Universiteit Leiden

Master Computer Science

AutoSR-RS: an AutoML Approach to Super-Resolution for Remote Sensing

Name:	Julia Wąsala
Student ID:	s1620444
Date:	28-06-2022
Specialisation:	Artificial Intelligence
1st supervisor:	Dr. M. Baratchi
2nd supervisor:	Dr. S.M. Marselis
Collaborators:	Laurens Arp (LIACS)
	Prof.dr. Holger Hoos (LIACS)
	Dr. Nicolas Longépé (ESA)
	Dr. Gurvan Lecuyer (ESA)

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

Super-resolution can be useful to improve the performance of downstream tasks that use remote sensing images. Remote sensing images are highly diverse as compared to other types of images, which gives rise to a need for super-resolution methods that can adapt to diverse datasets. AutoML could provide a solution, with its ability to create methods specifically for a given situation. Moreover, it would be valuable to automate and thus accelerate the process of network design. This could make it easier to create custom end-to-end pipelines for the analysis of remote sensing images. Therefore, we propose AutoSR-RS, an AutoML method that constructs the best neural network for a wide variety of datasets, and uses pre-trained weights from remote sensing datasets for training speedup. The search space is constructed with custom blocks based on state-of-the-art super-resolution methods. We evaluated our method by comparing the PSNR and SSIM to those of two state-of-the-art baselines and a simpler AutoML approach. The comparisons were performed on 3 remote sensing datasets in addition to a new dataset we propose, SENT-NICFI. SENT-NICFI adds to the small collection of multi-sensor single image super-resolution datasets. Our results show that AutoSR-RS achieves significantly higher performance than the baselines on 1 dataset, and comes second on the remaining 3 datasets. Statistical testing ranks it as the first, making it the most general of the methods considered. Analysis of the search space shows that there is a split between the datasets in terms of parameter values in the final models found by AutoSR-RS, indicating that future research on the relationship between these choices and dataset characteristics could help to further improve the search space. New model blocks and pretrained weights can be added easily, making it possible to adapt AutoSR-RS to future developments in the field. AutoSR-RS outperformed the simpler AutoML method every time, emphasizing the merit of a more customized search space for automated super-resolution of remote sensing images.

Contents

Chapter 1

Introduction

Our Earth is surrounded by satellites that serve different purposes like communication and intelligence. Scientific programs over the years have launched more and more satellites that serve to study our planet, for example by making measurements of the atmosphere or by taking optical images of the surface of the planet. The quality of those measurements is always increasing, which also creates new possibilities for using the data. Some examples include deforestation monitoring, where the tree cover of vital environments like the Amazon is constantly monitored, or satellites that measure the carbon-monoxide concentration of the gases emitted by gas flares to determine whether the combustion is complete.

Though the spatial resolution of satellite images is increasing with technological advancements, its further development is constrained by factors like bandwidth and hardware limitations and other factors like the tradeoff between swath width, and spatial and temporal resolution. The images can also suffer from degradations like noise and blurring.

Nonetheless, there is a need for higher resolution images for tasks like object detection or tree cover mapping. The spatial resolution of optical satellites is in the range of meters (E.g., 10m for Sentinel-2, 30m for Landsat-8, 4.77m for PlanetScope), while aerial images are in the range of centimetres, sometimes as high as 2.5cm (e.g, Houston dataset [4]). Increasing the resolution by a factor of 2 can make the difference in which types of buildings or plants you can discern. For example, Goldblatt et al. [14] have found that higher resolution images yield better results for ecosystem assessment for regions without very dense tree cover.

The resolution of images can be increased by using super-resolution techniques. These are neural networks that are used to increase the number of pixels in an image, effectively decreasing the size of each pixel. The effects of super-resolution on object detection have been studied by Shermeyer and Van Etten [41], who have found that in remote sensing images, object detection performance decreases with lower resolutions, and that super-resolution can offer a small increase in accuracy. Super-resolution is used as a preprocessing step for the remote sensing object detection method proposed by Zou et al. [56], which is competitive or superior to state-of-the-art methods. In a similar approach, Haris et al. [15] propose a framework containing a task-driven super-resolution sub-network which is shown to improve object detection results on low-resolution datasets.

Super-resolution methods are crafted by design experts. Though reliable, the careful curation of neural network architectures by experts is a time-consuming process. Since super-resolution is only a single part of possible remote sensing pipelines, manually designing these models can form a bottleneck to creating end-to-end pipelines for analysing remote sensing imagery. It is possible to use pre-trained super-resolution models, but due to the high visual diversity of remote sensing imagery and differences between individual datasets, this might not achieve the desired performance.

Based on state-of-the-art methods, we can automate the process of constructing super-resolution models for a wide array of datasets. Singular super-resolution models might have high performance for a subset of available datasets, and instead of manually selecting models for a given task. An AutoML approach could formalize this process.

Automated machine learning models have been used for object detection in remote sensing images ([33], [23], [37], [31], [34]), but to the best of our knowledge, no attempt at automating super-resolution has been done yet.

We propose an automated machine learning approach to single-image super-resolution *AutoSR-RS*, which aims to automatically find the best model as well as tune the hyperparameters. The method is tailored towards remote sensing images by using pre-trained weights from remote sensing datasets. Furthermore, it could easily be extended by including new state-of-the-art methods or additional sets of pre-trained weights.

The challenges of this approach are to find the computer vision and super-resolution paradigms that work well specifically for remote sensing images. Moreover, the approach has to work with different remote sensing datasets: images consisting of different optical bands and images taken of various regions can be very different, and might each need a different approach. An automated machine learning approach could be better able to address these problems than a single hand-crafted model could.

An additional challenge is that of the benchmarks: many super-resolution methods are evaluated on synthetic datasets, which are obtained by downsampling high-resolution remote sensing imagery in order to obtain high-resolution – low-resolution pairs of the same region. The downsampling procedure is often much more simple than how the loss of resolution occurs when capturing real-life images, and thus if a model is trained on such a dataset, the resulting performance might be overestimated compared to using real data [24]. Therefore we also introduce a new dataset from matching Sentinel-2 [9] and Planet [1] [42] images, called *SENT-NICFI*. To the best of our knowledge, at the time of compilation only one other such dataset existed: OLI2MSI [47].

Our contributions are the following:

- We propose a new automated machine learning approach for super-resolution of remote sensing images, called AutoSR-RS. The method is based on state-of-the-art super-resolution methods and uses pre-trained weights from remote sensing datasets for training speedup.

- We evaluate our methods in terms of peak signal-to-noise ratio and structural similarity index on 4 different remote sensing datasets.

- We compare our methods to state-of-the-art super-resolution methods and a simpler automated machine learning approach. In this comparison, AutoSR-RS beats the baselines on 1 of the 4 datasets and comes second on the remaining 3. However, pairwise statistical tests of the results per dataset have ranked it highest, indicating that the other methods yield more inconsistent results. This makes AutoSR-RS the most generally applicable method out of the ones considered.

- We propose SENT-NICFI, a novel super-resolution dataset consisting of paired images of Sentinel-2 and Planet. This dataset adds to the small set of multi-sensor super-resolution datasets for remote sensing, which are not constructed by simply downsampling a set of high-resolution images.

- We make available to the public the source code of AutoSR-RS and instructions on construction of SENT-NICFI, allowing both to be used for future super-resolution research for remote sensing.

The remainder of this thesis is organized as follows: first, background information on remote sensing, super-resolution and AutoML will be given in Chapter 2. This is followed by a discussion of the related work in Chapter 3. Next, the datasets used are described in Chapter 4, after which Chapter 5 outlines the methods used. Chapter 6 defines the experiments conducted, the results of which are presented in Chapter 7. The work is concluded with Chapter 8 which covers the conclusion and future work.

Chapter 2

Background Information

In this chapter, we will provide the reader with background information on the topics of remote sensing, super-resolution and automated machine learning.

2.1 Remote Sensing Images

Remote sensing (RS) is the measurement of the physical characteristics of an area from a distance. Different areas can be observed, including other planets. For this work, the scope will be limited to the Earth in which case it is also referred to as Earth Observation (EO).

There are many different techniques for obtaining remote sensing measurements. For this research, we will focus on simultaneous multi-spectral platforms which can take images of an area in multiple wavelengths at once. This results in multi-band imagery. The materials and particles in and on the atmosphere and surface of the Earth transmit, absorb and reflect different types of light. These types of light, both visible and invisible can be placed on the electromagnetic spectrum, as shown in Figure 2.1. Different instruments are required to measure different ranges of wavelengths (i.e., wavelength bands). The wavelength bands used depend on the task, because each band provides different information. For example, near-infrared (NIR) is used for vegetation mapping due to its sensitivity to vegetation type [13].

The resolution of an image can be quantified in different ways. In remote sensing images, the spatial resolution indicates the size of the area described by each pixel, the smaller the area, the higher the spatial resolution.

Spectral resolution relates to the detail in an image in terms of the spectral characteristics. Narrower bands (i.e., bands with a smaller wavelength range) have a higher spectral resolution, allowing to differentiate more between different wavelengths than with wider bands.

In addition to spatial and spectral resolution, remote sensing images can also be described by their temporal and radiometric resolution. The temporal resolution refers to the amount of time between subsequent images of the same region. This is different for each satellite and depends on the orbit. Radiometric resolution refers to the number of bits per pixel, and thus how detailed the pixel information can be.

Multi-spectral images can be different from "regular" or natural images that are taken of subjects like cars, streets and people. Natural images are commonly composed of RGB bands, which represent the human visual spectrum. However, satellite images can record wavelengths which are not visible to humans, and visual light can be divided into bands different from RGB bands.

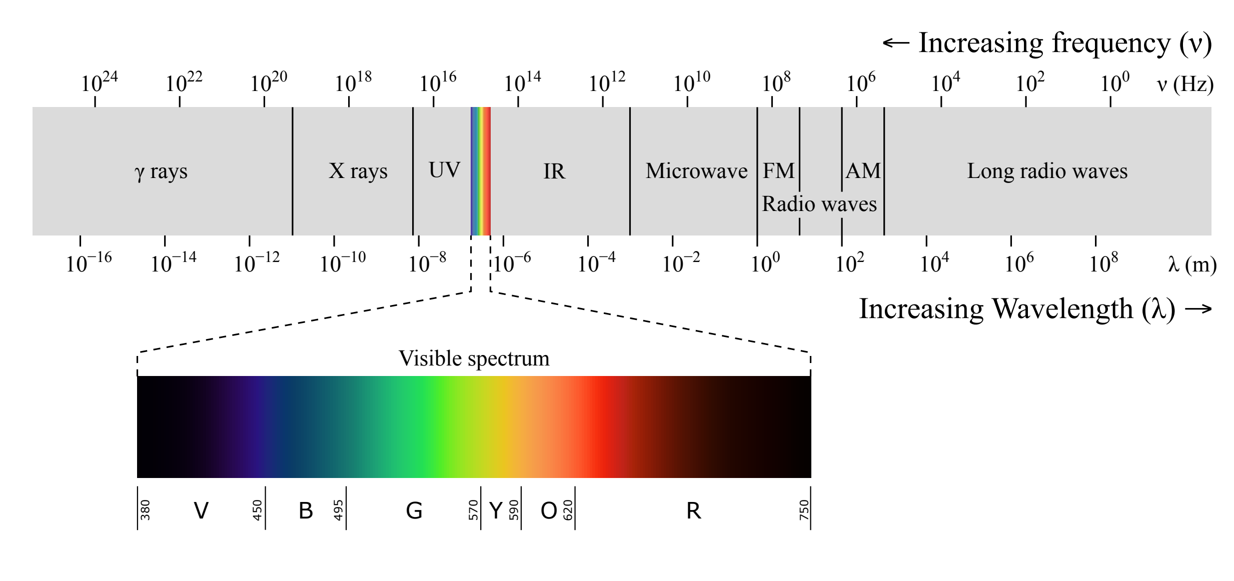

Figure 2.1: The electromagnetic spectrum [39]

2.2 Super-resolution for remote sensing

Super-resolution is the task of increasing the resolution of an image. It is a low-level computer vision task that is usually not used on its own, but in combination with a downstream task. Super-resolution could potentially increase the accuracy of tasks like classification or regression. For remote sensing specifically, super-resolution can support tasks like object detection or change detection, especially in cases where the features that have to be observed are very small on the original images. For example, the resolution of Sentinel-2 is 10 m per pixel, at that resolution one cannot always distinguish features like single trees. However, when detecting deforestation in a region with sparse vegetation, small patches of trees might matter much more than when detecting deforestation in the Amazon. Super-resolving the image might not solve the problem entirely, but having a resolution of 5m or even 2.5m will allow for capturing more information about smaller patches of vegetation. Super-resolution methods can be split into two categories:

- *Single image super-resolution* (SISR): one-to-one mapping of a single image to a higher resolution image.

- *Multi-image super-resolution* (MISR): many-to-one mapping where multiple images are fused into a higher resolution image. An example scenario is when a satellite passes over the same region at regular intervals. For instance, the revisit cycle of Sentinel-2 takes 5 days [9]. In other cases, an instrument might make multiple images of the same location at once, such as PROBA-V [10].

2.2.1 Datasets for super-resolution

The data necessary for supervised training of super-resolution models consists of pairs of low-resolution and high-resolution images of the same region (or for the case of MISR: multiple low-resolution images for 1 high-resolution image) and ideally also very close in time because of changing weather conditions and human activity. The latter would in theory be easier to achieve for SISR than for MISR, since fewer images are needed. Two types of super-resolution datasets exist:

- *Synthetic data*: low resolution – high-resolution pairs are created by downsampling a high-resolution image. The problem can then be described as learning the degradation function, to recover the high-resolution image. However, the performance of such methods can be worse on real-world data, because the degradation of images in real life cannot always be described by a simple degradation function [24]. Training and evaluating super-resolution methods exclusively with synthetic data has the disadvantage that the performance of these models can be overestimated [24]. There also does not appear to be a commonly used protocol for creating synthetic data for MISR.

