Image-to-Image Translation to learn SSIM Quality Map

Jinghan Zhou, Xinyu Guo, Yashesh Dasari
SYDE 675 – Group 16, Option B
Electrical and Computer Engineering Department
University of Waterloo, Canada
{j263zhou, x227guo, ydasari}@uwaterloo

***Abstract*—We conduct an empirical evaluation in the area of image-to-image translations using different machine learning models, specifically those based on deep neural networks (DNN). We focus on comparing these models using an Objective Image Quality Assessment (IQA) tool called Structural SIMilarity index (SSIM). SSIM is based on the degradation of structural information in an image after it is translated or distorted. We train and validate the selected deep neural networks using the Waterloo Exploration I Database.**

***Index Terms*—Image-to-image learning, Image Quality Assessment, Quality Map, SSIM**

I. Introduction

In image-to-image translation, we “translate” an input image into a corresponding output image, and many problems in image processing, computer graphics and computer vision can be treated as an image-to-image translation problem [1]. There are different ways to record a scene. For example, one can use an RGB image, a gradient field, an edge map, a semantic label, etc. Usually, we are given one of these representations in a typical image-to-image translation task, and we are required to get another representation. An example of Image-to-image translation is shown in Fig.1.

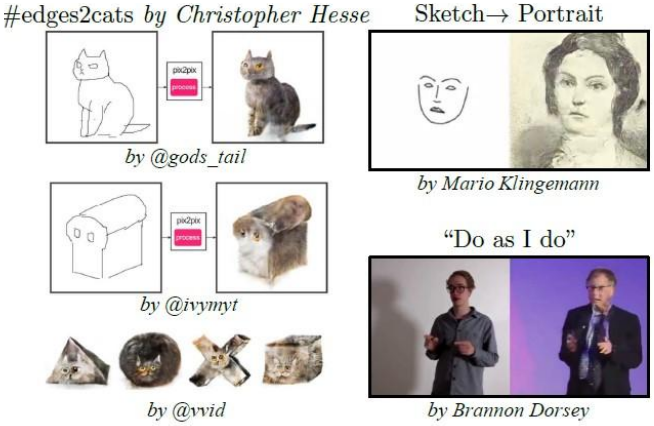

Fig. 1. Examples for the image-to-image translation [1]: $\#edges2cats$ [2] by Christopher Hess, ”$Do\ as\ I\ do$” [3] by Brannon Dorsey and $Sketch \rightarrow Protrait$ [4] by Mario Klingemann.

Image quality assessment (IQA) is a part of the quality of experience measure, which is a very fundamental problem in computer vision. Full-reference image quality assessment (FR-IQA) and no-reference image quality assessment (NR-IQA) are two very important methods used in IQA. (An example of reference images and distorted images used in IQA can be found in Fig.2) Some FR-IQA methods like SSIM[1] tried to build a model to simulate the human visual system (HVS) and they can give a very specific quality map to show the quality of every single pixel in the image and then use the map to predict the quality score of the whole image. While some NR-IQA models also tried to predict the quality depending on the natural scene statistics (NSS) but got obviously worse results than FR-IQA models.

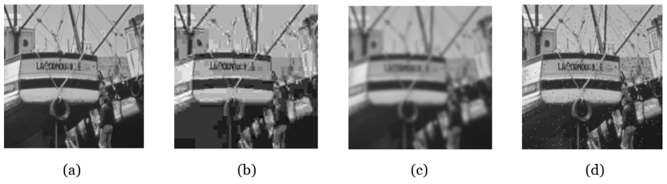
(a) (b) (c) (d)

Fig. 2. An example of reference image and its distorted images [5]. (a) is the original image, (b) is the JPEG compressed image, (c) is the contrast stretched image, and (d) is the salt-pepper impulsive noise contaminated image.

Recently, researchers have used deep neural networks (DNN) to predict the quality scores, which is named Deep-IQA. So based on this recent development, we want to use DNN to learn the NSS features captured by the FR-IQA model to predict the quality maps of images without references. Then we are expecting to get better results by using the quality maps to predict the quality scores of the images. In this work, we use the very famous SSIM as the FR-IQA model to learn, and we basically focus on predicting the quality maps of the images with distortion.

