CSC420H1 Project Report
Image Superresolution

Huakun Shen*
University College
University of Toronto
Toronto, ON, Canada
huakun.shen@mail.utoronto.ca

Weiqing Wang*
University College
University of Toronto
Toronto, ON, Canada
weiqing.wang@utoronto.ca

Yuan Xu*
University College
University of Toronto
Toronto, ON, Canada
xuyuan.xu@mail.utoronto.ca

***Abstract*—Image super-resolution is a classical problem in the field of image computing. Computer scientists have been trying hard to provide an estimate for the information that was not present in smaller images. There have been many attempts to apply convolutional neural networks to the image super-resolution problem.**

In this project, the team attempted to construct, augment, and train these model to test out their pros and cons, as well as to apply these neural network structures to novel situations such as text super-resolution.

***Index Terms*—Image super-resolution, Convolutional Neural Network, Residual Neural Network**

I. Introduction

Image super-resolution is the process of upscaling and improving the details within an image. It is a classical problem in the field of computer science. In the past, computer scientists have invented Duckworth-Lewis methods such as sparse-coding-based method and bicubic interpolation to perform the super-resolution task but the result often has high pixel signal-to-noise ratio (PSNR) [1].

Convolutional Neural Network (CNN) is a blend of artificial neural networks comprised of neurons [2]. Each neuron serves as a pixel in one of the convolution masks that self-optimise through learning. Convolutional Neural Networks are primarily used in the field of pattern recognition within images.

In recent years, computer scientists have been attempting to apply Convolutional Neural Networks to the image super-resolution problem [1].

II. Literature Review

In past twenty years, with neural network being used more broadly in the field of image computations, convolutional neural network has been broadly used in detection, segmentation and recognition of objects and regions in images [3], and recently used to perform the image super-resolution task, which has outerperformed the classical non-Duckworth-Lewis (DL) Methods [1].

* Equal contribution.

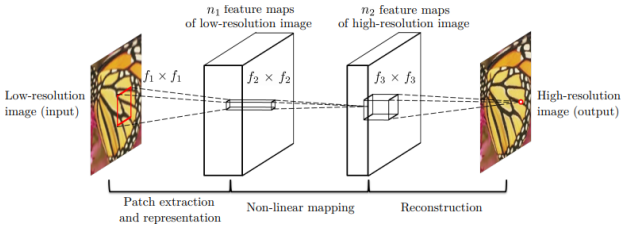


Fig. 1: Structure of SRCNN [1].

A. Super-resolution Convolutional Neural Network

SRCNN is one of the pioneers in this field [4]. The SRCNN [1], which is short for Super-Resolution Convolutional Neural Network, introduced by a team of researchers from the Chinese University of Hong Kong.

As demonstrated in Figure 1, SRCNN is a lightweight structure. It is a fully convolutional neural network that directly learns an end-to-end mapping between low- and high-resolution images. SRCNN has one input layer, two hidden layers, and one output layer. All layers are convolutional and uses Rectified Linear Unit (ReLU) on the filter responses. The first hidden layer is for Patch extraction and representation which extracts (overlapping) patches from the low resolution input represents each patch as a high-dimensional vector. The second layer is for Non-linear mapping: this operation nonlinearly maps each high-dimensional vector onto another high-dimensional vector. The third layer is for reconstruction, which aggregates the high-resolution patch-wise representations to generate the final high-resolution image.

B. Fast Super-resolution Convolutional Neural Network

The same team from the Chinese University of Hong Kong 2016 introduced a FSRCNN [5], which is short for Fast Super-resolution Convolutional Neural Network. Their proposed network is 10 times faster than SRCNN and achieved better super-resolution quality as measured by pixel signal-to-noise ratio.

