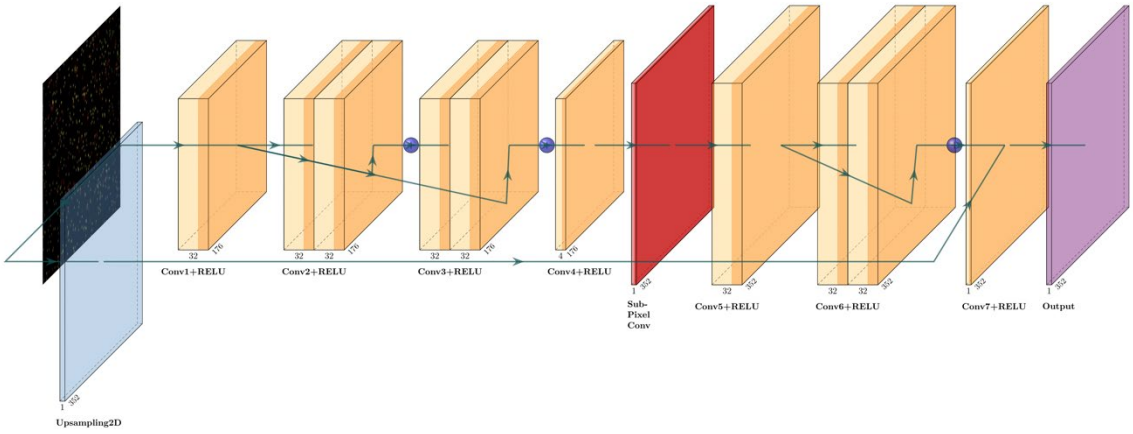

Figure 3.1: Final NN Structure Diagram

3.2.2 Measured Metrics

When training the final model, the same metrics were recorded, with the difference being the size and type of the training data. Using the methods mention in previous sections, the following metrics were measure after training.

Table 3.3: Average metrics of VCD-Net dataset trained models

Upscaling Method	PSNR (dB)	SSIM	Avg Epoch Training Time (s)	Average Computational Time (s)
Final NN model	49.22	0.992	20.96	0.7
Bicubic Interpolation	38.67	0.928	NA	NA

Table 3.4: Results of final model visual quality poll

Upscaling Method	% 1st Place Rankings	% 2nd Place Rankings	% 3rd Place Rankings	% 4th Place Rankings
Ground truth Image	96.88%	3.13%	0.00%	0.00%
Low-resolution input image	0.00%	0.00%	12.25%	87.25%
NN Upscaled image	1.56%	93.38%	3.13%	1.56%
Bicubic Upscaled Image	1.56%	3.13%	84.38%	10.94%

From the results above, we can see our final model outperformed bicubic interpolation by 27% in PSNR and 7% in SSIM. We also can see from the visual quality testing that our NN was ranked 2nd in terms of visual quality in over 90% of cases, only coming behind the ground truth image.

3.3 Understanding the Model

During the optimization process, the network changed from a low to high complexity structure with low transparency. To gain a better understanding of why the model produces the outputs it does, this section will use the XAI visualization techniques mentioned previously to analyze the final model. To do this, an input image is passed through the model, producing an output image which may then be analyzed. The input and output images from the final model are shown below.

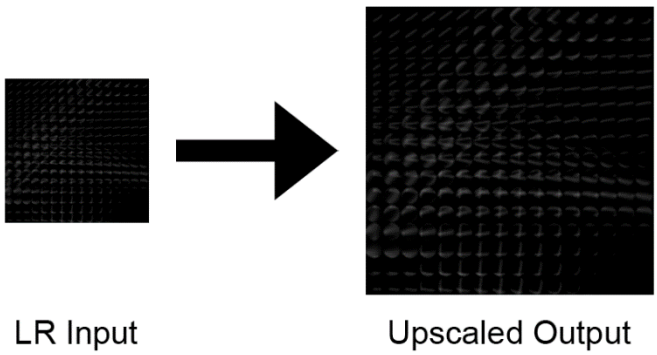

Figure 3.2: Input and output image from the trained NN model

3.3.1 Feature Maps

As mentioned in a previous section, the flow of an image through the final network can be analyzed using a feature map. By taking the output of the convolutional layers, the following results were obtained.

