Image Super Resolution - Report

Computer Vision – CSE578

Team 26, =*3*

Introduction

Abstract

The objective of image super-resolution (SR) is to enhance the resolution of a given low-resolution (LR) image, which has always been a continuous ongoing process in image technology, through up-sampling, de-blurring, de-noising, etc. In order to restore an image into a high-resolution (HR) image correctly, it is necessary to infer high-frequency components of a low-resolution image. In some applications like video surveillance, forensic investigation, face recognition, medical diagnosis, satellite images, and pattern recognition, it becomes essential to extract useful information from the images.

Methods

There are broadly three approaches to perform image super-resolution:

1. Interpolation based
2. Example based
3. Reconstruction based

Interpolation based approach

This is the simplest way to provide super-resolution by applying interpolation on the sampled visual data acquired from the sensor. This technique has the advantage of less computational complexity due to its simplicity and also real-time applications are possible. The low-frequency (LF) band of the high-resolution image will be more or less accurately reconstructed. For example, bicubic interpolation can be sued to increase the resolution of an image.

High Resolution

Bicubic Interpolated

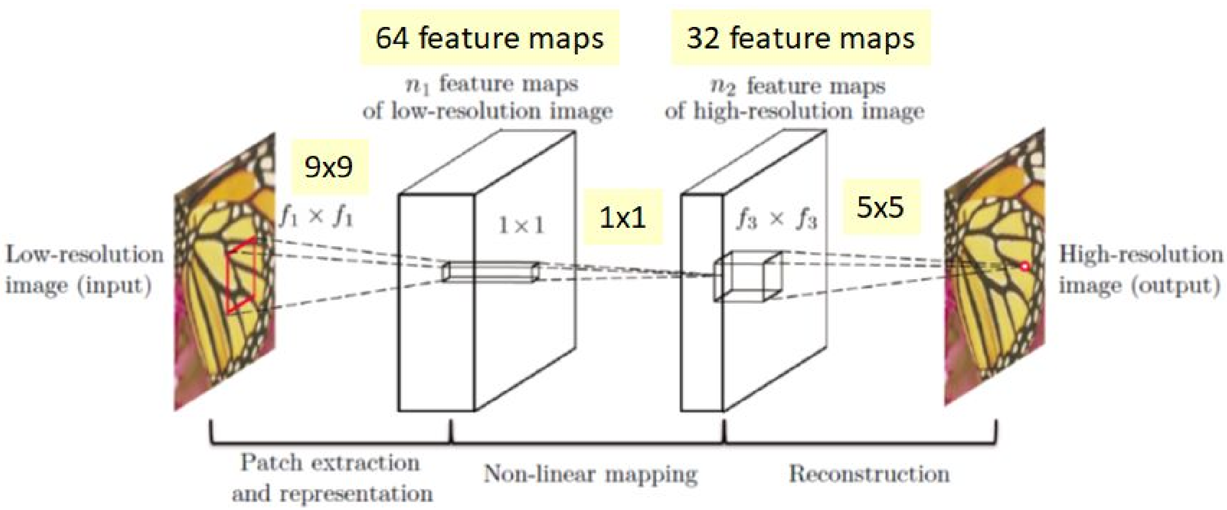
Example based approach

This approach estimates the high-resolution version of a low-resolution single-image by exploiting examples. This generally involves a model trained using a varied training set of images.

Sparse Coding
This approach is one of the external learning methods. It learns a sparse dictionary which is used to map low-resolution patches in the low-resolution image to high-resolution patches in the high-resolution image. To upsample an image, it finds the most similar patch in the learned dictionary and maps it to the corresponding high-resolution patch.

SRCNN
As part of this project, we implement an SRCNN model proposed by *Dong et al* in this paper presented at ECCV in 2014.
SRCNN is a very shallow convolutional neural network which broadly comprises of 3 parts:

1. Patch extraction and representation
2. Non-linear mapping
3. Reconstruction



The first step involves patch extraction and representation. In the above image, f1=9, i.e., 9x9 filters are used with an overlapping stride of 1 over the original input image. We obtain 64 features from each 9x9 match. We use zero padding=4 in this step to maintain the dimensionality of the input image.
The second step involves a non-linear mapping such that the 64 features obtained from the first step are now mapped to a set of 32 features. The final step is the reconstruction phase where the set of 32 features are mapped to the total number of channels in the input image with the usage of 5x5 filters, which is represented as f3xf3 in the above image. We use zero padding=2 in this step to maintain the dimensionality of the input image.

