We appreciate for all three review teams and they did a really good job! Their kind and insightful advices make our project stronger and more solid. We make some modify in our project and final report base on their useful commons. Thanks again for their review!
The reply of each individual common can be found here:

A. *reviewG43_80*

1. It Is unclear how you generated the Dataset. You talk about using a commercial package to generate ultrasonic signals, but then you mention changing the shape of the device. To what device does this refer? Is this verasonics system generating an ultrasonic image of a physical device or does "changing the shape of the device" refer to changing the characteristics of the virtual ultrasonic signals generated?
   We are sorry for the unclear introduction of the dataset part and we did a detailed discussion in our final report (III. DATASET AND FEATURES).
   In our video, the Verasonics system will generate the simulated ultrasonic signal and the signal need to be processed by beamforming algorithm to finally form a ultrasonic image, for the beamforming part, we made a python script to achieve it and the python script will be called after the simulated ultrasonic signal generated by Verasonics system.
   The 'device' we refer is the soft ultrasonic probe.
2. You do a good job of presenting traditional de-noising methods as well as the old school ML methods of thresholding, k-means, etc. How does your solution compare to these methods, is it better or worse, and why?
   This is a good question and we did an introduction about the old school denoising algorithms in final report (II. RELATED WORK) and we also mentioned this part in our video. The old school denoising algorithms essentially will try to focus on some special and certain noise domain but in our case the noise come from the uncertain positions of sensors and the beamforming algorithm, so our distorted image is much more complicated and does not have an obvious frequency domain, it need an entirely reconstruction but not any sample filters or clusters. Therefore, the convolutional autoencoder can did a better job than the other traditional approaches.
3. Wouldn't hurt to mention your training and test set sizes, especially since you mention you need a larger dataset in future work to avoid overfitting.
   We agree with this advice and actually we mentioned the size of our dataset in the video and also in the slides, which is ~3000 images. In order to make a quick try we use a smaller training set which contains 1000 training images and 100 test images and these parameters were also shown in the code introduction part.
   Since both of us think the size is important, so next time we maybe need to speak more loudly about these parameters.
   For the overfitting problem, the consideration is that when compared with professional image dataset(usually have more than 10k images), our homemade simulation dataset is so small and less of diversity, so maybe overfitting will still occur when our model is applied in real human body and a larger dataset can definitely give us a better performance.
4. You stated that the output was low resolution because of the low resolution input (64 x 128 pixels), but why is the resolution for the input so low?
   Because pictures from soft ultrasonic probe are 64 x 128 pixels.

B. *reviewG52_80*

1. It wasn't exactly clear to me what the image segmentation part of the project was trying to accomplish. I assume to segment whatever was in the ultrasound image, but the results seemed unclear as there were no bounding boxes. Additionally, it was unclear where this training set came from.
   The segmentation part is used to detect /classify carotid artery. The initial tag is simply blood vessel. The train set came. from pictures generated from U-net, which is our first part of the project.
2. For the YOLO code it was said that this was directly forked from GitHub. It was unclear if any modifications were then done on this code. I think it is important to talk about how the model is changed for their specific task. For example, were fully connected layers retrained? Was the original model pretrained?. Additionally for the UNet model it was unclear if this was a pretrained model or a model they built themselves.
   The YOLO model is converted to Tensorflow keras with pre-trained weights (trained in coco set) and then train it again with no modification on our own dataset.
   The U-Net model is not a pretrained model and we built it then train the model with random initial premasters to avoid 'lucky initiation'.
3. There wasn't much talk about tuning of the models or the data. For example, how did they preprocess the images? Would have been good to hear a discussion of parameters that were changed such as learning rate, optimizer, batch size, etc.
   We totally agree with this kind advice and due to the time limitation, we did not have the opportunity to introduce too much detail about our model, but we show as much as we can in our final report and we are happy to have more discussion about any problem.

C. *reviewG58_80*

1. The choice of UNET for the denoising: was there any specific reason to choose this architecture?
   Yes, we first did a survey about the denoising algorithms and then chose U-Net based on the pros and cons of different methods. We did a detailed discussion in our final report about this and thanks for this helpful advice!

