Computer Vision Project

Burin Naowarat, Chayanont Eamwiwat and Tharn Phongsavaree

December 20, 2018

Abstract

In this experiment we studied recovery text method using deep learning. We came up with the 2 different models which were train on our generated dataset and evaluated performance of models on MSE, MAE and PSNR metric against SRCNN as a baseline model. The model3 is the best model which has MSE, MAE and PSNR equal to 100.618, 3.096, 29.728. There is also a small problem on gray-scale image about background that need to be improved. The model tends to use the averaged color on pixel and represent it as a background color.

1 Introduction

Nowadays, technology is growing day by day and we can't deny that it involves with every parts of our life especially data technology since we are living in Big Data era. Image and text are the media which is commonly used and can be easily comprehended and utilized but there are always have problem with the resolution of image which make the image hard to utilize such as image in Fig 1.

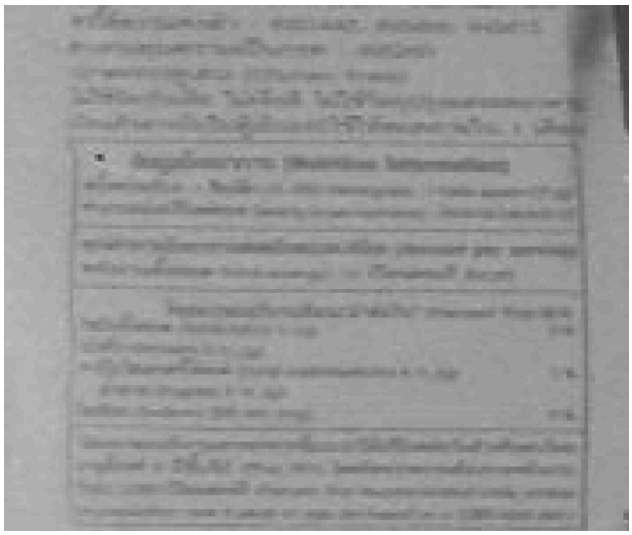


Figure 1: Example of low resolution image

Low resolution text in image is a common problem which have many causes such as image is cropped or the camera is not focusing. These images may lead to many problems when we bring it to use in other tasks such as Optical Character Recognition(OCR) that may cause a bad result when working with text in low resolution image.

Hence, we propose the denoising image method with deep learning approach combining with computer vision techniques to improve low resolution image to have a better resolution.

2 Dataset

Figure 2: Sample images in the dataset

We used our generated images. Images contains a single line of text in various font family, font size, words, background colors, and position.

To explain images generating steps. First, canvases of each data were created. Their size were fixed at 300px-width and 50px-height. All canvases were the same size in order to make it easy to be used in latter implementing step. Background colors of these canvases were randomly generated they can be any combination of 8-bit RGB color.

Figure 3: Canvas size of each dataset

Next, three to five Words were randomly selected from A word bank contains 3000 English words, which are 3000 most common words in English according to this website [1]. Fonts from fonts family consisting of Arial, Calibri, Tahoma, LiberationMono LiberationSan, and LiberationSerif were randomly applied to these selected words. Also, fonts size and position to place in canvas were randomly gave to the selected words.

After saving the generated images as original image from the previous step, we blurred them using three types of blurring method: downsampling, median blur, and Gaussian blur. In downsampling method, we resized the image to 40% - 50% of its original image size (this 10% range was selected randomly for each data). For median blur, the kernel size were

[1] https://www.ef.com/wwen/english-resources/english-vocabulary/top-3000-words/

randomly selected from 3 to 5 for each data. And for Gaussian blur, the kernel size and sigma value were randomly selected from 3 to 7 for each data.

We generated 30,000 images for training, 20,000 for developing, and 20,000 for testing. Total datasize is 2.1GB.

3 Model

3.1 Model1 (SRCNN)

We use SRCNN [4] as our baseline model since this model was easily implenmented, use mininal time to train and also not hard to comprehend. This model use the 3 convolutional layers as shown in Fig. 4 to map features from input to output as tranditional convolutional neural network(CNN) does.

