
Enhancing Single Image Super-resolution

Soumyadeep Dasgupta

Department of Materials Science & Engineering
University of Michigan at Ann Arbor
Ann Arbor, MI 48109
soumyadg@umich.edu

Robert J Jones

Department of Electrical and Computer Engineering
University of Michigan at Ann Arbor
Ann Arbor, MI 48109
rjjones@umich.edu

Daksh Maheshwari

Department of Electrical and Computer Engineering
University of Michigan at Ann Arbor
Ann Arbor, MI 48109
daksh@umich.edu

EECS 556 W24

April 26, 2024

Honor Code Statement:

Our team will give attribution for any figures used in our documents and will cite all code sources (beyond those already built into Julia Base or Matlab toolboxes).

Abstract

Our project focuses on the method introduced in [7] for single image super-resolution using sparse representation. The method involves jointly training two over-complete dictionaries on low-resolution (LR) and high-resolution (HR) image patches. By exploiting the sparsity of image patches, the coupled dictionaries use the sparse representation of the LR image patch to recover the HR image patch. The method provides a compact representation of patch pairs that reduces computational complexity, and possesses inherent sparse local modeling that offers robustness to noise, which have become important elements of many subsequent super-resolution approaches. This project will focus on investigating the image processing concepts of the sparse representation super-resolution framework in [7]. We will implement the method and replicate the results in [7] and explore different extensions to the method. Code can be found at <https://github.com/dmdaksh/sparse-coding-super-resolution>

1 Introduction

Super-resolution (SR) image reconstruction has emerged as a dynamic field of research, offering a pathway to overcome the intrinsic limitations of low-cost imaging sensors, such as those found in cell phones and surveillance cameras. By enhancing image resolution, SR enables better utilization of the increasing capabilities of high-resolution displays, including high-definition LCDs. This technology holds promise not only in consumer electronics but also in critical domains such as medical imaging and satellite imaging, where accurate diagnosis or analysis from low-quality images is challenging.

Conventional super-resolution techniques typically rely on multiple low-resolution (LR) images of the same scene that are meticulously aligned with sub-pixel accuracy. The SR problem is formulated as

an inverse problem, aiming to reconstruct the original high-resolution (HR) image by amalgamating information from the low-resolution images, guided by reasonable assumptions or prior knowledge about the observation model mapping high-resolution to low-resolution images. However, SR image reconstruction is inherently challenging due to several factors, including insufficient input images, ill-conditioned registration, unknown blurring operators, and non-uniqueness of solutions. To address these challenges, various regularization techniques have been proposed, but they often falter when faced with large magnification factors or limited input images, resulting in overly smooth outputs lacking crucial high-frequency details[5, 3].

To mitigate these limitations, researchers have explored different approaches to super-resolution, including interpolation-based methods[6, 2] and machine learning techniques[4, 11]. While interpolation methods suffer from producing overly smooth images with artifacts, machine learning approaches aim to capture the co-occurrence prior between low-resolution and high-resolution image patches. However, many learning-based methods demand extensive databases of paired high-resolution and low-resolution patches, making them computationally intensive.

In this project, our aim was to extend the insightful work of Yang et al. [7]. The authors focused on advancing the state-of-the-art in super-resolution by leveraging sparse signal representation techniques. Their approach was motivated by recent successes (at the time of publication in 2010) in sparse signal representation, which suggested that linear relationships among high-resolution signals can be accurately recovered from their low-dimensional projections. They learned a compact representation of pairs of image patches from features extracted from low-resolution images, thereby capturing the co-occurrence prior and significantly enhancing algorithmic efficiency. The super-resolution via sparse representation method presented in [7], which we will refer to as ScSR, demonstrated superior performance in both generic single-image super-resolution and domain-specific applications, such as face hallucination.

In the subsequent sections, we detail the formulation and solution to the single-image super-resolution problem based on sparse representation. We discuss how we jointly learn two dictionaries for high- and low-resolution image patches and present experimental results showcasing the efficacy of sparsity as a prior for single-image super-resolution. We describe how we implemented the ScSR method proposed in [7], as well as discuss a number of original extensions to the ScSR method that we investigated. We refer to the quantitative results from other super-resolution methods [16] subsequently introduced since the publication of [7] in order to compare the performance of our extensions to other super-resolution methods. Through our research, we aimed to contribute to the advancement of super-resolution techniques by offering practical solutions that are efficient, generalizable and robust to noise, thereby addressing key challenges in the field.

2 Quantitative performance prediction

The primary focus of our project was to investigate single image super-resolution. The method introduced in [7] (which we will refer henceforth as SCSR) serves as a seminal work in the domain of single image super-resolution using sparse representations. Since the publication of [7], various extensions have been implemented based on this methodology. We will implement the ScSR method and compare its performance to other super-resolution approaches (see Plan section for implementation details).

To quantitatively evaluate and compare the performance of super-resolution methods, we will examine commonly used image quality metrics, namely the peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM). Higher PSNR and SSIM indicate better performance. Analysis of PSNR and SSIM metrics were not presented in [7]. Instead, they used the root-mean-square-error (RMSE) between HR ground truth and super-resolution images. Thus, we will also use the RMSE for quantitative evaluations. Lower RMSE indicates better performance. For all image quality metrics, the results obtained from standard (bicubic) interpolation will serve as a baseline for assessing the super-resolution methods.

The definitions of these metrics are as follows. Let $f[m, n]$ be an (original) $M \times N$ high-resolution image, and $\hat{f}[m, n]$ be a restored version (e.g., an SR reconstructed image) of f . The mean squared error (MSE) is defined as:

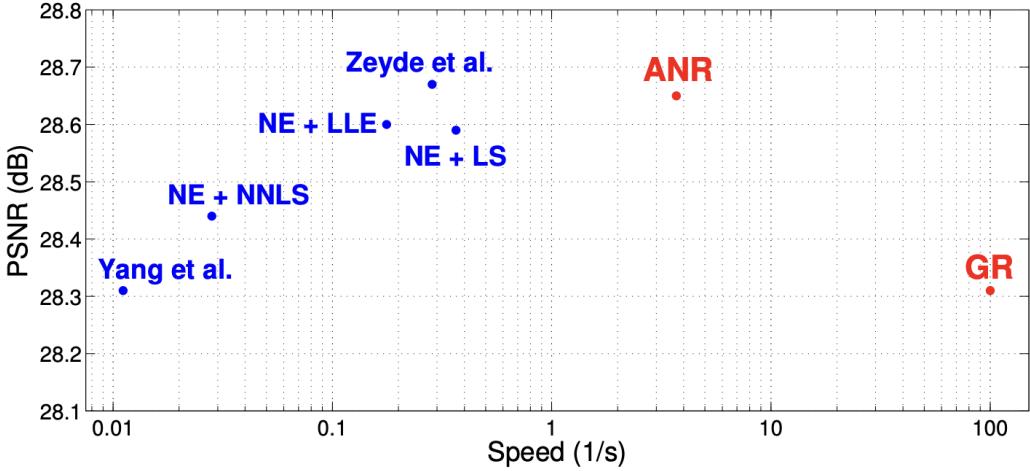


Figure 1: Figure 1 from [12], showing speed vs. PSNR for different super-resolution methods. The point Yang et al. show the SCSR method from [7].

$$\text{MSE} = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} |\hat{f}[m, n] - f[m, n]|^2 \quad (1)$$

Then, the RMSE is defined as:

$$\text{RMSE} = \sqrt{\text{MSE}} \quad (2)$$

The PSNR is defined as:

$$\text{PSNR} = 20 \log_{10} \left(\frac{\|f\|_\infty}{\text{RMSE}} \right) \quad (3)$$

The SSIM is defined as:

$$\text{SSIM} = \frac{(2\mu_f\mu_{\hat{f}} + C_1)(\sigma_{f\hat{f}} + C_2)}{(\mu_f^2 + \mu_{\hat{f}}^2 + C_1)(\sigma_f^2 + \sigma_{\hat{f}}^2 + C_2)} \quad (4)$$

where μ_f , $\mu_{\hat{f}}$ and σ_f^2 , $\sigma_{\hat{f}}^2$ are the mean and variance of f and \hat{f} respectively, $\sigma_{f\hat{f}}^2$ is the covariance of f and \hat{f} , and C_1 , C_2 are scalar variables stabilizing the division.