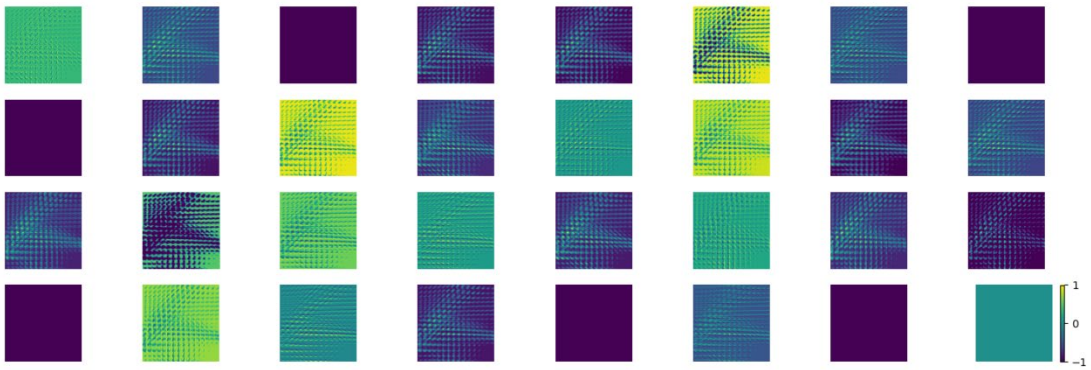

Figure 3.3: Feature map of the model's 2nd convolutional layer

The image above represents the intermediate output of the NN at the 2nd convolutional layer within its structure. Each image shows the application of a filter on the input image, with 32 total filters. The image shows that the model produces many versions of the input with various features highlighted. Because the input images lack color, the network is only able to pick up on a certain number of features. The network seems to

identify multiple structures, like the highlights within each lenslet, the dark surrounding area of the lenslets, and edges to name a few. This generally makes sense, and the feature map shows that the model seems to have learned the general structure of raw LFM images. Continuing looking at deeper layers, another pattern emerges.

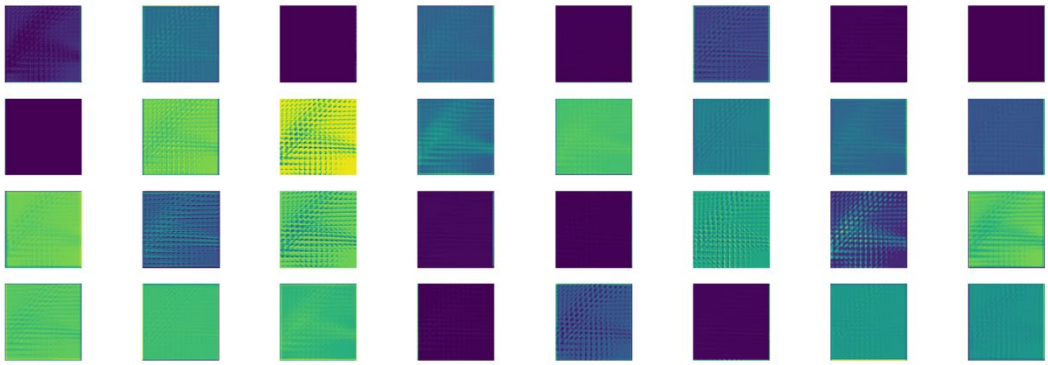

Figure 3.4: Feature map of the model's 5th convolutional layer

From this image, it can be seen that this layer shows much less detail than the previous layers and occurs right before the image is upscaled to the target resolution. Though the model still picks up on features important to the raw LFM image, like the lenslets, much less detail is shown. This is understandable, as the model is converting these features into much more abstract forms but removes the ability for humans to interpret these deeper map features. One of the most interesting facts about the network can be seen later in the network layers however, as shown below.