Different from SRCNN, FSRCNN has two more hidden layers as shown in Figure 2, FSRCNN has a deconvolution layer at the end of the network to learn the output image directly from the low-resolution one. Moreover, after feature

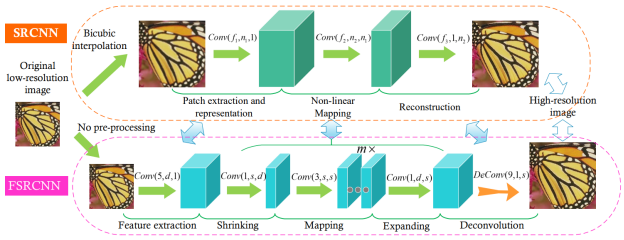


Fig. 2: Structure of FSRCNN [5]

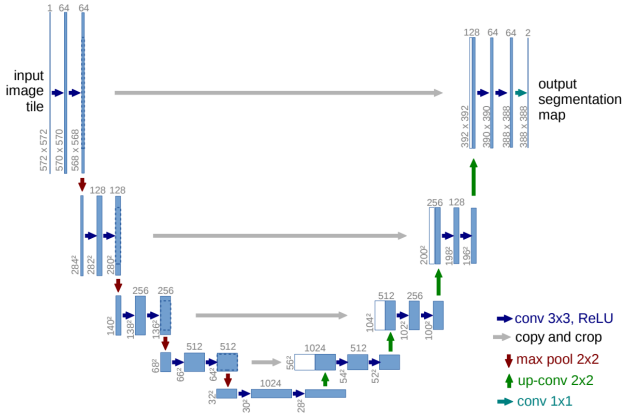


Fig. 3: Structure of U-Net [10].

extraction, it shrinks input image dimension before upscaling it [5].

C. U-Net

The "fully-convolutional neural network" has also been applied to specific class of images such as medical images. "U-Net" is a convolutional neural network designed for Biomedical image segmentation, extending the work by Ciresan et al [6] and Long et al [7]. It modified Long et al's work such that the upsampling layer have a large number of feature channels, enabling the network to propagate context information to higher resolution layers [8]. Moreover, it has residual connections between the layers, which helps easing the training process for such deep structure [9].

D. Very Deep Convolutional Network

Kim et el from National Seoul University in Republic of Korea invented a network named "Very Deep Convolutional Network" (VDSR) in 2016. It uses the structure as demonstrated in Figure 4. Its major difference from FSRCNN is that it has a residual connection before the output layer.

Kim et al recognised that end-to-end relation requires long-term memory and consequently leads to vanishing/exploding gradients problem, which they proposed solving with residual-learning [10]. Their design could use a learning rate that is 10^4 times higher than SRCNN [10]. Kim et el also recognised that the low frequency part of the low-resolution input image is similar with the low frequency part of the high-resolution output image, so the network only needs to learn the high frequency part [10].

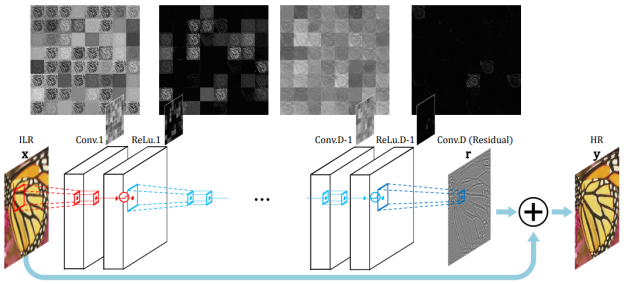


Fig. 4: Structure of VDSR [10]

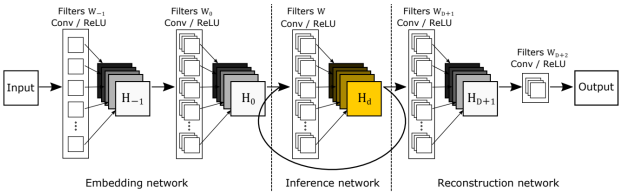


Fig. 5: Structure of DRCN [11].

E. Deeply-Recursive Convolutional Neural Network

Besides residual network, Kim et al have proposed an SR method using deeply-recursive convolutional neural network. Kim et al proposed that increasing recursion depth could improve the performance without introducing new parameters, and their proposed network has a very deep recursive layer up to 16 recursions.