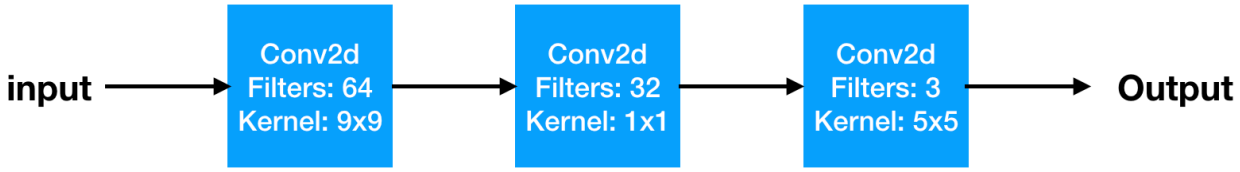


Figure 4: Architecture of SRCNN

3.2 Model2

Model2 was improved from model1. The size of kernels were reduced and layers of cnn were added more. The model had the same concept. It did not reduce any input's dimensions. Images were padded by zero. All convolution filters' stride was set to 1 for maintaining the size of images. ReLu was an activation function. The architecture of the model is shown in Fig. 5.

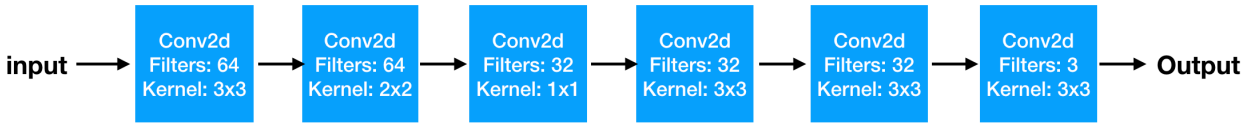


Figure 5: Architecture of model2

3.3 Model3

In this model, we had inspired that the layer on the top of model should capture the features from bottom layer. So, we decided to pass the lower-space features from bottom layer to the top layer as shown in Fig. 6. Residual highways were added in the model to solve the common problem such as gradient vanishing since we add pass the feather to top layer.

Hence, this model can capture and see the whole features in the layers all at once not just pass features like layer to layer just as traditional CNN does.

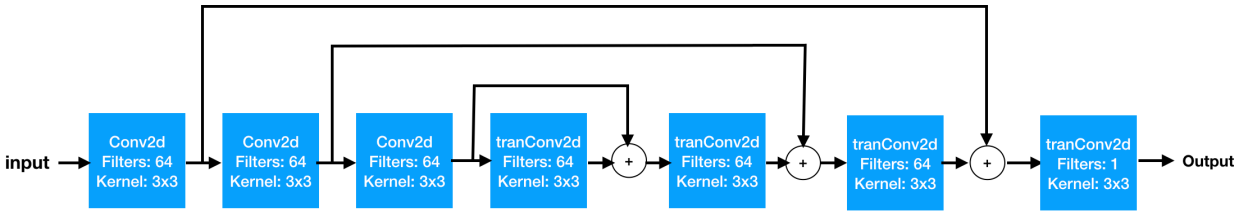


Figure 6: Architecture of model3

4 Results

In the experiment, we downgraded the orginal images by using three types of blurring algorithms, which are median blur, gaussian blur and downsampling The models were evaluated performance from test dataset(20k) in these following metrics:

- Average of mean square error of each pixel (MSE)
- Average of mean absolute error of each pixel (MAE)
- Average of peak signal-noise-ratio of each pixel (PSNR)

We did an experiment on RGB and gray scale images. Both of experiments used nearly the same model architecture. Difference is that the input and output layer dimensions were adjusted for suiting input images. During training, all types of blurring images were used as inputs. During evaluating, we both evaluate separate type and mixed type.

In our experiment, doing nothing to images surprisingly result in strong scores. This is because our backgrounds are clean and text occupies only a small portion of an image area. So, averaging metrics can be very high for doing nothing.

Besides, there are problems about the similar character such as i and l character that model seem to quite confuse to recover these character. The background of grey-scale image tends to use the averaged color on pixel and represent it as a background color.