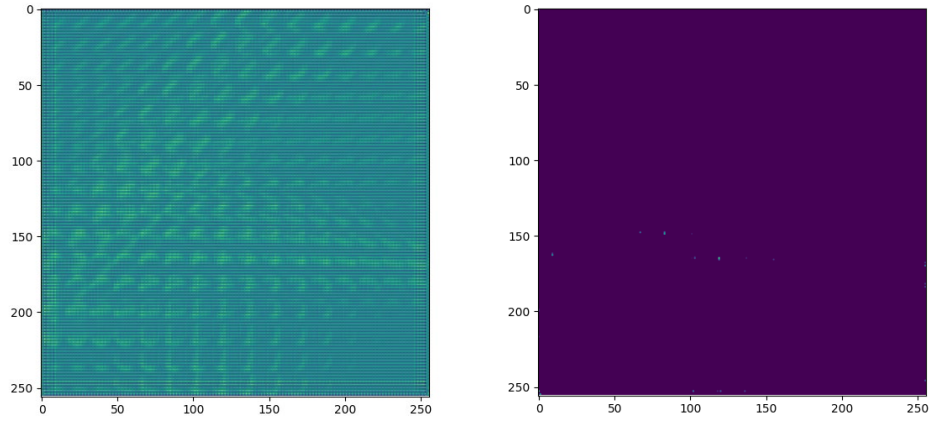

Figure 3.5: Feature map of the model's sub-pixel and final convolutional layers

The figure above shows the image at the sub-pixel convolutional layer on the left followed by at the final convolutional layer at the right. The sub-pixel convolutional layer, which upscales the input image to the target resolution, produces an image similar to the final output image, but with a noticeable grid pattern across the image. It would seem that the following convolutional layers would remove this and produce the final output, but instead the last convolutional layer seems to have very little information passed to the following layer, except for a few brighter specs. This seems to suggest that most of the information for the final upscaled image comes from the upsampling2D layer, which is added to the final convolutional layer to produce the final output image. From this, it seems that the rest of the NN's structure only produces a small number of changes to add onto normal upsampling, which produce a higher quality final image. For the feature maps of all convolutional layers, see Appendix A.

3.3.2 Filter Visualization

By visualizing the learned filters of the model, the importance of each section of a layer may be shown. These filters represent the value of learned weights of the model and can give an idea of what portions of the input the model deems as important.

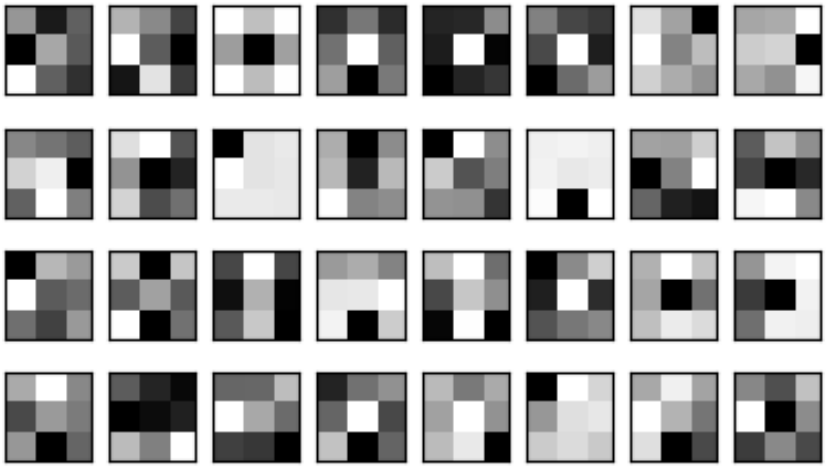

Figure 3.5: Visualization of learned weights of the model's 1st convolutional layer

This figure shows a visualization of the normalized weights of the 1st convolutional layer of the final NN model, with each 3x3 square representing one filter in the layer. In the image, dark squares indicate a small or inhibitory weight, while light sections indicate a large or excitatory weight. Looking at this, there does not seem to be a set pattern when viewing the input image, however, the network seems to look towards the center of the image more often than the edges. Unfortunately, due to the increasing number of channels as the image passes through the network, it is unfeasible to show and analyze the filters for each of the remaining layers. This does seem to suggest, however, that the network prioritizes the center of the image.