2. How was the final architecture of UNET followed by YOLO arrived at?
No, the distorted image will go through U-Net first and then input to Yolo to do object detection. (As shown in S1)

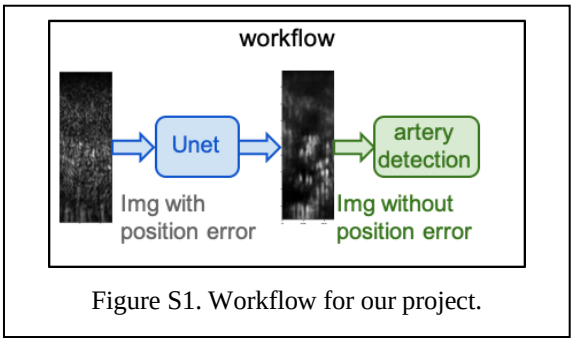

Figure S1. Workflow for our project.

3. A little more explanation about the dataset and if there are any scores available for them.

Yes, we totally agree with the reviewers and we did more explanation about our dataset in our final report.

Ultrasound Imaging Optimization via Machine Learning

Grpup 80

***Abstract*—This project is aiming at solving denoise and object detection problems on biomedical images generated by the state-to-art soft wearable ultrasonic probe. Two machine learning based frameworks are used in this project, one is the convolutional autoencoder neural network for image denoising, another one is the YoloV3 to detect the position of ceroid artery. The dataset will be established by ultrasonic simulation system and the models will be trained on GPU. The design, analysis and result will be shown in this repost.**

Keywords—ultrasonic image denoising, object detection, convolutional autoencoder, YOLOv3

II. INTRODUCTION

Ultrasound (US) imaging is a safe and powerful tool for providing detailed still and moving images of the human body. From the ultrasonic images, clinic doctors can clearly find out the statement of different organs and diagnose varies of disease. However, most of today's US systems are designed for use only in hospital. This configuration hinders its use in locations lacking clinical settings and professional doctors. In recent years, more and more researchers try to innovate a new approach to break through the limitation of the traditional US technique, in 2018, benefiting from the development of soft electronic technology, using a stretchable wearable ultrasonic probe to do daily health monitoring becomes feasible. This research group from UCSD demonstrate a fantastic soft US probe that can closely fit the human skin and transmit US signal as shown in Figure 1. However, they still face challenges in algorithms.

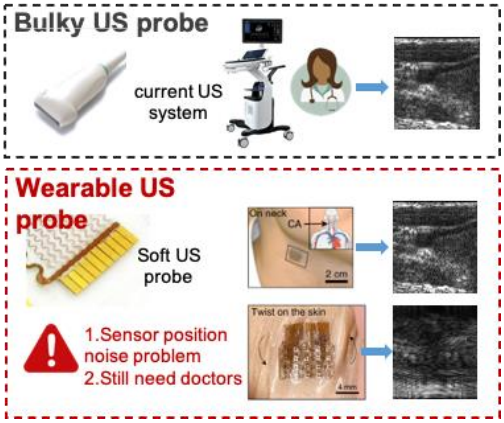

Figure 1. The soft US probe and the challenges.

A major challenge of using soft probes to perform US imaging is that the locations of transducer elements are uncertain for most application scenarios. When integrated into the human skin, the soft probe will be laminated on dynamic curvilinear surfaces and the transducer locations will be ever-changing. Due to the Delay And Sum (DAS) imaging algorithm, images reconstructed from the soft probe will be significantly distorted by the position noise. Another challenge is that it is hard for untrained users to get information from the US images directly without any professional guidance from clinical doctors because the output images have only black and white colors for them and it is challenging for new users to distinguish different organs from the B-mode ultrasonic images.

Our goal is to solve these two problems via machine learning algorithms. We expect our final model can handle the distorted images with uncertain sensor's position and fix the image via a convolutional neural network called U-Net to output clear images. Moreover, we will propose an autonomous method to detect target organs(we will use carotid artery as an example target) from the clear ultrasonic images to make the soft ultrasound device more friendly to any users who lack the guidance from trained clinic experts.

III. RELATED WORK

A. Image Denoising Algorithms

From the literature, image denoising algorithms can be summarized into three main groups: classical denoising method, transform techniques and machine learning-based denoising methods.

For the classical denoising methods, most of them aim to remove noise by calculating the grayscale of each pixel based on the correlation between image patches in the original image. Similar to classical methods, the transform techniques also focus on correlations but will first transform the given noisy image to another domain, and then they apply a denoising procedure on the transformed image according to the different characteristics of the image and its noise. The classical and transform technique usually have a good performance when processing images with certain destitution noise, but in our case, the noise inside the ultrasonic images is related to uncertain sensor's positions and does not have an obvious distribution or special domain, which means the classical and transform technique may not a good choice to solve our challenges.