In our experiments, we perform super-resolution on color (RGB) images. The super-resolution approach from [7] does this by first converting the images from RGB to YCbCr format, which stores image luminance or brightness in one channel (Y) and stores image chrominance or color (Cb, Cr) in two channels. SR is then performed only on the luminance channel, while the chrominance channels are both upsampled to the high-resolution image dimensions using bicubic interpolation. The three channels are then merged together to produce the final magnified SR image. When computing image quality metrics using this SR approach, the RMSE is computed only on the luminance channel. This RMSE is then used in the PSNR calculation. Thus, the RMSE and PSNR are calculated on only the luminance channel. The SSIM is calculated by first converting the color images to grayscale and then computing the SSIM.

Despite the exclusion of PSNR and SSIM analyses from [7], there are other papers that compared SCSR with other super-resolution methods [12, 15, 16] and that included PSNR and SSIM (see Figure 1). These papers give us an external reference for what we might expect in terms of the performance of SCSR with respect to other super-resolution methods.

We predicted that the SCSR method [7] will have an average of 1-3dB higher PSNR than bicubic interpolation (baseline) and about 1dB lower PSNR compared to other subsequently proposed methods for magnification factor 3. This prediction is based on using the training dataset from [7]

for dictionary training, which contains 91 real-world images of various objects and targets (animals, landscapes, faces, fruits, etc.) and was also used in other works [15, 9]. There are a number of dictionary training parameters that must be determined, including dictionary size and patch size. We plan to optimize these parameters by an exhaustive grid search.

We found that the best results we got had at least 15dB higher PSNR compared to the proposed method for the magnification factor 3. This result was from using learnt feature maps from a deep model. We also received about 1-5dB lower PSNR for experiments where we changed the blurring filter or modified the objective.

Overall, we proposed that our extensions will outperform ScSR, but that other more recent single image super-resolution techniques [9, 8, 10, 14] will outperform ScSR and our proposed extensions, in terms of both image quality metrics and computation time (1). ScSR was introduced over a decade ago and represented a seminal work on sparse representation-based SISR, and various extensions and deep learning based approaches have since emerged.

3 Methods

3.1 Established Method: Image super-resolution from sparsity

The goal of single-image super-resolution is to take a given low-resolution image \mathbf{Y} and recover a higher resolution image \mathbf{X} of the same scene. To solve this ill-posed problem, the process established in this article enforces two constraints, namely reconstruction constraints and sparsity priors, in the SR reconstruction.

3.1.1 Modeling constraints and assumptions

Due to the ill-posed nature of single-image SR, assumptions are made to simplify the problem. We first outline the constraints enforced by Yang et al. [7] in the formulation of the SR problem.

Reconstruction Constraint

The relationship between low-resolution and high-resolution images is described by the observation model:

$$\mathbf{Y} = S H \mathbf{X} \quad (5)$$

where H represents a blurring filter and S is the downsampling operator. Thus, the observed low-resolution image \mathbf{Y} is modeled as a blurred and downsampled version of the high-resolution image \mathbf{X} . The reconstruction constraint requires that the recovered image \mathbf{X} be similar to the observed image \mathbf{Y} with respect to the observation model. However, for a given low-resolution image \mathbf{Y} , there exists infinitely many high-resolution images \mathbf{X} that satisfy the reconstruction constraint. This motivates the use of the sparsity prior as an additional regularizer in the SR problem.

Sparsity Prior

The sparsity prior relies on the assumption that image patches are sparse. Specifically, image patches \mathbf{x} of the high-resolution image \mathbf{X} can be sparsely represented using a linear combination of atoms from an overcomplete dictionary D_h :

$$\mathbf{x} \approx D_h \boldsymbol{\alpha} \quad \text{for some } \boldsymbol{\alpha} \in \mathbb{R}^K \text{ with } \|\boldsymbol{\alpha}\|_0 \ll K$$

The sparse representations $\boldsymbol{\alpha}$ can be recovered from observed low-resolution patches \mathbf{y} of the input image \mathbf{Y} using a dictionary D_l , and used to recover the SR image \mathbf{X} . One of the key ideas behind this single-image SR approach is that the same sparse representation $\boldsymbol{\alpha}$ is used to represent both the low-resolution patch \mathbf{y} from D_l and represent the high-resolution patch \mathbf{x} from D_h . This is due to the constraining of dictionaries D_h and D_l , which is described in following sections.

3.2 Algorithm Overview

The generic single-image super-resolution formulation consists of two steps. First, a local model is employed to estimate a high-resolution patch x from an observed low-resolution patch y . Second, a global model enforces the reconstruction constraint in equation 5.

1. **Local Model:** Sparse representations for each patch are found using D_l .

$$\min \|\alpha\|_0 \quad \text{s.t.} \quad \|F\mathbf{D}_l\alpha - F\mathbf{y}_2^2\| \leq \epsilon$$

For each feature extracted path \mathbf{y} , we are finding a sparse representation w.r.t. to \mathbf{D}_l . Later, the corresponding HR patches can be generated by \mathbf{D}_h along with these coefficients. The F is a linear feature extraction operator.

Using the Lagrange multiplier, we can formulate the above as:

$$\min_{\alpha} \|F\mathbf{D}_l\alpha - F\mathbf{y}\|_2^2 + \lambda \|\alpha\|_1$$

where λ controls the trade-off between the sparsity of the solution and data consistency.

2. **Global Model:** To ensure the HR image X_0 produced by the local model satisfies the reconstruction constraint (equation 5, the initial solution X_0 is projected onto the solution space of $SHX = Y$:

$$\mathbf{X}_{t+1} = \mathbf{X}_t + \mu [H^T S^T (\mathbf{Y} - SH\mathbf{X}_t)]$$

The solution can be found iteratively using gradient descent via the update:

$$X_{t+1} = X_t + \nu [H^T S^T (Y - SHX_t)]$$

where X_t is the estimate at iteration t and ν is the step size.

3.2.1 Super-Resolution via Sparse Representation

The entire SR algorithm stated in [7] can be summarised compactly as follows:

SR via Sparse Representation:

Input: Training dictionaries D_h and D_l , a low-resolution image Y .

