Abstract

The aim of this work is to enhance super–resolution satellite imaging by using data augmentation techniques based on deep learning algorithms. Super–resolution is a technology that enables upscaling images to a higher resolution with more refined details and improved quality. Such image–enhancing techniques are nowadays undergoing rapid development thanks to advancements in deep learning and convolutional neural networks. Deep learning is an approach in which training data plays a key role in the outcome and quality of the solution. Size and quality of the dataset used to train super–resolution networks are crucial to achieve best results. This is especially significant when working with satellite images, which are effortful to acquire in large numbers. Thus, when training a super–resolution network, it may be worth incorporating data augmentation techniques. Data augmentation is a process that intends to enlarge and improve training datasets for machine learning by transforming, multiplying or generating data. This process has been traditionally done using trivial techniques, however this work aims to use deep learning to generate datasets for training super–resolution algorithms. Following chapters provide an overview of modern super–resolution solutions and a proposal of deep learning algorithms to enhance the training datasets. Results of the work are evaluated by testing super–resolution networks which were trained on the datasets created during the project.

Contents

Chapter 1

Introduction

1.1 Super–resolution technology

Super resolution algorithm is a solution that upscales images and improves their quality. Such an algorithm can be treated as a function that takes an image and returns it with a resolution n times larger. Algorithms taken into account in the process usually upscale images two or three times.

It is important to distinguish between super–resolution and traditional upscaling algorithms. The later use interpolation to enlarge images, however they hardly improve the quality of the image. The intent of super–resolution is not only to upscale images, but to improve the quality and detailing. Nowadays such an effect is achieved using machine learning, precisely—deep learning—a technology that utilizes multi–layered neural networks trained with large datasets. Deep learning networks that process image data usually utilize convolutional layers. Such layers contain a number of image filters that are tuned during training. Like all the rest of machine learning algorithms, the deep learning based super–resolution works in a statistical manner. This means that the extra details created during the image enhancement process state an imaginary approximation of image features.

Two kinds of super–resolution algorithms can be outlined: *one–to–one* and *many–to–one*. The first one is the obvious approach, where one low resolution image is translated into high resolution one. The latter is more advanced technique, which utilizes multiple low resolution images of the same scene to produce one high resolution picture. The usual approach is to have multiple low resolution images that are slightly shifted. Data from these multiple images is merged together to produce image of greater quality. This approach can lead to best results in super–resolution, in some scenarios the data fusion can lead to recreation of high resolution details, that are hardly visible in any single low resolution image. It should be noticed that the super–resolution networks trained on domain–specific data often cannot be used to enhance images with different contents. For example, if a network was trained on a dataset with human faces it is likely to perform poorly on satellite images. Network architecture can also be domain–specific, for example utilizing different bands of a multispectral image.

Super–resolution is a technique relevant in the field of satellite imaging and geoscience. The most common reason for image enhancement is for aesthetic reasons. This application is viable in satellite imaging, however super–resolution can lead to other practical advantages. Image enhancing techniques can be used as a preprocessing step in remote sensing pipelines. For this reason super–

resolution can be especially useful when considered in the context of satellite imagery.

1.2 Purpose of data augmentation and available solutions

Deep learning, utilized in the modern super–resolution techniques, requires a lot of data to train successfully. Increase in quality and size of dataset can lead to far better results when training a neural network. This is why data augmentation techniques are often used to improve performance of deep networks. Data augmentation incorporates various transformation to improve, multiply or generate training data. Common techniques to improve image data include: zooming, resizing, shifting, flipping, rotating, distorting, adding noise, modifying colors and exposure. These operations may be application–specific. To give an example, one should beware distorting or flipping data containing with constrained geometry, like road signs. The mentioned augmentation techniques can be considered classic and rather trivial. However more advanced approach can be taken to generate data. It is possible to create deep neural networks to create data augmentation transformations. This approach can be especially useful when a dataset is available, that it is too small to train the desired network. A smaller network can be made and trained on existing small dataset to multiply the data. Then the augmentation network can be used to generate more data for the original model to learn. With deep learning capabilities networks can learn to multiply, transform or even generate data without direct input.

1.3 Aim of the work and motivation

The objective of the work is to create a set of augmentation networks for enhancing super–resolution training data. Subsequent chapters will present considered super–resolution architectures with greater details and propose neural network models for data augmentation.

The nature of super–resolution technology and satellite imagery impose certain ways in which data augmentation should be applied to the training data. Super–resolution is trained using pairs of data—low resolution image with high resolution image (or a set of low resolution images with high resolution image in case of many–to–one network). This requirement renders compiling training sets a challenge, especially in the field of satellite imagery. In such scenario the most common technique is to use resizing algorithms on single image datasets. A set of training pairs can be created by downscaling high–resolution images. In the case of many–to–one networks, single high–resolution image can be multiplied and shifted before shrinking to create more low–resolution images. Such technique may work well, however it infuses the data with information about resizing algorithms. The way high–resolution and low–resolution images relate in such a set depends heavily on the interpolation algorithm (e.g. bicubic, bilinear, nearest, lanczos). Network trained on such datasets will likely learn to invert given interpolation methods. This does not match exactly real–life scenarios, most images are not created using resizing algorithms. Another approach, utilizes pairs of real low–resolution and high–resolution images of the same scene, taken by cameras of different quality. The con of this method is challenging data acquisition process—satellite images are rarely taken in pairs. Such a dataset has to be deliberately made with super–resolution in mind, which makes such data less common. The main idea of the project is to use such a dataset of real–life data to train an augmentation network. Such a network would learn to create low–resolution images for high–resolution, without imprinting resampling algorithms mechanisms into the data. The relation between low and high–resolution images in such an augmented dataset would resemble relation