4.1 Gray scale

Table 1-4 are shown performance of each model in gray scale mode. The best model is model3. Model's performance q It has the highest mse, mae, and psnr in all of blurring types.

4.2 RGB

In RGB colorspace, the best model is still model3. The scores are shown in Table 5-8. Our two models surpasses SCRNN, which is the baseline. The notice is that, there is only slightly

Model	Down Sampling		
	MSE	MAE	PSNR
Do nothing	838.8225	9.564	19.171
SRCNN	364.103	7.840	22.769
model2	293.607	5.721	24.618
model3	115.388	3.512	28.176

Table 1: Downsampling blur in gray scale

Model	Median		
	MSE	MAE	PSNR
Do nothing	1630.040	17.731	16.558
SRCNN	835.383	9.328	20.911
model2	464.693	6.022	23.531
model3	113.581	2.451	30.448

Table 2: Median blur in gray scale

Model	Gaussian		
	MSE	MAE	PSNR
Do nothing	1521.938	8.999	18.0476
SRCNN	552.408	9.688	21.6715
model2	293.607	6.415	24.569
model3	72.886	3.324	30.559

Table 3: Gaussian blur in gray scale

Model	All		
	MSE	MAE	PSNR
Do nothing	1330.621	12.0985	17.926
SCRNN	583.965	8.952	21.784
model2	334.135	6.053	24.239
model3	100.618	3.096	29.728

Table 4: All blur types in gray scale

difference between gray scale and RGB. Model with highest score are model3, model2, and SRCNN respectively. Except median blur images, SRCNN gave worse mae than do nothing.

Model	Down Sampling		
	MSE	MAE	PSNR
Do nothing	840.600	9.55	19.167
SCRNN	382.5456	8.082	22.517
model2	259.949	6.091	24.392
model3	113.609	3.391	28.260

Table 5: Downsampling blur in RGB

Model	Median		
	MSE	MAE	PSNR
Do nothing	1521.370	8.986	18.039
SCRNN	1006.996	10.036	20.113
model2	544.254	6.695	22.931
model3	113.693	2.338	30.223

Table 6: Median blur in RGB

Model	Gaussian		
	MSE	MAE	PSNR
Do nothing	1638.306	17.776	16.533
SCRNN	700.795	10.891	20.838
model2	368.536	7.018	23.963
model3	112.865	3.993	28.024

Table 7: Gaussian blur in RGB

Model	All		
	MSE	MAE	PSNR
Do nothing	1333.425	12.104	17.913
SCRNN	696.778	8.082	22.517
model2	390.913	6.601	23.762
model3	113.389	3.241	28.836

Table 8: All blur types in RGB

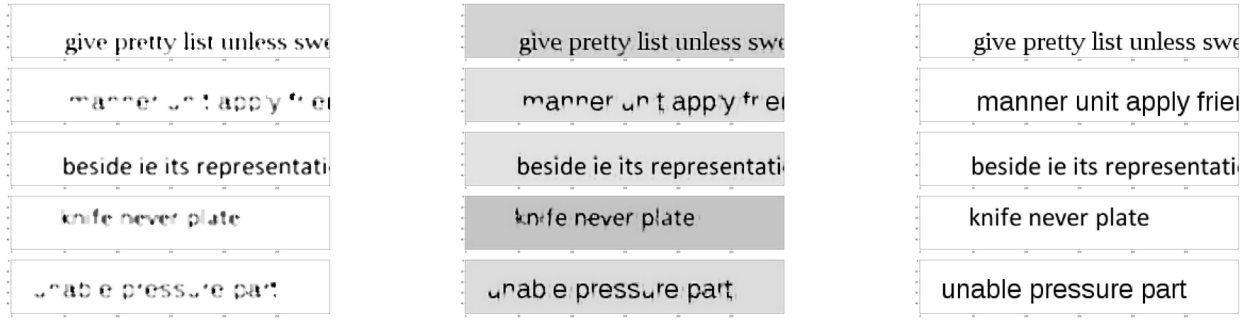
References

[1] Tian, Jing. *Kai-Kuang Ma. A survey on super-resolution imaging. Signal.* Image and Video Processing 5.3 (2011): 329-342.