One major drawback of this network is that recursive neural networks are extremely tough to train due to exploding/vanishing gradients [11]. In order to ease the training process, Kim et al proposed an advanced model which adds recursive supervision and skip-connection features to the structure shown in Figure 5. To computer science students, this means the network is hard to implement, and more importantly, even with the two proposed improvements, the model converges slowly especially when the dataset size is relatively small.

F. Deep Recursive Residual Network

In 2017, Tai et al. proposed a "Deep Recursive Residual Network" (DRRN) which outperformed DRCN with at least 70% smaller number of model parameters. Tai et el observed that increasing the number of layers of SRCNN does not yield a better performance so deeper network does not have an implication of better output for image super-resolution [4].

The structure of DRRN is shown in Figure 6. The red dashed box refers to a recursive block consisting of two residual units. In the recursive block, the corresponding convolutional layers in the residual units (with light green or light red colour) share the same weights. In all four cases, the outputs with light blue colour are supervised, and $\bigoplus$ is the element-wise addition. This solution combines residual unit from ResNet and recursive block from VDSR.

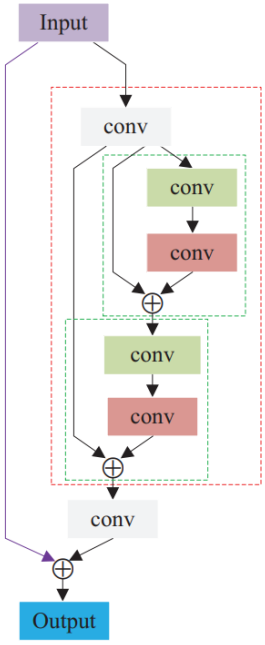


Fig. 6: Structure of DRRN [4].

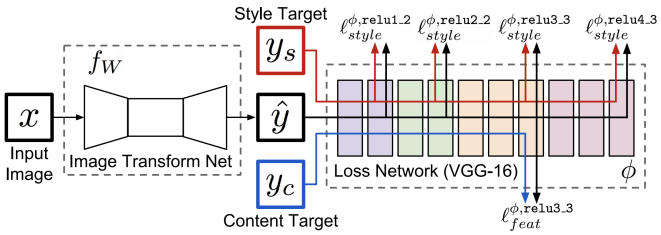


Fig. 7: System overview [13].

G. Loss Functions

Many of the neural networks above uses Mean Squared Error MSE as their loss function [1], [5]. Studies have shown that MSE does not correlate well with human's perception of image quality [12]. Johnson et el from Standford University have introduced a "Perceptual Loss" based on high-level features extracted from pretrained networks as opposed to per pixel loss [13]. See Figure 7 for how the training system with perceptual loss look like. Instead of computing per-pixel loss, this method computes the loss by passing the output and target into a pretrained VGG network, extracting the feature map from an intermediate layer and calculating the feature difference.

III. Methodology, Results, and Experiments

A. Methodology

One of the goals of this project is to perform a experiment on the existing models we have conducted literature reviews attempt to retrain it using our own dataset and perceptual loss, and test their performance on other tasks, in our case, text super-resolution.

One thing we have recognised when conducting literature review is that many network that have given us satisfying results were trained on a dataset that is vast in size. Due to the limit in time and computing resources we have, this is often not possible, we wanted to see if this model could perform well with limited data (800 images in our case), and if there are techniques to improve the model's result without training it with larger training data. For this reason, we decided to reconstruct the models we saw in our literature review and train it on our own dataset.

Another thing we have recognised was that Convolutional Neural Networks were initially designed for image classification tasks, and experiments have shown that many CNNs perform well in other image computing tasks including image super-resolution. We wanted to see if the existing neural network architectures could perform well on other tasks such as text-image restoration.

1) Dataset: We prepared two datasets to fulfill our task, **DIV2KCustom** and **TEXT**, respectively for our aforementioned two goals.

The first dataset, **DIV2KCustom**, was constructed on the basis of DIV2K dataset [14]. DIV2K dataset consists of 1000 2K resolution images with 800 for training, 100 for validation, and another 100 for testing. The original dataset have images of various sizes. Although the networks we reconstructed are image-size invariant, i.e. the networks work for all sizes of images, having various sizes of images disallows us from using batch-training. For this reason, the DIV2K were customised such that the dataset was cropped to the same size at various level (such as 50×50, 100×100, etc).

