Article

AutoSR4EO: An AutoML Approach to Super-Resolution for Earth Observation Images

Julia Wąsala [1,*], Suzanne Marselis [2], Laurens Arp [1], Holger Hoos [1,3], Nicolas Longépé [4] and Mitra Baratchi [1]

1 Leiden Institute for Advanced Computer Science (LIACS), Leiden University, 2333 CA Leiden, The Netherlands
2 Institute of Environmental Sciences (CML), Leiden University, 2333 CL Leiden, The Netherlands
3 Chair of AI Methodology (AIM), RWTH Aachen University, 52062 Aachen, Germany
4 Φ-Lab Explore Office, European Space Research Institute (ESRIN), European Space Agency (ESA), 00044 Frascati, Italy
* Correspondence: j.wasala@liacs.leidenuniv.nl;

Abstract: Super-resolution (SR), a technique to increase the resolution of images, is a pre-processing step in the pipelines of applications of Earth observation (EO) data. The manual design and optimisation of SR models that are specific to every possible EO use case is a laborious process that creates a bottleneck for EO analysis. In this work, we develop an automated machine learning (AutoML) method to automate the creation of dataset-specific SR models. AutoML is the study of the automatic design of high-performance machine learning models. We present the following contributions. (i) We propose AutoSR4EO, an AutoML method for automatically constructing neural networks for SR. We design a search space based on state-of-the-art residual neural networks for SR and incorporate transfer learning. Our search space is extendable, making it possible to adapt AutoSR4EO to future developments in the field. (ii) We introduce a new real-world single-image SR (SISR) dataset, called SENT-NICFI. (iii) We evaluate the performance of AutoSR4EO on four different datasets against the performance of four state-of-the-art baselines and a vanilla AutoML SR method, with AutoSR4EO achieving the highest average ranking. Our results show that AutoSR4EO performs consistently well over all datasets, demonstrating that AutoML is a promising method for improving SR techniques for EO images.

Keywords: super-resolution; neural architecture search; automated machine learning

1. Introduction

Many applications require high-resolution satellite imagery, such as land and forestry management, agricultural observations and crop monitoring [1–3], high-accuracy mapping, civil engineering and disaster relief and emergency response operations [4]. Technological advancements have increased the spatial resolution of optical images collected by satellites. Still, different factors constrain this resolution, including the size, power and cost of the satellites and trade-offs between swath width and spatial vs. temporal resolution.

Super-resolution (SR) techniques increase the spatial resolution of images with the goal of improving performance in downstream EO use cases, such as object detection [5–7]. Three requirements are considered when selecting SR models fitted to downstream Earth observation (EO) tasks.

Firstly, the SR method needs to be able to model the data at hand. Different approaches have been designed for different types of data. Edge-maintaining SR models work well for imagery with many sharp edges, such as buildings. However, other models are better suited for smoother images with more gradients than sharp edges (e.g., large bodies of water or desert landscapes).

Secondly, the choice of training dataset impacts the final results. SR models can be trained with images from other sensors if we lack the high-resolution reference images needed for supervised learning. This process of transferring knowledge by training on one dataset and evaluating on another is called transfer learning. However, results can degrade when we train a model on a dataset that is very different from the target dataset. For instance, trained models transfer poorly to the target data [8] if the difference in spatial resolution is too large. This issue relates to domain transfer and arises from differences in image characteristics, like the modulation transfer function (MFT), signal-to-noise ratio (SNR), spatial resolution and spectral characteristics.

Thirdly, we need the ability to evaluate the performance of SR frameworks in different pipelines. SR frameworks—either single, fixed models or algorithms that can automatically design SR models—need to be versatile because SR is a low-level computer vision task followed by high-level tasks with different requirements for the model, data and evaluation.

Current approaches to SR (e.g., SwinIR [9], DeCoNAS [10] and CARN-M [11]) fail to fulfil the requirements related to EO pipeline design.

Firstly, a single, well-performing SR model is often used for all scenarios (Figure 1a).

Secondly, many SR methods (e.g., SRDCN [12] and DMCN [13]) are trained and evaluated on synthetic datasets because these are easier to obtain than real-world datasets. Real-world datasets require matching images from different sensors, as shown in Figure 2. However, the performance of a model trained on synthetic data overestimates the model's performance on real-world data [14]. The simple downsampling procedures that are used for creating synthetic data are unable to capture the complicated patterns occurring in real-world data. The complex systems encountered in EO produce data that are often noisy and unpredictable. Differences in reflectance values in low-resolution inputs and high-resolution ground truths may bias the loss and training process, and the time lag between two matching images, the presence of clouds and small pixel shifts due to image co-registration all further complicate the picture.

Manual model design and training data selection can overcome these issues, but carrying this out for every target and application (Figure 1b) significantly increases the time and effort required for designing end-to-end pipelines.

We can satisfy the three requirements of good SR systems by automating the process of SR model design (Figure 1c) using automated machine learning (AutoML) approaches. AutoML is a recently growing research area studying the automatic design of high-performance machine learning models. Neural architecture search (NAS) systems are a specific group of AutoML systems that automatically design neural networks to find better architectures.

NAS systems consist of three components [15], as shown in Figure 3. The first component in creating neural networks is a *search space*, which is the set of all available design choices encoded by hyperparameters, including architectural parameters, like the number and type of layers in the neural network, and training parameters, like the learning rate.

The second component is a *search strategy*, which determines how to traverse the search space and selects suitable combinations of hyperparameter values. These combinations of values determine the architecture of the candidate network architecture to be evaluated. The vast search spaces typically encountered in NAS systems require sophisticated search strategies for effective exploration.