For each 3×3 patch y of Y , taken starting from the upper-left corner with 1 pixel overlap in each direction,

- Compute the mean pixel value m of patch y .
- Solve the optimization problem with \hat{D} and \hat{y} defined in (8): $\min_{\alpha} \|\hat{D}\alpha - \hat{y}\|_2^2 + \lambda \|\alpha\|_1$.
- Generate the high-resolution patch $x = D_h\alpha^*$. Put the patch $x + m$ into a high-resolution image X_0 .

End: Using gradient descent, find the closest image to X_0 which satisfies the reconstruction constraint

$$X^* = \arg \min_X \|SHX - Y\|_2^2 + c\|X - X_0\|_2^2.$$

Output: SR image X^* .

3.2.2 Learning the Dictionary Pair

1. **Single Dictionary Training:** Learn a compact dictionary D from training examples \mathbf{X} using sparse coding. The steps are as follows:

- (a) Initialize D with a Gaussian random matrix, with each column unit normalized.
- (b) Fix D , update Z by

$$Z = \arg \min_Z \|X - DZ\|_2^2 + \lambda \|Z\|_1 \tag{6}$$

solved efficiently through linear programming.

- (c) Fix Z , update D by

$$D = \arg \min_D \|X - DZ\|_2^2 \quad \text{s.t.} \quad \|D_i\|_2 \leq 1, i = 1, 2, \dots, K \tag{7}$$

which is a Quadratically Constrained Quadratic Programming.

- (d) Iterate between steps b and c until converge. .

2. **Joint Dictionary Training:** Simultaneously learn dictionaries for high and low-resolution patches to ensure shared sparse representations. The individual sparse coding problems in the high-resolution and low-resolution patch spaces are

$$D_h = \arg \min_{\{D_h, Z\}} \|X^h - D_h Z\|_2^2 + \lambda \|Z\|_1 \quad (8)$$

and

$$D_l = \arg \min_{\{D_l, Z\}} \|Y^l - D_l Z\|_2^2 + \lambda \|Z\|_1 \quad (9)$$

The objectives are then combined, forcing the high-resolution and low-resolution representations to share the same codes, which allows us to write,

$$\min_{\{D_h, D_l, Z\}} \frac{1}{N} \|X^h - D_h Z\|_2^2 + \frac{1}{M} \|Y^l - D_l Z\|_2^2 + \lambda \left(\frac{1}{N} + \frac{1}{M} \right) \|Z\|_1 \quad (10)$$

where N and M are the dimensions of the high-resolution and low-resolution image patches in vector form. Here, $1/N$ and $1/M$ balance the two cost terms of (8) and (9). (10) can be rewritten as

$$\min_{\{D_h, D_l, Z\}} \|X^c - D_c Z\|_2^2 + \lambda \left(\frac{1}{N} + \frac{1}{M} \right) \|Z\|_1 \quad (11)$$

3.2.3 Feature Representation

First- and second-order derivatives are used as features for low-resolution patches to enhance prediction accuracy. The features are extracted from low-resolution image patches using filters $[-1 \ 0 \ 1]$ and $[1 \ 0 \ -2 \ 0 \ 1]$, respectively. A total of 4 features have been extracted for each low-resolution image patch and used for subsequent dictionary training. Third-order derivatives and learned feature maps are also used for feature extraction for the low-resolution images, which will be discussed in the later experiments section [4.4, 4.6]

3.3 Implementation

We explored implementing the SR approach from [7] in MATLAB and Python, using available toolboxes in both languages as starting points. We observed differences in performance between the two languages in the dictionary training and SR reconstruction steps.

While no code was available from the paper's first author, several other resources with available code implemented the single-image SR work by Yang et al. [7]. We found two MATLAB toolboxes and one Python implementation that we used as a starting point for our implementation:

1. Zeyde et al. [15] introduced a direct extension of the work from [7], and provided Matlab code available at <https://elad.cs.technion.ac.il/software>.
2. A MATLAB GitHub repository with demo codes for single-image SR available at <https://github.com/tingfengainiaini/sparseCodingSuperResolution>. The readme cited [7, 13], and listed the first author (Yang) as a contact.
3. A Python GitHub repository for Sparse Dictionary Training and Sparse Coding Super Resolution is available at <https://github.com/BrunoVox/ScSR>. The README cited [7], and listed the first author (Yang) as a contact.

These Matlab toolboxes and Python code contained almost identical implementations of the dictionary training and super-resolution reconstruction described in [7], with minor variations. The downloaded code did not run out of the box for a few test cases and didn't work as expected for the rest of the test cases. However, several functions required edits to work properly for both the MATLAB and Python implementations.

We modified Python code to a great extent, as the author translated the MATLAB code into Python. We made the code more Pythonic and re-formulated matrices sums and products to make it more efficient, as well as reformulated the expressions given in the paper [7] to make it more efficient for our implementation. We were able to speed up our implementation by 30% for dictionary training as well as for learning sparse representation.

4 Experimental Results

4.1 Effect of dictionary parameters on SR performance

As a first step, we implemented the dictionary training and ScSR super-resolution method from [7] and performed test reconstructions using different dictionaries. We compared dictionaries with 256, 512 and 1024 atoms, that were trained on 25,000, 50,000, and 100,000 randomly sampled patches from natural images (using the training dataset from [7]). We then used the dictionaries to perform ScSR reconstructions of a test image from [7] (the Child image). The results in table 4.1 show the PSNR, RMSE and SSIM of SR reconstructions from different dictionaries.

Table 1: Experiment results. Effect of dictionary training parameters on ScSR performance.

Metric	Number of patches	256 atoms	512 atoms	1024 atoms
PSNR	25k	33.312	33.377	33.394
	50k	33.362	33.350	33.324
	100k	33.370	33.372	33.360
RMSE	25k	5.507	5.467	5.456
	50k	5.476	5.484	5.500
	100k	5.471	5.469	5.477
SSIM	25k	0.798	0.799	0.799
	50k	0.799	0.799	0.798
	100k	0.799	0.799	0.799

Our results showed that the dictionary size (number of atoms) and the number of randomly sampled training patches used in training did not have a significant impact on SR performance. While minor improvements could be seen when using more randomly sampled patches or using more dictionary atoms, there were no apparent trends, as all dictionaries produced relatively similar quantitative metrics. We should note that while we varied the number randomly sampled patches, we did not vary the set of training images from which we randomly sampled patches from.

4.2 Extracting 3rd order features from LR image patches

As a straightforward extension of the original ScSR method [7], we investigated whether using 3rd derivative features extracted from LR images would improve SR performance. The original implementation in [7] used 1st and 2nd order derivative features from LR images, which were extracted from training images during the coupled dictionary training and extracted from LR test images during super-resolution reconstruction. We modified the dictionary training and super-resolution code to also extract numerical approximations of 3rd derivatives from LR image patches. In [7], the first derivatives of image patches were extracted using the filters $\mathbf{f}_1 = [-1, 0, 1]$, $\mathbf{f}_2 = \mathbf{f}_1^T$, and second derivatives of image patches were extracted using the filters $\mathbf{f}_3 = [1, 0, -2, 0, 1]$, $\mathbf{f}_4 = \mathbf{f}_3^T$. To extract 3rd derivatives from image patches, we used the filters:

$$\begin{aligned} \mathbf{f}_5 &= \mathbf{f}_3 * \mathbf{f}_1 \\ &= [-1, 0, 3, 0, -3, 0, 1] \\ \mathbf{f}_6 &= \mathbf{f}_5^T. \end{aligned} \tag{12}$$

The motivation behind this extension was that incorporating additional, higher-order features could provide more detail on the content in LR image patches that would be beneficial for establishing the relationship between pairs of LR and HR image patches, leading to improved SR results. The authors of [7] did not mention whether they had investigated such higher-order features in their work, thus we were interested to explore this area in our project.