The machine learning-based denoising methods have been made great achievements in recent decades, due to the strong feature extraction capability of convolutional neural network, the machine learning-based denoising methods especially the convolutional autoencoder technique always can reach a better performance in most of the denoising applications when compared with traditional methods. According to the complexity of the transducer position noise, the U-Net, which is an powerful convolutional autoencoder based machine learning algorithm, can be applied to our project to reduce the noise in US images.

B. Image Segmentation/Classification Methods

There are many existing image segmentation and object detection approached, and can be summarized into two main groups: Machine Learning based approaches and Deep Learning approaches. For Machine Leanring approaches, it is necessary to statistically compute features of object and then apply a technique such as support vector machine to do the classification/segmentation. Without specifically defining features under the object, Deep Learning approaches can provide end-to-end object detection and are typically based on Neural Network, more specifically, Convolutional Neural Network.

ML based approahces	DL based approaches
Viola-Jones framework	R-CNN
HOG features	Fast R-CNN
SIFT	Single Shot Multiple Detector

IV. Dataset and Features

Since the US images from soft probe are special, we established the dataset by ourselves and the wearable ultrasound research group at UCSD (Prof.Sheng Xu's group) provided a lot guidance for this part. Here are the details of the dataset establishment:

A. Sensor Position Nosie Generation

The dataset is established via two steps: First, a commercial US simulation system called Verasonics is applied to generate the US signal with different shapes of US probes. Second, the US signal is processed by a DAS module to form the US images with/without position errors. The DAS module is developed by python and the process is shown in Figure 2.

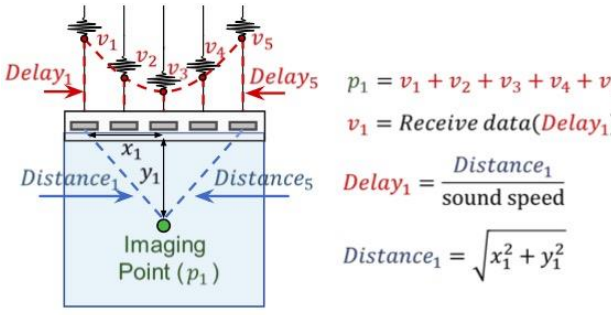

Figure 2. The DAS beamforming.

From the Figure 2, we can see that if the positions of sensors or transducers are uncertain, the delay calculation would has some error and let to a wrong signal added into the pixel in one US image. Therefore, the image with uncertain positions of sensors can have lots of noise and distortion.

For the object detection task, we use our simulation system to generate the artery position file for each corresponding image so that we can use our simulation dataset to train the detection network.

The final dataset will contain ~3,000 US simulated images of carotid artery and will be split by 7:2:1 to form the training dataset, validation dataset and test dataset and all images are reshaped to 176X64 and preprocessed via min-max normalization. Figure 3 shows the image inside the dataset.

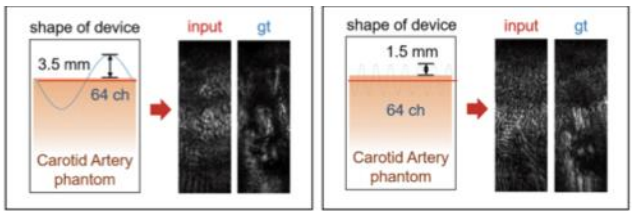

Figure 3. The input and ground truth data in our dataset.

V. Methods

A. U-Net

U-Net is the one of the convolutional neural network architectures proposed by Ronneberger et al. The architecture of U-Net looks like an alphabet 'U' which justifies its name. As shown in Figure 4, this architecture consists of three sections: The encoder side, The bottleneck, and the decoder side. The encoder side is made of three contraction blocks. Each block takes an input applies two 3X3 convolution layers followed by a 2X2 max pooling. Here the function of 3X3 convolution layers is to extraction the features, the 2X2 max pooling layers are aim to down-sample the hidden-layer output matrix to reduce the dimensionality and allow for assumptions to be made about features contained in the sub-regions binned. The number of kernels or feature maps starts form 16 and after each block and doubles so that architecture can learn the complex structures effectively. The bottommost layer mediates between the encoder layer and the decoder layer. It uses two 3X3 CNN layers followed by 2X2 up convolution layer.