The third component is an *evaluation strategy*, which efficiently assesses the candidate network until the search strategy finds a suitable architecture, i.e., the best architecture found after a pre-determined number of evaluations or the first architecture to reach a target metric, like minimum accuracy score.
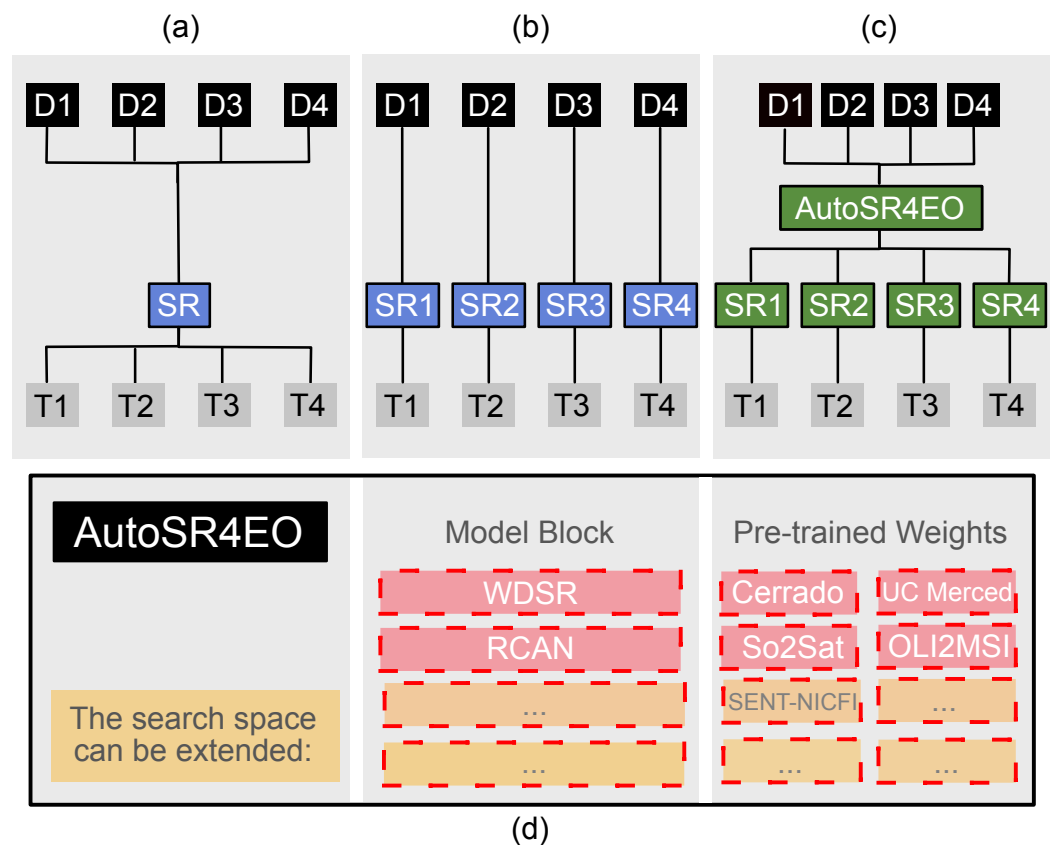
Figure 1. Illustration of the three options for selecting SR methods, where **D** denotes a dataset, **T** is a downstream task and **SR** represents an SR method: (**a**) the scenario where the same model is used for all pipelines, possibly with lower performance than desired; (**b**) the case when an SR model is manually selected or designed for each pipeline, which is time-consuming; (**c**) our proposed approach, AutoSR4EO, which can automatically construct a custom neural network for each dataset; (**d**) the currently available model blocks and sets of pre-trained weights in our proposed approach, the search space of which could easily be extended in the future.
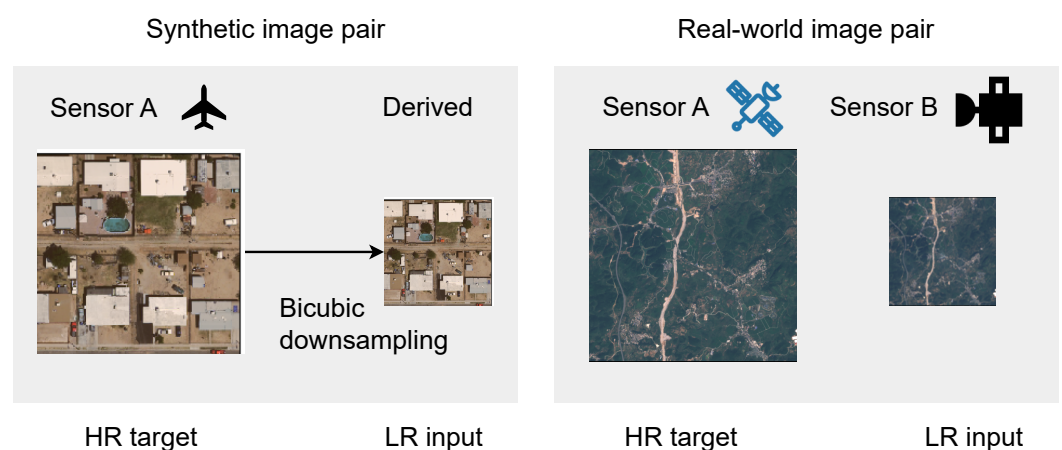

Figure 2. A synthetic image pair compared to a real-world image pair. The synthetic pair is obtained by bicubically downsampling an image using a pre-determined scaling factor. The real-world pair consists of two co-located images obtained by different sensors: (**left**) UC Merced [16]; (**right**) OLI2MSI [17].
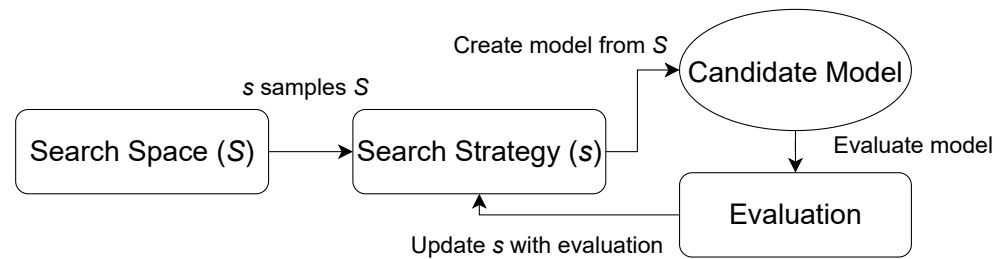
Figure 3. Diagram of the components of an NAS framework. A search strategy *s* samples candidate models from the search space *S*. The candidate model is evaluated. The search strategy is updated with the evaluation results.

While NAS systems create high-performance neural architectures, several challenges arise when creating NAS systems for EO tasks. Several NAS approaches have been proposed in the past few years, including approaches for the EO domain. To the best of our knowledge, none have yet been applied to SR for EO images. Moreover, designing a good search space for this task is a challenging problem. On the one hand, the search space must be large and diverse enough to design well-performing SR models for each dataset; on the other hand, if the search space is too large, it can become too computationally expensive to search.

