Low Resolution Image Classification

Taoan Lu Tianjun Li Tiancheng Zhang Yanfu Guo Yifan Dong
University of Michigan, Ann Arbor
{taoanlu, ltianjun, zhangtc, yanfuguo, spikedyf}@umich.edu

1. Introduction

The progress of the image classification is to categorize the images given into different categories based on their semantic contents in the picture. The features in the images, i.e. the image quality and resolutions, are important factors in image classification. The accuracy of image classification can be performed pretty well if the images are well depicted in a high resolution(HR) image. However, many features of the image can be missing when low resolution(LR) images compared to the high resolution images that are applied to the classifier.

Therefore, such condition raises the problem on how to improve the performance when the low resolution image are applied to the image classifier. Our solution is to combine the super resolution techniques to enrich the image details of the low resolution images. Our main principle is to verify that a good Super Resolution tool can supplement the missing details in the low resolution images so that the image classification can achieve better performance.

1.1. Background Information

Super Resolution: Super Resolution(SR) is a process to obtain high-resolution from one or more low-resolution observations [7], the SR techniques mainly focus on giving more details than the given grids by increasing the pixels in per unit area. Different from the interpolation, which uses the unblurring, sharpening methods to restore the image given [7], the super resolution method, a model of the HR scene together with the imaging parameters are given to improve the quality of the output and increase the the size(the pixels per unit area) [7]. During the process, the iterative back-projection, projecting on the convex sets and maximum a posterior image prediction is the way what the common SR algorithm does [8]

Image Classification: Image Classification has wide range of applications. Deep Convolutional Neutral Network brings great improvement in the image classification [5]. To get a great neural network training model, the network needs the degradation process that get the accuracy saturated. When considering the model to learn, it should be important to avoid the some over-fitting problem. Therefore, an easy to optimize and the enjoyable accuracy model is expected for the designed model [5].

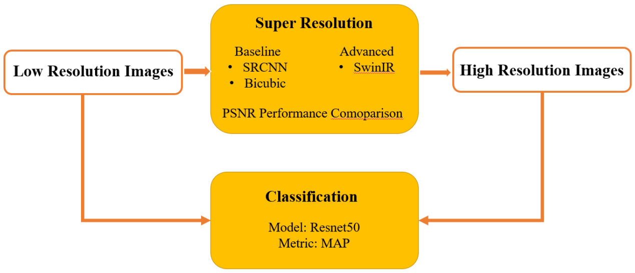


Figure 1. High Level Architecture

2. Approach

Our project firstly uses several super resolution models to restore the resolution of some low resolution images, and then compare their peak signal-to-noise ratio (PSNR) performance on some test dataset. After that, we feed the resulting super resolution images into downstream classification models to check whether super resolution will increase the performance for classification. The high-level workflow of our project is shown in Figure 1.

2.1. Dataset

Our project uses DIV2K dataset for training the super resolution models and performs super resolution tests on the Set5 dataset. DIV2K dataset is a dataset that has 800 images for training, 100 images for validation, 100 images for testing [1]. Set5 dataset is widely used for testing performance of Image Super-Resolution model and only contains five images for baby, bird, butterfly, head, and woman [2]. After training process, we let these baseline and advanced SR models perform SR on the PASCAL VOC2007 dataset with 20 object classes, and then feed the resulting super solution images into our downstream classification models [4]. In the VOC2007 dataset, there are 5000 images for trainval set and 4952 images for the test set. Since the VOC2007 dataset only has high resolution images, we follow other paper's method to first resize the images to smaller scales [9], and then use these resized images as our lower resolution image dataset.

2.2. Super Resolution

2.2.1 Model

Our models in the super resolution part are two baselines: 1) bicubic Interpolation, 2) Super-resolution convolution neural network (SRCNN), and one advanced model: 3) SwinIR.

Bicubic Interpolation: Bicubic interpolation is a method for interpolating data points as an extension of cubic interpolation and is often widely used in image processing. The algorithm we use in this project is called bicubic convolution algorithm where the kernel used in the project us designed as: By applying the kernel to the image, a prediction of the point $p(x, y)$ can be calculated as with:

$$W(x) = \begin{cases} (a+2)|x|^3 - (a+3)|x|^2 + 1 & \text{for } |x| \leq 1, \\ a|x|^3 - 5a|x|^2 + 8a|x| - 4a & \text{for } 1 < |x| < 2, \\ 0 & \text{otherwise}, \end{cases}$$