The rest of the paper is structured as follows. Section II introduces the background knowledge of this work, including Image-to-Image Translation and Image Quality Assessment. Section III introduces the methods we use in our work. The training details, the results and analysis of the experiments are introduced in Section IV. And finally, the conclusions with potential future work are in Section V.

II. Background

A. Image-to-Image Learning

Image-to-image learning tasks can be considered to be per-pixel classification or regression problem [1]. In the early development phase in this area, researchers were able to develop task-specific algorithms using special-purpose machinery [1]. However, in the recent years successful advances have been made to develop more generic networks that are capable of handling such tasks of pixel-to-pixel analyses. One of the common models used is the convolutional neural nets (CNNs). To further automate the manual inputs associated with CNNs, a new network was proposed called the Generative Adversarial Networks (GANs) [1]. There are other networks too which use similar quality maps, for example, fully convolutional neural network (FCNNs). A more recently proposed network which has shown promising results in image processing tasks is Deep Convolutional Neural Networks (DCNNs) [6].

In this work, we investigate some of these popular networks to conduct an empirical evaluation in terms of performance parameters. The models investigated are: FCN, DCNN, U-Net, U-Net32, Densenet121-UNet, Resnet18-UNet, Resnet50-UNet, and GAN(U-Net). These methods are explained in Section III.

B. Image Quality Assessment

Image Quality is a measure of the level of accuracy of the scene captured. When an image is translated from one form to another, there are several losses that accompany the translation. Image Quality Assessment (IQA) is a metric to quantify the errors between the original image and the translated image during an image-to-image translation task. There are broadly two ways of assessing image quality, namely, subjective evaluation, and objective evaluation. Subjective evaluations are usually inconvenient, expensive, and time-consuming [5]. Additionally, often it is difficult to completely automate subjective evaluations. Therefore, objective evaluation techniques are used to develop quantitative measures of image quality [5].

Objective IQA is considered to be a fundamental problem in tasks related to computer vision and image processing. Therefore, building an accurate IQA model is of high importance in such tasks [7].

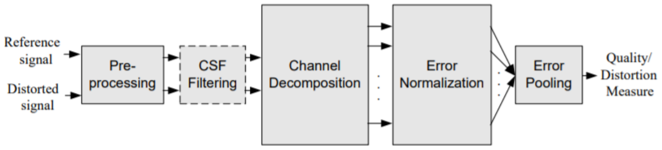

Fig. 3. An image quality system based on error sensitivity [5].

III. Method

In this section, we introduce the different neural networks we have used in this experiment, including the Deep Learning (DL) models coming from different fields. In addition to exploring some common DL models for the task, we also investigate conditional GAN as a part of this project.

A. FCN

The Fully Convolutional Networks (FCNs) [8] are originally created for image segmentation which needs dense prediction. Thus, this method allows mapping from image to image with arbitrary size. For extracting features from input image, the VGG architecture is used, then transposed convolution is used to recover the size of the image. For simplicity and speeding up the training process, we remove the fusion process used in the original paper.

B. U-Net

The U-Net was originally proposed for medical image segmentation. U-Net typically contains a contraction section, a bottleneck section, and an expansion section. In the contraction section, each basic block is formed by two consecutive convolutional layers and a down-sample layer. The contraction section and the expansion section is connected by several convolutional layers which were formally called bottleneck. The basic block in the expansion section is formed by two consecutive convolutional layers and an up-sample layer. [9] The structure is shown is the Fig.4. For our experiment, we have used U-Net and U-Net32 which denote a 4-layer and a 5-layer U-Net respectively.