The second dataset, **TEXT**, was constructed on the basis of DDI-100 dataset [15]. We utilised Gaussian Blur to create a blurred input images, and the original high-resolution images were used as desired target for the model to restore.

2) Preprocessing Data: At the first stage of coding, Weiqing only used 10 images as training set (solely for the purpose of making sure the training framework we developed works). He surprisingly found that the validation error well exceeded the training error, and the model works well solely on the 10 images in the training set. We realised that overfitting could be an issue in the later training process.

In STA413H1, we were introduced to generlization problem, where we were introduced to a bag of tricks for avoiding data-overfitting problem. One of them is data-augmentation. Therefore, we augmented every image twice – there were seven methods of manipulating an image without changing its content, e.g. mirroring it horizontally, mirroring it vertically, rotate the image, etc, and added them to the training set. This task was performed using Python Image Library (PIL) and torchvision's transforms module.

3) Constructing the Neural Network: The next step we took was to construct the neural network. PyTorch was the framework we chose to use as it is which we are familiar with and found easy to programme. The networks were constructed according to the introductions in the research papers. A Recti-

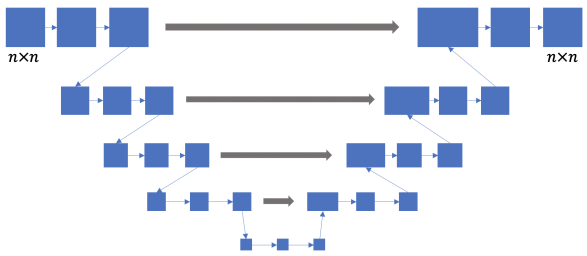


Fig. 8: Structure of UNetSR.

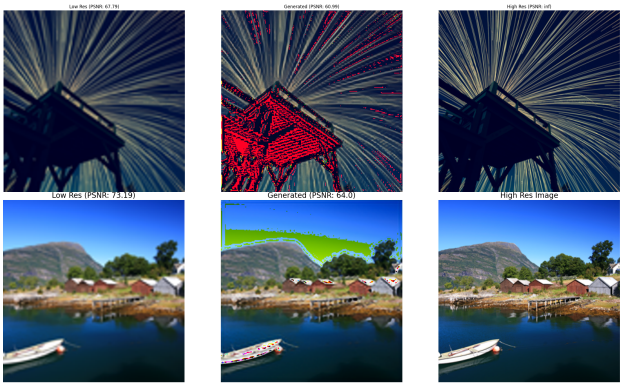


Fig. 9: From left to right: The input image, the Output of trained model with perception loss function, the ground truth. We could see that this model automatically added "neon light effect" and "green sky effect" to the output, which looks beautiful, but isn't the result we want.

fied Linear Unit (ReLU) was used as the activation function if the activation function was not specified in the research paper.

4) Augmenting the Neural Networks : We implemented several neural network models based on some research papers, and made modifications to them when necessary.

Take U-Net as an example, U-Net was initially designed for Biomedical Image Segmentation. A U-Net consists of a contracting path and an expansive path. The Contracting path keep reducing the spatial size and increasing the number of features (number of channels), while the expansive path combines extracted features and spatial information by concatenating layers in the contracting path to the expansive path [8]. A very deep network may not work well for a Super Resolution problem as some details may be lost across many layers. As our output is a high resolution image with lots of details, it is intuitive that the skip connection used in a U-Net may be able to bring some details from earlier layers. We were inspired by the idea of U-Net and decided to develop a model on the basis of that. We modified the U-Net to such that it would have the same input size and output size, i.e. input a $(n \times n \times 3)$ image output a $(n \times n \times 3)$ image. See Figure 8 for the structure of the network. We call it UNetSR.