But the heart of this architecture lies in the decoder side,

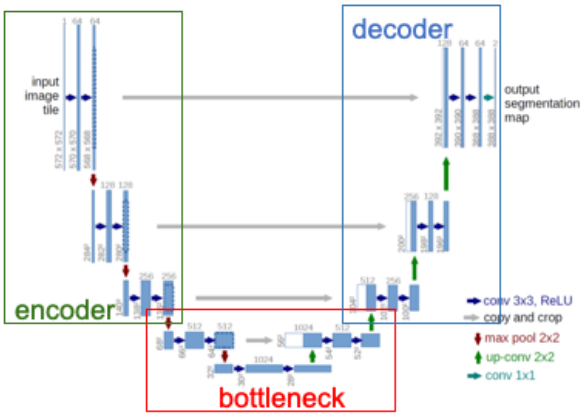

Figure 4. The U-Net architecture.

which is the right part in the Figure 4. Similar to encoder layers, it also consists of three expansion blocks. Each block passes the input to two 3X3 CNN layers followed by a 2X2 up sampling layer. The up sampling here is in order to make sure that the final dimension of the output image is as same as the input image. Also after each block number of feature maps used by convolutional layer get half to maintain symmetry. However, the critical part is that every time the input is also get appended by feature maps of the corresponding contraction layer. This action would ensure that the features

that are learned while contracting the image will be used to reconstruct it, which can benefit the output performance.

B. YoloV3

You Look Only Once is a state-of-the-art object detection algorithm which can detect multiple objects in a picture or video very fast and accurately. Unlike prior object detections, who apply the model to a picture or video at multiple locations and different scales, YOLO uses a single neural network to the full picture. The output picture is divided into regions. YOLO then predict the picture with multiple bounding boxes on it (if there are multiple objects detected) with confidence score for each region. YOLO v3 is much faster than R-CNN and Fast R-CNN since they both require thousands of pictures to train. The underlying mechanism of YOLO is the same from version 1 to 3, they all apply just one single neural network. The differences are loss functions, bounding boxes initialization and output format. Therefore, in this report, we only explain the basic ideas of YOLO v3.

YOLO v3 uses a brand-new network called Darknet-53. It is based on some ideas from ResNet such as Residual module of ResNet. The residual module can help resolve gradient problems in deep network. Every residual module consists of two convolutional layers and one shortcut connection. There is no maxpooling layer or fully connected layer. Downsampling from the pictures in Darknet-53 is done by setting stride=2. Every time a picture pass a convolutional layer, the picture is resized to half size of its last size. Every convolutional layer includes a Convolutional layer, Batch normalization and Leaky ReLu.

	Type	Filters	Size		Output	
	Convolutional	32	3	3	256	256
	Convolutional	64	3	3 / 2	128	128
1	Convolutional	32	1	1		
	Convolutional	64	3	3		
	Residual				128	128
	Convolutional	128	3	3 / 2	64	64
2	Convolutional	64	1	1		
	Convolutional	128	3	3		
	Residual				64	64
	Convolutional	256	3	3 / 2	32	32
8	Convolutional	128	1	1		
	Convolutional	256	3	3		
	Residual				32	32
	Convolutional	512	3	3 / 2	16	16
8	Convolutional	256	1	1		
	Convolutional	512	3	3		
	Residual				16	16
	Convolutional	1024	3	3 / 2	8	8
4	Convolutional	512	1	1		
	Convolutional	1024	3	3		
	Residual				8	8
	Avgpool		Global			
	Connected		1000			
	Softmax					

Figure 6. Darknet-53

Above is the feature extractor part of YOLO v3. There is also a bounding boxes initialization part along with feature extraction called Anchors. Basically, it learns the ratio of the trained object and predict the size of Anchors of possible objects in a given picture using Logistic Regression.