We address these challenges and create an NAS SR approach for EO. Our contributions are as follows:

- We propose AutoSR4EO, the first AutoML system for SR for EO, by designing a customised search space based on state-of-the-art research in SR;
- We further propose to use pre-trained weights generated from EO datasets in AutoSR4EO to facilitate knowledge transfer and speed up the training of SR methods for EO tasks;
- We introduce a vanilla baseline AutoML system for SR, dubbed AutoSRCNN, based on existing NAS search spaces consisting exclusively of convolutional layers, which is useful as a lower-bound baseline for comparison to future AutoML approaches for SR;
- We evaluate the performance of AutoSR4EO on four EO datasets in terms of peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) and compare our methods to four state-of-the-art SR methods and AutoSRCNN;
- We introduce SENT-NICFI, a novel SR dataset consisting of paired images obtained by Sentinel-2 [18] and Planet [19].

2. Related Work

In this section, we discuss the related work on the topics of SR and AutoML for EO tasks. We conclude with a discussion of the relevance of this work.

2.1. Super-Resolution

Super-resolution can be addressed using either traditional model-based approaches (e.g., regression [20–22] and kernel estimation [23]) or, more recently, neural networks. Deep learning methods, which train neural networks to map low-resolution images to higher-resolution images, are currently state-of-the-art in super-resolution [24]. These methods range from classical convolutional neural networks (CNNs) (e.g., SRCNN [25–27]) to newer generation SR methodologies that use residual neural networks, generative adversarial networks (GANs) and vision transformers.

Residual approaches (including EDSR [27], WDSR [28] and CARN-M [11]) train deeper networks than earlier fully convolutional approaches, like SRCNN [25], using residual connections. Residual connections preserve the information of previously extracted features, alleviating the problem of features disappearing as network depth increases [29].

Attention-based models (e.g., RCAN [30], AWSRN [31] and HBPN [32]) weigh features according to their importance using channel attention.

GAN-based approaches are gaining popularity in SR. These architectures consist of a generator and a discriminator that are trained alternatingly. One of the first approaches based on a GAN was SRGAN [33]. Other GAN-based SR methods include EEGAN [34], ESRGAN [35], ESRGAN+ [36], EnhanceNet [37] and OpTiGAN [38]. GANs are also used for SR for EO tasks: MA-GAN [39] combines a GAN with multi-attention and a pyramidal structure; TE-SAGAN [40] reduces artefacts and improves texture with self-attention and weight normalisation; NDSRGAN [41] uses pairs of images taken at different altitudes instead of bicubically downsampled images.

GANS are difficult to include in automated frameworks as they face training challenges, like mode collapse, non-convergence, instability and vanishing gradients [42], and they come with the risk of hallucination. NAS frameworks that are specifically developed for GANs do exist (e.g., [43–45]), but our goal was to create a rich search space comprising different types of architectures. The two-network architectures of GANS make it very challenging to include any other types of architectures because of the significant differences in both training and architecture.

Other notable work comes from the recent area of vision transformers (e.g., [46–48]). Liang et al. [9] applied vision transformers to super-resolution, taking inspiration from the Swin Transformer [49], and achieved state-of-the-art results while using similar amounts of data as convolutional baselines.

Recently, SR approaches using diffusion techniques have been proposed. For instance, Han et al. [50] used diffusion to create detailed super-resolved images and used feature distillation to reduce inference time. Wu et al. [51] used diffusion together with contrastive learning to estimate the degradation kernels of images, without making assumptions about the kernels. Ali et al. [52] combined diffusion models with vision transformers in a two-step approach.

We took a different approach to SR. Instead of designing a new SR algorithm, we designed a framework that can automatically generate a network architecture for a given dataset. The advantage of this approach is that architectures can be created and optimised automatically for any dataset at hand. Moreover, such an approach goes beyond model selection and can yield new architectures. We built our search space based on existing residual- and attention-based SR approaches.

2.2. AutoML for EO Tasks

Auto-sklearn [53,54], AutoGluon [55] and FLAML [56] are examples of popular off-the-shelf AutoML systems for tabular data. These frameworks allow users to easily optimise their machine learning pipelines using classic machine learning algorithms. AutoKeras [57] and Auto-Pytorch [58] automatically design neural networks and also support image data.

The EO community is interested in using AutoML in their applications; for example, César de Sá et al., compared the performance of auto-sklearn and AutoGluon to a manual design approach for grass height estimation [59]. In atmospheric science, Zheng et al., employed the FLAML framework to estimate particulate matter concentrations in satellite measurements [60]. In image classification, Palacios Salinas et al., proposed a network architecture search (NAS) system optimised for classifying EO images with blocks that were pre-trained on four EO datasets (e.g., [16,61–63]) by customising the search space of AutoKeras [64]. Another approach for object recognition in EO images was presented by Polonskaia et al., who proposed an automated evolutionary NAS approach for designing CNNs implemented in Auto-Pytorch [65].

Even though existing AutoML systems have been successfully applied to EO tasks, the available frameworks have yet to cover all tasks related to EO data. As we show in this work, we can create better and more accurate ML pipelines for EO data by extending or creating AutoML frameworks that focus on the requirements for EO tasks.

NAS Systems for SR

One of the first examples of an NAS system for SR, MoreMNAS [66], was designed for mobile devices by optimising both the peak signal-to-noise ratio and FLOPS. The search strategy is a separate reinforcement learning (RL) neural network that selects candidate networks. Similarly, FALSR [67] uses an RL search strategy and evolutionary search at the micro and macro levels. A downside of using a neural network for the search is the added overhead of training the network in addition to training the candidate architecture.

MoreMNAS and FALSR have relatively high search costs: 56 and 24 GPU days, respectively [10]. DeCoNAS [10] only requires 12 GPU hours for the search as it uses parameter sharing during the training of candidate networks. MBNASNet [68] further improves these results because it captures multi-scale information better with the help of its multi-branch structure. Nevertheless, state-of-the-art SR techniques still outperform current NAS approaches in terms of PSNR and SSIM [24].

A key aspect differentiates these works from our proposed methods: we search and train on the target dataset. In most super-resolution works, the generated models are not optimised for the target dataset [10,66–70]. Instead, the networks are often searched and trained on the DIV2K dataset [71] (a large-scale dataset with multiple scaling factors for the development of SR methods) and evaluated on a different set of benchmark datasets (e.g., Set5 [72], Set 14 [73], and Urban100 [74]). This is less computationally expensive than repeating the search and training with multiple datasets, but it comes with the risk that the resulting pre-trained model is not suitable for each task because it is optimised for another dataset.