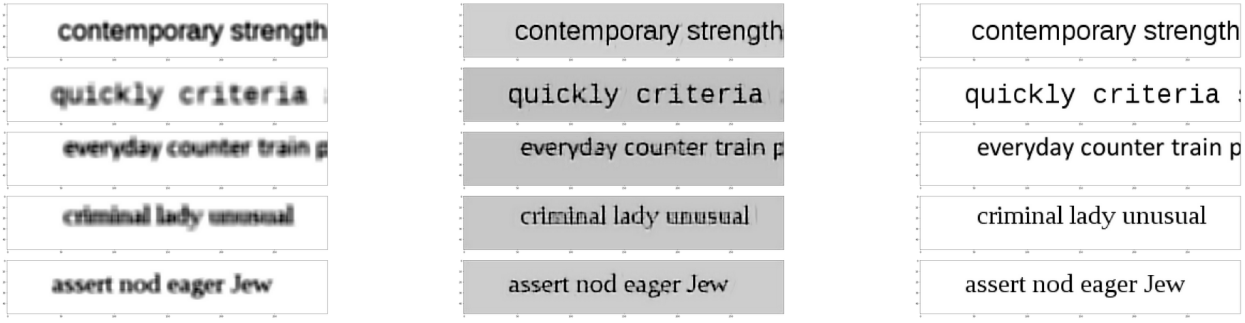
[2] Dong, Chao, et al. "Learning a deep convolutional network for image super-resolution." European conference on computer vision. Springer, Cham, 2014.

[3] Christian Ledig, Lucas Theis, Ferenc Huszr, Jose Caballero,Andrew Cunningham, Alejandro Acosta, Andrew Aitken,Alykhan Tejani, Johannes Totz, Zehan Wang, Wenzhe Shi *Photo-Realistic Single Image Super-Resolution Using aGenerative Adversarial Network*

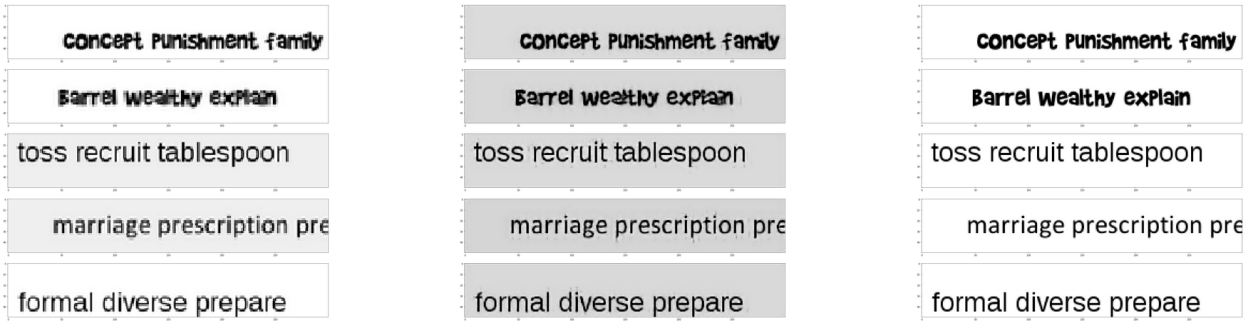
[4] Chao Dong, Chen Change Loy,and Kaiming He, Xiaoou Tan, *Image Super-Resolution Using Deep Convolutional Networks*, in Proceedings of European Conference on Computer Vision (ECCV), 2014