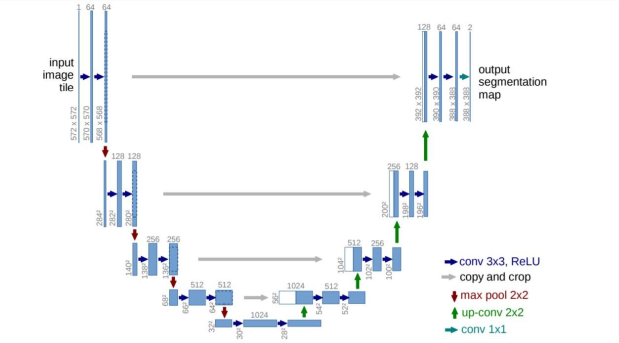

Fig. 4. The U-Net Architecture [9]

C. Densenet and Resnet

The Densely Connected Convolutional Network (DenseNet) was originally proposed for visual object recognition. The DenseNet are featured with dense blocks and transition layers. For the dense blocks, each layer takes the feature map from all previous layers, and the transition layer is formed by a convolutional layer and a pooling layer. [10] For applying this network to the image to image translation task, we use a U-Net decoder. [11] As mentioned in the U-Net section, the expansions section in U-Net is responsible for expanding the feature size. Thus, we learn representative features using DenseNets and then up-sample the feature map using a U-Net decoder to map from input image to its SSIM quality map. In our experiment, we have used 121 layered DenseNet.

The ResNet model is also proposed for object recognition. The Resnet is featured with its identity shortcut connection. In this connection, a new branch directly connects the input with the output [12]. Usually, when you increase the number of layers in a Deep Neural Network (DNN), the gradient in the front layers can go to infinitely small. By introducing this direct connection, the gradient can be preserved when constructing a very deep neural network. We introduce this network to our image to image translation task. For ResNet, we also use a U-Net encoder which is the same as what we have done when introducing DenseNets, and in our experiment, we have tried 18-layer, 50-layer ResNet.

D. DCNN

Deep Convolutional Neural Networks (DCNNs) have pushed the performance of machine learning models in high-level tasks like computer vision and image processing, like image classification. DCNNs were widely used conventionally for document recognition related tasks. But recent experiments have been able to produce state of art performance in high level tasks such as image classification and object detection. DCNNs trained in an end-to-end manner perform better and have shown much better results in comparison to other models which based on manually crafted features [6].

An interesting feature of DCNNs is their built-in invariance to local image transformations and this allows the network to learn more abstract data representations. Most of the successful image segmentation developed in the previous decade relied on hand crafted features. Improvements were achieved over the years but the performance was limited because of the limited feature capabilities. Over the past few years, Deep Learning algorithms in image classification were successfully transferred to semantic segmentation tasks [6].

E. GAN

The Generative Adversarial Network (GAN) is featured with a generator which is used to generate an image from input noise, a discriminator which is responsible for discriminating whether a input is real or fake. For the purpose of image-to-image translation, we use a conditional version of GAN. [1]. Both the generator and the discriminator can observe the input image, and try to make a real output SSIM quality map. The conditional GAN framework is shown in Fig.5

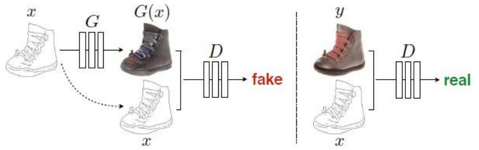

Fig. 5. The Conditional GAN Framework [1]

x denotes the input into the framework and y denotes the output image we want to produce. The objective of the Conditional GAN is given by following formula.

$$\mathcal{L}_{cGAN}(G,D) = \mathbb{E}_{x,y}[\log D(x,y)] + \mathbb{E}_{x,z}[\log(1 - D(x, G(x,z))]$$

Where G tries to minimize this objective. The task for D is to maximize this objective. Additionally, the $L1$ distance between y and the fake y generate by G is also considered. Thus the final objective is.

$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G,D) + \lambda \mathcal{L}_{L1}(G). \quad (1)$$

IV. Results & Analysis

A. Dataset & Training Details

Dataset: For training and validation, we use the dataset generated from Waterloo Exploration I Database [13]. The Waterloo Exploration Database is a large-scale database with 4744 pristine images and 94880 distorted images created from them. All the pristine images in this database are captured from the real world. And there are four different distortion types in this database, which are Gaussian blur, Gaussian white noise contamination, JPEG and JPEG2000, with five distortion levels each. We can see some examples of the images in this database in Fig.6.