2.3. Relevance of Our Work

Previous work has demonstrated the successes achieved using NAS systems in the context of EO image classification. NAS systems may enable similar gains for other EO tasks, including SR. To the best of our knowledge, our work is the first to propose an NAS method for SR specifically for EO data. Our goal is to leverage an NAS system to improve the current state-of-the-art SR approaches for EO images by automatically designing a network for each dataset. While others have mainly considered transferring knowledge from natural image datasets, such as ImageNet [75], we studied how transfer learning can be efficiently used by considering transferable knowledge within the EO domain.

3. Materials and Methods

In this section, we describe our methods, the data used in this study and the experimental setup.

3.1. Methods

We now describe our new automated SR method for EO images, called *AutoSR4EO*. Our goal was not to propose a new SR neural network architecture; instead, we aimed to devise a system that can produce a new and high-performance neural network architecture for each dataset. The three main components of an NAS system (as described in Section 1) are (i) a search strategy, (ii) a search space and (iii) an evaluation procedure (also called performance estimation strategy) [15].

To reach our goal of creating an NAS system for SR, our main focus was on the first component: a new customised search space specifically designed for the task of SR. We designed this search space by including all design choices (i.e., the hyperparameters involved in designing neural networks) from previously proposed SR architectures (see Section 3.3.1).

The second component is the search strategy, which is used to sample from this search space. Different search strategies can be used for this purpose, including Bayesian optimisation or random search. We propose using an existing search strategy (see Section 3.1.2). The search strategy may stop the search early if it does not find new candidate architectures that improve the validation loss of the best candidate found so far.

The third component of NAS systems, the evaluation procedure for efficiently evaluating sampled architectures, is simple: candidate architectures are created and trained one by one using early stopping on the validation set until a maximum number of epochs (100) is reached. The final validation performance of each candidate network is saved. The best candidate is retrieved and evaluated on the test set when the maximum number of trials is reached or when the search strategy stops the search early due to lack of improvement.

We implemented our methods in AutoKeras, like the work on image classification by Palacios Salinas et al. [64]. The AutoKeras library is a natural choice for SR because it already contains functionality for image tasks.

The combination of a search strategy and our custom search space leads to the selection of a pre-determined number of candidate architectures, referred to as trials. In the following subsections, we specify the SR blocks that are the basis of the search space of AutoSR4EO and we describe the search strategy.

3.1.1. Search Space

AutoKeras implements a search space in the form of configurable blocks that are used as the basic units to build candidate architectures. A block is a smaller collection of possible design choices. For instance, the ConvBlock consists of convolutional layers and corresponding hyperparameters, like the number of kernels and filters. Multiple blocks combine to create larger search spaces. The search strategy builds candidate networks by selecting and stacking different blocks, and the blocks themselves can also be morphed, i.e., the hyperparameters of the blocks can be optimised.

AutoKeras offers different types of blocks that are mostly used for image classification, such as ResNet [76]. However, SR is a complex image task that requires different architectures.

To define a new SR framework for EO tasks, we propose a search space that includes relevant architectural hyperparameters for creating SR networks. We define this search space based on existing deep learning models for SR tasks, namely, RCAN [30] and WDSR [28]. The model blocks form the foundation of AutoSR4EO. We selected WDSR and RCAN as the basis of AutoSR4EO because of the representative nature of these methods in the domain of non-GAN-based SISR methods. Both WDSR and RCAN achieve high performance for SR tasks with natural images [77–80].

In our implementation, the RCAN block had just 1 residual group instead of the original 10 because our initial experiments indicated that the version with 1 residual group achieved significantly higher scores than the original.

Figure 4 illustrates the search space of AutoSR4EO. The search strategy selects a type of model block and modifies it by choosing the number of residual blocks in the selected model. We based the ranges of the numbers of residual blocks on the original papers [28,30]: the maximum number of blocks for RCAN in search space *S* was 20, while in search space *L*, the maximum number of 40 reflected the range of residual blocks in WDSR.
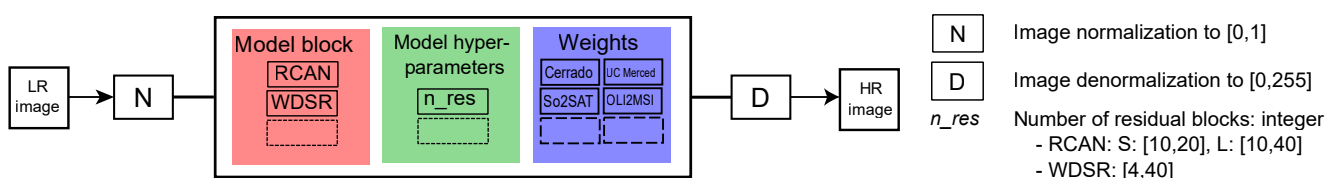

Figure 4. The architecture and search space hyperparameters of search spaces *S* (small) and *L* (large) in AutoSR4EO. The large search space allows for more residual blocks in RCAN. The range for WDSR is equal in both search spaces. The search spaces can be abstracted into three components that are tuned automatically by the search strategy: (i) the model block; (ii) the model hyperparameters; (iii) the set of pre-trained weights. The currently available options are shown for each component, but all three components can be extended.

Finally, the search strategy selects a set of pre-trained weights for the residual blocks. The shape of the upscaling module at the end of the network depends on the upscaling

factor. This limits the usage of pre-trained weights to weights obtained from datasets with the same upscaling factor as the training dataset. Therefore, we restricted the pre-trained weights to the residual stack to make the weights transferable to datasets with other scaling factors. We trained WDSR and RCAN on EO datasets to obtain these weights.

Figure 4 shows a schematic of AutoSR4EO and the hyperparameters defining its search space.

Design choices. In practice, a user is more likely to have access to a model trained on a different dataset than the target dataset. Therefore, AutoSR4EO cannot select pre-trained weights from the dataset on which it is trained and evaluated.

We set hyperparameters, such as the kernel sizes, the number of filters, the learning rate and residual block hyperparameters, including linear scaling factor and expansion, by following the recommendations of the authors of RCAN and WDSR. Thus, we limited the search space to increase the likelihood of finding high-performing solutions.

The maximum number of blocks was one of the most important design choices in defining this search space. We investigated two search spaces: *S*, with a maximum number of 20 residual blocks in the RCAN model block, and *L*, with a maximum number of 40 residual blocks in the RCAN block. These choices determine the maximum depths of the models generated by AutoSR4EO: low values result in shallower models and higher values result in deeper models. Deeper models may model more complex patterns, but they are also more prone to overfitting.