To assess the effect of using 3rd order features on ScSR, we trained two dictionaries using identical parameters, except that one extracted just 1st and 2nd order features from LR patches, and the other extracted 1st, 2nd and 3rd order features from LR patches. The training for both dictionaries used a patch size of 5, an overlap of 4 pixels, 512 dictionary atoms, $\lambda = 0.15$, and training on 100,000 randomly sampled image patches from the training dataset provided by [7]. We then performed

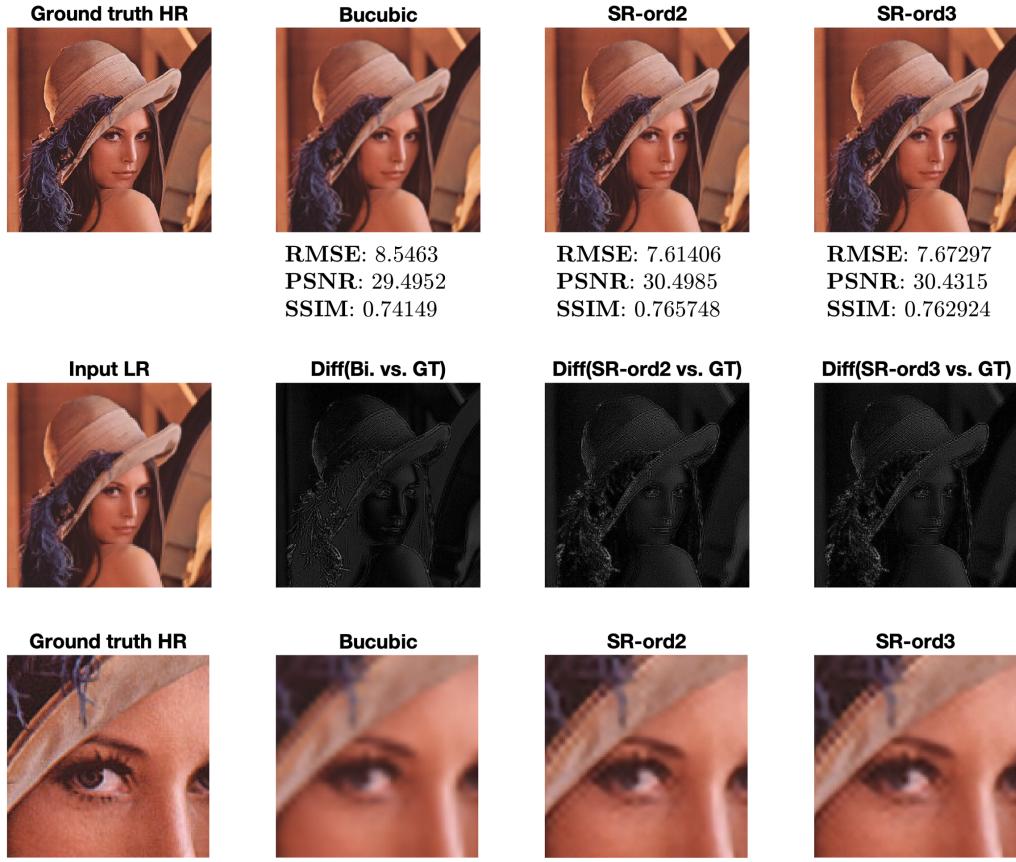


Figure 2: Results from the test using 3rd order features. Top row: Ground truth HR image, Bicubic, SR-ord2 (ScSR [7] 1st and 2nd order features), SR-ord3 (ScSR [7] 1st and 2nd order features). Middle row: Input LR image, and grayscale difference images between the bicubic or SR images and the true HR image. Bottom row: Zoom-ins of the images shown in the top row.

super-resolution reconstructions of the Lena test image (not included in the training data), with a magnification of 4¹

Figure 2 shows results from the proposed 3rd order features test. The input LR image are shown in the middle row, left-most column. The top row shows, from left to right, the ground truth HR image, bicubic interpolation of the LR input, ScSR using just the 1st and 2nd derivative LR features (SR-ord2), and ScSR using 1st, 2nd and 3rd order LR features (SR-ord3), with the RMSE, PSNR, and SSIM metrics given below each image. Compared to bicubic, both ScSR images yielded reduced RMSE and higher PSNR and SSIM. However, SR-ord3's performance (RMSE 7.67, PSNR 30.43, SSIM 0.76) was worse than that of SR-ord2 (RMSE 7.61, PSNR 30.50, SSIM 0.77). These quantitative results, as well as visual inspection of the full and zoom-in SR images (bottom row), indicated that incorporating the third derivative features from LR image patches did not benefit SR reconstructions. It is possible that reaching beyond 2nd order features results in noisy higher-order features that capture more noise than true image patch features. This would explain why our tests using 3rd order features produced worse results than using only 2nd order features. The results from this test motivated our exploration in other areas of feature extraction, which we explain in the next sections.

¹To achieve this, we performed an initial SR with a magnification of 2 on the input LR image, then performed another SR with a magnification of 2 on the initial SR image. This was the approach used in the code available online, thus was the approach we used here.

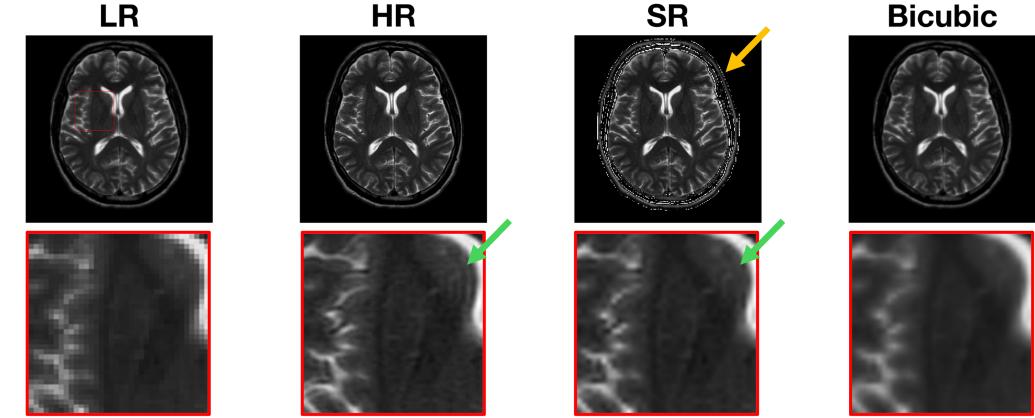


Figure 3: Results from the SR test of the brain MRI image. Top row, left to right: Input LR image, ground truth HR image, ScSR result, and bicubic interpolation result. Bottom row: zoom-ins of the images in the top row.