$$b_{-1} = p(t_x, f_{(-1,-1)}, f_{(0,-1)}, f_{(1,-1)}, f_{(2,-1)}),$$

$$b_0 = p(t_x, f_{(-1,0)}, f_{(0,0)}, f_{(1,0)}, f_{(2,0)}),$$

$$b_1 = p(t_x, f_{(-1,1)}, f_{(0,1)}, f_{(1,1)}, f_{(2,1)}),$$

$$b_2 = p(t_x, f_{(-1,2)}, f_{(0,2)}, f_{(1,2)}, f_{(2,2)}),$$

$$p(x, y) = p(t_y, b_{-1}, b_0, b_1, b_2).$$

SRCNN: Super-resolution convolution neural network (SRCNN) is a deep convolution neural network that learns end-to-end mapping of low resolution to high resolution images [3]. SRCCN's processes consists of patch extraction and representation, non-linear mapping, reconstruction. For the patch extraction part, the first layer $F_1(Y) = max(0, W_1 * Y + B_1)$, where W_1 and B_1 are respectively the filters and biases, and $*$ represents the convolution operation. For the non-linear mapping part, the second layer is $F_2(Y) = max(0, W_2 * F_1(Y) + B_2)$, where W_2 and B_2 are respectively the filters and biases. For the reconstruction part, the third layer is $F(Y) = W_3 * F_2(Y) + B_3$, where W still stands for filters and B stands for biases. Its architecture is visualized in figure 2.

SwinIR: SwinIR is a model based on Swin Transformer and has three parts, which are respectively shallow feature extraction, deep feature extraction and high-quality image reconstruction modules [6]. For shallow feature extraction, a 3 × 3 convolutional layer is used. For deep feature extraction, several connected residual Swin Transformer blocks (RSTB) and a 3 × 3 convolutional layer are used. As for the reconstruction process, it is obtained by aggregating the shallow and deep features. The loss function for image super resolution is $\mathcal{L} = \|I_{RHQ} - I_{HQ}\|_1$, where I_{RHQ} is obtained by taking the low quality image as the input of SwinIR, I_{HQ} is the corresponding ground-truth high quality image. Its architecture is visualized in figure3.

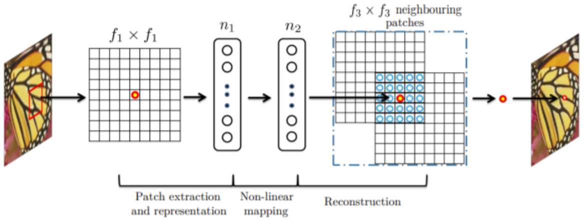

$f_3 \times f_3$ neighbouring patches
$f_1 \times f_1$
n_1
n_2
Patch extraction and representation
Non-linear mapping
Reconstruction

Figure 2. SRCNN Architecture [3]

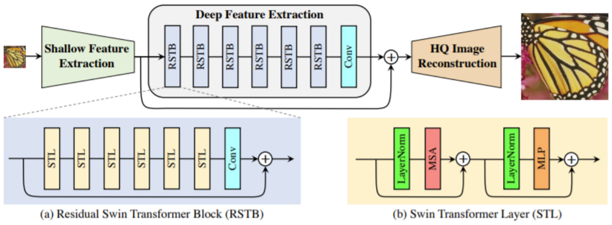


Figure 3. SwinIR Architecture [6]

2.2.2 Metric

PSNR: Peak signal-to-noise ratio (PSNR) is an expression for the ratio between the maximum possible value (power) of a signal and the power of distorting noise that affects the quality of its representation. Specifically, $PSNR = 20\log_{10}\left(\frac{\mathrm{MAX_f}}{\sqrt{\mathrm{MSE}}}\right)$ where $\mathrm{MSE} = \frac{1}{\mathrm{m \cdot n}}\sum_0^{\mathrm{m}-1}\sum_0^{\mathrm{n}-1}\|\mathrm{f(i,j)} - \mathrm{g(i,j)}\|^2$. In the above equations, f represents the matrix data of our original image, g represents the matrix data of our degraded image in question, m represents the numbers of rows of pixels of the images, n represents the number of columns of pixels of the image. The larger the PSNR is, the higher quality the image has.

2.3. Classification

For classification task, we use Resnet-50 to train, and Mean Average Precision (mAP) to evaluate the performance.

2.3.1 Model

Resnet-50: ResNet represents for residual networks. It is a classic neural network used for computer vision tasks.

ResNet-50 is a variant of ResNet model which has 48 Convolution layers along with 1 MaxPool and 1 Average Pool layer. Our project applies ResNet-50 for classification.