3.1.2. Search Strategy

Search strategies sample hyperparameter values from search spaces to select candidate network architectures. We used AutoKeras' default search strategy: a combination of greedy and random search. This stops further exploration early if the search converges to a local optimum. In each trial, the search strategy builds a candidate network by sampling the search space and the blocks and block hyperparameters are selected and combined into a network. The network is then trained and evaluated. AutoKeras saves the performance and returns the highest-scoring neural network after the final trial. Other NAS frameworks (e.g., NNI [81]) could implement the same concepts. The networks generated by AutoSR4EO vary in depth and the number of parameters, depending on the choices of the search strategy.

Time Complexity. The time complexity of NAS systems is $O(nt)$, where n is the number of trials and t the average trial time [57]. More trials may be necessary to achieve convergence if the search space size is increased. However, the number of trials is much lower than the number of configurations in the search space; therefore, we doubted that the number of trials would increase significantly. Combined with the linear complexity of NAS systems, the cost of adding new methods is relatively small.

3.2. Data

We selected both types of datasets that are used for SR: synthetic datasets created by downsampling existing images and real-world datasets created by matching acquired images from different sensors. We used the five datasets shown in Table 1 for both evaluation and pre-training.

3.2.1. Synthetic Datasets

Table 1 lists the EO image datasets used to create the synthetic data. The low-resolution images were generated by downsampling the images in the data sources using a bicubic kernel [28–30,71] with a scaling factor of 2 (i.e., the resolution of the high-resolution images is twice that of the low-resolution images). We used the following data sources:

- **UC Merced** [16], which is a dataset for land use image classification containing 21 different classes of terrain in the United States;
- **So2Sat** [62], which is a dataset comprising images of 42 different cities across different continents. The RGB subset of So2Sat consists of Sentinel-2 images. We used this

dataset exclusively to generate the pre-trained weights because the large size of the dataset made it infeasible to evaluate AutoSR4EO on this dataset using the current experimental setup;

- **Cerrado-Savanna** [82], which consists of images of Brazil's Serra do Cipó region and has a wide variety of vegetation and high variations between classes.

3.2.2. Real-World Datasets

The simple downsampling procedure used to generate synthetic data can oversimplify differences between high- and low-resolution images from different sensors. We can avoid this problem by using real-world datasets with different resolutions. However, these datasets are much more difficult to obtain due to the limited availability of freely accessible satellite data with different resolutions. Additionally, neural networks have to account for differences in images that can occur due to non-strict overlapping between the spectral bands of different sensors and different signal-to-noise ratios. Discrepancies can also occur during radiometric calibration when estimating reflectance from radiance. Additionally, atmospheric conditions can change over time and data providers provide images at different production levels, for instance, either top of atmosphere (TOA) or bottom of atmosphere (BOA) [83]. We used the following two real-world datasets:

- **OLI2MSI**, proposed by Wang et al. [17], which consists of low-resolution images taken by Landsat-8 and Sentinel-2 of a region in Southwest China and contains 10,650 training pairs;
- **SENT-NICFI**, which is a novel SR dataset we constructed using images from Sentinel-2 and Planetscope that were taken in June 2021. The Planetscope images are part of the NICFI programme. We selected images of countries around the equator, covering an area of about 45 million square kilometres. We selected high-resolution (HR) images from five scenes from each of the following ecosystems from countries on the African continent: urban, desert, forest, savanna, agriculture and miscellaneous (i.e., outside of the previous categories). The low-resolution (LR) images were Sentinel-2 images from around the same month, producing 12,000 training pairs. We aligned the HR image colours to the LR images via histogram matching. We provide code for the reconstruction of this dataset.

Table 1. Overview of the data sources and the synthetic (Syn.) and real-world datasets (Real). The resolution is given in metres (m) and the image size is given in pixels (px). The size of the synthetic LR images is left out because these images were derived by bicubic downsampling. * The bands are near-infrared (NIR), green (G) and red (R).

Dataset	Type	Source	Bands	# Images	Resolution (m)	LR Size (px)	HR Size (px)
UC Merced [16]	Syn.	USGS (aerial)	RGB	590 k	0.3	-	256 × 256
So2Sat [62]	Syn.	Sentinel-2	RGB	376 k	10	-	32 × 32
Cerrado-Savanna [82]	Syn.	RapidEye	NIR, G, R *	27 k	5	-	64 × 64
OLI2MSI [17]	Real	Landsat and Sentinel-2	RGB	10.65 k	30 and 10	160 × 160	480 × 480
SENT-NICFI	Real	Sentinel-2 and Planetscope	RGB	2.2 k	10 and 5	100 × 100	200 × 200

3.3. Experiments

This section describes the baselines, training configurations, evaluation procedures and experimental setup.

3.3.1. Baselines

We considered the following baseline methods:

- **RCAN** [30], which introduces channel attention modules that give more weight to informative features. The network consists of stacked residual groups, with an

upscaling module at the end of the residual stack after merging the two branches. We used the Keras implementation of RCAN, made available by Hieubkset (https: //github.com/hieubkset/Keras-Image-SR, accessed on 28 June 2022);

- **WDSR** [28], which is a residual approach, like RCAN, but the residual blocks lack the channel attention mechanism. The two branches are merged after upsampling on each of them. Convolutions with weight normalisation replace all convolutional layers. We used the Keras code for the WDSR model released by Krasser (https: //github.com/krasserm/super-resolution, accessed on 28 June 2022);

- **SwinIR** [9], which is a state-of-the-art adaptation of the Swin Transformer [49] for image reconstruction and super-resolution. We used the DIV2K and Flickr2K pre-trained models (https://github.com/JingyunLiang/SwinIR, accessed on 1 September 2023). We followed the original work and selected the "Medium" configuration, which is comparable in complexity to RCAN, with a patch size of 64 and a window size of 8. We directly inferred on our test sets (as defined in Section 3.2, as in the original work, but with natural image test sets (Set5 [72], BSD100 [72], Set14 [73] Urban100 [74] and Manga109 [84]);

- **HiNAS** [69], which is a state-of-the-art NAS framework for super-resolution and image denoising. It is computationally efficient due to its gradient-based search and architecture sharing between layers. We searched and trained the best networks for the upscaling factors of 2 and 3, following the original work. The evaluation was the same as for SwinIR;

- **AutoSRCNN**, which is an AutoML SR approach inspired by SRCNN [25]. We implemented AutoSRCNN exclusively with convolutional layers, without residual connections, pre-trained weights or specialised blocks. The search space (shown in Figure 5) is much smaller than that of AutoSR4EO; thus, it served as a control to ensure that a more extensive search space is beneficial for solving the problem of SISR for EO images. AutoSRCNN found networks comparable to SRCNN, which are less complex than the state-of-the-art alternatives. As such, AutoSRCNN served as a vanilla baseline to AutoSR4EO.