4.3 Dictionaries trained on natural images generalize to medical images

We performed dictionary training on sets of natural images, as done in [7]. To investigate the generalizability of the ScSR dictionaries, we performed dictionary training and super-resolution as described in [7] to reconstruct a brain MRI image (from our 556 HW). The motivation behind this experiment was to see whether such dictionaries trained on natural images could be effectively applied to perform SR of test medical images that were outside of the domain of the training images. The training for the dictionary applied here used a magnification factor of 3, patch size of 5, an overlap of 4 pixels, 512 dictionary atoms, $\lambda = 0.15$, and training on 100,000 randomly sampled image patches from the training dataset provided by [7]. We then took the brain MRI image and downsampled it by a factor of 3 to obtain a LR input test image. We then performed bicubic interpolation and ScSR super-resolution reconstruction with a magnification of 3 on the LR input image.

Figure 3 shows the results from our SR test of the brain image. Compared to bicubic, the ScSR image provides more detail and less smoothing of the gray and white matter (zoom-in). In addition, the ScSR image appears to have removed some ringing artifacts present in the ground truth HR image (green arrows), which may indicate denoising benefits of ScSR. In particular, it is possible the artifacts in the HR image were smoothed out when downsampling to generate the LR image, and the SR reconstruction produced a HR image devoid of artifacts thanks to the sparse representations in the dictionary. We did not compute quantitative metrics for this test because there were artifacts in ScSR reconstructions on the periphery of the brain, where there is pia matter and skull and little MRI contrast (orange arrow). For many neuroimaging purposes, the quality of MRI images outside of the brain and in the ventricles (the bright white regions in the center of the image) are irrelevant and/or less important than the image quality inside the gray and white matter of the brain.

4.4 Blurring filter

We're using a Gaussian 2D filter to satisfy the reconstruction constraint while updating our Super-Resolution image. The updated equation for this iterative method:

$$\mathbf{X}_{t+1} = \mathbf{X}_t + \mu[H^T S^T (\mathbf{Y} - SH\mathbf{X}_t)]$$

Where H is the blurring filter.

For super-resolution (SR) tasks, the choice of blurring technique depends significantly on the specific characteristics of the input images, the desired output quality, and the computational constraints. However, a few methods stand out for their performance and suitability in super-resolution pipelines, focusing on the goals of enhancing image quality, preserving details, and efficiently handling noise and artifacts.

We conducted experiments using various blurring filters and found that, except for the 2D Gaussian filter, only the Bilateral blurring filter produced satisfactory results. We observed that the Bilateral

Table 2: Experiment results. Image results can be found in figure [4, 5, 6]

Image	Test Name	PSNR	RMSE	SSIM
Jaguar	Bicubic	29.680	0.048	0.870
	Proposed SR	28.426	0.055	0.863
	Reconstruction Constraint	27.398	0.062	0.846
	Bilateral Blur	28.283	0.056	0.838
	Learnt Feature Maps	35.227	0.025	0.961
Worker	Bicubic	26.790	0.067	0.790
	Proposed SR	25.810	0.075	0.780
	Reconstruction Constraint	25.216	0.080	0.765
	Bilateral Blur	25.548	0.077	0.744
	Learnt Feature Maps	31.989	0.037	0.930
Camel	Bicubic	31.347	0.029	0.819
	Proposed SR	31.438	0.029	0.834
	Reconstruction Constraint	31.031	0.030	0.828
	Bilateral Blur	30.827	0.031	0.803
	Learnt Feature Maps	34.503	0.020	0.909

filter produced smoother images with better edges compared to other filters. Additionally, this filter reduced noise in the super-resolution images, resulting in fewer artifacts. Our motivation for experimenting with different blurring filters was to determine their impact on super-resolution. As we modeled the image as $Y = SHX$, where S is the downsampling matrix, H is the blurring matrix, X is the high-resolution image and Y is the low-resolution image, we wanted to assess how different blurring filters would affect the super-resolution. We obtained diverse results and found that the best filter was 2D Gaussian, followed by the Bilateral filter.

4.5 Reconstruction Constraint

We added a reconstruction constraint so that our super-resolution image stays close to the actual image.

$$\mathbf{X}_{t+1} = \mathbf{X}_t + \mu[H^T S^T (\mathbf{Y} - SH\mathbf{X}_t) + c(X - X_0)]$$

We found that enforcing the reconstruction constraint pushed the super-resolution image closer to the upsampled low-resolution image. This contradicts the desired result, where we would like to have a super-resolution image closer to the ground truth, i.e., a high-resolution image. As the effect of reconstruction constraint can be seen from results images [4, 5, 6], where all performance metrics are worse than our learned feature map method as well as proposed method in [7] when compared to the ground truth.

4.6 Learnt feature maps vs hand-crafted features

Filters like Edge [-1 0 1] or Contours [1 0 -2 0 1] are useful for detecting edges and contours in an image. However, they have limitations and are only able to identify specific patterns that they were designed to find. They might miss or misinterpret features that don't match their predefined patterns. A better method is to use learned feature maps to recognize a wide range of patterns, from simple edges to complex textures and objects within an image. In our experiments, we utilized learned convolutional kernels from the ResNet18 model to extract features from low-resolution images. This allowed sparse representation to capture more complex patterns. We chose the ResNet18 model because it is small yet good enough to provide better results while still maintaining the velocity needed to perform experiments.

ResNet18 has a 4-layer CNN block, and we experimented with all four blocks. We found that the layer1 block provided the best performance per computational cost. We passed the low-resolution patches through ResNet18 until layer1 and then used the feature maps for dictionary learning and super-resolution via sparse representation. From figures [7, 8, 9], we can observe that different



Figure 4: **HR** is the ground truth and high-resolution image. **Bicubic** is the bicubic interpolation by upsampling low-resolution image by a factor of 3. **Proposed SR** is the image re-produced using the algorithm proposed in the [7] paper. **Reconstruction** is the image produced by enforcing reconstruction constraint between the low-resolution image and the super-resolution image. **Bilateral Blur** is the image where the blurring filter was changed to the Bilateral blurring filter. **Learnt Feature Maps** is the image where ResNet18 layer1 block is used for feature extraction from low-resolution images

channels of the layer1 block capture different features, which leads to much better results than the proposed method in the original ScSR[7] paper.

5 Conclusion and Future Work

To quantitatively compare our results to other super-resolution methods, we used the results presented in [16], which compared their proposed deep learning SR method to other SR techniques, including the ScSR method from [7]. There, they showed that ScSR [7] yielded improvements in PSNR of between 0.75-1.75 dB compared to bicubic and improvements in SSIM of between 0.01-0.04 compared to bicubic. They reported that their deep learning-based SR method yielded improvements in PSNR of between 3.0-3.5 dB compared to ScSR and improvements in SSIM of between 0.01-0.07 compared to ScSR. Our results (Table 4.3) also showed a similar performance gain for ScSR over



Figure 5: **HR** is the ground truth and high-resolution image. **Bicubic** is the bicubic interpolation by upsampling low-resolution image by a factor of 3. **Proposed SR** is the image re-produced using the algorithm proposed in the [7] paper. **Reconstruction** is the image produced by enforcing reconstruction constraint between the low-resolution image and the super-resolution image. **Bilateral Blur** is the image where the blurring filter was changed to the Bilateral blurring filter. **Learnt Feature Maps** is the image where ResNet18 layer1 block is used for feature extraction from low-resolution images

bicubic. The proposed Learnt Feature Maps method introduced here yielded improvements over ScSR that were of similar order 4 as those shown by the deep learning-based methods introduced in [17].