During experiments, we found that UNetSR occasionally performs poorly. We suspect that it may be due to over-fitting, we then made a variant of UNetSR, UNetD4, namely UNetSR with a depth of 4. Depth of UNetSR is reduced to 4 intending to reduce the architecture complexity. It turns out that UNetD4 sometimes indeed outperforms UNetSR.

5) Developed a Framework for Model Training : The limit of computing power led us to come up with solutions to move this project forward. With limited computing power we have, both on our personal laptop and `teach.cs` machines, the completion of this project is only possible with external computing resources. Unfortunately, the project budget was CAD$0.00 which disallows us from using commercial computing engines.

Google Co-laboratory (CoLab) provides free GPUs for use up to 12 hours a day. There is a demand for us to start training the neural net from where we last ended. Moreover, we want to simplify the process of training setup to make better use of the precious 12-hour runtime.

For this purpose, we have decided to develop a reusable framework such that

- We could (re)train a neural network by specifying the model object and configuration and clicking the run button.
- Save the training results every few iterations automatically, such that retraining a model from 0 is never needed.

The early version developed by Weiqing only achieved the first function. Huakun improved Weiqing's implementation of first function and implemented the second function. This framework saved a lot of time for us in the training process.

B. Results

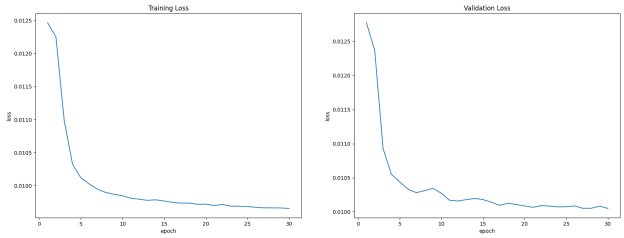
1) A General Comment : : We have identified that all neural networks we trained failed to reconstruct the parts of images that are extremely white or dark, as demonstrated in Figure 9.

We believe that this is due to the fact that extremely dark/bright pixels have too low frequency, such that the filters would fail to learn the details missing in these areas.

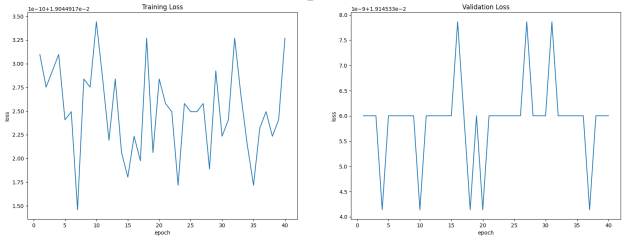
2) Choice of Loss Function: One thing we observed was that with a relatively small dataset (of size 800), training with Mean Square Error as loss function could be hard to converge. In general, training using MSE error takes a long time to converge, and invariably gives us a poorer result.

Take the loss data in VDSR model's training process as an example, Figure 10 shows the loss vs. iteration in the training process. It would be seen that the perceptual loss in Figure 10a decreases steadily over time, while the MSE loss as shown in Figure 10b did not converge even we have already given a small learning rate (5×10^{-3}).

3) Pure Convolution Neural Networks – SRCNN, FSRCNN: On a training set of size 800, we observed that SRCNN achieved a much more satisfying result in comparison to FSRCNN. As domonstrated in Figure 11, the output shown in Figure 11a have a higher PSRN compared to Figure 11b. We see signal noises on the edges of lemons in output of FSRCNN. This is a possible demonstration of the general shortcoming of all our models as mentioned in Section III-B1.



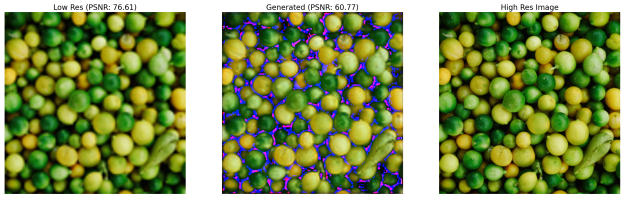
(a) Perceptual Loss



(b) Mean Square Error Loss

Fig. 10: Training and Validation Loss of VDSR model.