- *Real data*: real-world datasets consist of matched low resolution – high-resolution pairs, with the images often taken by different instruments. Training and evaluating a model on this type of data gives a more realistic picture of how well the model would perform in real-life applications. One of the downsides is that it is not always possible to find matching image pairs of sufficient regions to train a model on, considering that the time at which the low-resolution and high-resolution images were taken should be as close as possible. Other challenges include differences in coverage between sensors, differences in preprocessing between sensors. The latter requires resampling at least one of the images, with information loss as a result.

Few real datasets exist due to the challenges involved in creating them. Additional considerations have to be taken into account when obtaining real data for MISR methods:

- *Theoretical advantages*: The images are often slightly misaligned, which could give more information about the region because the pixels of the different images all have slightly different coverage. Theoretically, multiple images of the same scene would provide more information about a scene than a single image would. Thus, it could inform the super-resolution process.

- *Challenges*: the time in between subsequent fly-overs of a satellite may be too large depending on the task the remote sensing images are used for. For example, in monitoring vegetation, the change of the seasons may have a large effect on the appearance of a region. This issue makes the method less general, since not every satellite will pass over the same region multiple times or even often enough. Other challenges include changing cloud coverage and atmospheric conditions, as well as possible deviations in image brightness. All of these factors combined can lead to unexpected behaviour at testing time.

Because of the above factors and the low availability of datasets related to the MISR problem, we shall be addressing the SISR problem in this work. We will use both synthetic datasets and an existing real dataset, as well as a new, real dataset constructed by us. The goal is to evaluate both baselines and proposed methods on a variety of datasets.

2.3 Automated Machine Learning & Neural Architecture Search

The field of automated machine learning studies the automatic optimization and configuration of machine learning models and pipelines. Design choices that have to be made for a typical deep learning pipeline are data preprocessing, data augmentation, architecture design and training parameter optimization. Extensive domain knowledge is required to manually design deep learning architectures and hyperparameters. Smart methods are needed for NAS, because each new layer expands the search tree further with new parameters. The design of such methods is not trivial, but important for making deep learning more accessible to non-experts.

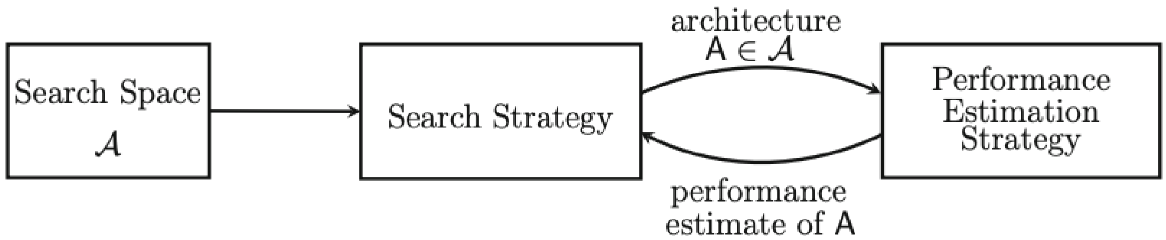

Figure 2.2: Schematic of NAS. Architectures are sampled from search space $\mathcal{A}$ by a search strategy. The performance of sampled architectures is used to select new ones. [18]

An AutoML system consists of three parts. The first is the *search space*, the set of design choices to be searched. The second part is the *search strategy* used to explore the search space. Finally, an *evaluation metric* is required to assess the performance.

There are three problems addressed by AutoML: algorithm selection, hyperparameter optimization and the combined algorithm selection and hyperparameter optimization problem (CASH) (Thornton et al. [43]). In this work, we will be addressing joint neural architecture search and hyperparameter optimization, which is a special case of CASH. Both architectural and training parameters are optimised.

Neural architecture search (NAS) is the development of neural network architectures by iteratively (i) selecting an architecture from the search space, (ii) estimating the architecture performance and (iii) saving the estimated performance for reference to the search strategy. The results of previously explored architectures help inform the selection of the next architecture. NAS search spaces can be divided into two categories: macro-level, micro-level and a combination of the two. Macro-level search optimizes the whole structure of the network, on a higher level. AutoKeras [20] is an example of a macro-level AutoML system: the search space consists of hyperparameters like the number of layers, number of filters, kernel sizes, etc.
NAS typically has a high computational complexity, since each candidate architecture is trained such that it can be evaluated. For macro search specifically, there is still a large parameter space to be explored. Micro-level strategies fix the high-level architecture and focus on optimising smaller, repeated motifs. An example is NASNet [55].

Chapter 3

Related Work

This section will discuss deep learning approaches to single image super-resolution for remote sensing images as well as examples of automated machine learning in remote sensing research.

3.1 Single-image super-resolution

This section will cover super-resolution methods, categorized by the architecture type.

3.1.1 CNN-based models

Convolutional neural networks (CNNs) are suitable for many tasks involving images. Thus, it is not surprising that many super-resolution techniques are CNN-based. One of the earlier approaches on which many more recent ones are based, is SRCNN Dong et al. [7], which was applied to remote sensing by Ducournau and Fablet [8]. It is a simple CNN with only a few convolutional layers. SRCNN showed improvement over previous CNN architectures, because it was able to capture features at different scales [28]. Very Deep Super-Resolution (VDSR) [22] consists of many more convolutional layers than SRCNN and has a skip connection. VDSR is shown to have a significant advantage compared to SRCNN for natural images, but little improvement with regards to the bicubic interpolation baseline [45]. Enhanced Deep Super-Resolution (EDSR) [27] outperforms VDSR with residual blocks replacing vanilla convolutional layers, adding more skip connections at different levels. Residual channel networks (RCAN), proposed by Zhang et al. [51], introduces a channel attention mechanism weighting the image features as well as the concept of residual-in-residuals. The authors indicate that it has been shown that upscaling the image at the end reduces computational complexity and achieves higher performance, rather than at the beginning which has been done with earlier approaches (e.g. SRCNN [7]).

Finally, WDSR [11] outperforms EDSR with similar architecture to EDSR but with wider activation and weight normalization [40]. Weight normalization is shown to achieve higher performance than batch normalization or no normalization. The key idea of WDSR as compared with EDSR [27] is that a wider activation, which means expanding the image features before an activation layer, allows for preserving more information of the original image. Lim et al. [27] have found that batch normalization impacts the model performance negatively, and this is also confirmed by Fan et al. [11].

3.1.2 GAN-based models

In addition to CNN-based models, several super-resolution approaches are based on generative adversarial networks. One of the first of such approaches is SRGAN [25]: the discriminator is tasked to differentiate between the original high-resolution image and the super-resolved image, pushing the generator to approximate the original high-resolution image. An improvement of SRGAN specifically for remote sensing is proposed by Ma et al. [29]. This approach addresses the frequency domain rather than the spatial domain and combines the wavelet transform and ResNet [16] to super-resolve low-resolution input. Edge-enhanced GAN (EEGAN) proposed by Jiang et al. [19] shows even better performance. This super-resolution approach targeted for remote sensing enhances the edges of the super-resolved image, which can offer advantages for remote sensing use cases such as building detection.

3.2 Automated Machine Learning in Remote Sensing

Two AutoML frameworks for deep learning include AutoKeras [20] and Auto-PyTorch [54], these frameworks jointly perform neural architecture search and hyperparameter tuning. Both frameworks have been used for AutoML in remote sensing. Palacios Salinas et al. [33] have proposed a NAS system optimized for classifying remote sensing images by using blocks pre-trained on remote sensing datasets, implemented in AutoKeras. Other works also base their methods on the same AutoKeras image classifier but with fewer modifications, including Koh et al. [23] who classify plants based on remote sensing imagery using the classifier, but with blocks pre-trained on ImageNet [5] instead of remote sensing datasets. Renza et al. [37] classify landslides and Nguyen et al. [31] forecast aerosol concentrations in the atmosphere, both using the AutoKeras image classifier.

An example of research conducted using Auto-Pytorch, is the work by Polonskaia et al. [34] proposing an automated evolutionary approach to NAS for designing CNNs for the task of object recognition in remote sensing images.

All these examples are of classification of satellite images. Neither AutoKeras nor Auto-Pytorch have functionality for image-to-image approaches like super-resolution: there are many different blocks, but they lack an image output module which is necessary for super-resolution. The two frameworks also address the search space differently. AutoKeras's approach is to morph neural networks to varying degrees: adding or removing layers, but also switching out parts of the architecture for an alternative version. The architecture can be quite complicated. In contrast, Auto-PyTorch only offers support for simpler architectures. We decided to use AutoKeras based on the options offered by both frameworks; the level of automation and that more remote sensing examples were found for AutoKeras than for Auto-Pytorch. Generally, AutoKeras is a macro-level search strategy, but the possibility to define a custom high-level architecture and blocks makes it possible to perform micro-search as well and even combine macro- and micro-search.

3.3 Relevance of this work

We construct a neural architecture search specifically for the task of super-resolution of remote sensing data, which, to the best of our knowledge, has not been researched yet. Previous work shows that AutoML can be applied successfully to the classification of remote sensing images, indicating AutoML might provide similar gains to other remote sensing tasks like super-resolution. The aim is to achieve higher performance than the state-of-the-art to show that automated machine learning can successfully be applied to the problem of super-resolution. The proposed method is based on

RCAN and WDSR, which were both based on EDSR but developed independently from each other. These models are consistently cited as the state-of-the-art in SISR works and are among the best performing models listed. Since in many cases the results presented in papers are obtained by training and evaluating models only once, the performance may differ. However, with the available information, these methods present themselves as suitable work to build upon.

Chapter 4

Data

This chapter is organized by the type of datasets used for super-resolution: synthetic and real data. Both sections describe the data sources and procedures used to construct the datasets used in the research.

4.1 Synthetic Data

A set of high-resolution images is needed for the synthetic datasets. Table 4.1 lists the satellite image datasets used for creating the synthetic data. Originally, these were image classification datasets, but for this research, only the images and not the labels are needed. To avoid confusion, these datasets will be referred to as the data sources and the final result containing the pairs of low-resolution and high-resolution images will be called datasets.
The datasets are made available as TensorFlow datasets at `https://universiteitleiden.data.surfsara.nl/index.php/s/MBZs6OyPOiELRNZ`.

4.1.1 Data sources

- **UC Merced** [50] is a dataset for land use image classification. It contains 21 different classes including agricultural, buildings, rivers and many other types of terrains in the United States.
- **So2Sat** [53] is a dataset with images of 42 different cities across different continents and regions. The RGB-subset of So2Sat consists of Sentinel-2 images.
- **Cerrado-Savanna** [32] consists of images of the Serra do Cipó region in Brazil. It has a wide variety in vegetation and high variation between classes, though it only covers vegetation in a single region.

Table 4.1: Overview of HR datasets. The resolution is given in metres [m] and the image size in pixels [px]

Dataset	Source	# Images	Resolution [m]	HR size [px]
UC Merced [50]	USGS (aerial)	590k	0.3	256×256
So2Sat [53]	Sentinel-2	376k	10	32×32
Cerrado-Savanna [32]	RapidEye	27k	5	64×64

Table 4.2: Overview of LR-HR pair datasets. The resolution is given in metres [m] and the image sizes in pixels [p]

Dataset	Satellites	# Images	Resolutions [m]	LR size [px]	HR size [px]
OLI2MSI [47]	Landsat & Sentinel-2	10.65k	30 & 10	160×160	480×480
SENT-NICFI	Sentinel-2 & Planetscope	2.2k	10 & 5	100×100	200×200

4.1.2 Creating synthetic datasets

For each data source described in Table 4.1, all the images are gathered and the images with aberrant sizes are removed. Then, the low-resolution images are generated by downsampling all the images in the data source with a factor of 2 with a bicubic kernel. This simple downsampling procedure does not always accurately reflect the differences between a high- and a low-resolution image in real life. These differences often occur due to that in practice, different sensors are used to obtain images of different resolutions. These sensors have different characteristics. The risk of downsampling images like this, is that the situation will be over-simplified. Real datasets can be used to avoid this problem.

4.2 Real data

This section describes the real datasets used. Real datasets are harder to obtain than synthetic datasets, due to the limited availability of freely accessible satellite data with different resolutions. We will discuss two datasets, as well as the strengths and limitations. The first dataset is OLI2MSI [47], which is an existing dataset. Secondly, we will present the SENT-NICFI dataset which has been compiled from Sentinel-2 and Planetscope data. Details of these datasets are shown in Table 4.2

4.2.1 OLI2MSI

OLI2MSI is proposed by Wang et al. [47] and consists of low-resolution images taken by Landsat-8 (courtesy of the U.S. Geological Survey) and Sentinel-2 of a region in the southwest of China. Images with the same production level are taken, with less than an hour of time difference. Final images are selected based on the cloud cover. The data is available for download on the GitHub page of the authors [1].
This dataset is compiled into a TensorFlow dataset, like the synthetic data.

4.2.2 SENT-NICFI

SENT-NICFI has been constructed using images from Sentinel-2 and Planetscope, taken in June of 2021. The Planetscope images are part of the NICFI program. The NICFI dataset is part of Norway's International Climate and Forests Initiative Imagery Program. It contains images of tropical forests used for tasks like monitoring deforestation and biodiversity. It contains images of forests in different countries around the equator, covering an area of about 45 million square kilometres. In addition, images of countries with no or limited forest cover outside of this region are included.
A similar dataset for MISR has been constructed by Razzak et al. [36], who combine Sentinel-2 images and Planetscope images from the SpaceNet-7 challenge [46].
The NICFI dataset consists of basemaps which are averaged over either a month or biannually.

[1] `https://github.com/wjwjww/OLI2MSI`

Therefore, it is not possible to select LR images from a narrow window as with OLI2MSI. Each basemap consists of so-called quads, which are raster files with a resolution of 4.77 m per pixel. The visual basemaps are selected (the other option is normalised basemaps).

The dataset is consists of 5 scenes from each of the following ecosystems: cities, desert, forest, savanna, agriculture and miscellaneous (not falling in any of the previous categories). The images are split into smaller 200 by 200 pixel training images, yielding 12000 training pairs.

Under the licencing limitations regarding NICFI, we are unable to make the SENT-NICFI dataset publicly available. However, the dataset can easily be reconstructed by using the code we have provided. Detailed descriptions on how the dataset was obtained and how it can be reconstructed, are provided in Appendix A.

Chapter 5

Methods

This chapter will describe the proposed methods used for the super-resolution task. First, we define the AutoML search space followed by a description of the proposed model for super-resolution of remote sensing images, AutoSR-RS.