CHAPTER 4

DISCUSSION

In the development of the SRCNN, the initial model started very simple, but grew into an increasingly complex structure that gave higher metrics and increased the visual quality of the output upscaled images. In the development of the final model structure, however, it seems that the rationale behind the structure did not fully map to how the final model functions. When initially creating the model structure, the NN was divided into two sections, the upscaling half, consisting of the input-to-sub-pixel convolutional layers, and a quality enhancement half, consisting of the convolutional and addition layers after the sub-pixel convolutional layer. While initially intended to upscale the image, then use the following layers to improve the quality using residual connections, the model instead seems to extract important details from the sub-pixel convolutional layer and add those to the upsampling layer, producing a better-quality final image. Given the number of layers with feature visualizations showing little to no activation of filters, it seems possible that much of the convolutional layers do not contribute to the final output image, and the model could be simplified by increasing the number of filters, while decreasing the number of convolutional layers. Further testing could be done to see if a model structure with reduced dependance on the upsampling layer gives better or worse results than the current model.

CHAPTER 5

CONCLUSION

In this project, a SRCNN was developed that upscales an input raw 2D LFM image by a factor of 2 times. XAI techniques were used during the training, testing, and analysis of the model. Image augmentation and K-fold cross-validation were used during the training of the model, ensuring the final product would be more robust, and produce better results on novel unseen data than a model which was not trained using these techniques. After an initial round of optimization of model structure, the network was trained on LFM images. This NN was able to outperform bicubic interpolation for PSNR and SSIM of 27% and 7% respectively. Additionally, polling was done, which showed the final model was ranked 2nd in visual quality compared to the ground truth, low resolution, and bicubic interpolated images, only falling behind to the ground truth image. Analysis of the final structure showed the model learned the structure of raw LFM images, focusing primarily on the lenslets within the images. The convolutional layers identified multiple structures in the image, including the background, lenslet highlights, and edges of the lenslets. A visualization of the input filters showed the model focused more on the center of the image, with a lower priority on the edges. Feature maps of the final layers in the model also showed the model seemed to isolate important features to add to the upsampled original input image, producing the final output image with increased image clarity. This implies the model may be able to be simplified, either by reduction in the number of convolutional layers and increasing the number of layer filters, or by removal of some convolutional

layers altogether. Further testing is required to see the level to which the final trained model depends on the upsampling layer, and what effect reducing its impact on the overall model would have on the final image quality.

APPENDIX A

FEATURE MAPS OF NEURAL NETWORK CONVOLUTIONAL LAYERS

This appendix contains a collection of feature maps from each convolutional layer in the final trained NN model. It consists of the input image, followed by 5 sets of convolutional layers with 32 filters each. After this is the subpixel convolutional layer, with a filter size of 4, and then 3 convolutional layers with 32 filters again. Finally, the last convolutional layer is shown, which has a filter size of 1.