2.3.2 Metric

Mean Average Precision (mAP): For object detection, mAP is widely used as an evaluation metric. Rather than taking the average of precision, mAP is average of average precision, where average precision(AP) is the area under the precision-recall curve. Precision and recall are respectively calculated by $\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP+FP}}$, $\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP+FN}}$.

3. Experiments

Super Resolution: For the super resolution part, we trained, tested, and compared three models: 1) Bicubic, 2) SRCNN, 3) SwinIR.

Bicubic performs data interpolation on the low resolution images and generates high resolution images accordingly. So for the Bicubic method, we just feed in the test dataset Set5 into it and calculate its performance. For SRCNN and SwinIR, we trained the two deep neural networks on dataset DIV2k, and tested the trained models' performance on the Set5 dataset. For bicubic method, the coefficient a used in the process is 0.5. The SRCNN model use a batch size of 4, the channel of the feature extraction are 64 and 32, and the kernel size are set as 9 and 5. The best performing parameters for SwinIR that we used are: the RSTB number set to default 6, the STL number set to default 6, window size set to 8, channel number set to 180, and attention head number set to 6.

We used PSNR (peak signal-to-noise ratio) to measure and compare the performance of our three models. This metric measures the image enhancement quality of the high resolution images generated from the low resolution ones by the SR models. The average PNSR results of the three trained models Bicubic, SRCNN, and SwinIR are summarized in table 1. We can see that the SwinIR model outperforms the two baseline models–SRCNN and Bicubic to a great extent.

	SwinIR	Bicubic	SRCNN
Average PSNR	36.15	28.69	31.73

Table 1. Average PSNR of Different Super Resolution Technique

Downstream Classification: In this part, we use Mean Average Precision (mAP) to measure the overall success of the multi-label classification results. We first train a 3-conv-layer CNN based on 442 assignment. However, the test mAP is only 0.1796, which can hardly be used. Therefore, we train a Resnet-50 model on VOC2007 dataset and use it to test the performance of our SwinIR model. To be specific, the trained Resnet-50 model is a multi-label image classifier and it takes 5 test sets as input: the original pictures in VOC2007 test set, the low resolution pictures, the SwinIR results, the SRCNN results and the Bicubic Interpolation results. Figure 4 shows the visual comparison of those 5 sets. We can see that the resolution has been improved to a great extent by using SwinIR. Table 2 shows the quantitative comparisons among those 5 test sets and Table 3 shows the Average Precision(AP) for each class.

	Original Pictures	Low-Resolution Pictures	SRCNN Result	Bicubic Result	SwinIR Result
Mean Average Precision	0.9091	0.9050	0.8907	0.8998	0.9021

Table 2. Mean Average Precision under Different Test Set

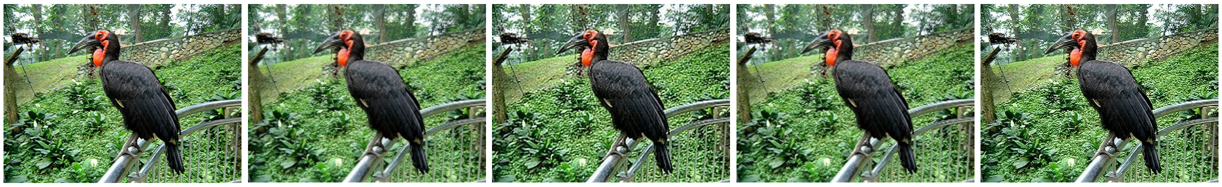
As one can see, when tested on the original test set of the VOC2007, we reach a mAP of 0.9091, which can also show that the performance of the model is great. The mAP for low resolution test set is 0.9050, higher than 0.8907 for SRCNN, 0.8998 for Bicubic and 0.9021 for SwinIR. It is counter-intuitive that the mAP for super resolution results are lower. It means that the SwinIR is not helping the classification. One possible explanation is that some useful features in the low resolution image are lost. SwinIR model may add too many artificial details.

To help figure out the reason, we try Class Activation Mappings (CAM) which overlay a heatmap over the original image to show us where our model focus on. Figure 5 shows the focus of the final convnet layer when the input is the low resolution picture and SwinIR result. We can see that those two heatmaps are slightly different from each other, especially in the bird's head and body regions. Further analysis is still needed to determine the exact reason of why SwinIR cannot help the classification.