5.1 Search Space

AutoML methods explore a search space to find optimal hyperparameter settings. The search space consists of tunable hyperparameters and the range these parameters can take. In the context of neural architecture search, examples of hyperparameters are the number of layers, the layer type, and the number of nodes in a layer, but also other factors like whether to use dropout or not, or which optimizer to use. Ideally, one would have an infinitely large search space that doesn't exclude any feasible solutions and an algorithm that can find the best hyperparameters for the problem in a finite amount of time. In practice, this is not feasible. Having a very large search space increases the probability that the AutoML algorithm will need many iterations to find a good set of hyperparameters. To reach good results faster, it is important to constrain the search space with the help of domain knowledge. For instance, by excluding settings that research has found to be detrimental to the model performance, and including settings that are shown to be beneficial to model performance.

In the next section, we will describe the proposed AutoSR-RS model and define the search space in more detail.

5.2 AutoSR-RS architecture

The architecture of AutoSR-RS can be abstracted in terms of smaller units that make up the complete model. We will refer to these units as blocks. A block is a module of the neural network, it can be a single layer or even a complete model like ResNet [16].

During each trial, a different model is built from the available blocks, and these blocks can also be mutated. These mutations can include changing existing layers, e.g., changing the kernel size, or adding/removing layers or adding/removing blocks of layers like a set of convolutional layers or an LSTM block [17].

In the case of AutoSR-RS, the available blocks are implemented based on super-resolution methods. AutoKeras [20] (see Sections 2.3 & 3.2), the library AutoSR-RS is implemented in, has a wide array of available blocks, but none of them are specifically suited for super-resolution. This is also true for other NAS frameworks like Auto-Pytorch [54]. For instance, AutoKeras offers a ResNet [16]

Table 5.1: Parameters and possible values in the search space of AutoSR-RS. There are two different ranges for the numbers of residual blocks: these numbers are based on the numbers of residual blocks in the original RCAN and WDSR methods.

Parameter	Values	
	RCAN	WDSR
Method	(RCAN,WDSR)	
pre-trained weights	(Cerrado, UC Merced, So2Sat, OLI2MSI)[1]	
Number of residual blocks	$[10, 20]$	$[4, 40]$

block. ResNet is used for classification, but Liu et al. [28] have shown that it does not perform well for super-resolution when adapted directly. Additionally, there often are options to use pre-trained weights, but these pre-trained weights are usually from more general datasets like ImageNet [5]. To address this problem, we propose AutoSR-RS, an automated machine learning model with a search space tailored to super-resolution for remote sensing by using pre-trained weights from remote sensing datasets. One of the challenges is to choose the size of the search space such that it is large enough to be able to offer good solutions, but not so large that it would take a prohibitively long time to find them. Moreover, some methods from super-resolution of natural images may not be well-suited to remote sensing imagery, and vice versa. The following section will describe the composition of the search space.

5.2.1 Search space parameters

We introduce a search space specifically crafted for super-resolution of remote sensing images. AutoSR-RS is based on state-of-the-art SISR methods: RCAN [51] and WDSR [11]. The latter is the winner of the NTIRE 2018 Challenge on Single Image Super-resolution on all three realistic tracks. Both of these methods achieve high performance on super-resolution tasks with natural images. These methods are extensions of earlier work like EDSR [27], and reflect the shift towards using increasingly deep residual networks for super-resolution. We selected these two methods as the basis of AutoSR-RS, because of the representative nature of these methods in the domain of non-GAN SISR methods. With these two methods as the base of AutoSR-RS, we aim to test the merit of the approach and whether this is an avenue worth exploring by adding more methods in the future.
A schematic of AutoSR-RS is shown in Figure 5.1: the global structure is that the search algorithm can choose between WDSR and RCAN as a base, then can choose the number of residual blocks and thus the depth, and finally choose from a set of pre-trained weights from different remote sensing datasets. This makes this method specifically tailored towards remote sensing images and can help speed up training as opposed to not using pre-trained weights or using pre-trained weights from a non-remote sensing dataset like ImageNet [5]. The search space parameters are also shown in Table 5.1. Our dataset, SENT-NICFI, is not included in the set of pretrained weights, because it was not yet completed at the time of obtaining the sets of pre-trained weights. We will now go into more detail about the underlying methods of AutoSR-RS: RCAN and WDSR (both of which have also been briefly discussed in Section 3.1.1).

RCAN RCAN, which stands for Residual Channel Attention Network, introduces channel attention. Channel attention modules give more weight to informative features instead of considering all features to be equal. The network consists of stacked residual groups, with an upscaling module at

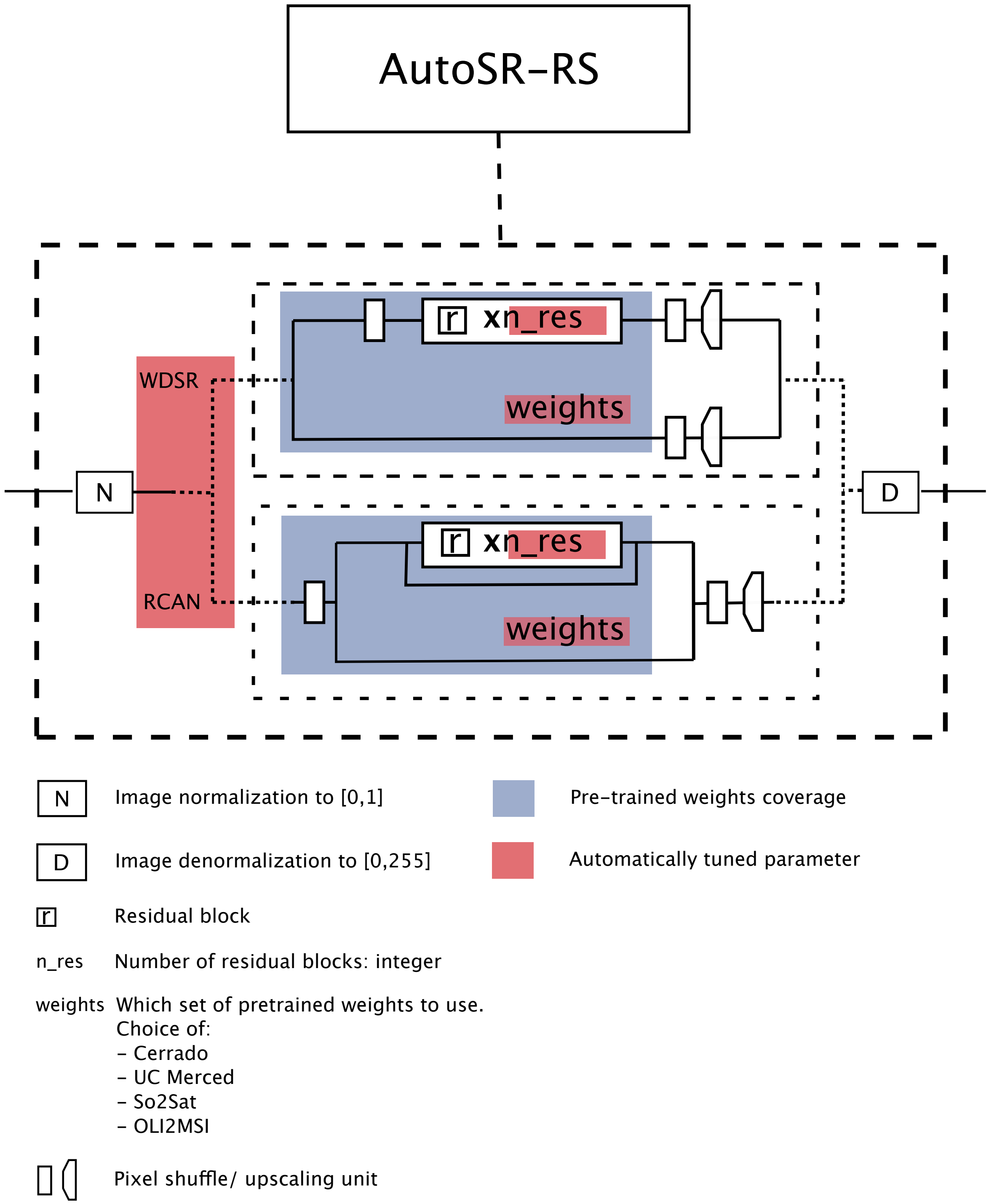

Figure 5.1: The architecture of AutoSR-RS.

the end of the residual stack.

WDSR WDSR's [2] main branch also exists of residual blocks like RCAN, but does not have the residual-in-residual structure. Moreover, the residual blocks do not contain the channel attention mechanism. The convolutional layers in the whole model are replaced by convolutions with weight normalization.

Differences between RCAN & WDSR There are some differences between the high-level architectures of the models. For example, the skip branch of RCAN first passes through an activation layer (which is the first convolutional layer), which Fan et al. [11] have chosen to omit. Additionally, the upscaled image produced by RCAN's upscale module is decoded using a convolutional layer, whilst in WDSR, the two branches are simply added together and no decoding is taking place. Finally, an important difference is that RCAN is much deeper than WDSR, due to the residual-in-residual structure.

5.2.2 Super-resolution blocks

We implement blocks for the task of super-resolution of remote sensing images. These blocks are used by the NAS framework to build candidate networks. To the best of our knowledge, current NAS frameworks do not offer such blocks. We propose a model with a search space built upon RCAN [51] and WDSR [11]. The optimizer can choose between these two models, and modify them by changing the number of residual blocks in each model. We observed that RCAN has many more trainable parameters, and can take up to 10 times as long to train. However, RCAN does not outperform WDSR in all cases. Preliminary experiments with a shallower version of RCAN, with just 1 residual group instead of 10, showed that this setting of RCAN (which we refer to as RCAN_U, for undeep RCAN) can achieve good results on the Cerrado dataset even with much fewer parameters than the original proposed by Zhang et al. [51]. These results, along with the number of parameters of various models can be found in Appendix B. The resulting model still has more parameters than WDSR, but is more comparable, and trains faster than the full-size RCAN.

In addition to modifying the number of residual blocks, another parameter that can be tuned by the AutoML search algorithm is which set of pre-trained weights to use. These weights are obtained by training WDSR and RCAN on the datasets presented in Chapter 4. The dataset on which the model is trained during a particular experiment will be excluded from the choices. This is done by removing the name of the current training data from the list of available pre-trained weights for the experiment.

[2]The acronym appears to be inspired by its predecessors VDSR [22] and EDSR [27], the paper is titled "Wide Activation for Efficient and Accurate Image Super-Resolution"

Chapter 6

Experiments

The sections in this chapter will describe the experiments we defined to answer our research questions:

> *Q1*: Can AutoSR-RS achieve better performance on remote sensing datasets than the state-of-the-art models?
> *Q2*: Which blocks (RCAN or WDSR) and block hyperparameters (i.e., number of residual blocks, set of pre-trained weights) are used the most in well-performing models found by AutoSR-RS?

Since baseline methods are included in the search space, it is expected that the model should be able to perform better than the baselines. By answering Q1, we do not just want to answer which method achieves the highest scores, but which one performs the most consistently on all the datasets, since we are looking for a method that can be applied to diverse datasets. Answering Q2 could give insight into how to improve the search space of AutoSR-RS.

Before going into the details of the experiments, we will first describe the baseline methods in Section 6.1. Section 6.2 outlines hyperparameter choices of AutoSR-RS, followed by an outline of the evaluation methods in Section 6.3.

6.1 Baselines

We use the following baseline methods:

- **WDSR** [11], (described in Sections 3.1.1 & 5.2.1), consists of a skip branch and a main branch with a block consisting of smaller residual blocks. Instead of standard convolutional layers, weight normalization is used [40] for faster convergence and higher accuracy. We used the Keras code for the WDSR model released by Martin Krasser[1]. The network is trained with the same parameters as WDSR-B in the original paper: 32 residual blocks, Adam optimizer with piecewise constant decay of the learning rate.

- **RCAN** [51], also described in Sections 3.1.1 & 5.2.1, consists of a skip branch and a main branch with residual blocks. The residual blocks are organized in residual groups, with skip connections from group to group as well. The residual blocks have a channel attention unit,

[1]Single Image Super-Resolution with EDSR, WDSR and SRGAN (2020) `https://github.com/krasserm/super-resolution`

which weights features according to their importance. The implementation of RCAN in Keras that we used is made available by Hieubkset[2]. The parameter settings maintained from the original paper are: 10 residual groups with 20 residual blocks each, using the Adam optimizer with an initial learning rate of $1 \cdot 10^{-4}$, with a learning rate decay of 0.5 and decaying every 10 and 15 steps.

- **AutoSRCNN**: a simple autoML super-resolution approach. The method is inspired by SR-CNN [7]. AutoSRCNN is a CNN built from convolutional layers exclusively, without residual connections, pre-trained weights or specialized blocks. The only parameters are the number of layers, the kernel size of each layer and the number of filters of each layer. The model consists of an upsampling layer which scales the input image to the required output size using interpolation, followed by a ConvBlock (parameters in Table 6.1). Details on the blocks we added to the original AutoKeras code to implement AutoSRCNN and AutoSR-RS are found in Appendix D. The search space of this method is much smaller compared to AutoSR-RS. Thus, this method serves as a control to make sure that a larger search space is indeed necessary to solve the problem of SISR for remote sensing images. AutoSRCNN in general finds networks comparable to SRCNN, which are much simpler than the current state-of-the-art. The search space is also smaller, allowing AutoSRCNN to traverse more of its search space in the same number of trials than AutoSR-RS could. As a result, training is fast and pre-trained weights are not needed to speed up the weights.

We selected WDSR and RCAN among the methods described in Chapter 3, because we were limited to using Keras [2] in order to keep the environment consistent for all considered methods, and it was not possible in all cases to find a Keras implementation. Other methods like EDSR [27] were also available, however, RCAN and WDSR both outperform this approach [51][11]. Thus, it would not help answer the question whether AutoSR-RS is better than the state-of-the-art.

Experimental settings such as the number of epochs and batch sizes for different datasets are determined by considering factors like validation loss, memory and time limits on the usage of computational cluster. All of the experiments are run with an early stopping callback, which halts the training if the validation loss has not decreased for 10 epochs. The details on these experimental settings can be found in Appendix C.