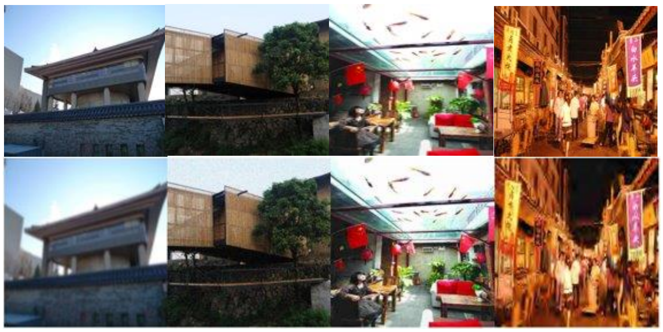

Fig. 6. Examples of the pristine images and distorted images in Waterloo Exploration Database [13]

To make it more convenient to train the models, we create a subset dataset from the Waterloo Exploration Database with 500 pristine images and their distortion images. To balance the distortion types of data, we over-sample the pristine images to the new dataset. This makes the dataset contains a total of $500 \times 5 \times 5$ (including pristine) images.

Training Details: We split the dataset by pristine images into two parts: $400 \times 5 \times 5$ (including pristine) images for training and $100 \times 5 \times 5$ (including pristine) images for validation. We generate all the SSIM [5] quality maps for the images before the training to save time and we crop the quality maps the same way we crop the images when training to get the quality maps for the patches. The size of patches used in training is 256×256. The adam optimizer is used and the initial learning rate is set as 0.001. The learning rate decay epoch is set as 100 with a rate of 0f 0.5. We try to train all the models as many epochs as possible and stop the training after they being converged or overfitting.

Evaluation Criteria: We evaluate the training result by calculating the mean absolute error (MAE) between the predicted quality maps and the SSIM quality maps since they are both

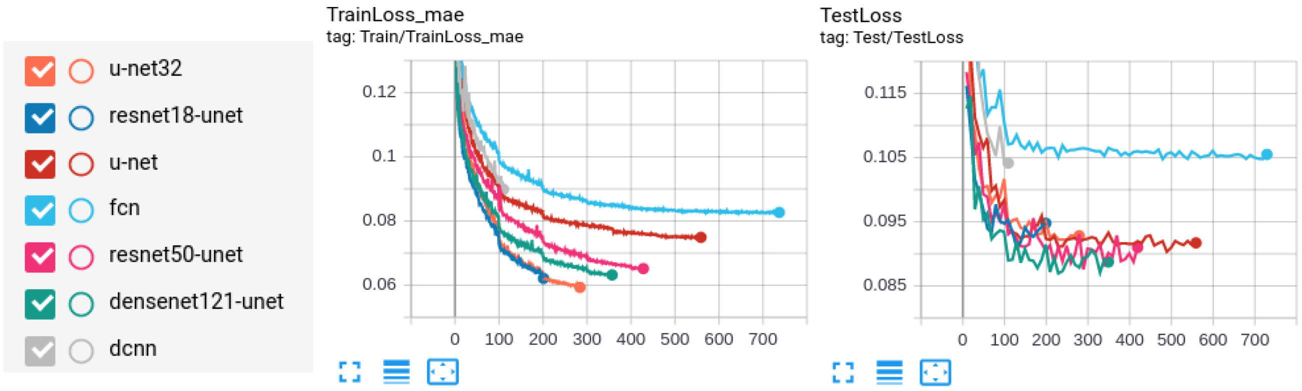

Fig. 7. Train and test curves for deep neural networks

images and MAE can show the average error of quality for every single pixel in the predicted quality maps.

B. Training Result

For the training, we trained all the models mentioned above and there are 8 models in total. There are basically three types of models. The first two models, FCN, and DCNN are both fully convolutional networks. The next five models are U-Net based networks, and finally, the last one is a GAN network. The network used in GAN is also an U-Net based network.

As the computational complexities increase with deeper networks, it becomes difficult to train them [12]. For the DCNN model, we only trained 80 epochs since this network was too big for the computational resources available, and it was also very hard and slow to train it. The training results are shown in Table I.