Relationship with Sparse Coding - For Sparse Coding (SC), in the view of convolution, the input image is convoluted by f1 and project to onto an n1-dimensional dictionary. n1=n2 usually is the case of SC. Then mapping of n1 to n2 is done with the same dimensionality without reduction. It is just like a mapping of low-resolution vector to a high-resolution vector. Then each patch is reconstructed by f3. And overlapping patches are averaged instead of adding together with different weights by convolution.

Dataset - We experimented with the following datasets for training:

1. **T91**: Preprocessed to generate 22092 images of size 32 *x* 32
2. **BSD300:** dataset generated for boundary detection which can be used to learn edge upsampling.

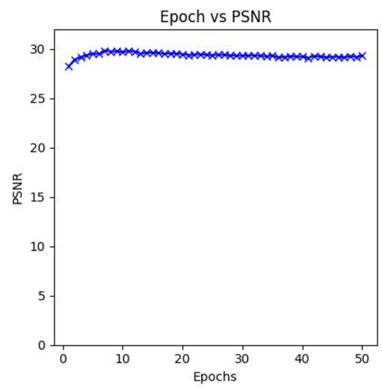


Fig: Training convergence. Each epoch has ~22K backprops

Loss function - We experiment with the following loss functions for our model:
1. PSNR
2. Perceptual Loss using 7th layer output of VGG19

Results

	2x	**3x**	**4x**
PSNR	36.21	32.41	30.12
Perceptual Loss	35.43	31.91	28.96

Table 1: PSNR values with SRCNN trained on different loss functions

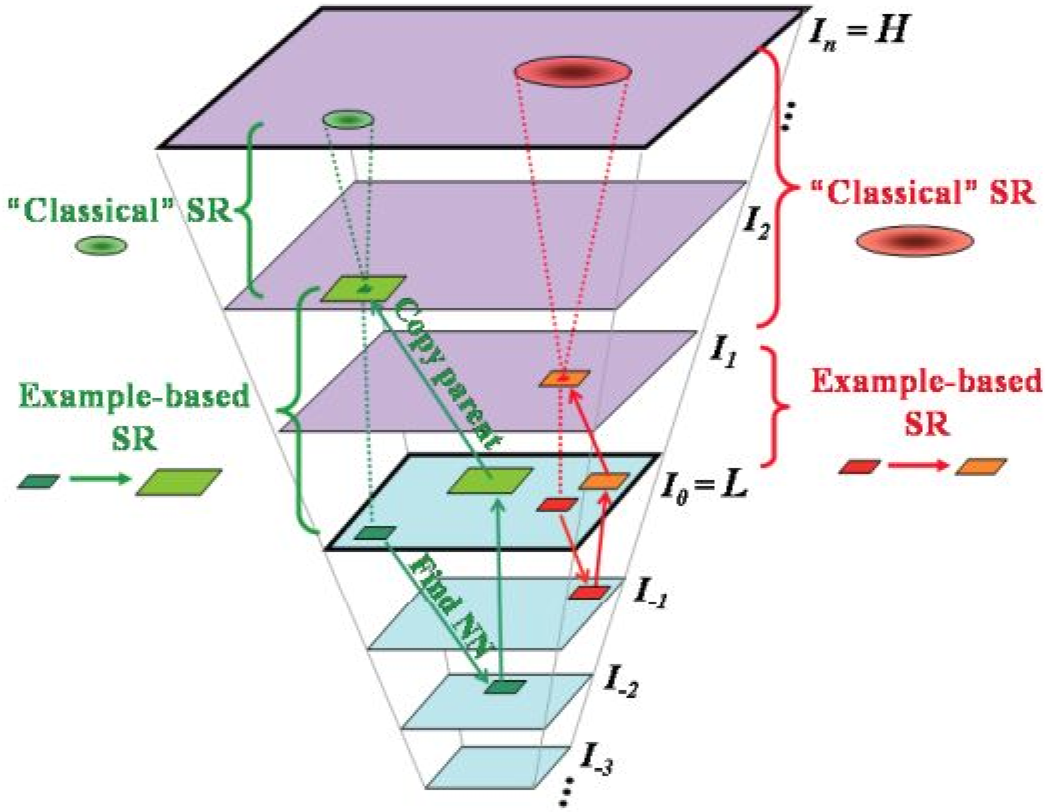
Experiments
We ran the model on recursively on the query image with the intuition that the image quality will improve recursively.