(a) SRCNN



(b) FSRCNN

Fig. 11: From left to right: image input, output of purely convolutional networks, ground truth.

Experiments from researchers in CUHK showed that FSRCNN have faster training process [5], this was not observed in our experiments. We have found that FSRCNN took approximately 40 iterations to converge, while SRCNN took only 13. The training time for the two models were approximately the same. We suspect that this is due to the fact that FSRCNN is deeper and have more parameters in comparison to SRCNN.

4) Models with Residual Connections – U-Net and VDSR: As we mentioned in the literature review, residual connections between the layers ease the training process for such deep structure [9]. U-Net and VDSR are residual networks. This statement is confirmed by our result as UNetSR we designed took less than 10 iterations for the training loss to converge and the validation loss decreases steadily over time. The losses for UNetD4 and VDSR models are similar to UNetSR's.

As presented in Figure 12, the three networks provide similar outputs that at least we could not differentiate with

(a) UNetSR

(b) UNetD4

(c) VDSR

Fig. 12: From left to right: image input, output of convolutional networks with residuals, ground truth.

eyes. The two outputs share a similar PSNR. While all three outputs still look blurred compared to the original one, we could obviously find that the generated images contains much more details than the low-resolution input.

5) Models with Recursion Block – DRRN: DRRN was the only network with recursion block we tested. One natural downside of networks with recursion blocks is that it is hard to train due to exploding/vanishing gradients [11]. As restricted by the computing power, our implemented DRRN has a recursion depth of 2 only, and with perceptual loss, both training and validation loss converges quickly, as demonstrated in Figure 13.

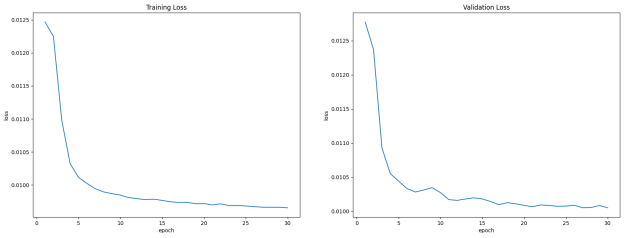


Fig. 13: Training and validation losses of DRRN model.

However, since our implemented version does not have a large recursion depth, in other words, does not have a deep-enough network, we did not see that recursion block improved the result on the basis of models with residual connections. Figure 17 is a representative example. The magnitude of the model's improvement is moderate. We suspect that in order to get the result as demonstrated by the authors from

Nanjing University of Science and Technology and University of Michigan, we had to increase the recursion depth.

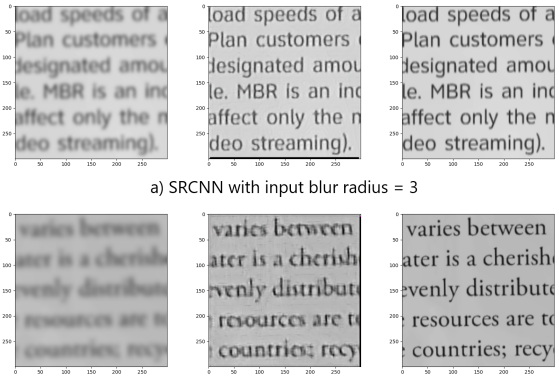
Fig. 14: From left to right: image input, output of DRRN, ground truth.

C. Experiments

In order to test if the neural networks we re-constructed could solve a broader class of problems, we applied this model to the text superresolution problem.

1) SRCNN: We used PIL gaussian kernel to blur the target images to get our input datasets. When the kernel radius is 3, the input images are still readable to human eyes. We can see in Fig. 15 that SRCNN performs well on this dataset. On the other hand, we do not get satisfactory results when the kernel radius is 5.