(a) Median blur input images

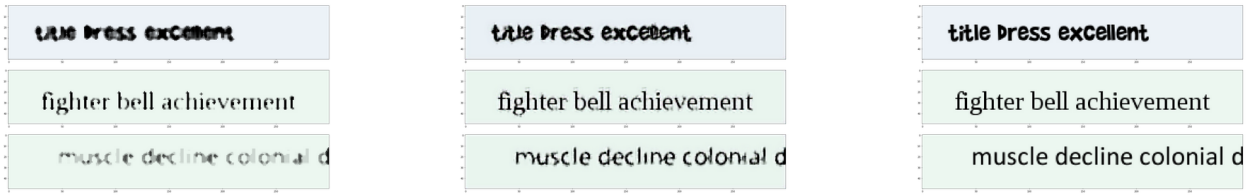


(b) Gaussian blur input images

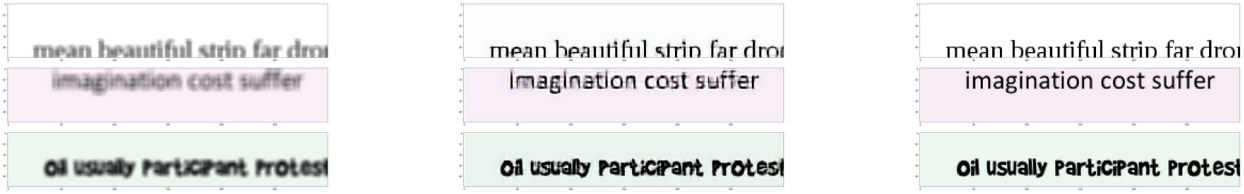


(c) Downsampling blur input images

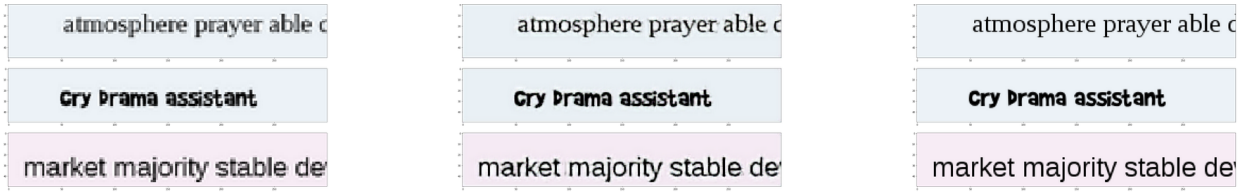
Figure 7: Gray scale super-resolution results from model2. The left row has input images. The right row contains label images. The middle row represents outputs from the model.