The number of epochs for WDSR and RCAN (equal for both) was determined per dataset by considering the decrease of the validation loss and computational resources (the larger datasets like So2Sat need more time to train, and training time was limited by available resources). Both AutoSRCNN and AutoKeras are run for 20 trials, with the number of epochs set at a maximum of 100 (AutoKeras might cut the training of before that number of epochs is reached if no improvement is expected).

6.2 Hyperparameter tuning

The search space can be narrowed down by considering choices made in state-of-the-art papers. We list settings that have been taken from Fan et al. [11] and Zhang et al. [51] in Table 6.2.

Batch normalization and dropout are rarely used in super-resolution methods. These measures are often used to prevent overfitting. However, overfitting is rare for super-resolution [11], and batch normalization can cause artefacts to occur in the super-resolved images [48]. L1 loss or MAE is chosen over L2. Both loss functions are popular choices for image-related tasks. However, L1 loss

[2]Hieubkset. Single Image Super-resolution, EDSR, SRGAN, SRFeat, RCAN, ESRGAN and ERCA (ours) benchmark comparison (2020) `https://github.com/hieubkset/Keras-Image-Super-Resolution`

Table 6.1: Settings of AutoSRCNN

Parameter	Value
Pooling	False
Dropout	0
Padding	same
Filters	64
Kernel size	$[3, 5, 7, 9]$

Table 6.2: Hyperparameter settings from [11][51]

Setting	Value
Batch Normalization	None
Dropout	0
Loss function	L1

yields better results for super-resolution than L2 loss [52]. A possible explanation could be that L2 is a squared loss function, which might penalise faulty pixels very strongly. These pixel errors naturally occur in remote sensing images, and can result in high losses through no fault of the model. The L1 loss would not give as much weight to these faulty pixels because it is a linear loss. L1 loss is given by:

$$\text{L1} = \sum_{i=1}^{N} |I(i) - \hat{I}(i)| \tag{6.1}$$

where I is the ground-truth image, $\hat{I}$ is the super-resolved image, and N is the number of pixels in the image.

AutoKeras offers multiple optimisers to tune the hyperparameters. We chose to use the default optimiser, which is a combination of random search and greedy search. We use this optimiser for both AutoSRCNN and AutoSR-RS for a fair comparison.

6.3 Evaluation

In super-resolution research, two common evaluation metrics are peak signal-to-noise ratio (PSNR) and structural similarity (SSIM). Both are widely used. ([28],[38],[26],[44])

- **PSNR** is related to the MSE between images. It evaluates the image exclusively at pixel-level. Because it does not take visual perception into account, it can be a poor metric of super-resolution quality. It is popular due to the lack of better alternative metrics for comparing methods [49]. The PSNR is defined as follows [49]:

$$PSNR = 10 \cdot \log_{10} \frac{L^2}{\frac{1}{N}\sum_{i=1} N(I(i) - \hat{I}(i))^2} \tag{6.2}$$

  Where I is the ground-truth image, $\hat{I}$ is the super-resolved image, L is the maximum pixel value, which in this case is 255, and N is the number of pixels.

- **SSIM** computes the similarity of two images based on the luminance, contrast and structure that are calculated from the RGB pixel values [49]. The SSIM is given by [26]:

$$SSIM = \frac{(2\mu_I\mu_{\hat{I}} + c_1)(2\sigma_{I\hat{I}} + c_2)}{(\mu_I^2 + \mu_{\hat{I}}^2 + c_1)(\sigma_I^2 + \sigma_{\hat{I}}^2 + c_2)} \tag{6.3}$$

  where $\hat{I}$ is the super-resolved image, I ground-truth image, μ is the average luminance and σ the standard deviation of the luminance and c_1 and c_2 are constants.

6.4 Experimental setup

We have designed the following experiments to answer the research questions presented at the beginning of this chapter. We will train and evaluate AutoSR-RS, as proposed in Chapter 5, on the datasets described in Chapter 4. The test set is created by taking 20% of the dataset, the remaining data is split with 80% for training and 20% for validation. The same splits are maintained for all experiments.

The experiments are run on the ALICE cluster[3], with two GeForce RTX 2080TI GPUs and 10GB of CPU RAM. Each experiment is repeated 5 times, for every dataset. The running time ranges from 30 minutes to 5 days, depending on the number of parameters of the model and the number of images and image sizes of the dataset. The results of the experiments are compared by first bootstrapping the results with 1000 samples of size 3 followed by a Wilcoxon signed-rank test [3] for non-normally distributed samples.

A version of AutoKeras which was adapted to include image output and custom metrics is available at `https://github.com/JuliaWasala/autokeras` (release `1.0.16.post1`) and the code at `https://github.com/JuliaWasala/autoSR-RS_SENT-NICFI`. The code repository provides a detailed description of the environment and required software and packages. The pre-trained weights necessary to run the code, can be downloaded from `https://universiteitleiden.data.surfsara.nl/index.php/s/i9oF0o8R36ZXZ5X`.

[3] `https://www.universiteitleiden.nl/en/research/research-facilities/alice-leiden-computer-cluster`

Chapter 7

Results

This chapter shows the results of our experiments. The results will be discussed per research question, as presented in Chapter 6. The chapter is concluded with a discussion of the results of the SENT-NICFI dataset.

7.1 Q1: Can AutoSR-RS outperform the state-of-the-art?

Tables 7.1 and 7.2 list the average results from training and evaluating each method–dataset combination 5 times. The significantly highest scores are in **boldface** [1]. From these tables, we can read that AutoSR-RS outperforms all baselines on the SENT-NICFI dataset, and comes in as a close second on the other datasets. The results of the SENT-NICFI dataset are especially interesting. The high variety of images, including 5 types of ecosystems, makes it a challenging dataset to model by a neural network. The fact that AutoSR-RS scores the highest on this dataset, means that it is best able to handle highly diverse datasets. It is not clear from these results why AutoSR-RS does not outperform all of the baselines on the other datasets.

Figures 7.1 and 7.2 show how the PSNR and SSIM scores are distributed across a sample of 100 random images of the test set of each dataset. These distributions can show how consistently the models super-resolve images from different datasets. The narrower the distribution, the more consistent the results.

In Figure 7.1 we can see that the most outliers occur in the Cerrado dataset results. In most cases, the prediction consistencies are comparable, except for SENT-NICFI, where WDSR has the narrowest distribution. In Figure 7.2, we can see that for Cerrado, RCAN and AutoSR-RS have perfectly recovered a HR ground truth image, achieving a score of 1.0. We can also see that there is more difference in consistency of prediction between the methods than with the PSNR score: for example, AutoSRCNN has a wider distribution for the UC Merced dataset than the other methods, meaning that the prediction scores are less consistent than for other methods. Additionally, AutoSRCNN and WDSR have the widest distributions for the OLI2MSI datasets.

Now, focusing on the AutoSR-RS scores specifically, there is not much difference in terms of consistency of PSNR scores. However, the SSIM scores obtained by AutoSR-RS on the Cerrado and OLI2MSI datasets have narrower distributions and thus are more consistent than the other methods. To conclude, from these figures we can not see that any method yields much more consistent scores than any other, in terms of spread of PSNR and SSIM scores of the sample of test images.

[1] The results of the So2Sat dataset are excluded: due to excessive resource use, AutoSR-RS could not be evaluated on this dataset.

Table 7.1: PSNR results of all methods, each experiment has been run 5 times. The significantly best results are in **boldface**.

Method	Cerrado	UC Merced	OLI2MSI	SENT-NICFI
WDSR	**40.96** $\pm$ 0.39	28.00 $\pm$ 11.55	43.89 $\pm$ 0.04	28.17 $\pm$ 1.20
RCAN	38.48 $\pm$ 0.38	**33.77** $\pm$ 0.02	**44.45** $\pm$ 0.01	30.12 $\pm$ 0.02
AutoSRCNN	38.80 $\pm$ 0.89	30.82 $\pm$ 1.44	43.13 $\pm$ 0.68	28.85 $\pm$ 0.22
AutoSR-RS	40.61 $\pm$ 1.91	33.57 $\pm$ 0.23	44.42 $\pm$ 0.68	**30.20** $\pm$ 0.42

Table 7.2: SSIM results of all methods, each experiment has been run 5 times. Best results are in **boldface**.

Method	Cerrado	UC Merced	OLI2MSI	SENT-NICFI
WDSR	**0.9729** $\pm$ 0.0016	0.7414 $\pm$ 0.3956	0.9719 $\pm$ 0.0002	0.7843 $\pm$ 0.0525
RCAN	0.9544 $\pm$ 0.0067	**0.9252** $\pm$ 0.0002	**0.9749** $\pm$ 0.0000	0.8537 $\pm$ 0.0007
AutoSRCNN	0.9507 $\pm$ 0.0083	0.8825 $\pm$ 0.0269	0.9680 $\pm$ 0.0045	0.8223 $\pm$ 0.0009
AutoSR-RS	0.9645 $\pm$ 0.0191	0.9238 $\pm$ 0.0024	0.9741 $\pm$ 0.0090	**0.8550** $\pm$ 0.0097

Table 7.3: Ranking of the methods, showing the number of times a method has a significantly higher score on a dataset than another method. The experimental results are bootstrapped and then pairwise wilcoxon signed-rank tests are performed. The average ranking is calculated across the 4 datasets, with 1 being the highest.

Model	# outperformed Based on		Average ranking Based on	
	PSNR	SSIM	PSNR	SSIM
AutoSR-RS	9	9	1.75	1.75
RCAN	8	9	2.00	1.75
WDSR	4	4	3.00	3.00
AutoSRCNN	3	2	3.25	3.50

Table 7.4: Results of longer experiments with AutoSR-RS. Each experiment has been run 5 times.

Trials	Dataset	PSNR	SSIM
100	Cerrado	42.10 $\pm$ 0.83	0.9763 $\pm$ 0.0031
50	SENT-NICFI	30.45 $\pm$ 0.26	0.8612 $\pm$ 0.0124

The complete test results (shown in Tables 7.1 and 7.2) have been analysed by first bootstrapping the results by selecting 3 measurements 1000 times. Then, pairwise Wilcoxon [3] tests are conducted, counting the number of times it has achieved a significantly higher result than other methods for each dataset. The Wilcoxon test is chosen, because the data points are not normally distributed. The methods are then ranked according to these counts. There were no ties in these comparisons: all results are statistically significant. The results are shown in Table 7.2. According to this ranking, AutoSR-RS ranks first in terms of the PSNR score, and shares the first place with RCAN in terms of SSIM. This table also shows the average ranking, calculated across the 4 datasets. Because of the small number of datasets, any statistical tests regarding the ranking of methods across all datasets (as opposed to comparing results of the methods per dataset), such as CD diagrams, would not be very meaningful. Thus, no CD diagrams can be presented.

From the rankings we can also see that AutoSRCNN consistently ranks last: this shows that a simple automated machine learning method is not enough for the problem of super-resolution of remote sensing images. This justifies the use of methods with more carefully crafted search spaces like AutoSR-RS.

7.1.1 Additional experiments with higher numbers of trials

Additional experiments have been conducted with Cerrado for 100 trials, and SENT-NICFI for 50 trials. The goal of these experiments was to find whether there is more room for improvement of AutoSR-RS results. The results are shown in Table 7.4. These results are significantly better than the results for 20 trials shown in Tables 7.1 and 7.2 and also outperform the baselines. Mainly, running AutoSR-RS yields high results more consistently, as can be read from the difference in standard deviations of the results between 20 trials and longer. Running AutoSR-RS for more trials makes it more likely to find a good result. To illustrate this, Figure 7.3 plots the highest PSNR achieved on the validation set so far at each trial. The runs stop at different trials, ranging from around 30 to 75 (for Cerrado) or 45 (for SENT-NICFI), because AutoKeras might terminate training before the maximum number of trials is achieved. This mechanism is similar to how AutoKeras can terminate training of candidate networks before the epoch budget is depleted. It helps to reduce use of resources when no improvement is expected to be found by the optimizer.

The improvement of validation scores, steep at first, starts to flatten around 20 trials, which is equal to the number of trials in the experiments shown in Tables 7.1 and 7.2. Better results can be obtained, though at a decreased rate of improvement. By giving AutoSR-RS more trials, it can in some cases help escape local minima. The result of this is that more runs achieve high results, with a higher average PSNR and SSIM and lower standard deviations.

In summary, AutoSR-RS either has the highest or second-highest performance when run for 20 trials. AutoSR-RS ranks first on statistical tests of the PSNR score, and has no significant difference with RCAN with regards to SSIM. Running AutoSR-RS for more trials achieves better results, also outperforming a baseline that was not beaten within 20 trials, though at the cost of additional iterations with a decreased rate of improvement. The ranking based on the PSNR score suggests that even though AutoSR-RS does not achieve the highest score in each dataset, it is more general than the baselines. As a first AutoML approach to super-resolution of remote sensing images, this is promising. Future improvements to the method could yield even better results. For instance, AutoSR-RS might be better able to find networks suited to specific situations by including more model blocks and sets of pre-trained weights. However, a downside of AutoSR-RS compared to the baselines is the computational complexity. 20 networks are trained instead of just 1 (in the case of RCAN and WDSR). The implication is that it requires much more machine time to train, though it may not necessarily be 20 times as long, because AutoSR-RS can terminate the training of a network if it does not expect to find improvement.