VI. Experiments and Results

A. U-Net Trainig Process

If we define the noisy image as $\overline{u_i}$ and the output image as u_i , we can get:

$$h(\overline{u_i};\beta) = u_i \tag{1}$$

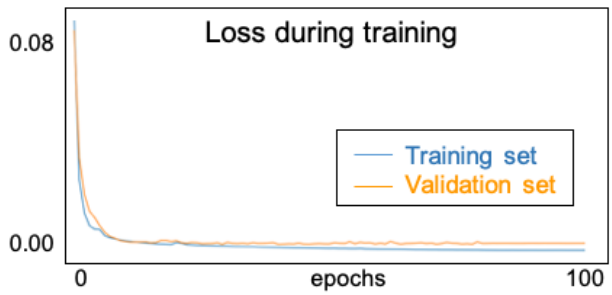

Figure 7. The training result of the U-Net.

where β is the parameters of the U-Net and h is the model. For the loss function $L(\beta)$, we chose the Mean Square Error (MSE) since MSE is the most commonly used regression loss function and used to show a good performance in our previous 2D convolutional autoencoder project. MSE is the sum of squared distances between our target variable and predicted values and the entire training process actually is to minimizes the MSE:

$$L(\beta) = \frac{1}{N}\sum(h(\overline{u_i};\beta) - u_i)^2 \tag{2}$$

Minimization of (2) was done with Adam under TensorFlow. Adam is a replacement optimization algorithm for stochastic gradient descent for training deep learning models. We try 10^{-2}, 10^{-3} and 10^{-4} as the learning rate and found that the best parameter here is 10^{-3}. In order to get an effective and sufficient training process, the batch size we chose is 32. Training and test losses were computed during training and early stopping was adopted to avoid overfitting. Usually we would get an early stop at around 100 epochs for training. Figure 7 shows the loss decrease during the training process.

B. U-Net Trainig Result

To test the performance of our U-Net model, we calculate the difference between the ground truth and the image processed by our U-Net model and also the difference between the ground truth and the image with noise which actually is the input of our U-Net model. Figure 8 shows the comparison of these two differences. From the comparisons

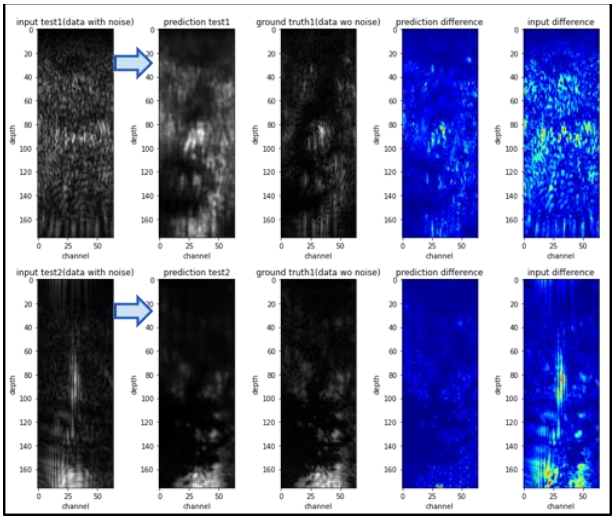

Figure 8. The testing result of the U-Net.

we can observed that the input distorted image is reconstructed by the U-Net successfully and although the output image is a little blurry, but the noise level decrease a lot and we can distinguish different organ and tissue structure from it easily.

C. *YOLO v3 Result*

We trained our model on 3279 samples and validate on 364 samples with batch size 4, finally we test our data on 24 samples. The loss function is called YOLO loss and defined

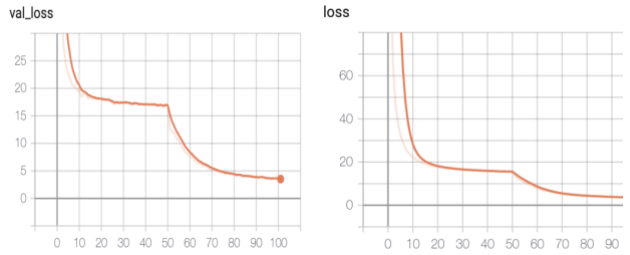

Figure 9. Validation loss and Train loss

as:

$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$
$$+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$
$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2$$
$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2$$
$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2$$

where it consists of two parts. The first part is **Localization loss**:

$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$
$$+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$

where

$\mathbb{1}_{ij}^{obj} = 1$ if the j th boundary box in cell i is responsible for detecting the object, otherwise 0.

λ_{coord} increase the weight for the loss in the boundary box coordinates.

The second part is Confidence loss, the confidence loss is computed in two cases, when it finds the object:

$$\sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2$$

where

$\hat{C}_i$ is the box confidence score of the box j in cell i.