Both WDSR and RCAN are based on residual neural networks. Figure 6 shows diagrams of the residual blocks of WDSR and RCAN.
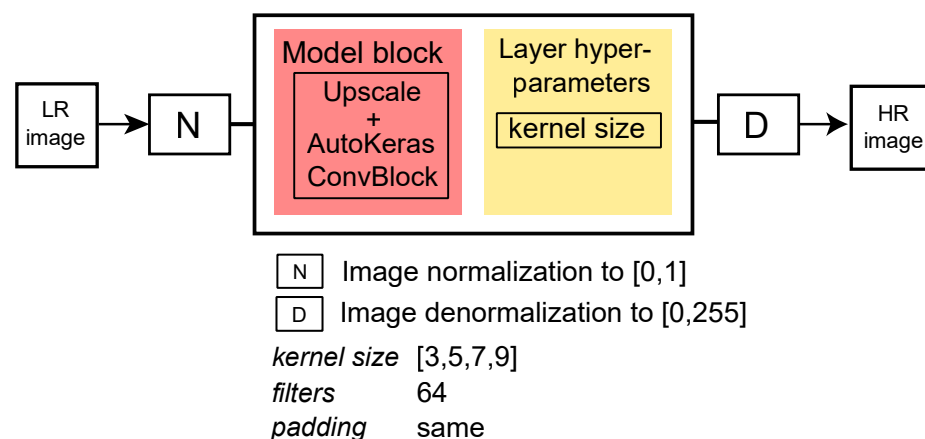

Figure 5. The fixed architectural hyperparameters and the search space hyperparameters that are changed during the AutoSRCNN search. AutoSRCNN scales images up at the beginning, as in SRCNN [25], followed by convolutional layers (implemented using AutoKeras' ConvBlock).
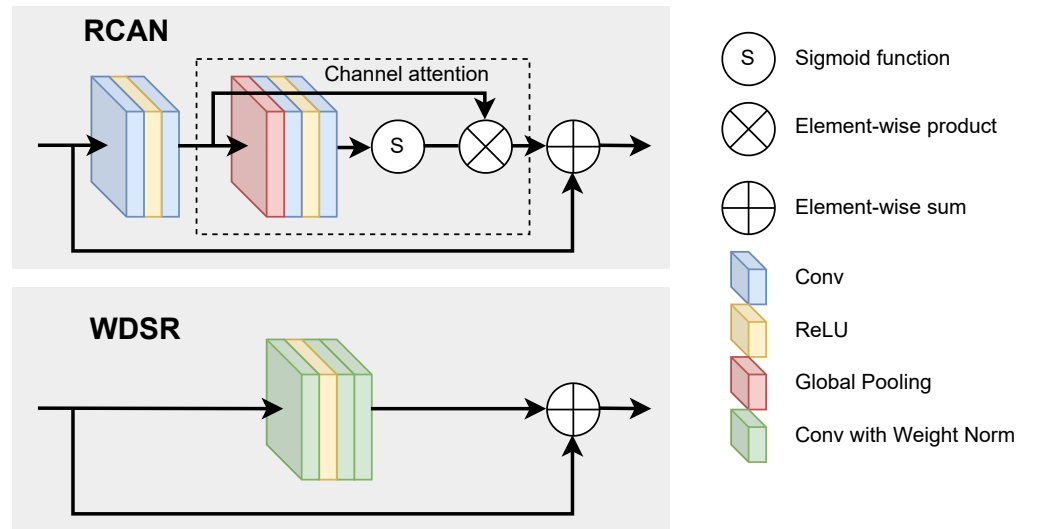
Figure 6. The architectures of the residual blocks of RCAN [30] and WDSR [28]. Both use blocks with residual connections, where the output of the residual block is the sum of the input of the block and the final result within the block. The sizes of the kernels and the numbers of filters are left out for simplicity. Figure created by authors.

3.3.2. Training Details

We set the number of epochs and batch sizes per method and dataset, depending on the validation loss, memory and time limit for the computational cluster used in our experiments. We used early stopping with a patience of 10 epochs. AutoSRCNN, AutoSR4EO$_S$ and AutoSR4EO$_L$ evaluated a maximum of 20 candidate networks per run, with a maximum of 100 epochs per candidate network. We used L1 for all methods because it yielded better results than L2 loss for SR [85]. The networks were trained on images with three channels (the spectral bands per dataset are listed in Table 1).

3.3.3. Evaluation

We evaluated AutoSR4EO and the baseline methods using two metrics: peak signal-to-noise ratio (PSNR), a pixel-wise metric related to the MSE, and the structural similarity index measure (SSIM), a perception-based metric that considers the contrast, luminance and structure of images to better reflect human visual interpretation. Both metrics are widely used for evaluating SR approaches [80,86–88]. The PSNR is given by

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

where I is the ground truth image, $\hat{I}$ is the super-resolved image, L is the maximum pixel value (which is 255 in this case) and N is the number of pixels. The SSIM is given by

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

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

3.3.4. Experimental Setup

The experiments were run on two GeForce RTX 2080TI GPUs with 10 GB of CPU RAM. We differentiated between the two types of baseline experiments: WDSR, RCAN and AutoSRCNN were each trained and evaluated on the datasets presented in Section 3.2.

We trained and evaluated each combination of baseline method and dataset five times to facilitate a more thorough comparison between AutoSR4EO and the baseline methods underlying its search space. The results of the experiments were compared by first bootstrapping the results with 1000 samples of size 3, followed by a Wilcoxon signed-rank test [89] for non-normally distributed samples. We used the pre-trained models SwinIR and HiNAS and evaluated them on UC Merced, Cerrado, OLI2MSI and SENT-NICFI. This evaluation strategy, common in SR, yielded a single result per combination of method and dataset. Though it is possible, we did not fine-tune the models, since this is not customary in the evaluation of NAS baselines. For instance, both HiNAS and SwinIR were trained on the DIV2K [71] dataset and evaluated on different datasets without fine-tuning (Set5 [72], BSD100 [72], Set14 [73], Urban100 [74] and Manga109 [84]).

The test set consisted of 20% of the dataset. The remaining data were split into 80% for training and 20% for validation. The same splits were maintained for all experiments.