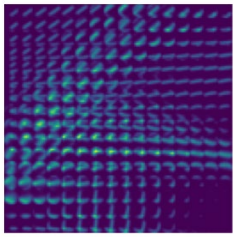

Figure A.1: Input image passed through the final model

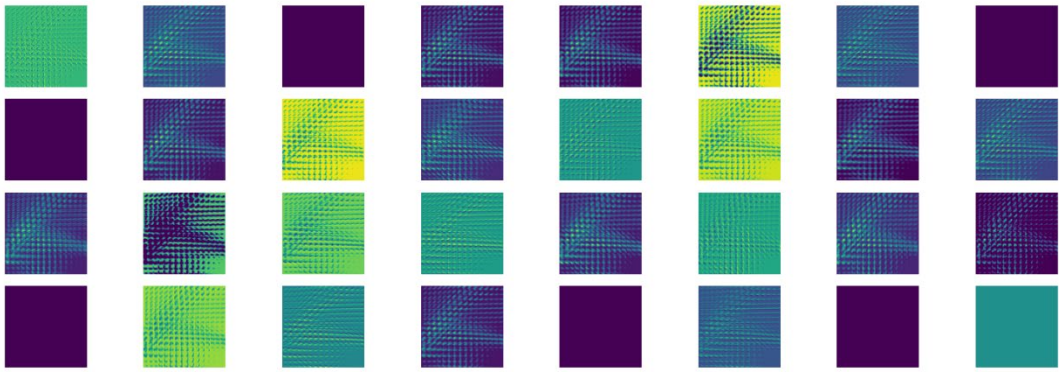

Figure A.2: Feature map of the model's 1st convolutional layer

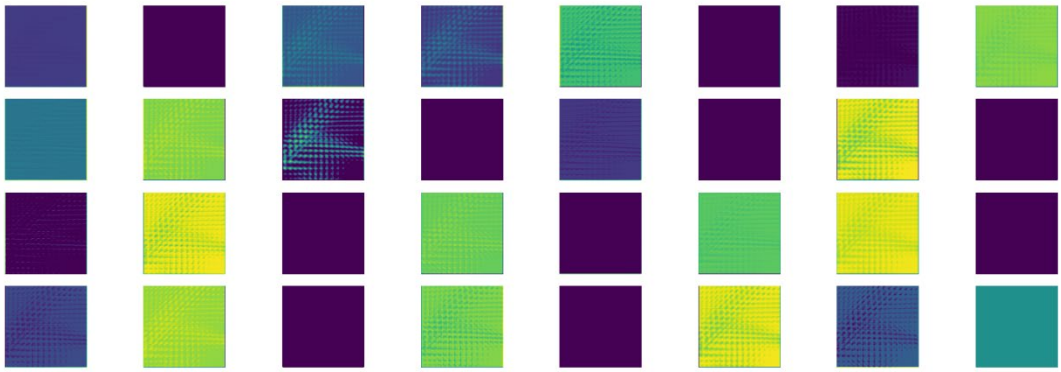

Figure A.3: Feature map of the model's 2nd convolutional layer

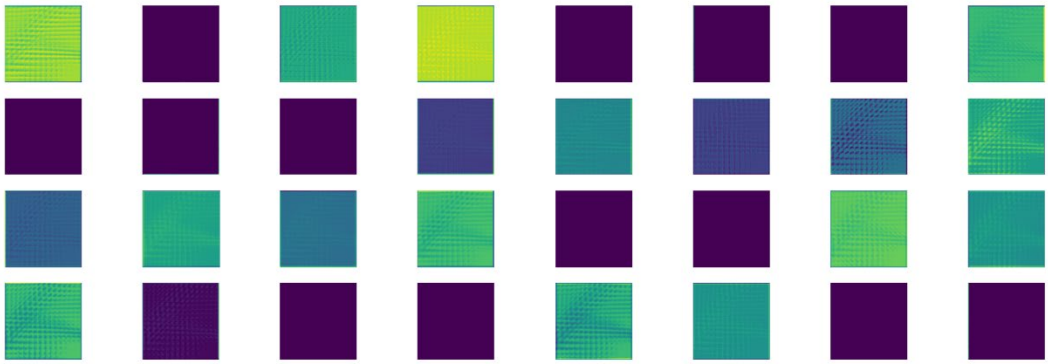

Figure A.4: Feature map of the model's 3rd convolutional layer

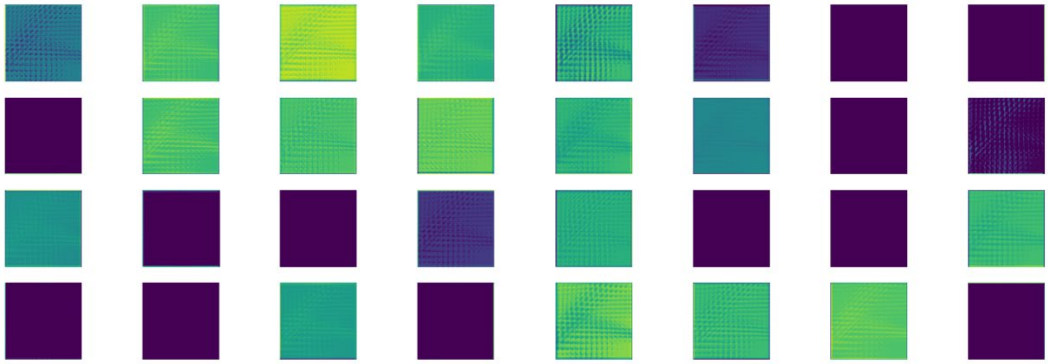

Figure A.5: Feature map of the model's 4th convolutional layer

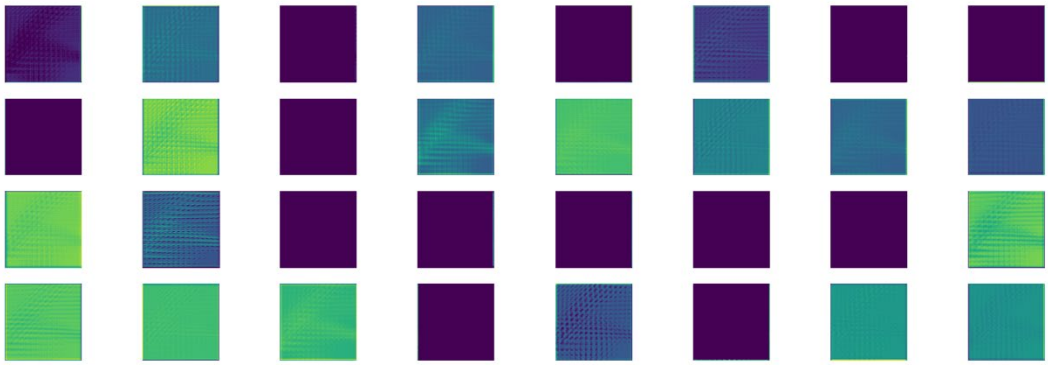

Figure A.6: Feature map of the model's 5th convolutional layer

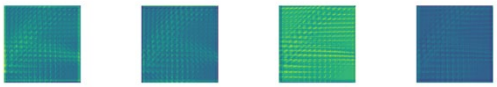

Figure A.7: Feature map of the model's subpixel convolutional layer

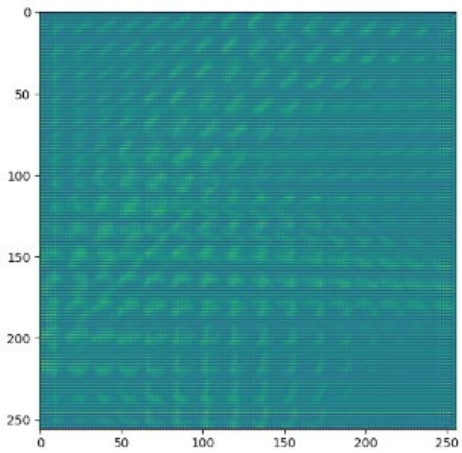

Figure A.8: Feature map of the model's 6th convolutional layer

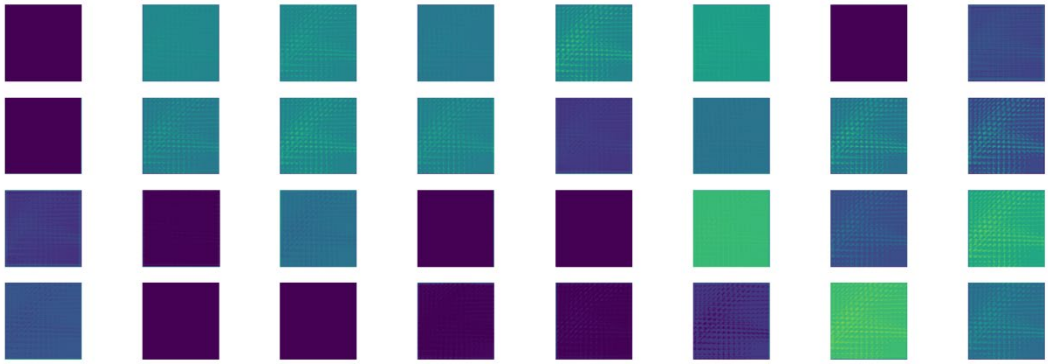

Figure A.9: Feature map of the model's 7th convolutional layer

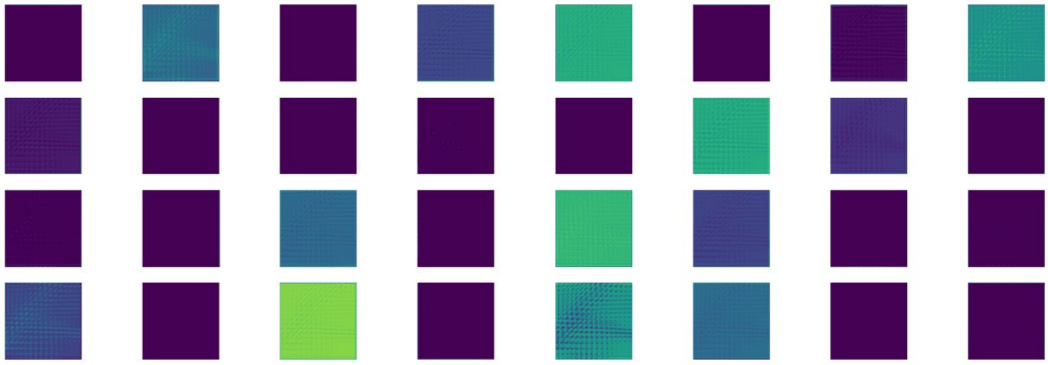

Figure A.10: Feature map of the model's 8th convolutional layer

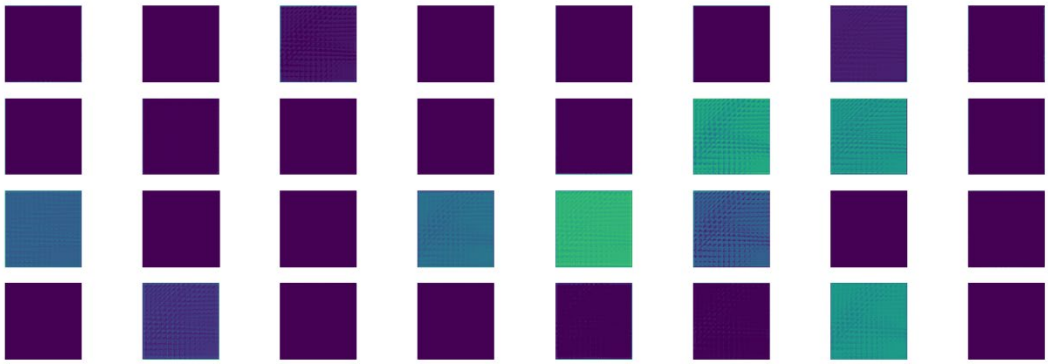

Figure A.11: Feature map of the model's 9th convolutional layer

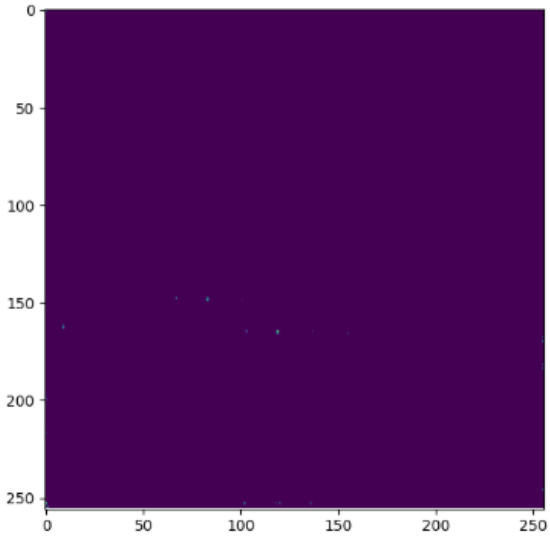

Figure A.12: Feature map of the model's 10th convolutional layer

APPENDIX B

CONTRIBUTIONS BEYOND THE SCOPE OF THE SENIOR DESIGN PROJECT