We were able to achieve better results in our experiments for image super-resolution by using feature maps learned from a deep model. Specifically, for the magnification factor of 3, we obtained at least 15dB higher PSNR than the proposed method. However, we found that changing the blurring filter or the objective resulted in lower PSNR by 1-5dB. In addition, adding the reconstruction constraint did not improve the results and actually gave poorer outcomes than the standard ScSR method. We believe this is because our objective brings the super-resolution image closer to the low-resolution image, which goes against the goal of producing images that are closer to the ground truth. We attribute our success to the use of learned feature maps, which we used for dictionary training and super-resolution via sparse representation. The inclusion of these nuanced feature maps from deep

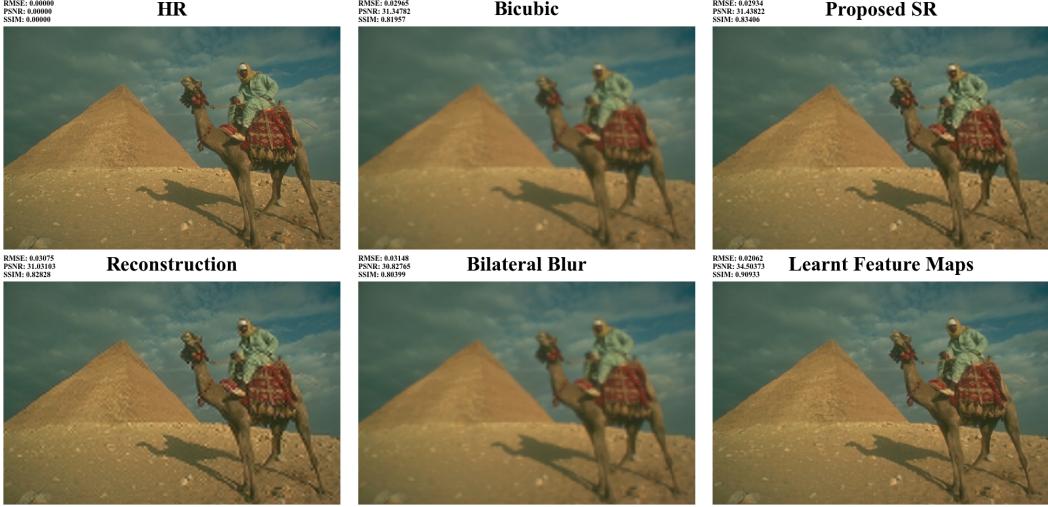


Figure 6: **HR** is the ground truth and high-resolution image. **Bicubic** is the bicubic interpolation by upsampling low-resolution image by a factor of 3. **Proposed SR** is the image re-produced using the algorithm proposed in the [7] paper. **Reconstruction** is the image produced by enforcing reconstruction constraint between the low-resolution image and the super-resolution image. **Bilateral Blur** is the image where the blurring filter was changed to the Bilateral blurring filter. **Learnt Feature Maps** is the image where ResNet18 layer1 block is used for feature extraction from low-resolution images

models allowed us to capture complex patterns, resulting in superior outcomes than those presented in the [7] paper.

We faced some challenges while implementing our approach and code to perform super-resolution. After resolving the issues, we were able to achieve super-resolution results that were comparable to those reported in the literature. However, we did observe some discrepancies in terms of performance and computation time. Additionally, we faced convergence issues while performing super-resolution using the learned feature map. We were able to solve the convergence issues by preconditioning the feature matrix, which resulted in better super-resolution images than the ScSR method [7], which can be seen in [4, 5, 6].

Future work in this direction would continue to explore the incorporation of deep learning into the SR framework to take advantage of the plethora of data and images available, similar to the work introduced in [16]. As demonstrated in their work, the use of deep learning elements can produce improved results over classical SR techniques like ScSR. In terms of classical SR methods, the use of newer dictionary learning methods such as KSVD [1] and sparse coding algorithms can improve the computation time and efficiency of dictionary learning, which is a topic that has been previously explored [15]. As the key idea of this work was based on sparse representations of images, thus methods for more efficient and sparser representations of images would provide benefit to single image SR applications.

References

- [1] M. Aharon, M. Elad, and A. Bruckstein. “\$rm K\$-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation”. In: *IEEE Transactions on Signal Processing* 54.11 (Nov. 2006), pp. 4311–4322. ISSN: 1053-587X. DOI: 10.1109/TSP.2006.881199. URL: <http://ieeexplore.ieee.org/document/1710377/> (visited on 02/23/2024).
- [2] Shengyang Dai et al. “Soft Edge Smoothness Prior for Alpha Channel Super Resolution”. In: *2007 IEEE Conference on Computer Vision and Pattern Recognition*. 2007, pp. 1–8. DOI: 10.1109/CVPR.2007.383028.
- [3] S. Farsiu et al. “Fast and robust multiframe super resolution”. In: *IEEE Transactions on Image Processing* 13.10 (2004), pp. 1327–1344. DOI: 10.1109/TIP.2004.834669.
- [4] William T. Freeman, Egon C. Pasztor, and Owen T. Carmichael. “Learning Low-Level Vision”. en. In: *International Journal of Computer Vision* 40.1 (Oct. 2000), pp. 25–47. ISSN: 1573-1405. DOI: 10.1023/A:1026501619075. URL: <https://doi.org/10.1023/A:1026501619075> (visited on 02/24/2024).
- [5] R.C. Hardie, K.J. Barnard, and E.E. Armstrong. “Joint MAP registration and high-resolution image estimation using a sequence of undersampled images”. In: *IEEE Transactions on Image Processing* 6.12 (1997), pp. 1621–1633. DOI: 10.1109/83.650116.
- [6] Hsieh Hou and H. Andrews. “Cubic splines for image interpolation and digital filtering”. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 26.6 (1978), pp. 508–517. DOI: 10.1109/TASSP.1978.1163154.
- [7] Jianchao Yang et al. “Image Super-Resolution Via Sparse Representation”. In: *IEEE Transactions on Image Processing* 19.11 (Nov. 2010), pp. 2861–2873. ISSN: 1057-7149. DOI: 10.1109/TIP.2010.2050625. URL: <http://ieeexplore.ieee.org/document/5466111/> (visited on 02/23/2024).
- [8] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. “Accurate Image Super-Resolution Using Very Deep Convolutional Networks”. In: 2016, pp. 1646–1654. URL: https://openaccess.thecvf.com/content_cvpr_2016/html/Kim_Accurate_Image_Super-Resolution_CVPR_2016_paper.html (visited on 02/24/2024).
- [9] Xuesong Li et al. “Single image super-resolution via adaptive sparse representation and low-rank constraint”. In: *Journal of Visual Communication and Image Representation* 55 (Aug. 2018), pp. 319–330. ISSN: 1047-3203. DOI: 10.1016/j.jvcir.2018.06.012. URL: <https://www.sciencedirect.com/science/article/pii/S104732031830138X> (visited on 02/24/2024).
- [10] Tomer Peleg and Michael Elad. “A Statistical Prediction Model Based on Sparse Representations for Single Image Super-Resolution”. In: *IEEE Transactions on Image Processing* 23.6 (June 2014), pp. 2569–2582. ISSN: 1941-0042. DOI: 10.1109/TIP.2014.2305844. URL: <https://ieeexplore.ieee.org/document/6739068> (visited on 02/24/2024).
- [11] Jian Sun et al. “Image hallucination with primal sketch priors”. In: *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings. Vol. 2*. 2003, pp. II–729. DOI: 10.1109/CVPR.2003.1211539.
- [12] Radu Timofte, Vincent De Smet, and Luc Van Gool. “Anchored Neighborhood Regression for Fast Example-Based Super-Resolution”. In: 2013, pp. 1920–1927. URL: https://openaccess.thecvf.com/content_iccv_2013/html/Timofte_Anchored_Neighborhood_Regression_2013_ICCV_paper.html (visited on 02/24/2024).
- [13] Jianchao Yang et al. “Image super-resolution as sparse representation of raw image patches”. In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*. ISSN: 1063-6919. June 2008, pp. 1–8. DOI: 10.1109/CVPR.2008.4587647. URL: <https://ieeexplore.ieee.org/abstract/document/4587647> (visited on 02/24/2024).
- [14] Wenming Yang et al. “Deep Learning for Single Image Super-Resolution: A Brief Review”. In: *IEEE Transactions on Multimedia* 21.12 (Dec. 2019), pp. 3106–3121. ISSN: 1941-0077. DOI: 10.1109/TMM.2019.2919431. URL: <https://ieeexplore.ieee.org/abstract/document/8723565> (visited on 02/24/2024).
- [15] Roman Zeyde, Michael Elad, and Matan Protter. “On Single Image Scale-Up Using Sparse-Representations”. en. In: *Curves and Surfaces*. Ed. by Jean-Daniel Boissonnat et al. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2012, pp. 711–730. ISBN: 9783642274138. DOI: 10.1007/978-3-642-27413-8_47.