The wall-clock time for training and evaluating WDSR, RCAN, AutoSR4EO and AutoSRCNN (and finishing all trials, in the case of the NAS methods) on a single dataset ranged from 30 min to 2 days, with outliers of 5 days, depending on the number of parameters of the model and the number of images and the image sizes in the dataset. The training time of AutoSR4EO encompassed two components: the design time and the training of the candidate architectures. The design time is the time taken to find an effective architecture. The training time is the total time taken to train all candidate architectures. The design time of WDSR and RCAN is not easily quantifiable because it is not defined as the runtime of the algorithm but is instead the time that was implicitly invested by the experts that crafted these methods. As a result, a direct comparison between the training times of AutoSR4EO and these baselines was not appropriate.

We customised AutoKeras to include image output and custom metrics (https://github.com/JuliaWasala/autokeras (release `1.0.16.post1`, accessed on 28 June 2022)). The code for our methods and experiments has been made publicly available (https://github.com/JuliaWasala/autoSR-RS_SENT-NICFI, accessed on 28 June 2022).

4. Results

In this section, we present the results of the experiments described in Section 3.3. In the first subsection, we present the performance of AutoSR4EO compared to that of the state-of-the-art alternatives, followed by a subsection describing the analyses of the performance of search spaces S and L.

4.1. Performance Evaluation

This section describes the results of the comparisons between AutoSR4EO$_S$, AutoSR4EO$_L$ and the baseline methods on the four training datasets: Cerrado, UC Merced, OLI2MSI and SENT-NICFI. Figure 7 shows samples of an image predicted by the different methods. AutoSR4EO produced much sharper images than AutoSRCNN. Table 2 presents the results of the PSNR and SSIM scores. Firstly, we considered the set of baselines trained on the target datasets: WDSR, RCAN and AutoSRCNN. AutoSR4EO$_L$ outperformed the baselines on UC Merced and OLI2MSI. RCAN achieved a higher score than AutoSR4EO$_L$ on SENT-NICFI, but this difference was not statistically significant. However, AutoSR4EO$_S$ performed significantly better than RCAN and the other baselines on this dataset.
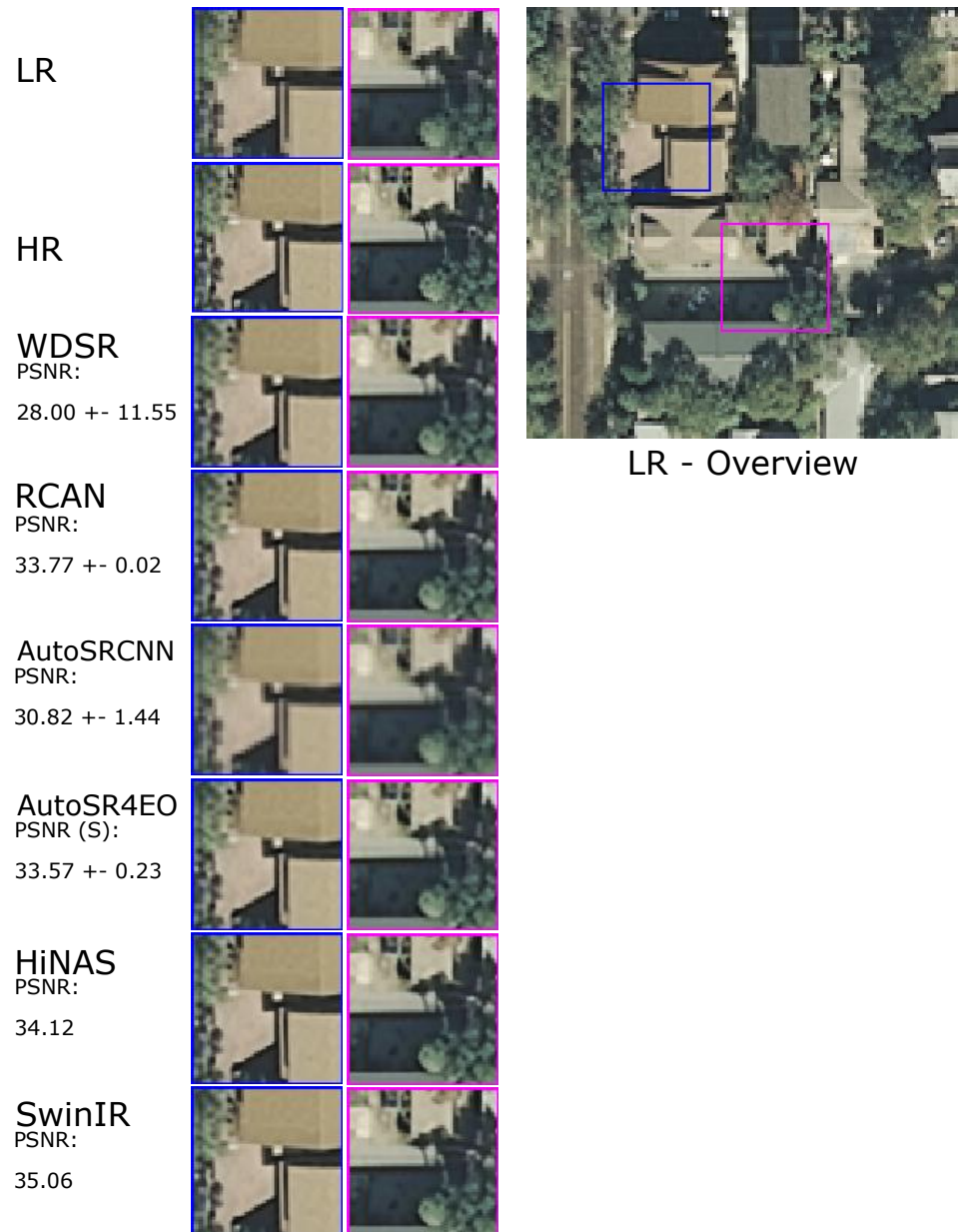
Figure 7. Samples of a super-resolved image from the UC Merced [16] dataset. The LR image was obtained by bicubically downsampling the HR image with a scaling factor of 2. The presented samples are parts of a single image, an overview of which is shown on the right. The images with blue and magenta borders are crops of the original image. The PSNR values are the averages of the whole dataset, as shown in Table 2. Though there was a difference in PSNR, it can be difficult to visually distinguish the results at this image resolution and super-resolution factor. Still, AutoSR4EO clearly outperformed AutoSRCNN, showing that a simple AutoML approach is not enough to solve the problem of SR.