As mentioned before, this project is an addition to the work done for one of the bioengineering senior design projects over the past year. Though there is some overlap in what the projects entail, this appendix details the specific contributions added to this project that were not a part of the original senior design project. For this project, the main focus was on the implementation of Explainable AI (XAI) processes/methods to ensure that the final model would be trustworthy and increase confidence that any end-user would have in the model. The model's structure gradually increased in complexity as it was optimized this semester, which led to it becoming a black box model. As it would primarily be used in the research setting, transparency in the functionality of the model is key in allowing it to become a trusted tool for LFM image upscaling and deconvolution. Feature visualization through feature maps and filter visualization were the main XAI techniques used to show what features of LFM images the model focused on when producing the upscaled output. This analysis gave a better idea as to what decisions the network is making at each convolutional layer, which increased the transparency of the final model's structure. The analysis also identified a potential limitation of the network, that being its reliance on the upsampling layer for most of the image data when producing its output. This showed that the final model could produce similar results with a much less complex structure, as well as possible improvements and future iterations on the current structure. Generating this overview also provides a way that other end-users could implement and troubleshoot the network, modifying it for their own purposes. Lastly, using these XAI techniques increases awareness of them, and can hopefully influence future researchers to implement them as well, reducing the number of black box models in the future.

REFERENCES

Ahmed, M., & Zubair, S. (2022). Explainable artificial intelligence in sustainable smart healthcare. *Studies in Computational Intelligence*, 265–280. https://doi.org/10.1007/978-3-030-96630-0_12

Artificial Intelligence in medical science. (2014). *Medical Diagnosis Using Artificial Neural Networks*, 11–23. https://doi.org/10.4018/978-1-4666-6146-2.ch002

Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., & Herrera, F. (2020). Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, *58*, 82–115. https://doi.org/10.1016/j.inffus.2019.12.012

Brownlee, J. (2019, July 5). *How to visualize filters and feature maps in Convolutional Neural Networks*. Machine Learning Mastery. Retrieved from https://machinelearningmastery.com/how-to-visualize-filters-and-feature-maps-in-convolutional-neural-networks/