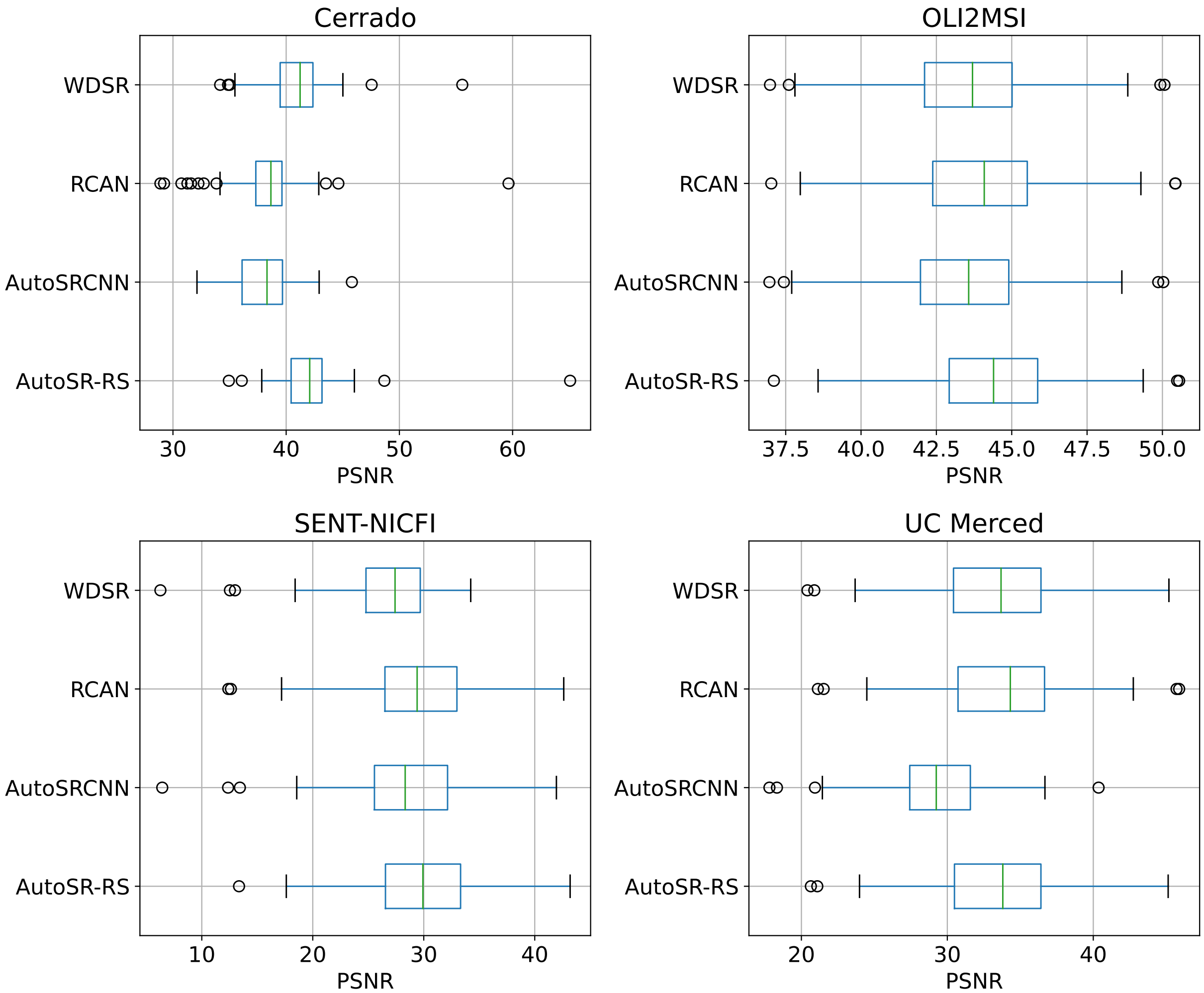

Figure 7.1: Boxplots of the PSNR scores of 100 random test images.

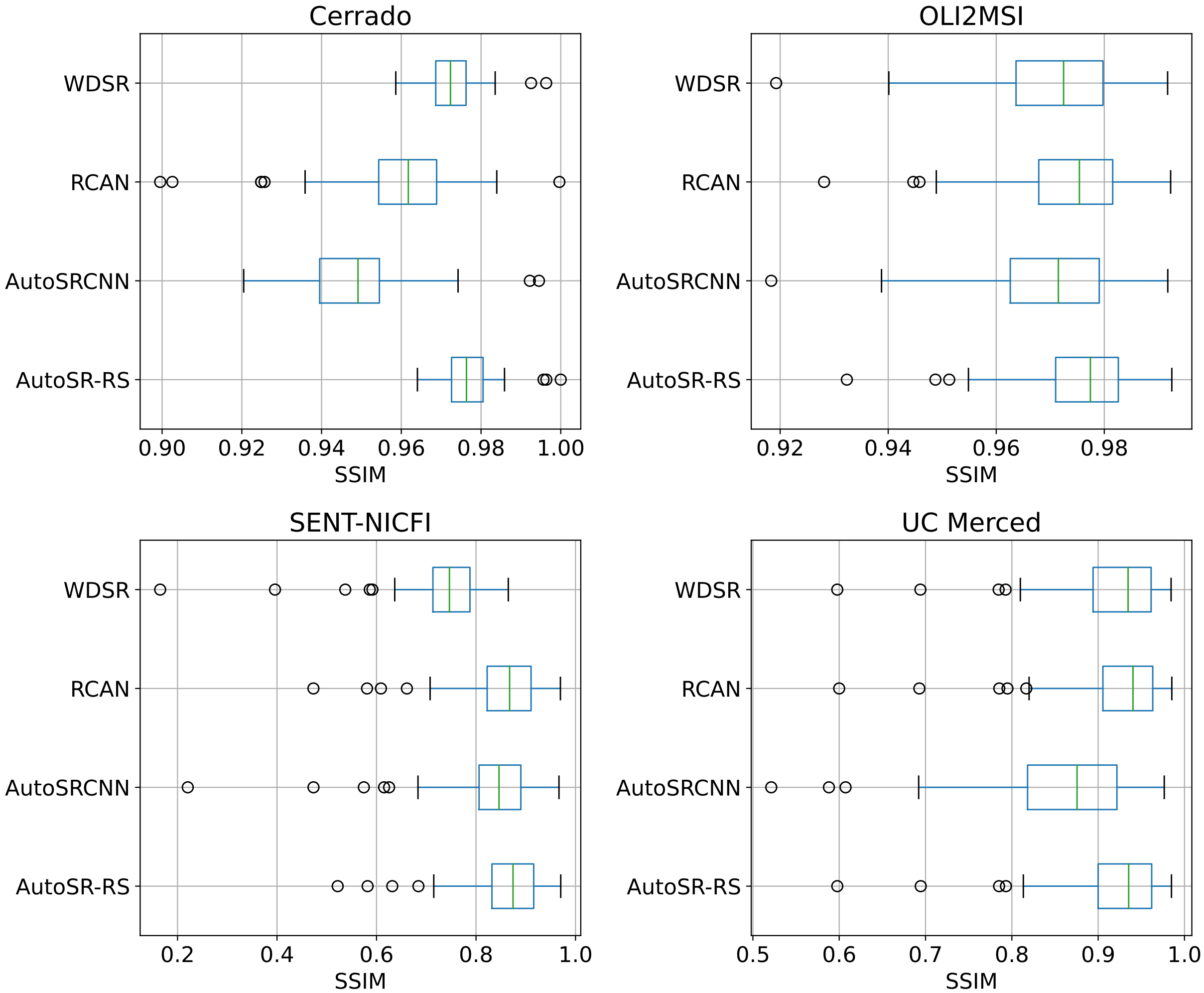

Figure 7.2: Boxplots of the SSIM scores of 100 random test images.

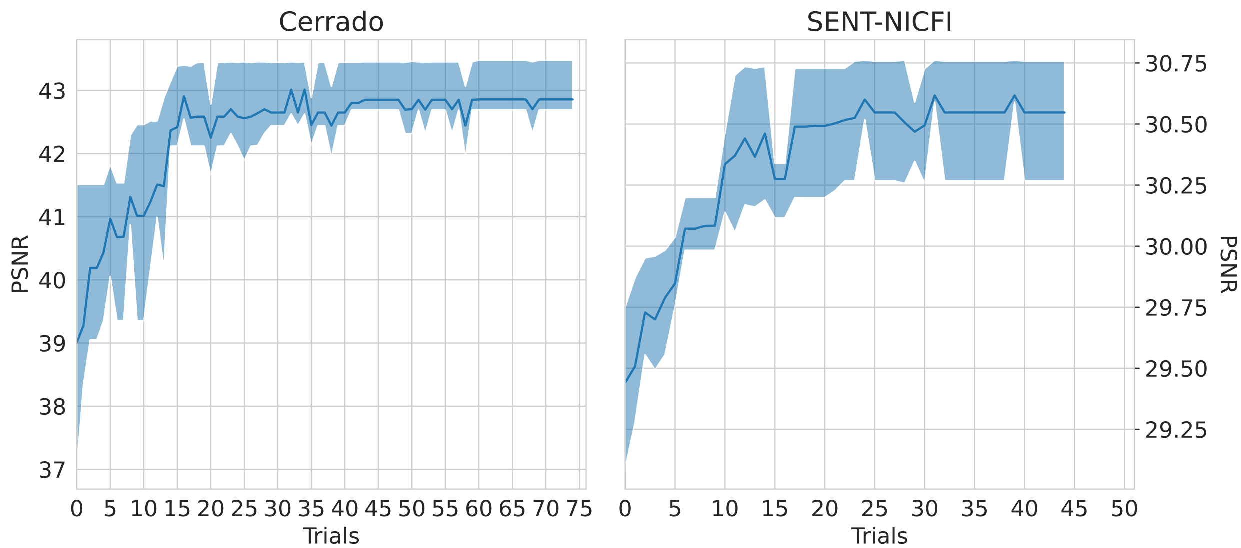

Figure 7.3: Evolution of the PSNR on the validation set for each trial of experiments with the dataset Cerrado and a maximum of 100 trials (left) and dataset SENT-NICFI and a maximum of 50 trials (right). Each point shows the mean of the best score achieved in each run up until that trial. The error bands show show the region between the lower and upper quantile.

7.2 Q2: Which hyperparameters values are selected by AutoSR-RS?

In this section, we analyze the hyperparameter values selected by AutoSR-RS: these are the hyperparameters of the best model returned by AutoSR-RS after each training. The hyperparameters are described in Section 5.2.1. Analyzing these results can help inform us how the search space can be improved.

A global overview of the selected values is shown in Figure 7.4. RCAN is chosen more than twice as often as WDSR as the model block. Considering that RCAN scores significantly higher than WDSR in our experiments, it raises the question of whether AutoSR-RS could perform better with just the RCAN block. However, this picture is nuanced by the left plot in Figure 7.5 shows which model blocks are chosen as a function of the training dataset. It shows that there is a strong preference for RCAN when AutoSR-RS is trained on UC Merced or SENT-NICFI. When trained on Cerrado and OLI2MSI, the difference is not so large. Thus, it might not be beneficial to the performance of AutoSR-RS to discard the WDSR block: there are still 2 datasets that select it almost half of the time, and this could also be true or even more for other datasets AutoSR-RS might be used for.

The center plot in Figure 7.4 shows the selected number of residual blocks, N_res. The most selected number is around the middle of the possible range, at 20 blocks, which coincides with the maximum number of residual blocks possible for the RCAN block. Thus, it could make sense to increase this maximum to a higher number. Finally, the right plot in Figure 7.4 shows that the sets of pre-trained weights that are chosen the most are UC Merced and Cerrado, while So2Sat and OLI2MSI are chosen slightly less often. Interestingly, the only weights obtained from a real dataset (OLI2MSI) are chosen the least often.

Figure 7.5 shows which set of pre-trained weights is selected as a function of the training dataset. The right plot in Figure 7.5 shows that the choice of pre-trained weights is split between Cerrado and OLI2MSI on the one hand, and UC Merced and SENT-NICFI on the other hand, as with the model blocks. Models trained on Cerrado exclusively select weights from UC Merced in the final model, while models trained on OLI2MSI select both weights from UC Merced and from Cerrado. Models trained on UC Merced or SENT-NICFI select weights from Cerrado, So2Sat or OLI2MSI without a strong preference for any of the three. These results show that different sets of pre-trained weights can benefit different scenarios to varying degrees. It could improve AutoSR-RS to provide a wider variety of pre-trained weights, so an appropriate set can be found for these varying scenarios. However, more research is needed to determine the cause of this split in results depending on the training dataset. Understanding better why specific values are chosen based on dataset characteristics could yield valuable insights in how to improve the search space.

Finally, Figure 7.6 shows the numbers of residual blocks for each dataset. These plots once again show that there is a peak at 20 residual blocks, indicating that the RCAN block could potentially perform better if higher numbers of residual blocks were possible.

From the results discussed in this section, we can conclude that the range of residual blocks can be restricted. Even though the WDSR block seems to be less relevant at first view, Figure 7.5 suggests that the WDSR block can be relevant to some datasets. Thus, it would make sense to keep the WDSR block for generality of AutoSR-RS. Finally, all of the available sets of pre-trained weights are used similarly, thus this set of choices is not to be restricted. Adding more model blocks and sets of pre-trained weights could allow AutoSR-RS to find models even better suited to a varying array of scenarios. Extending AutoSR-RS with new state-of-the-art methods is straightforward, thus the method could be adapted to discoveries in the field. Additional research is needed to better understand how dataset characteristics influence the choice of parameter values, so that the search space can be improved in a targeted way.

7.3 SENT-NICFI

Tables 7.1 and 7.2 list the results of different datasets. The scores vary a lot per dataset, with high scores achieved by all methods on OLI2MSI and much lower scores for SENT-NICFI. A possible explanation could be that the OLI2MSI images are more similar to each other, displaying a lot of tree cover. Examples of these images can be found in Appendix E. A similar statement can be made about the Cerrado images, which are all also very similar in colour. The UC Merced and SENT-NICFI datasets contain images of more varied scenes in terms of colours and features, which could make these datasets more challenging to model by a neural network. In general, RCAN and AutoSR-RS seem to be better able to adapt to these varied datasets.

This property of SENT-NICFI potentially makes it an interesting dataset to evaluate models on their ability to model more complex datasets. SENT-NICFI has the added benefit that it can be partitioned per land cover type, enabling the training of models for specific use cases. However, multi-sensor datasets like SENT-NICFI have the caveat that, in addition to "translating" between resolutions, the model will also need to learn to convert from one sensor to another. To reduce this effect, the colours of the NICFI images are matched to the Sentinel images. Another approach is taken by Razzak et al. [36], who propose a colour matching module as an addition to the super-resolution model to address this problem.