SRCNN is a convolution neural network with only one no-linear mapping layer. We believe that is not sufficient for the model to recognise and memorise the characters, the only capability of this model is sharpening edges. This is the reason why SRCNN performs well on mildly blurred inputs, but cannot reconstruct the characters in severely blurred images.



a) SRCNN with input blur radius = 3

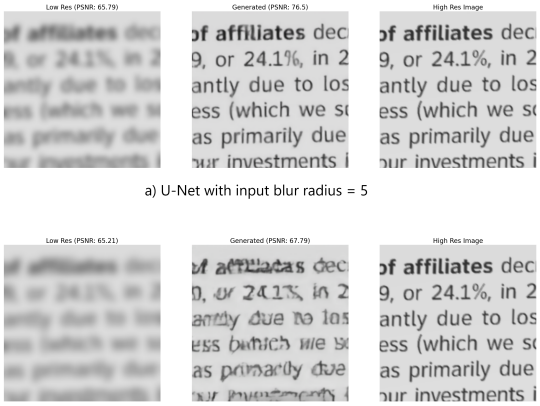
b) SRCNN with input blur radius = 5

Fig. 15: From left to right: image input, output of SRCNN, ground truth.

2) U-Net: Because U-Net has a deep structure comparing to SRCNN, its performance on inputs with blur kernel radius 5 is astonishing. We can see all characters are well reconstructed expect the percent sign, "%". This conforms our hypothesis that deep neural networks try to memorise each character, and the percent sign cannot be reconstructed because the frequency of its appearance is low.

Yet if we take a step further and change the blur kernel radius to 7, the result is terrible. But this is different with where the SRCNN model failed to reconstruct input images with radius 5. In that case, the model was only trying to sharp the edges, but the problem here is that the model cannot detect the characters correctly (or separately).

We think this is mainly due to falsely generating the input images instead of the capability of the U-Net model. The most accurate way to blur an image is to defocus the camera while capturing the images. While using Gaussian blur with large kernel radius, too much information overlaps at the same position.



a) U-Net with input blur radius = 5

b) U-Net with input blur radius = 7

Fig. 16: From left to right: image input, output of U-Net, ground truth.

3) VDSR and DRRN: Now we have learned the general performance of Super-Resolution models on text images with different blurring scales, we want to test if the more advanced models are faster and more stable on an easy task comparing to the basic model SRCNN.

We train models SRCNN, VDSR, and DRRN on images blurred with kernel radius 3. All of them gives correct and clean results. In Fig. 17, we can see that both the VDSR and DRRN models converge using fewer epochs and at lower validation loss.

Note that the initial training rate of SRCNN is 0.002, while the initial training rate for the other two models are 0.005. This is because a higher learning rate would make the loss of SRCNN oscillate. So we can also conclude that VDSR and DRRN are stabler comparing to the basic model, SRCNN.

IV. Conclusion

After testing the re-constructed (and augmented) models from research papers, we have concluded that Convolutional neural networks, including those with redisual connections and recursion blocks, are valid solutions to image super-resolution problem.

We have found that a large dataset is required for the nerual network models to train well. Techniques such as weight

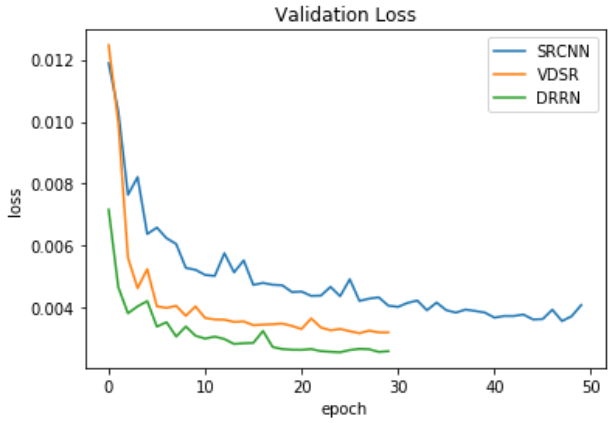


Fig. 17: The validation loss of models SRCNN, VDSR, DRRN

decay, appropriate learning rate, etc shall be chosen more prudently with small training dataset.

We recognised that given a small dataset, residual connections provide a better result in comparison to purely convolutional neural network and models with recursion blocks.

Based on our experiment, we have found that perceptual loss leads to quicker and more stable convergence for all models we tested.

Moreover, we have observed that these models can be utilised for text-image recognition. However, the shortcoming mentioned in Section III-B1 could become a problem.