Broxton, M., Grosenick, L., Yang, S., Cohen, N., Andalman, A., Deisseroth, K., & Levoy, M. (2013). Wave optics theory and 3-D deconvolution for the light field microscope. *Optics Express*, *21*(21), 25418. https://doi.org/10.1364/oe.21.025418

Shrivastava, M., & Kumar, D. (2022). The potential of artificial intelligence in public healthcare industry. *Impact of Artificial Intelligence on Organizational Transformation*, 349–360. https://doi.org/10.1002/9781119710301.ch20

BIOGRAPHICAL INFORMATION

Nicholas Laudermilk is a graduating senior at the University of Texas at Arlington's College of Engineering, pursuing an Honors Bachelor of Science in Biomedical Engineering with a minor in Mathematics and Physics. They first became involved in research on campus in Spring 2019, when they joined Dr. Nguyen's Nanomedicine and Tissue Engineering lab as a volunteer. During the following summer, they joined the UROP program, and led their own research project focused on the development of nanoparticle conjugated microbubbles. Over the past year, Nicholas' interest has shifted to the usage of AI systems in the healthcare setting. Their interest stems from their work on their senior design project, and they wish to research new use cases for neural networks in diagnosis and treating diseases. After graduation, Nicholas plans to attend grad school to further their knowledge of artificial intelligence and progress towards a career as a professor.