(a) Median blur input images

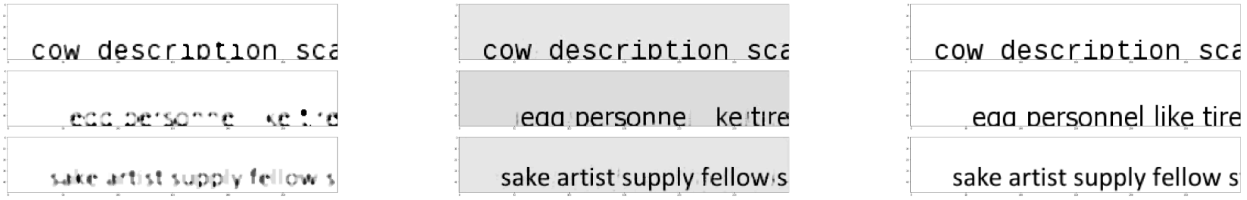


(b) Gaussian blur input images

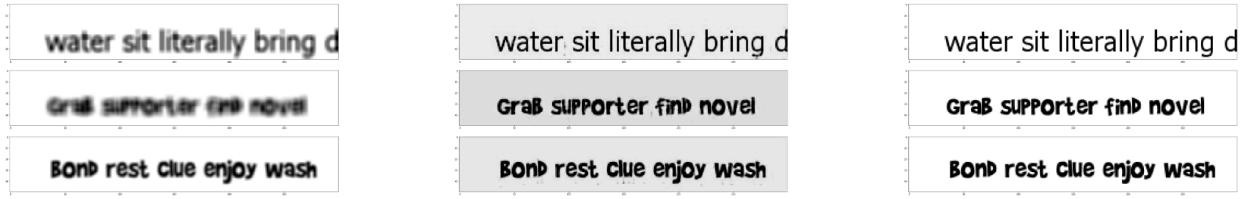


(c) Downsampling blur input images

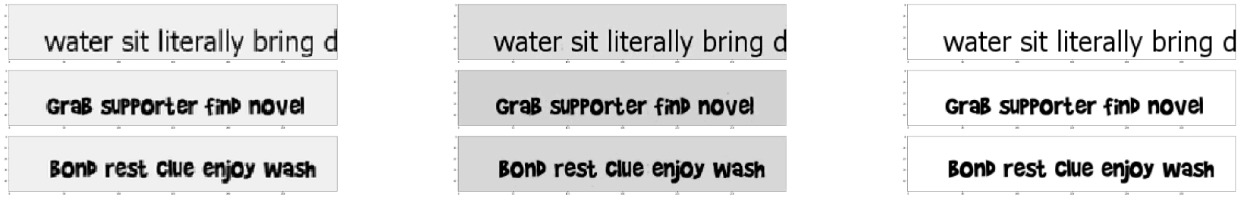
Figure 8: RGB super-resolution results from the model2. The left row has input images. The right row contains label images. The middle row represents outputs from the model.



(a) Median blur input images

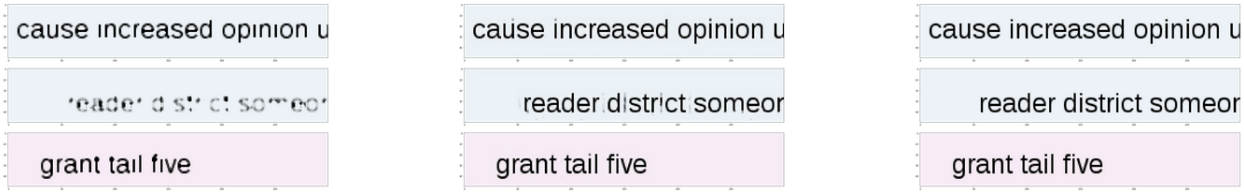


(b) Gaussian blur input images

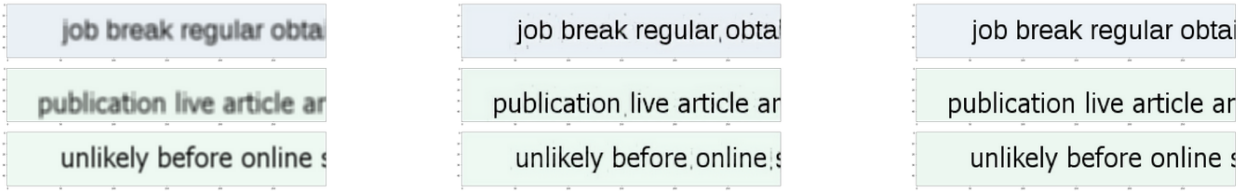


(c) Downsampling blur input images

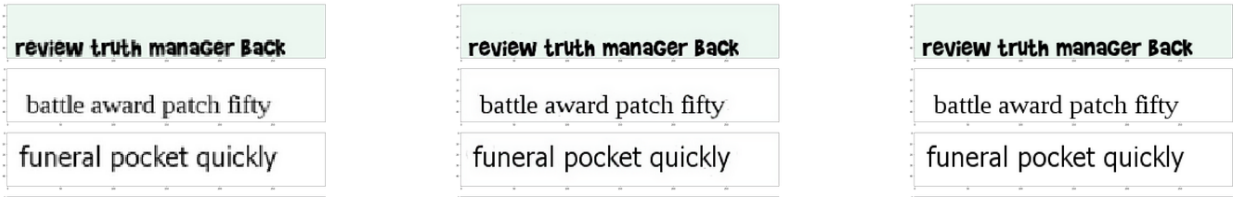
Figure 9: RGB super-resolution results from the model3. The left row has input images. The right row contains label images. The middle row represents outputs from the model.



(a) Median blur input images



(b) Gaussian blur input images



(c) Downsampling blur input images

Figure 10: RGB super-resolution results from the model3. The left row has input images. The right row contains label images. The middle row represents outputs from the model.