However, this wasn’t true since the image becomes progressively worse as we increase the number of iterations.

To test out the applicability of the model on practical applications, we also ran the model on CCTV footages and got the following results:

Glasner

For the purpose of comparison, we also implemented another example based approach introduced by *Glasner et al* in [this paper] presented at ICCV in 2009.



In this paper. The authors propose a unified framework for combining two families of methods - (i) The classical multi-image super-resolution (combining images obtained at subpixel misalignments), and (ii) Example-Based super-resolution (learning correspondence between low and high resolution image patches from a database). They further show how this combined approach can be applied to obtain super-resolution from as little as a single image (with no database or prior examples). Our approach is based on the observation that patches in a natural image tend to redundantly recur many times inside the image, both within the same scale, as well as across different scales. Recurrence of patches within the same image scale (at subpixel misalignments) gives rise to the classical super-resolution, whereas recurrence of patches across different scales of the same image gives rise to example-based super-resolution. This approach attempts to recover at each pixel its best possible resolution increase based on its patch redundancy within and across scales.

Dataset - We experimented with the following datasets for training:

3. **T91**: Preprocessed to generate 22092 images of size 32 *x* 32
4. **BSD300:** dataset generated for boundary detection which can be used to learn edge upsampling.

Results

	2x	**3x**	**4x**
PSNR	35.39	31.21	28.94

Reconstruction based approach

Reconstruction-based algorithms are widely used in super-resolution algorithms. They include maximum likelihood (ML), maximum a posteriori (MAP), projection onto convex sets (POCS), and other methods. Reconstruction-based algorithms have a common framework. A part of that common framework first sets the resolution enhancement ratio, called a magnification factor. People simply expect that they can set a larger magnification factor when using a larger number of LRIs. Through practical trials, they have found that this resolution enhancement is quite limited, even if using a large number of LRIs.

Quantitative Comparison

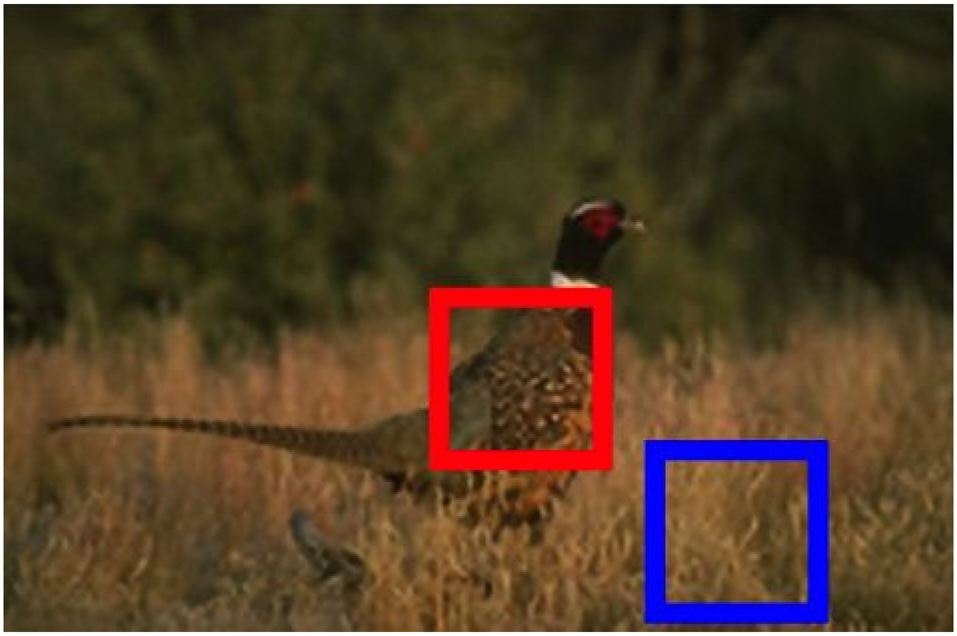
We compare the two methods we implemented (SRCNN vs Glasner) using PSNR values achieved. These are displayed across various scales in the table below.

	2x	**3x**	**4x**
SRCNN	36.21	32.41	30.12
Glasner	35.39	31.21	28.94

From the above table, it is clear that the SRCNN approach outperforms Glasner's approach by a considerable margin. This dominance in performance is further exemplified by some qualitative examples shown in the following section.