It was mentioned by the researchers from Nanjing and Michigan that deeper network does not have an implication of better output for image super-resolution [4]. This behaviour was observed in our experiments.

V. Authors' Contributions

Yuan Xu mainly focused on the experiment part of our project. Yuan researched and compared dataset that could potentially be used for training text super-resolution models. He strategically selected DDI-100 dataset [15] for the training purpose, performed data pre-processing which includes cutting the images to appropriate sizes, blur the images using Gaussian Kernel, splitting them into training and test sets, etc. Yuan also trained the models we re-constructed on the data he pre-processed, and performed analysis on the output. Yuan's high standards and exemplary skills were essential to Section III-C of this report.

Huakun Shen's contribution includes dataset generation, documentation, video demo and visualizer web app, development of the model training framework (as described in Section III-A5), as well as implementation of U-Net, SRCNN and FSRCNN models with help of resources online. Besides reconstructing neural networks, Huakun augmented the U-Net structure (as described in Section III-A4) making it capable of performing image super-resolution tasks. Huakun performed experiments to find the optimal hyper-parameters for model training, and tested out a few pretrained neural networks such as ResNet. Huakun participated in the preparation work of presentation video. Huakun also processed administrative works such as repository setup, team meeting arrangements, etc.

Weiqing Wang's contribution includes conducting literature reviews, early stage development of the model training framework (as described in Section III-A5), implementation of VDSR and DRRN models, and performing analysis on the models' performance. Besides the technical work, Weiqing took care of all project deliverables such as drafting and polishing the video presentation slides, writing the video presentation script, creating the presentation video, and writing this report.

References

[1] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *CoRR*, vol. abs/1501.00092, 2015. [Online]. Available: http://arxiv.org/abs/1501.00092

[2] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *CoRR*, vol. abs/1511.08458, 2015. [Online]. Available: http://arxiv.org/abs/1511.08458

[3] Y. LeCun, Y. Bengio, and H. Geoffrey, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015. [Online]. Available: https://doi.org/10.1038/nature14539

[4] Y. Tai, J. Yang, and X. Liu, "Image super-resolution via deep recursive residual network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[5] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," *CoRR*, vol. abs/1608.00367, 2016. [Online]. Available: http://arxiv.org/abs/1608.00367

[6] D. Ciresan, A. Giusti, L. Gambardella, and J. Schmidhuber, "Deep neural networks segment neuronal membranes in electron microscopy images," in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012, pp. 2843–2851. [Online]. Available: https://proceedings.neurips.cc/paper/2012/file/459a4ddcb586f24efd9395aa7662bc7c-Paper.pdf

[7] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *CoRR*, vol. abs/1411.4038, 2014. [Online]. Available: http://arxiv.org/abs/1411.4038

[8] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *CoRR*, vol. abs/1505.04597, 2015. [Online]. Available: http://arxiv.org/abs/1505.04597

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[10] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," *CoRR*, vol. abs/1511.04587, 2015. [Online]. Available: http://arxiv.org/abs/1511.04587

[11] ——, "Deeply-recursive convolutional network for image super-resolution," *CoRR*, vol. abs/1511.04491, 2015. [Online]. Available: http://arxiv.org/abs/1511.04491

[12] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for neural networks for image processing," *CoRR*, vol. abs/1511.08861, 2015. [Online]. Available: http://arxiv.org/abs/1511.08861

[13] J. Johnson, A. Alahi, and F. Li, "Perceptual losses for real-time style transfer and super-resolution," *CoRR*, vol. abs/1603.08155, 2016. [Online]. Available: http://arxiv.org/abs/1603.08155

[14] E. Agustsson and R. Timofte, "Ntire 2017 challenge on single image super-resolution: Dataset and study," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.

[15] I. Zharikov, F. Nikitin, I. Vasiliev, and V. Dokholyan, "DDI-100: dataset for text detection and recognition," *CoRR*, vol. abs/1912.11658, 2019. [Online]. Available: http://arxiv.org/abs/1912.11658