TABLE I
TRAINING RESULTS OF IMAGE-TO-IMAGE MODELS

Model	Type	#Param	Epoch	Train MAE	Valid MAE
FCN	FCN	5495617	479	0.0836	0.1048
DCNN	FCN	**39633751**	109	0.0878	0.1041
U-Net	U-Net	886625	899	0.0749	0.0909
U-Net32	U-Net	14123905	239	**0.0607**	0.0922
Densenet121-unet	U-Net	13607633	229	0.0671	**0.087**
Resnet18-unet	U-Net	14328209	129	0.069	0.091
Resnet50-unet	U-Net	**32521105**	289	0.069	**0.0876**
GAN(U-Net)	GAN	**44598000**	200	0.0703	0.129

C. Training Analysis

MAE Performance: As Table I shows, we can first see the performance of the models based on their MAE for training and validation.

For the three types of model, it is obvious that U-Net > FCN > GAN. Though we may expect the GAN network has the best performance since it has the largest number of parameters and a new training method different from the traditional neural network training, the GAN network does not perform well in this task and even faces a very serious problem of overfitting.

While for the U-Net based network, they all get very good results. We try different network structures for the convolution layers. We try normal U-Net with 4 convolution layers (U-Net) and 5 convolution layers (U-net32). And we also try to change the normal convolution layers into some typical networks like Densenet (Densenet121-unet) and Resnet (Resnet18-unet & Resnet50-unet). The networks will not necessarily get better performance but may get a more serious overfitting problem when having a larger number of parameters. The Densenet121-unet gets the best result among those networks without having the largest parameter number, so the performance is also related to the structure of the networks.

Training Performance: Just like what we mentioned above, the training of the deep neural networks to predict the quality maps faces a serious overfitting problem. The curves of training and validation loss are shown in Fig.7. We can see from the curves that most models' validation loss stops decreasing at about 100-150 epochs but their training loss is still decreasing and not converged even after 400 epochs.

And the training speed is also very important in this task since we are using networks with a large number of parameters. In the computer with two GTX 1080Ti GPUs, the training of FCN, U-Net, U-Net32, Densenet121-unet and Resnet18-unet will take about 60 seconds for each epoch while training of Resnet50-unet and GAN network will take about 120 seconds for each epoch. And for the most complicated network DCNN, it will take more than 900 seconds for each epoch, which makes it very hard to train the network.

Predicted Quality Maps: Besides the MAE between the SSIM quality maps and predicted quality maps, we also compare them directly by showing those maps as images. An example can be found in Fig.8. It is shown in the example that the U-Net32 and the Densenet121-unet networks have obviously better results. While the FCN network predicts very blur quality with grids because of the bottleneck in the middle of FCN network. And the GAN network gives some meaningless maps, which means they totally did not learn the information of quality maps. The U-Net32 and the Densenet121-unet networks predict very good quality maps with a lot of details on the Gaussian white noise and JPEG images. But even those two networks cannot predict very good maps on the Gaussian blur and JPEG2000 images.