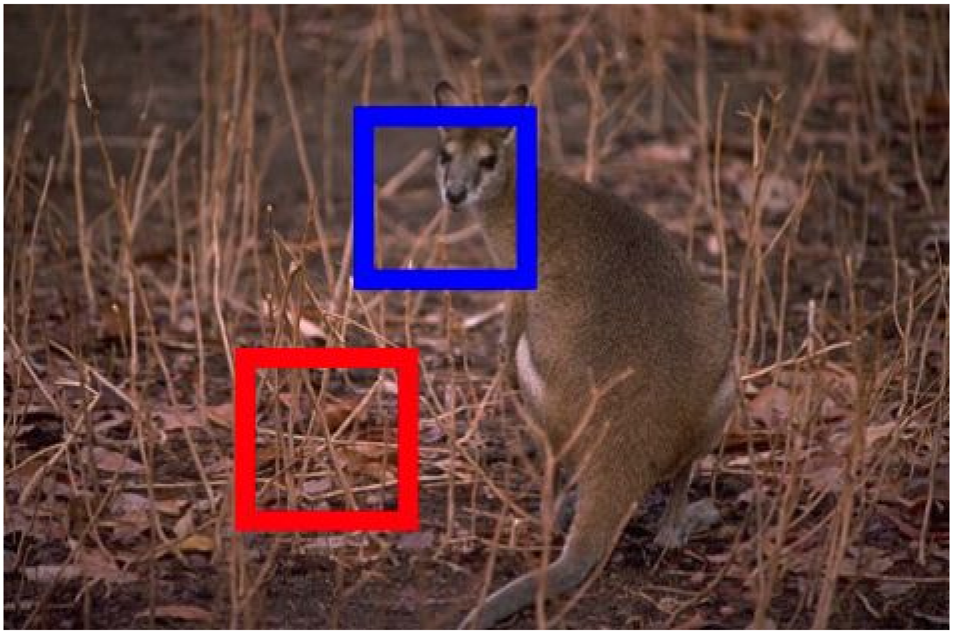
Qualitative Comparison

At 2x



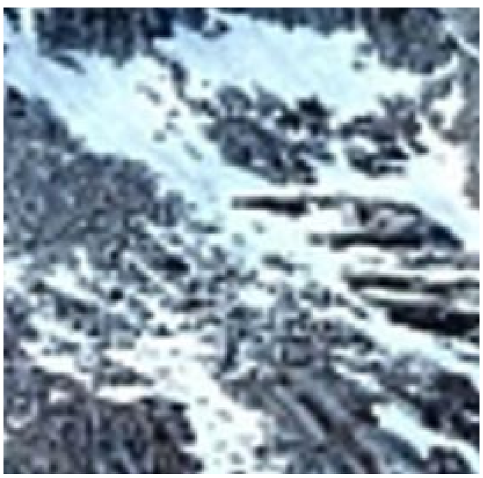
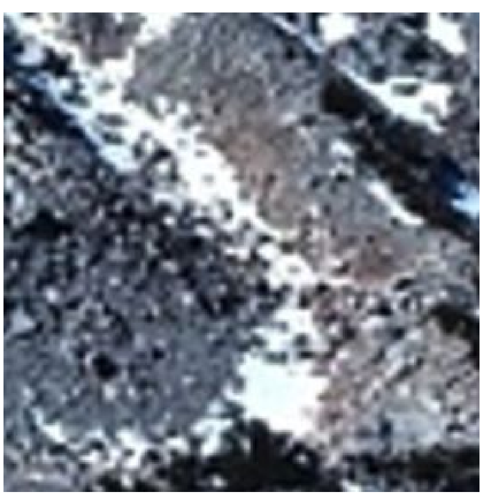
High Resolution	SRCNN	Glasner
		
		

At 3x



High Resolution	SRCNN	Glasner
		
		

At 4x

High Resolution	SRCNN	Glasner
		
		

Conclusion

It is clear (through both quantitative and qualitative analyses) that the SRCNN approach is superior to Glasner's approach to image super-resolution.

1. In areas of little detail, Glasner and SRCNN approaches have similar outputs with little to no difference.
2. In areas of fine details, the SRCNN approach is far superior to Glasner's approach.

The above two conclusions are easily noticed in the qualitative comparison section's first example at 2x scale. In this image, the body of the bird consists of fine details that are much clearer in the SRCNN generated image, but these details are blurred out in the image generated by Glasner's approach. In the same image, the background image consisting of fewer details is generated with equally good quality by both the SRCNN and Glasner approaches.

Code Base

The final code base can be found at - https://github.com/dheerajpreddy/image-super-resolution