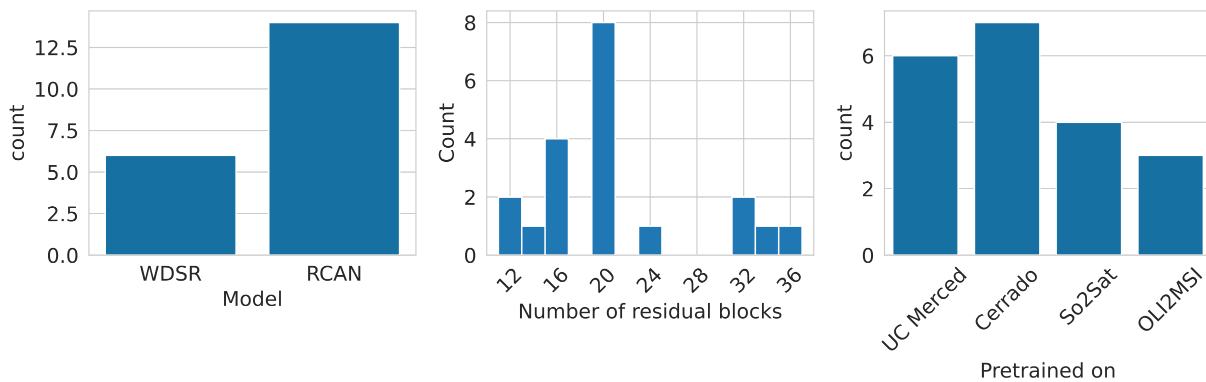

Figure 7.4: This figure shows the hyperparameter values chosen in the best networks returned by AutoSR-RS in each experiment. The plot on the left shows whether the WDSR or RCAN block is chosen. The middle plot shows which numbers of residual blocks (N_res) are chosen. The plot on the right shows how many times each set of pre-trained weights is selected.

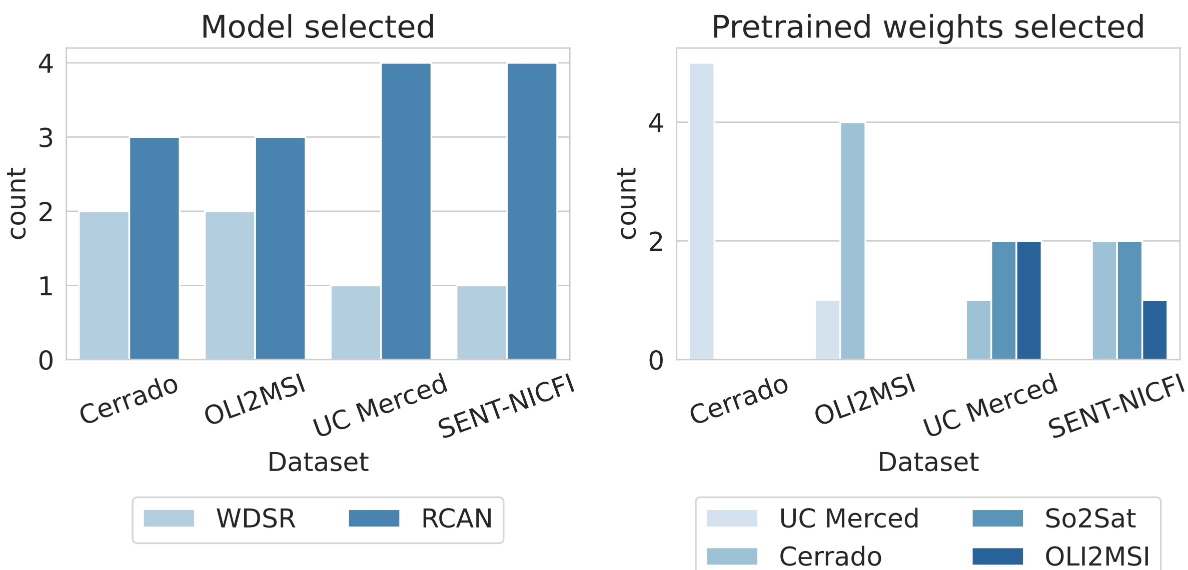

Figure 7.5: This figure shows hyperparameter values in models returned by AutoSR-RS as a function of the training dataset. The counts indicate how many times out of the 5 repetitions with each training dataset, a parameter value is chosen for the final architecture. The plot on the left shows which model block is selected, the plot on the right shows which set of pre-trained weights is selected.

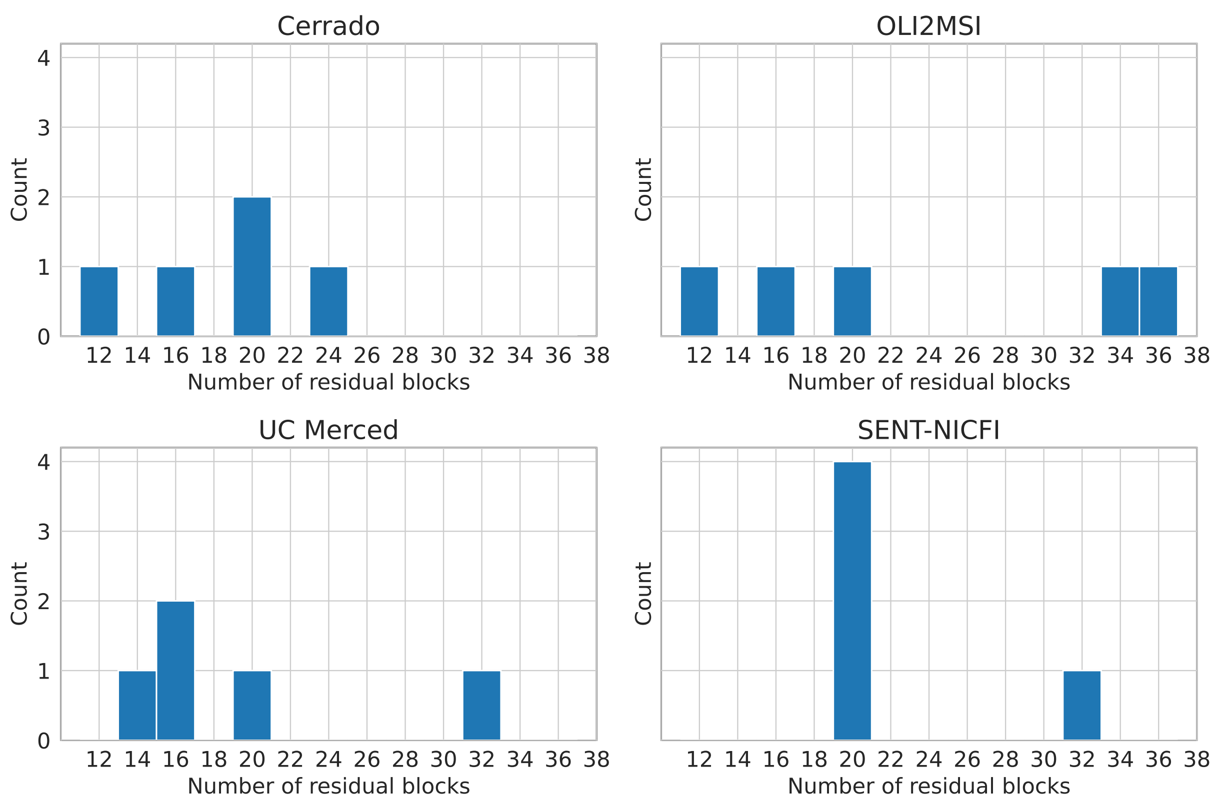

Figure 7.6: This figure shows the number of residual blocks that are selected for models produced by AutoSR-RS for each training dataset.

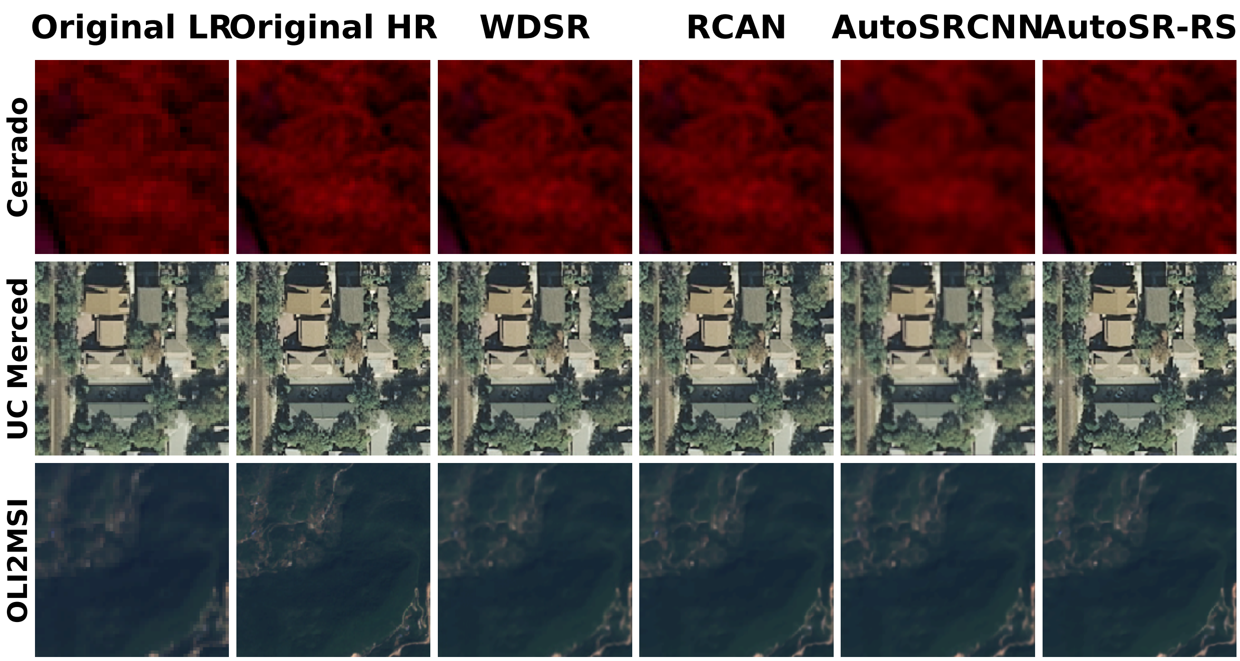

Figure 7.7: This figure shows examples from each training dataset of predicted test images produced by different methods. The OLI2MSI images have been brightened, because it would otherwise display as almost black in Python (but normal in QGIS [35].) The OLI2MSI images have also been cropped to a region of 60 by 60 pixels (for LR, 120 by 120 for the HR images) in the top left corner to show the resolution improvements more clearly.

Chapter 8

Conclusion & Future Work

In this chapter, we will summarize the results discussed in the previous Chapter. Finally, ideas for future research are presented.

8.1 Conclusions

In this thesis, we investigated an autoML approach to the problem of super-resolution in remote sensing, called AutoSR-RS. We sought to answer the question of *whether this method can outperform the state-of-the-art* and *which model hyperparameters are found most in the models returned by AutoSR-RS*.

We showed that automated machine learning can be used to super-resolve remote sensing images and achieve similar or better results than state-of-the-art methods. We used datasets with images from different sources, including different satellites and aerial imagery. We included both synthetic data as well as real data, and proposed a new real dataset for single image super-resolution called SENT-NICFI. SENT-NICFI adds to the small set of real datasets for SISR. It consists of many images of different types of land cover, and can be partitioned per land cover type for specific purposes.

AutoSR-RS is an automated machine learning approach with the search space constructed with findings from state-of-the-art methods like WDSR and RCAN, and uses pre-trained weights from remote sensing datasets to speed up the training. We compared the performance of AutoSR-RS to WDSR, RCAN and a simpler automated machine learning approach to super-resolution, called AutoSRCNN. We showed that the carefully crafted search space of AutoSR-RS outperforms AutoSRCNN on every dataset, motivating the use of a more intricate, specialized search space instead of a simple and more general one. Though AutoSR-RS does not have higher performance than RCAN on each dataset (AutoSR-RS outperforms all baselines on 1 of the 4 datasets, and comes second on the remaining 3), statistical testing has ranked the method higher. Thus, AutoSR-RS performs more consistently on various datasets than the baseline methods, achieving the goal of AutoML to be consistently better on all datasets than a single, non-automated method would.

We found that running AutoSR-RS for more trials, further increases the performance, though at the cost of longer running times and a decreased rate of improvement.

Furthermore, from analysing the hyperparameters chosen by AutoSR-RS, we have found that all sets of pre-trained weights and model blocks are used. The RCAN model block is used the most, which is to be expected from the experimental results that show that RCAN outperforms WDSR in 3 out of the 4 datasets (as shown in Tables 7.1 and 7.2 in Chapter 7). The results of parameter values appear to be split between training datasets: Cerrado and OLI2MSI have more similar results, and

UC Merced and SENT-NICFI are more alike. These differences could be explained by Cerrado and OLI2MSI containing more visually similar images, while the UC Merced and SENT-NICFI datasets contain more varying scenes. More research is needed to find why these differences in the chosen values occur.

AutoSR-RS could be further improved by expanding the possibilities for these hyperparameters, though results have shown that the maximum number of residual blocks could be increased.

AutoSR-RS is the first automated machine learning method for super-resolution of remote sensing images. It achieves high performance across different remote sensing datasets with the use of pre-trained weights. We hope the method can further be improved in the future, with the addition of more findings from super-resolution research.

8.2 Future research

In future work, AutoSR-RS could be improved by learning how the characterstics of different datasets influence the final model produced by AutoSR-RS, with the goal to improve the search space.

Alternatively, we would like to extend AutoSR-RS to MISR. MISR methods are gaining in popularity, in part because of the theoretical advantage of having more information when using multiple images to super-resolve an image. AutoSR-RS could be expanded to MISR, for example in a similar way as in the work by Kawulok et al. [21], HighResNet [6] or DeepSUM [30]. The latter two are both participants of the ESA PROBA-V Super-Resolution Competition [10]. These methods process each of the LR images independently and then merge them at some point in the model. This type of approach would require additional computational resources.

A different research avenue could be to focus on the loss function. Super-resolution is only a low-level task and usually not the end-goal by itself. Therefore, the goal should be to increase performance on the target task instead of approaching the original high-resolution image as much as possible. A loss function that takes the results of an application task like object detection into account could help boost the performance of remote sensing tasks.