- [16] Jing Zhang et al. “Image super-resolution reconstruction based on sparse representation and deep learning”. In: *Signal Processing: Image Communication* 87 (Sept. 2020), p. 115925. ISSN: 0923-5965. DOI: 10.1016/j.image.2020.115925. URL: <https://www.sciencedirect.com/science/article/pii/S092359652030117X> (visited on 02/24/2024).
- [17] Yulun Zhang et al. *Image Super-Resolution Using Very Deep Residual Channel Attention Networks*. arXiv:1807.02758 [cs] version: 2. July 2018. DOI: 10.48550/arXiv.1807.02758. URL: <http://arxiv.org/abs/1807.02758> (visited on 02/24/2024).

6 Supplemental Figures

Feature Maps from ResNet18 layer1

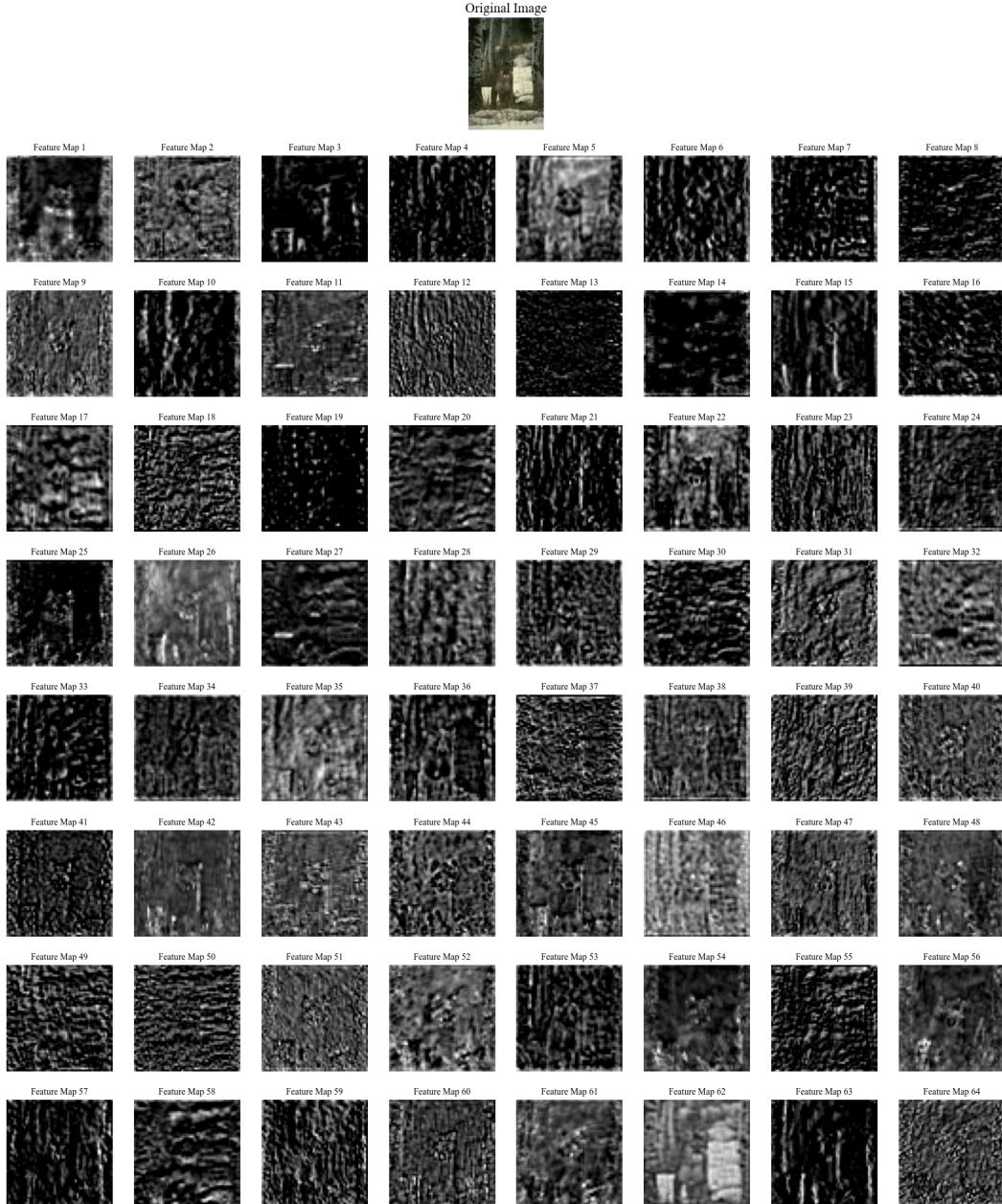


Figure 7: ResNet18 layer1 block output for Jaguar image.