4. Implementation

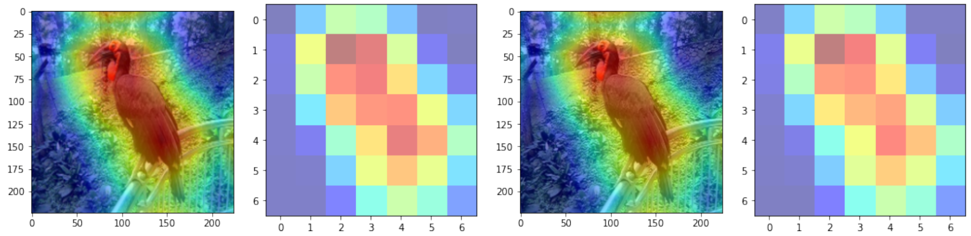
For the super resolution part, we implement the down sample code on our own. We then refer to the SwinIR network model from the following link (https://github.com/JingyunLiang/SwinIR). We modify the test and picture generation function to our needs. For the baselines super resolution method. The bicubic is implemented by ourselves based on the algorithm and the SRCNN makes reference to the following link (https://github.com/Mirwaisse/SRCNN). The dataset for train and test are modified and the parameters including the batch size and the number of training epoch are adjusted to achieve better performance.

For the classification part, we refer to a residual network model from the following link (https://github.com/lyz04551/voc2007_classification_pytorch). We add some functions calculating the average precision on all classes and the mean average precision. We redesign and simplify the randomnized data loader because the



(a) Original image (b) Low resolution image (c) SRCNN (d) Bicubic Interpolation (e) SwinIR

Figure 4. Visual comparison of SR methods on a test image



(a) Overlayed Heatmap of Low Resonlution Pictures (b) Heatmap of Low Resolution Pictures (c) Overlayed Heatmap of SwinIR result (d) Heatmap of SwinIR result

Figure 5. Heatmap on a test image

Average Precision	Original Pictures	Low-Resolution Pictures	SRCNN Result	Bicubic Result	SwinIR Result
Aeroplane	0.9857	0.9815	0.9794	0.9761	0.9810
Bicycle	0.9495	0.9530	0.9323	0.9400	0.9439
Bird	0.9492	0.9494	0.9453	0.9493	0.9451
Boat	0.9457	0.9523	0.9430	0.9482	0.9545
Bottle	0.6354	0.6068	0.5760	0.5853	0.6076
Bus	0.9320	0.9260	0.9022	0.9078	0.9252
Car	0.9354	0.9311	0.9280	0.9321	0.9313
Cat	0.9182	0.9006	0.8986	0.8967	0.9051
Chair	0.7546	0.7627	0.7636	0.7676	0.7563
Cow	0.8726	0.8560	0.8327	0.8324	0.8594
Dining Table	0.7776	0.7916	0.7564	0.7732	0.7557
Dog	0.9322	0.9362	0.9061	0.9409	0.9198
Horse	0.9676	0.9635	0.9569	0.9585	0.9632
Motor Bike	0.9006	0.8898	0.8717	0.8813	0.8866
Person	0.9676	0.9637	0.9569	0.9605	0.9617
Potted Plane	0.7412	0.7472	0.7355	0.7392	0.7589
Sheep	0.9069	0.8977	0.8735	0.8921	0.8880
Sofa	0.6550	0.6848	0.7005	0.6814	0.6864
Train	0.9817	0.9760	0.9726	0.9712	0.9759
Tv Monitor	0.8230	0.8172	0.8181	0.8252	0.8311

Table 3. Average Precision of Different Classification under Different Test Set

focus of the project is not on the image classification. We also add code for the heatmap generation to visualize the output of the model applied by following this tutorial (http://www.snappishproductions.com/blog/2018/01/03/class-activation-mapping-in-pytorch.html.html).

References

[1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 126–135, 2017.

[2] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie Line Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. 2012.

[3] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2015.

[4] Mark Everingham, Andrew Zisserman, Christopher KI Williams, Luc Van Gool, Moray Allan, Christopher M Bishop, Olivier Chapelle, Navneet Dalal, Thomas Deselaers, Gyuri Dorkó, et al. The pascal visual object classes challenge 2007 (voc2007) results. 2008.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[6] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1833–1844, 2021.

[7] Kamal Nasrollahi and Thomas B. Moeslund. Super-resolution: a comprehensive survey. *Machine Vision and Applications*, 25(6):1423–1468, Aug 2014.

[8] C. Papathanassiou and M. Petrou. Super resolution: An overview, Jul 2005.

[9] Yuanwei Wu, Ziming Zhang, and Guanghui Wang. Unsupervised deep feature transfer for low resolution image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.