Finally, an interesting direction would be to address the search space differently. Much like NASnet [55], the high-level architecture could be fixed and only the repeating residual blocks would be optimised. This approach would reduce the size of the search space and could offer a reduction in computational complexity. On a high level, many SR methods are quite similar. Thus, it could make sense to focus on the smaller units making up these models. The neural architecture search could also be split into two phases, where in one phase the high-level architecture is optimised and in the second phase the residual blocks are optimised.

Bibliography

[1] (2022). Norway's international climate and forest initiative (nicfi). `https://www.nicfi.no/`.

[2] Chollet, F. et al. (2015). Keras. `https://keras.io`.

[3] Conover, W. J. (1998). *Practical Nonparametric Statistics*. Wiley Series in Probability and Statistics. John Wiley & Sons, Nashville, TN, 3 edition.

[4] Debes, C., Merentitis, A., Heremans, R., Hahn, J., Frangiadakis, N., van Kasteren, T., Liao, W., Bellens, R., Pižurica, A., Gautama, S., et al. (2014). Hyperspectral and lidar data fusion: Outcome of the 2013 grss data fusion contest. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(6):2405–2418.

[5] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255.

[6] Deudon, M., Kalaitzis, A., Goytom, I., Arefin, M. R., Lin, Z., Sankaran, K., Michalski, V., Kahou, S. E., Cornebise, J., and Bengio, Y. (2020). Highres-net: Recursive fusion for multi-frame super-resolution of satellite imagery. *arXiv preprint arXiv:2002.06460*.

[7] Dong, C., Loy, C. C., He, K., and Tang, X. (2014). Learning a deep convolutional network for image super-resolution. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8692 LNCS:184–199.

[8] Ducournau, A. and Fablet, R. (2016). Deep learning for ocean remote sensing: an application of convolutional neural networks for super-resolution on satellite-derived sst data. In *2016 9th IAPR Workshop on Pattern Recogniton in Remote Sensing (PRRS)*, pages 1–6.

[9] European Space Agency. Orbit - Sentinel 2 - Mission - Sentinel Online - Sentinel Online. `https://sentinel.esa.int/web/sentinel/missions/sentinel-2/satellite-description/orbit`. Last visited on 16-05-2022.

[10] European Space Agency. Proba-v. `https://earth.esa.int/eogateway/missions/proba-v`. Last visited on 15-12-2021.

[11] Fan, Y., Yu, J., and Huang, T. S. (2018). Wide-activated deep residual networks based restoration for bpg-compressed images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 2621–2624.

[12] GDAL/OGR contributors (2022). *GDAL/OGR Geospatial Data Abstraction software Library*. Open Source Geospatial Foundation.

[13] Glass, K. (2013). *Interpreting Aerial Photographs to Identify Natural Hazards*. Elsevier.

[14] Goldblatt, R., Rivera Ballesteros, A., and Burney, J. (2017). High spatial resolution visual band imagery outperforms medium resolution spectral imagery for ecosystem assessment in the semi-arid brazilian sertão. *Remote Sensing*, 9(12).

[15] Haris, M., Shakhnarovich, G., and Ukita, N. (2021). Task-driven super resolution: Object detection in low-resolution images. In *International Conference on Neural Information Processing*, pages 387–395. Springer.

[16] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

[17] Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

[18] Hutter, F., Kotthoff, L., and Vanschoren, J. (2019). *Automated Machine Learning: Methods, Systems, Challenges*. Springer Publishing Company, Incorporated, 1st edition.

[19] Jiang, K., Wang, Z., Yi, P., Wang, G., Lu, T., and Jiang, J. (2019). Edge-enhanced gan for remote sensing image superresolution. *IEEE Transactions on Geoscience and Remote Sensing*, 57(8):5799–5812.

[20] Jin, H., Song, Q., and Hu, X. (2019). Auto-keras: An efficient neural architecture search system. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1946–1956. ACM.

[21] Kawulok, M., Benecki, P., Piechaczek, S., Hrynczenko, K., Kostrzewa, D., and Nalepa, J. (2019). Deep learning for multiple-image super-resolution. *IEEE Geoscience and Remote Sensing Letters*, 17(6):1062–1066.

[22] Kim, J., Lee, J. K., and Lee, K. M. (2016). Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1646–1654.

[23] Koh, J. C., Spangenberg, G., and Kant, S. (2021). Automated machine learning for high-throughput image-based plant phenotyping. *Remote Sensing 2021, Vol. 13, Page 858*, 13:858.

[24] Köhler, T., Bätz, M., Naderi, F., Kaup, A., Maier, A., and Riess, C. (2019). Toward bridging the simulated-to-real gap: Benchmarking super-resolution on real data. *IEEE transactions on pattern analysis and machine intelligence*, 42(11):2944–2959.

[25] Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al. (2017). Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690.

[26] Li, J., Pei, Z., and Zeng, T. (2021). From beginner to master: A survey for deep learning-based single-image super-resolution. *arXiv preprint arXiv:2109.14335*.

[27] Lim, B., Son, S., Kim, H., Nah, S., and Mu Lee, K. (2017). Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 136–144.

[28] Liu, H., Qian, Y., Zhong, X., Chen, L., and Yang, G. (2021). Research on super-resolution reconstruction of remote sensing images: a comprehensive review. *Optical Engineering*, 60:100901.

[29] Ma, W., Pan, Z., Guo, J., and Lei, B. (2019). Achieving super-resolution remote sensing images via the wavelet transform combined with the recursive res-net. *IEEE Transactions on Geoscience and Remote Sensing*, 57(6):3512–3527.

[30] Molini, A. B., Valsesia, D., Fracastoro, G., and Magli, E. (2019). Deepsum: Deep neural network for super-resolution of unregistered multitemporal images. *IEEE Transactions on Geoscience and Remote Sensing*, 58(5):3644–3656.

[31] Nguyen, P., Gite, R., Yang, Z., and Halem, M. (2021). Deep neural network architecture search for emulating physical parameterization of planetary boundary layer height. In *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, pages 7119–7122.

[32] Nogueira, K., Dos Santos, J. A., Fornazari, T., Freire Silva, T. S., Morellato, L. P., and S. Torres, R. D. (2016). Towards vegetation species discrimination by using data-driven descriptors. In *2016 9th IAPR Workshop on Pattern Recogniton in Remote Sensing (PRRS)*, pages 1–6.

[33] Palacios Salinas, N. R., Baratchi, M., van Rijn, J. N., and Vollrath, A. (2021). Correction to: Automated machine learning for satellite data: Integrating remote sensing pre-trained models into automl systems. In *Machine Learning and Knowledge Discovery in Databases. Applied Data Science Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part V*, page C1, Berlin, Heidelberg. Springer-Verlag.

[34] Polonskaia, I. S., Aliev, I. R., and Nikitin, N. O. (2021). Automated evolutionary design of cnn classifiers for object recognition on satellite images. *Procedia Computer Science*, 193:210–219. 10th International Young Scientists Conference in Computational Science, YSC2021, 28 June – 2 July, 2021.

[35] QGIS Development Team (2022). *QGIS Geographic Information System*. QGIS Association.

[36] Razzak, M., Mateo-Garcia, G., Gómez-Chova, L., Gal, Y., and Kalaitzis, F. (2021). Multi-spectral multi-image super-resolution of sentinel-2 with radiometric consistency losses and its effect on building delineation. *arXiv preprint arXiv:2111.03231*.

[37] Renza, D., Cárdenas, E. A., Jaramillo, C. M., Weber, S. S., and Martinez, E. (2021). Land-slide susceptibility model by means of remote sensing images and automl. *Communications in Computer and Information Science*, 1431 CCIS:25–37.

[38] Rohith, G. and Kumar, L. S. (2021). Paradigm shifts in super-resolution techniques for remote sensing applications. *Visual Computer*, 37:1965–2008.

[39] Ronan, P. (2007). EM Spectrum. `https://commons.wikimedia.org/wiki/File:EM_spectrum.svg`.

[40] Salimans, T. and Kingma, D. P. (2016). Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *Advances in neural information processing systems*, 29.

[41] Shermeyer, J. and Van Etten, A. (2019). The effects of super-resolution on object detection performance in satellite imagery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

[42] Team, P. (2017–). Planet application program interface: In space for life on earth. `https://api.planet.com`.

[43] Thornton, C., Hutter, F., Hoos, H. H., and Leyton-Brown, K. (2013). Auto-weka: Combined selection and hyperparameter optimization of classification algorithms. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Part F128815:847–855.

[44] Tsagkatakis, G., Aidini, A., Fotiadou, K., Giannopoulos, M., Pentari, A., and Tsakalides, P. (2019). Survey of deep-learning approaches for remote sensing observation enhancement. *Sensors 2019, Vol. 19, Page 3929*, 19:3929.

[45] Tuna, C., Unal, G., and Sertel, E. (2018). Single-frame super resolution of remote-sensing images by convolutional neural networks. *International journal of remote sensing*, 39(8):2463–2479.

[46] Van Etten, A., Hogan, D., Manso, J. M., Shermeyer, J., Weir, N., and Lewis, R. (2021). The multi-temporal urban development spacenet dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6398–6407.

[47] Wang, J., Gao, K., Zhang, Z., Ni, C., Hu, Z., Chen, D., and Wu, Q. (2021a). Multisensor remote sensing imagery super-resolution with conditional gan. *Journal of Remote Sensing*, 2021.

[48] Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Qiao, Y., and Change Loy, C. (2018). Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European conference on computer vision (ECCV) workshops*, pages 0–0.

[49] Wang, Z., Chen, J., and Hoi, S. C. (2021b). Deep learning for image super-resolution: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43:3365–3387.

[50] Yang, Y. and Newsam, S. (2010). Bag-of-visual-words and spatial extensions for land-use classification. In *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*, pages 270–279.

[51] Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., and Fu, Y. (2018). Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 286–301.

[52] Zhao, H., Gallo, O., Frosio, I., and Kautz, J. (2015). Loss functions for neural networks for image processing. *arXiv preprint arXiv:1511.08861*.

[53] Zhu, X. X., Hu, J., Qiu, C., Shi, Y., Kang, J., Mou, L., Bagheri, H., Häberle, M., Hua, Y., Huang, R., et al. (2019). So2sat lcz42: A benchmark dataset for global local climate zones classification. *arXiv preprint arXiv:1912.12171*.

[54] Zimmer, L., Lindauer, M., and Hutter, F. (2020). Auto-pytorch tabular: Multi-fidelity metalearning for efficient and robust autodl. *arXiv preprint arXiv:2006.13799*.

[55] Zoph, B., Vasudevan, V., Shlens, J., and Le, Q. V. (2018). Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8697–8710.

[56] Zou, F., Xiao, W., Ji, W., He, K., Yang, Z., Song, J., Zhou, H., and Li, K. (2020). Arbitrary-oriented object detection via dense feature fusion and attention model for remote sensing super-resolution image. *Neural Computing and Applications*, 32:14549–14562.

Appendix A

SENT-NICFI

This section describes how SENT-NICFI is constructed. First, the procedure for selecting the HR images is described, followed by the procedure of obtaining the matching LR images, as well as a short discussion of the dataset.

A.1 Selecting HR data

Locations have been selected manually using the Planet basemap explorer [1]. The selection criteria were to find diverse regions, including vegetation, desert habitation. Only complete quads were downloaded, excluding quads at the edges of regions covered by the NICFI study that have transparent pixels. The selected quad names are listed in Table A.1.

A.2 Obtain LR images

To complete the dataset, Sentinel-2 images corresponding to the Planet images have to be selected. The sentinelloader library[2] provides functionality to find Sentinel images based on criteria like the date, the desired bands, the region and cloud cover.

The GDAL library [12] makes it possible to extract the coordinates as well as the coordinate system from a GeoTiff image. In this case, both Sentinel-2 and PlanetScope use the same coordinate system for the images: WSG84, or EPSG 4326, thus the coordinates do not need to be converted. An area polygon is created with the extracted coordinates, and this polygon is passed to sentinelloader.

A narrow range of cloud coverage of 0%–10% is chosen, because the selected NICFI images do not have any clouds.

The sentinelloader package will then return images cropped to the specified region. For older images it is possible that they are still in the long term archive and not available for immediate download. Those files have to be requested and can be downloaded within 24 hours. It is likely that multiple images will be found for each HR image. The image with the least cloud cover is selected as the final LR image.

[1] `https://www.planet.com/products/basemap/`
[2] `https://github.com/flaviostutz/sentinelloader`

A.3 Resampling and splitting into tiles

The original PlanetScope images do not have an exact resolution of 5m, but rather 4.77m. However, an integer upsampling factor is required for this research. Thus, the HR images are downsampled to 5m using the GDAL library and bicubic downsampling algorithm.
The Sentinel-2 images obtained using the sentinelloader have a slightly lower resolution than 10m, so these have to be upsampled in order to reach the desired resolution. Once again, GDAL is used with the bicubic algorithm.

Next, because the original images are large, they are divided into smaller tiles to limit the memory usage of the super resolution model. The HR images are split into tiles of 200×200 pixels, the LR images into tiles of 100×100 pixels. The sizes of both images are not perfect multiples of these tile sizes, so the incomplete tiles at the edges of the images are discarded.

The final step is to bring the NICFI images closer to the Sentinel-2 images: these two image sources have different processing levels, and thus the colours appear differently. The processing of Sentinel-2 images is known to be helpful for analysis of the images. Thus, we aim to correct the colours of the NICFI images to more closely resemble the Sentinel-2 images. The simplest way to do this is by way of histogram matching, this is the manipulating of pixel values of an image such that the colour distribution histogram more closely matches the histogram of another image. This in effect will shift the colours of the image to be more similar to the reference image.

A.4 How to obtain the dataset

Due to the Terms of Use of the NICFI data, we cannot provide the full SENT-NICFI dataset. After applying for (free) access to the program at `https://www.planet.com/nicfi/`, the images used in this research can be downloaded using the script that is available at the repository at `https://github.com/JuliaWasala/autoSR-RS_SENT-NICFI`. The corresponding sentinel images can be downloaded from `https://universiteitleiden.data.surfsara.nl/index.php/s/dIrAZuB4PjcdUD2`. When all the images are obtained, one can use the code available at the repository to compile the images into a Tensorflow Dataset, or use as is.