Feature Maps from ResNet18 layer1

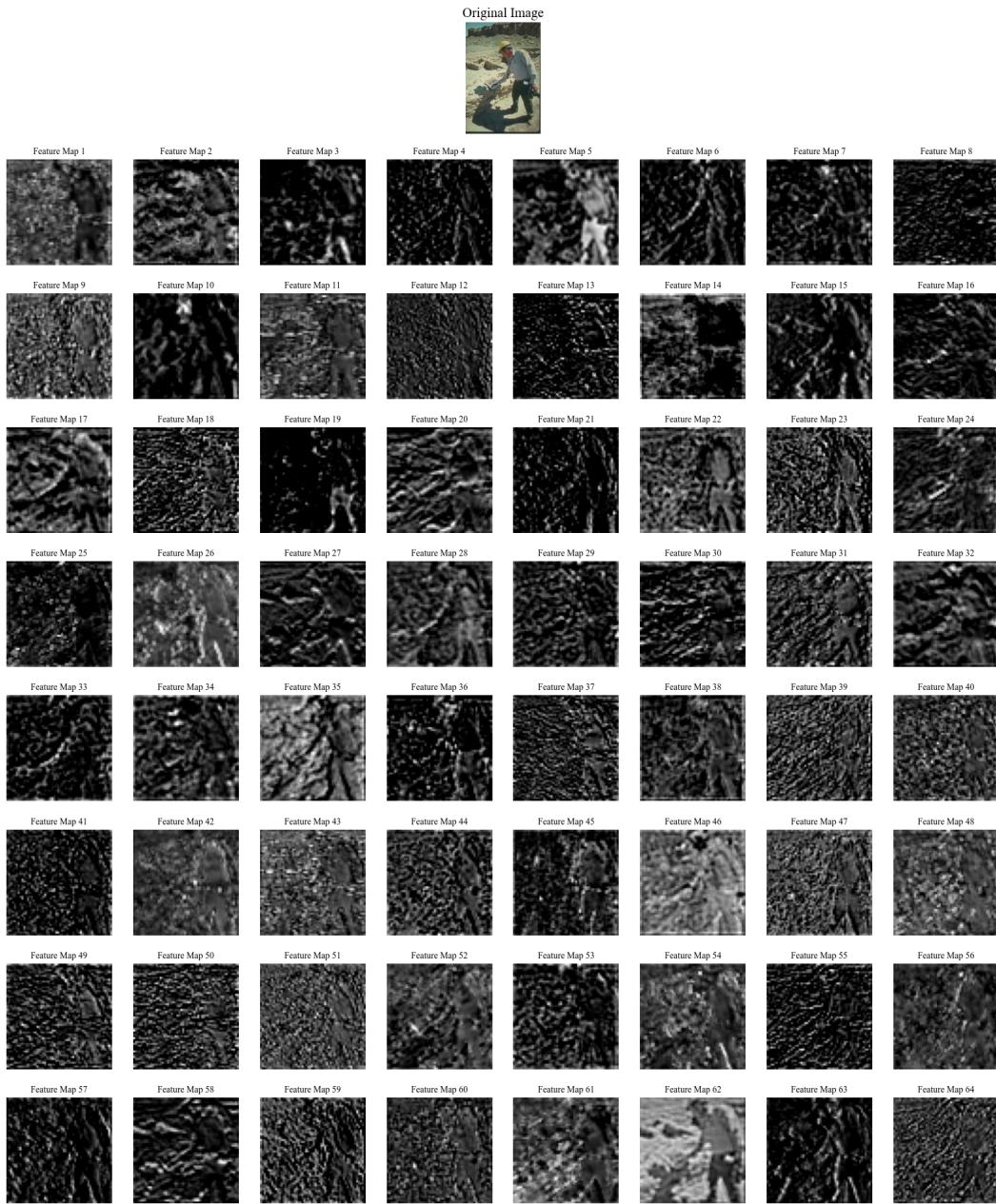


Figure 8: ResNet18 layer1 block output for Construction Worker image.

Feature Maps from ResNet18 layer1

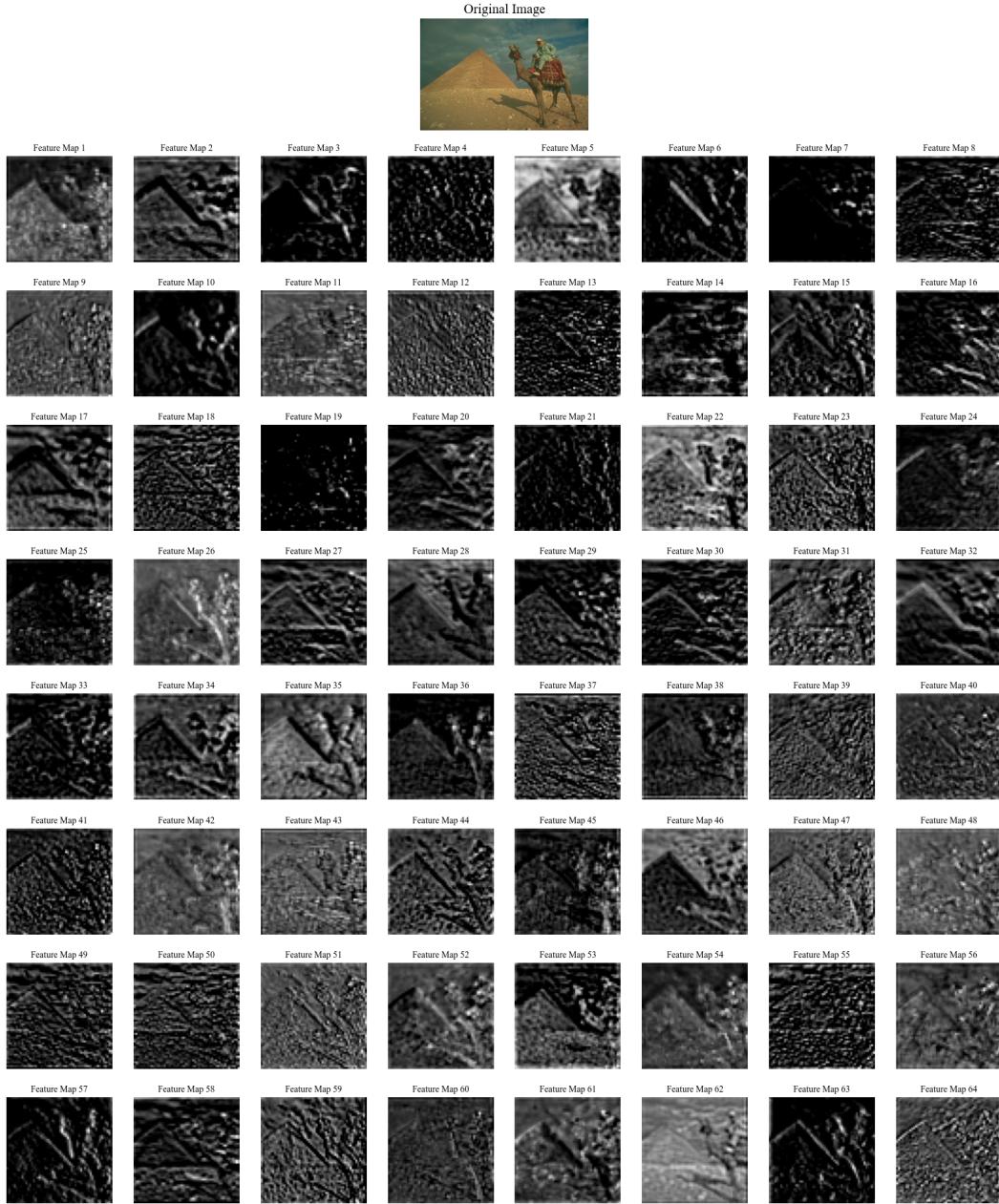


Figure 9: ResNet18 layer1 block output for Camel in a Desert image.