$\mathbb{1}_{ij}^{obj} = 1$ if the j th boundary box in cell i is responsible for detecting the object, otherwise 0.

when it does not find the object:

$$\lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2$$

where

$\mathbb{1}_{ij}^{noobj}$ is the complement of $\mathbb{1}_{ij}^{obj}$.

$\hat{C}_i$ is the box confidence score of the box j in cell i.

λ_{noobj} weights down the loss when detecting background.

By the deadline of submitting this report, we still haven't figured out a way to make the bounding boxes appear and label clear. The only way we can prove our result is accurate is to look through Annotations-export.csv to manually locate the boxes. The naïve purpose of YOLO training now become detecting if there is a blood vessel (carotid artery) in simulated US images.

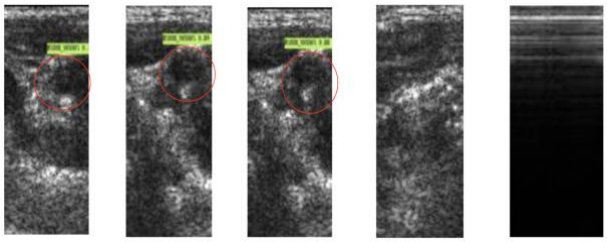

Figure 10. Test results

VII. CONCLUSION AND FUTURE WORK

In conclusion, we successfully built our own simulated dataset and established two machine learning model: U-Net and YOLO v3 to optimize the ultrasound imaging process of the advanced soft ultrasonic transducer probe. From our result, we believe that our U-Net model decrease the noise level and correct the uncertain sensor position error significantly, and the YOLO v3 model achieved an autonomous organ detection which can reduce the difficulty of this soft electronic technique when applied to new user without any guidance from clinical doctors. Benefit from the strong capability of feature extraction, our convulsion autoencoder method reach a better performance of denoising when the image noise is complex and has uncertain domain. The connection between encoder side and decoder also make sure the reconstruction image contains more details then the traditional autoencoder model, which further improve the denoising performance. The YOLO v3 model itself is an end-to-end model but we haven't integrated with the output from Autoencoder. However, by manually handling data, we can still easily achieve our object detection goal.

For future work, if we had the opportunity to get more time, more team members and especially more help from the clinical doctors, we expect to try larger and deeper leaning model to further reduce the noise level and did denoising and segmentation via only one large model, and can detect more organs under the human skin or even can do some disease prediction, which will make this project more valuable.

CONTRIBUTIONS

Xinyu Tian:

- Pre-processed dataset
- Simulating dataset
- Denoising Autoencoder training
- YOLO v3 training
- Report writing

Yudong Diao:

- Denosing Autoencoder training
- YOLO v3 training
- Report writing

REFERENCES

[1] Redmon, Joseph, and Ali Farhadi. YOLOv3: An Incremental Improvement. 2018, *YOLOv3: An Incremental Improvement*, pjreddie.com/media/files/papers/YOLOv3.pdf.

[2] "YOLO Deep Learning: Don't Think Twice." *MissingLink.ai*, missinglink.ai/guides/computer-vision/yolo-deep-learning-dont-think-twice/.

[3] Jain P, Tyagi V (2016) A survey of edge-preserving image denoising methods. Inf Syst Front 18(1):159–170. https://doi.org/10.1007/s10796-014-9527-0

[4] Nah S, Kim TH, Lee KM (2017) Deep multi-scale convolutional neural network for dynamic scene deblurring. In: Abstracts of 2017 IEEE conference on computer vision and pattern recognition. IEEE, Honolulu, pp 257–265. https://doi.org/10.1109/CVPR.2017.35

[5] Vincent, Pascal, et al. "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion." Journal of machine learning research 11.Dec (2010): 3371-3408.

[6] Gondara, Lovedeep. "Medical image denoising using convolutional denoising autoencoders." 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW). IEEE, 2016.

[7] Lessons, Python. "YOLO v3 Theory Explained." *Medium*, Analytics Vidhya, 23 Jan. 2020, medium.com/analytics-vidhya/yolo-v3-theory-explained33100f6d193#:~:text=A%20Fully%20Convolutional%20Neural%20Network,YOLO%20makes%20use&text=As%20it's%20name%20suggests%2C%20it,to%20downsample%20the%20feature%20maps.