Table A.1: The ids of the NICFI quads used for SENT-NICFI

Ecosystem	Country	Quad id
Cities	South Africa	1183 – 869
	Djibouti	1269 – 1090
	Zimbabwe	1200 – 920
	Mali	978 – 1096
	Angola	1137 – 956
Desert	Niger	1075 – 1128
	Somalia	1304 – 1079
	Namibia	1110 – 882
	Sudan	1110 – 882
	Niger	1032 – 1107
Forest	Congo	1134 – 1020
	Gabon	1102 – 1028
	Ghana	1013 – 1053
	Madagascar	1304 – 935
	Congo	1176 – 1016
Agriculture	Mozambique	1253 – 938
	Senegal	943 – 1098
	Nigeria	1056 – 1076
	Zambia	1171 – 946
	Somalia	1282 – 1037
Miscellaneous	Kenya	1235 – 1016
	Tanzania	1203 – 977
	Madagascar	1288 – 931
	Eritrea	1245 – 1109
	Angola	1128 – 953
Savanna	Central African Republic	1149 – 1085
	Burkina Faso	1019 – 1090
	Mali	982 – 1096
	Botswana	1188 – 895
	Burkina Faso	1005 – 1089

Appendix B

Preliminary experiments: RCAN_U

In this chapter we provide additional information on the preliminary experiments limiting with RCAN_U.

B.1 Experiment

Table B.1: Number of parameters per model. The number of parameters of the AutoML models is the average over all experiments.

Model	Parameters
WDSR	$601,975$
RCAN	$15.467,971$
RCAN_U	$2,641,795$
AutoSRCNN	$\approx 35,371$
AutoSR-RS	$\approx 1,255,938$

Table B.1 shows the number of trainable parameters of the baselines and AutoSR-RS. The numbers for WDSR and RCAN are obtained from a model with an upscaling factor of 2. The numbers for both AutoSRCNN and AutoSR-RS are approximate. They represent the means of the numbers of parameters of the models returned by AutoSR-RS in the experiments performed in Chapter 6. The number of parameters can vary significantly between runs.

Due to the large difference in numbers of parameters between WDSR and RCAN, we have performed a set of experiments on a single dataset with an undeep variant of RCAN with a single residual group and 20 residual blocks. We refer to this model als RCAN_U. The motivation for this experiment is that RCAN is much deeper than WDSR, and has a much longer running time. As shown in Chapter 7, RCAN mostly outperforms WDSR, but the performance gap between the two methods was not that large, that the question arose whether similar results could not be obtained with a simplified version of RCAN that would train much faster.

B.2 Results

The results of 5 runs of RCAN_U with dataset Cerrado are a PSNR of 39.51 ± 0.32 and SSIM of 0.9692 ± 0.0012, which is significantly better than the results of RCAN with the same dataset (as shown in Tables 7.1 & 7.2 in Chapter 7) following the Wilcoxon signed rank test[3].

Appendix C

Experimental settings

This chapter contains the details necessary to reproduce the experiments presented. Note that the running times are generally taken generously. RCAN runs about 10 times as slow as WDSR, but the same maximum running times were kept for simplicity.
The maximum running times for the AutoML methods were set to be generous, due to the variance in running times of trials. More successful trials will run for longer than less successful ones that will be stopped early by AutoKeras. Individual runs (consisting of multiple trials) can take much longer or shorter than others.

Table C.1: The settings of the baseline experiments. Time is the maximum running time.

Baseline	Dataset	Epochs	Time
RCAN [51]	Cerrado	60	2h
	UC Merced	100	24h
	So2Sat	18	120h
	OLI2MSI	50	24h
	SENT-NICFI	50	24h
WDSR [11]	Cerrado	60	2h
	UC Merced	100	24h
	So2Sat	18	120h
	OLI2MSI	50	24h
	SENT-NICFI	50	24h
Baseline	Dataset	Trials	Time
AutoSRCNN	Cerrado	10	2h
	UC Merced	10	24h
	So2Sat	10	48h
	OLI2MSI	10	48h
	SENT-NICFI	10	24h
AutoSR-RS	Cerrado	20	5h
	UC Merced	20	48h
	OLI2MSI	20	100h
	SENT-NICFI	20	48h

Appendix D

Modifications to AutoKeras

This section gives an overview of the modifications made to AutoKeras for this research. This information is limited to the changes made to the autokeras source code, as made available at `https://www.github.com/JuliaWasala/autokeras`. The implementations of AutoSR-RS and associated custom blocks are found at `https://www.github.com/JuliaWasala/AutoSR-RS_SENT-NICFI`.

D.1 Image output

Output blocks in AutoKeras are called *heads*, and can be found in the same folder as the implementation of the blocks. Specifically for this research an ImageHead is implemented, which is a convolutional layer with the default number of filters set to 3. The number of filters must be equal to the required number of channels in the output image. The kernel size is set to 1, such that it just passes along the information to the optimizer.

The loss function is set to L1, or mean absolute error.

D.2 Custom metrics

Super resolution performance is commonly measured in PSNR and SSIM (see Section 6). These metrics are not integrated in AutoKeras, but are implemented in tensorflow[1][2]. As per the AutoKeras documentation, it should be possible to pass custom metrics[3]. Unfortunately, in the AutoKeras version used for this research (`1.0.16.post1`). Thus, some changes have been made to the source code in order to make this possible.

The open GitHub issue can be found at `https://github.com/keras-team/autokeras/issues/1400`. The changes mostly involve passing a `custom_objects` dictionary to Keras save and load functions. These changes have been made at the following locations:

- autokeras.AutoModel.predict
- autokeras.AutoModel.tuner.get_best_model
- autokeras.AutoModel.export_model
- autokeras.AutoModel.evaluate

[1] `https://www.tensorflow.org/api_docs/python/tf/image/psnr`
[2] `https://www.tensorflow.org/api_docs/python/tf/image/ssim`
[3] `https://autokeras.com/tutorial/faq/#how-to-use-customized-metrics-to-select-the-best-model`

Appendix E

Sample predicted images

This appendix contains image samples from Cerrado [32], UC Merced [50], OLI2MSI [47] and SENT-NICFI, as well as predictions from WDSR [11], RCAN [51], AutoSRCNN and AutoSR-RS.

Figure E.1: Examples of the test set of the Cerrado dataset. Original images from the dataset are shown, as well as images predicted by WDSR, RCAN, AutoSRCNN and AutoSR-RS.

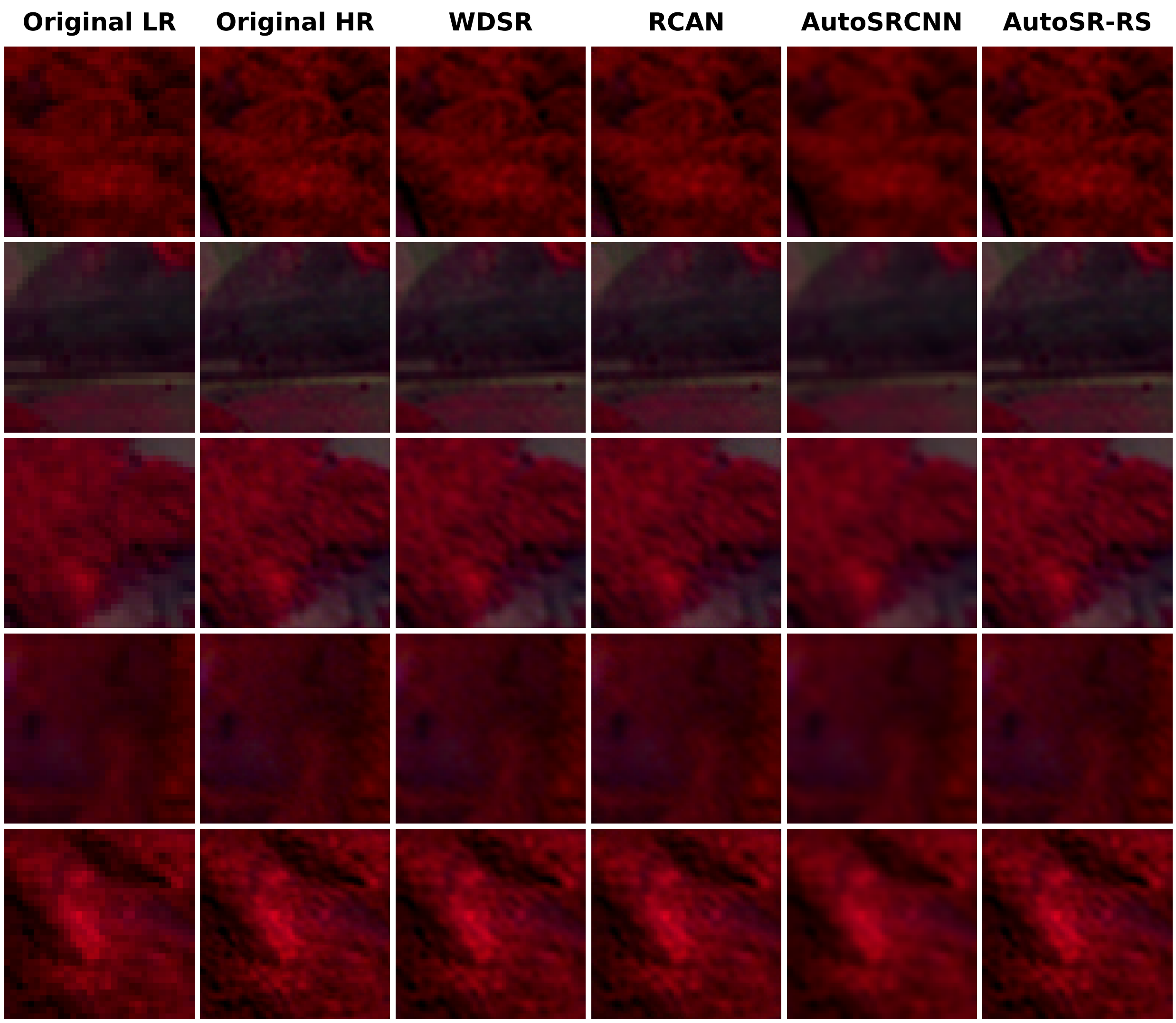

Figure E.2: Examples of the test set of the UC Merced dataset. Original images from the dataset are shown, as well as images predicted by WDSR, RCAN, AutoSRCNN and AutoSR-RS.

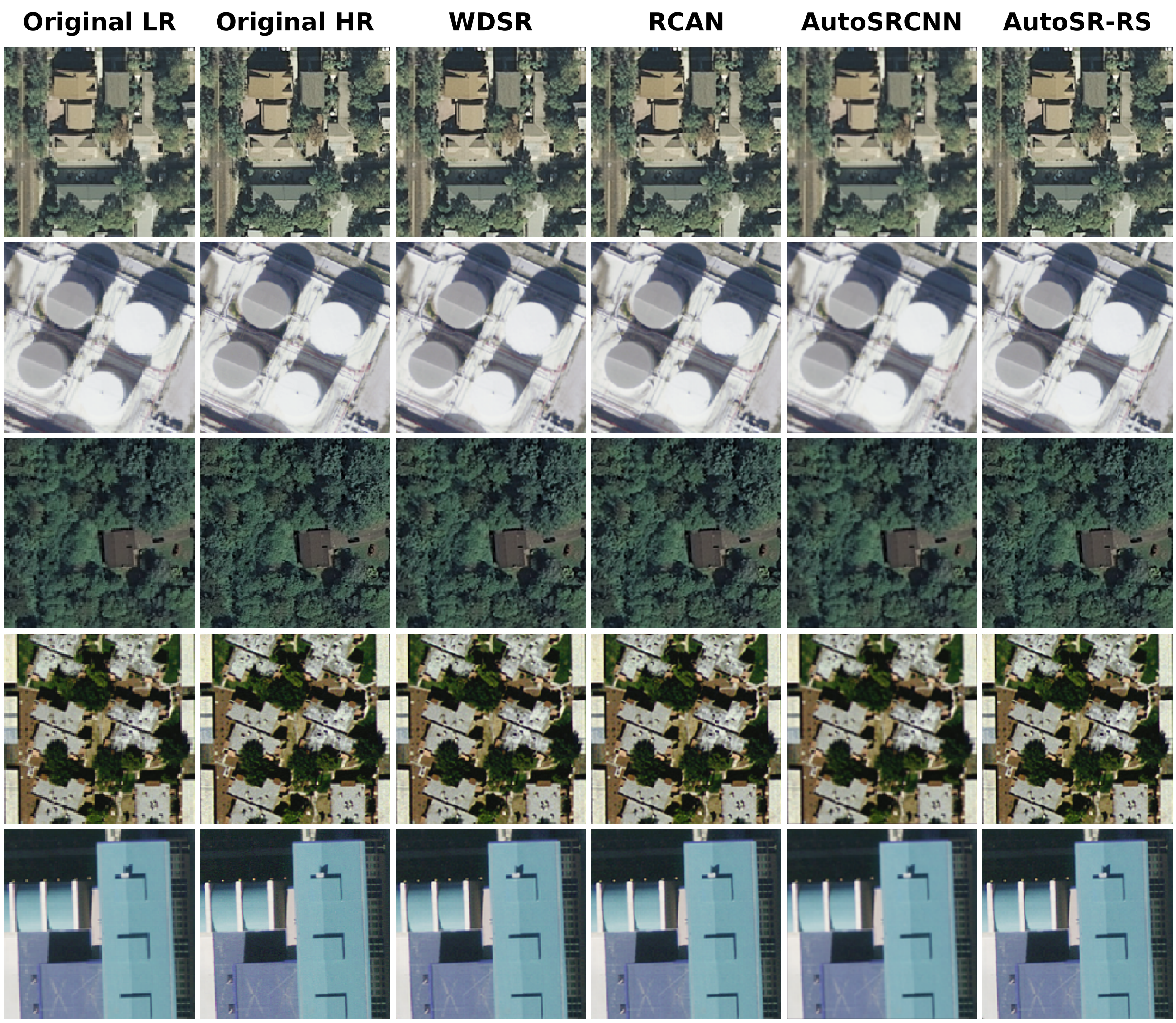

Figure E.3: Examples of the test set of the OLI2MSI dataset. Original images from the dataset are shown, as well as images predicted by WDSR, RCAN, AutoSRCNN and AutoSR-RS.