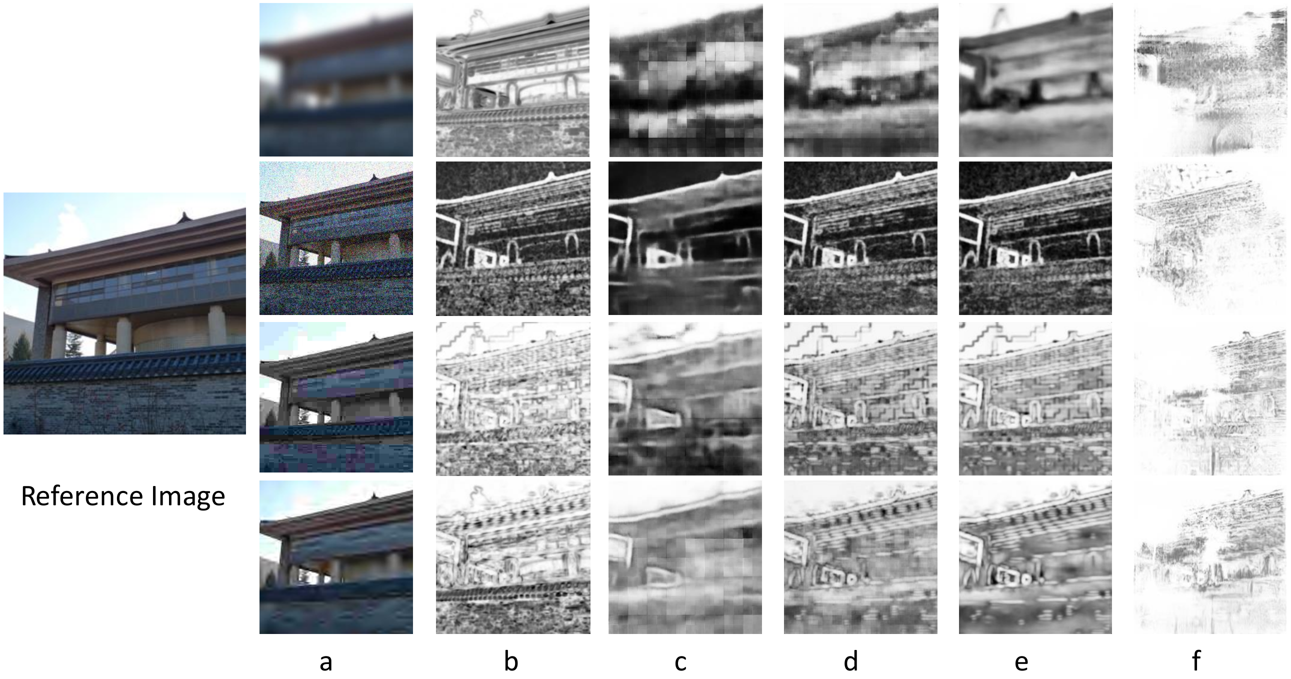

Fig. 8. This is an Example of the training result. The image on the left is the reference image. And the four rows of images are the distorted images and their quality maps for four different distortion types, which are Gaussian blur, Gaussian white noise, JPEG and JPEG2000. While the columns mean: (a) distorted image; (b) quality maps of SSIM; (c) quality maps predicted by FCN; (d) quality maps predicted by U-Net32; (e) quality maps predicted by Densenet121-unet; (f) quality maps predicted by GAN.

V. Conclusions

A. Training Conclusion

In this work, we conduct an empirical evaluation using different image-to-image translation models to learn the quality map generated by SSIM. From the training results, we get conclusions on both training and performance.

For the training itself, we find that using those famous image-to-image models to train the quality maps will face a very serious problem of overfitting so we cannot get very satisfied models from the training. And as we mentioned above, the training time is also very long, taking a lot of computing resources.

For the models' performance, we conclude that the U-Net based networks have better performance than the FCN network in this task. While the GAN network has the worst performance which does not reach our expectations. For the U-Net based networks, the Densenet121-unet and the Resnet-unet have the best performance. The difference of performance on different distortion types is also observed that networks have a good performance on images with distortion of Gaussian white noise and JPEG while they have relatively worse performance on Gaussian blur and JPEG2000.

B. Future Work

The future work can be divided in two parts as mentioned in the conclusion. First, we need to find some ways to solve the overfitting problems and reduce computing resource consumption to make the networks more ease to use. Potentially, we can also try to find smaller but more powerful networks for our image-to-image translation tasks. Secondly, we can try to use our training results to improve the current NR-IQA methods by using image-to-image networks to learn the quality maps generated by FR-IQA methods.

References

[1] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[2] Christopher hesse. Image-to-image demo. https://affinelayer.com/pixsrv/, 2017.

[3] Brannon dorsey. Full person-to-person image translation with machine learning. https://twitter.com/brannondorsey/status/806283494041223168, 2017.

[4] Mario klingemann. Generating faces from a sketch. https://twitter.com/quasimondo/status/826065030944870400, 2017.

[5] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[7] Da Pan, Ping Shi, Ming Hou, Zefeng Ying, Sizhe Fu, and Yuan Zhang. Blind predicting similar quality map for image quality assessment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6373–6382, 2018.

[8] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[9] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[10] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[11] Pavel Yakubovskiy. Segmentation models pytorch. https://github.com/qubvel/segmentation_models.pytorch, 2020.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[13] Kede Ma, Zhengfang Duanmu, Qingbo Wu, Zhou Wang, Hongwei Yong, Hongliang Li, and Lei Zhang. Waterloo exploration database: New challenges for image quality assessment models. *IEEE Transactions on Image Processing*, 26(2):1004–1016, 2016.