between same image taken by cameras of different quality. The augmentation neural network can be then used on other satellite image datasets to generate training data pairs for super–resolution. Different data, models and generation techniques can be used to achieve desired results. Possible variations are discussed in the course of this work to improve super–resolution datasets.

Chapter 2

Overview of super–resolution imaging techniques

2.1 Characteristics of satellite imagery

2.2 Encoder–decore mechanism

Encoder–decoder network architecture is a common pattern in generative image processing. It is used both in super–resolution models and in the data augmentation network presented in the latter chapters. Encoder–decoder translates input data into abstract state during encoding, then reconstructs it when decoding. The mid–point of the architecture usually bottlenecks the information containing compressed–like data. Convolutional interpretation of the encoder–decoder is usually used when working with images. During encoding process the depth of input is usually increased and spatial dimensions are shrunken. This is achieved by subsequent usage of convolutional and pooling layers. After encoding the compressed data can undergo some form of processing. For example it can be flatten and dense connected, although this is rarely applied in the super–resolution, because dense layers break the fully convolutional nature of a network (meaning that it can not process images of varying spatial size). The decoding process commonly reconstructs depth dimensions into spatial size by upsampling or transposed convolution. The output may match input dimension, however it is not necessary. In super–resolution it is common to output data of different size, than input. Encoder–decoder architecture is appropriate for image–to–image transformations in machine learning. The inner workings of such an architecture are shown in the figure **??**.

2.3 Measuring quality of image–generating neural networks

Both super–resolution networks and data augmentation networks input and output images. Quantitive evaluation of such networks require comparison of two images—the network output and the ground truth reference image. Images are usually compared using metrics like *mean absolute error*, *mean square error* and *peak signal to noise ratio (PSNR)*. These calculate error between pairs of corresponding pixels in different ways. However these metrics may be insufficient for super–resolution related problems. Calculating pixel–wise differences doesn't resemble the way humans

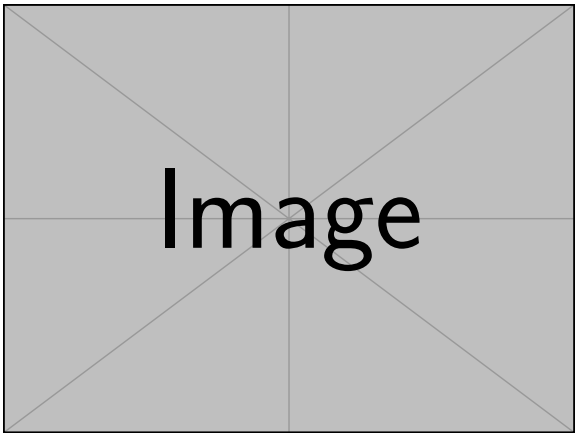

Figure 2.1: Schematic of encoder–decoder mechanism

estimate image quality. Images of varying perceived quality can have same *PSNRs* compared to the reference image.

To measure image similarity in more reliable way *structural similarity index (SSIM)* was introduce. *SSIM* calculates image quality in three components:

- Average *luminance*.
- *Contrast* as standard deviation of pixels.
- *Structure* as luminance difference divided by standard deviation.

However, these values are not calculated globally. Instead *SSIM* values are measured using windows with pixel weights determined by Gaussian distribution. Values of *SSIM* components are combined using a compound formula. Formal description of the *SSIM* metric can be found in. *SSIM* has a value between zero and one, where one means a perfect match between compared images. Having values in constrained a constrained range is another advantage of *SSIM* over metrics like *PSNR*. Advantages of *structural similarity index* render it suitable for super–resolution related image quality evaluation.

2.4 Super–resolution with HighRes–net

2.4.1 Architecture overview

HighRes–net is a super–resolution network based on generative deep learning. It falls into the category of *multi–frame super–resolution (MFSR)* algorithms, which takes *many–to–one* approach to output generation. In *MSFR* systems input is a series of images, taken with a slight shift, perhaps with a small time interval. The input series contains more information, then a single image, as a result of random displacements, noise disturbances and atmospheric conditions. *MSFR* tackles the problem of aliasing in sampled data. Low frequency parts of image, with large geometry and little detail don't differ much between many images. However *MSFR* is crucial when enhancing small

detailing. Upscaling small details from a single images can be non reliable due to aliasing. Applying *MSFR* techniques and multiple low–resolution images fusion leads to de–aliasing information contained in the images.

HighRes–net processing is divided into four subtasks:

1. **Co–registration**, which estimates relative geometric differences between input images. These include divergences, due to shifts, rotations, deformations, etc.)
2. **Fusion**, which combines multiple input images into single one, that is more refined.
3. **Up–sampling**, which upscales low into high–resolution image.
4. **Registration–at–the–loss**, which estimates relative geometric differences of high–resolution prediction nad ground truth, for more representative loss calculation. After calculating shift between super–resolution output and reference image, they are aligned using Lanczos resampling and then loss is measured. The registration and alignement are learned by a model inspired by a *ShiftNet* network architecture.

The unique feature of *HighRes–net* is that all of the above are learned in a single architecture in an end–to–end fashion.

2.4.2 Super–resolution inference process

The key element of *HighRes–net* is achieving *multi–frame super–resolution* by *recursive fusion*. Image generation is done by a neural network organized in an encoder–decoder scheme. The input of the encoder is constructed from a series of low–resolution images. If necessary the input set is padded with zero–valued images, to ensure that the number of low–resolution images in a power of 2, which is required by the network architecture. For each input series a *reference image* is computed using median values of images. Then the reference picture is paired with the input images. Each low–resolution and reference pair is processed through an embedding function. Embedding layer consists of a convolutional layer and two residual blocks with PReLu activations. For input of length n, output of the encoding consists of n images, each convolved with the reference image. In this scheme embedding learns to perform a process called *implicit co–registration*, which is responsible for adjusting geometric differences between images in the input. It is important to notice that the embedding block is a single instance shared between input pairs. The embedding process block diagram is shown in the figure 2.2.

The next step in the *HighRes–net* architecture is *recursive–fusion*. In this process output images are recursively fused together, pair by pair. Fusion operation consists of two steps—co–registration of input pair and the actual fusion. The co–registration of fused images is similar to the co–registration of input–reference pairs. It is done by convolutional layer with PReLu activation and two residual layers. Then the fusion itself is done, again by a combination convolutional layer and PReLu (this part doesn't include local residual layer). The whole co–registration–fusion includes a residual connection. Similarly to the embedding block, the fusion operator has a single instance that is shared for all steps of the recursion. This process is shown in the figure 2.3, where the output of sufficient number of recursive fusion steps is a single–image hidden state.

The last step of super–resolution process is to upscale the image, by decoding the hidden state. This is done with transposed convolutional layer with PReLu activation. The transposition of the output of convolution makes the data grow in spatial dimensions, instead of the usual increase of

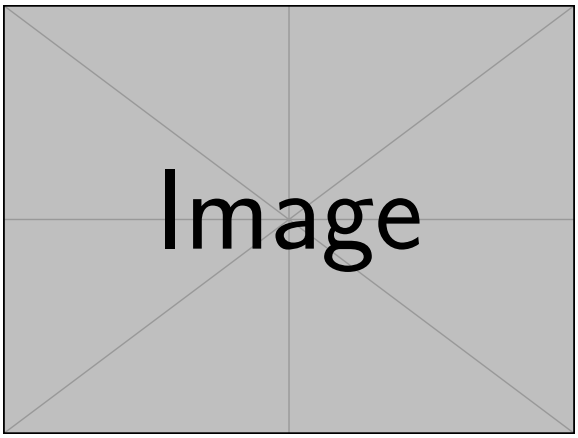

Figure 2.2: Schematic of embedding mechanism in *HighRes–net*

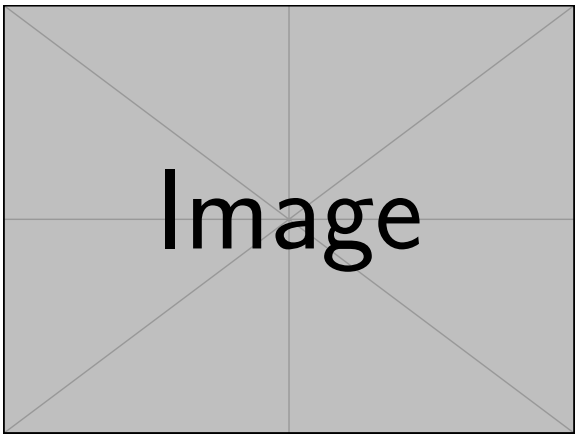

Figure 2.3: Schematic of fusion operation in *HighRes–net*

depth when convolving. The final image is constructed by applying convolution of size one, which doesn't change the size of image.

2.4.3 Registered loss calculation

As stated before registration is important part of super–resolution algorithms. It is especially crucial at loss calculation step, where comparing unaligned pixels often lead to network blurring image, in result of a shift between output and the ground truth. Previous steps of *HighRes–net* include an *implicit co–registration*, where registration mechanisms learned by the network don't have to be necessarily based on shifts, but also other geometric distortions. During evaluation it is desired to register image shifts explicitly, thus the *registration–at–loss* differs from registration performed during encoding and fusion. At the final step the sub–pixel registration is done by the *ShiftNet–Lanczos* network.

Chapter 3

Data augmentation with usage of deep learning

Chapter 4

Super–resolution training and evaluation

Chapter 5

Results

Appendices

List of Figures

List of Tables

List of Listings

Bibliography