Table 2. PSNR/SSIM results for all methods. Experiments for WDSR, RCAN, AutoSRCNN and AutoSR4EO were run five times per configuration, while for SwinIR and HiNAS, it was only possible to acquire one result since the results were obtained from pre-trained models. The highest and second-highest performances are shown in red and **blue**, respectively.

Method	Cerrado	UC Merced	OLI2MSI	SENT-NICFI
WDSR	40.96 ± 0.39 / 0.9729 ± 0.0016	28.00 ± 11.55 / 0.7414 ± 0.3956	43.89 ± 0.04 / 0.9719 ± 0.0002	28.17 ± 1.20 / 0.7843 ± 0.0525
RCAN	38.48 ± 0.38 / 0.9544 ± 0.0067	33.77 ± 0.02 / 0.9252 ± 0.0002	**44.45** ± 0.01 / **0.9749** ± 0.0000	**30.12** ± 0.02 / 0.8537 ± 0.0007
AutoSRCNN	38.80 ± 0.89 / 0.9507 ± 0.0083	30.82 ± 1.44 / 0.8825 ± 0.0269	43.13 ± 0.68 / 0.9680 ± 0.0045	28.85 ± 0.22 / 0.8223 ± 0.0009
SwinIR	42.85 / **0.9784**	35.06 / 0.9365	42.72 / 0.9687	27.79 / 0.7766
HiNAS	**42.67** / 0.9803	**34.12** / **0.9339**	42.75 / 0.9695	27.83 / 0.7897
AutoSR4EO$_S$ (Ours)	40.61 ± 1.91 / 0.9645 ± 0.0191	33.57 ± 0.23 / 0.9238 ± 0.0024	44.42 ± 0.68 / 0.9741 ± 0.0090	30.20 ± 0.42 / 0.8550 ± 0.0097
AutoSR4EO$_L$ (Ours)	39.84 ± 4.94 / 0.9414 ± 0.0783	33.91 ± 0.36 / 0.9266 ± 0.0024	45.01 ± 0.11 / 0.9780 ± 0.0005	30.10 ± 0.26 / **0.8541** ± 0.0179

Table 3 shows the average ranking of the methods. AutoSR4EO$_S$ and AutoSR4EO$_L$ achieved higher rankings than the baseline methods, with *L* achieving the highest overall ranking. AutoSRCNN consistently ranked last.

Table 3. The average ranking of the methods calculated across the four datasets, with 1 being the highest ranking. Both AutoSR4EO versions are ranked individually. The highest scores are in **boldface**. The rankings were calculated by ranking the methods per dataset and then taking the average rank across the datasets.

	AutoSR4EO						
	L	*S*	**HiNAS**	**RCAN**	**SwinIR**	**WDSR**	**AutoSRCNN**
PSNR	**3.00**	**3.00**	4	3.50	4	4.75	5.25
SSIM	**3.25**	**3.25**	**3.25**	3.50	4	5	5.75

Additional Trials

We performed additional experiments on Cerrado (a synthetic dataset) and SENT-NICFI (a real-world dataset) to select the optimal number of trials. Although the increase in computational complexity as a function of the dataset size created a bottleneck, we trained AutoSR4EO$_S$ for 100 trials on Cerrado and 50 trials on SENT-NICFI. These experiments monitored performance as a function of the number of trials. This information was essential for understanding the trade-off between performance gain and additional running time. We expected this to have little effect on the optimal number of trials because the size of the AutoSR4EO$_L$ search space only increased with a few possible values for the number of residual blocks, as discussed in Section 3.1.2.

Table 4 shows the results of these experiments. The results with more trials were significantly better than those for 20 trials (Table 2). The lower standard deviations indicate that high scores were obtained more consistently, consequently increasing the average scores. Figure 8 plots the highest validation PSNR values found so far for each trial, which can be different from the PSNR value of the current trial. The improvement in validation scores flattened around 20 trials: running the method for longer improved the results but at a decreased rate of improvement.

Table 4. Results of longer experiments with AutoSR4EO$_S$ compared to the original results with 20 trials. Each experiment was run five times. Significantly best results are shown in **boldface**.

Dataset	Trials	PSNR	SSIM
Cerrado	20	40.61 ± 1.91	0.9645 ± 0.0191
	100	$\mathbf{42.10} \pm 0.83$	$\mathbf{0.9763} \pm 0.0031$
SENT-NICFI	20	30.20 ± 0.42	0.8550 ± 0.0097
	50	$\mathbf{30.45} \pm 0.26$	$\mathbf{0.8612} \pm 0.0124$
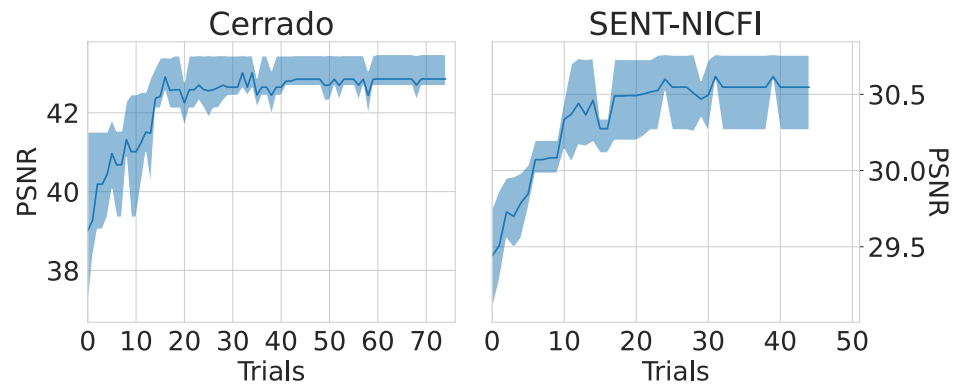
Figure 8. Evolution of the PSNR values on the validation set for each trial of experiments on the Cerrado (**left**) and SENT-NICFI (**right**) datasets, with a maximum of 100 and 50 trials, respectively. Runs could stop if no improvement was expected before the maximum number of trials was reached. Each point shows the mean of the best score achieved in each run up until that trial. The bands show the ranges between the lower and upper quantiles. The scores stabilises around 20 trials.

4.2. Search Space Analysis

We analysed the architectures returned by AutoSR4EO to compare the effectiveness of the AutoSR4EO$_S$ and AutoSR4EO$_L$ search spaces. We analysed the model blocks, model depths and the sets of pre-trained weights occurring in the constructed architectures. Figure 9 shows the numbers of residual blocks (N_res) chosen from search spaces S and L. The number of blocks peaked at 20 for search space S. RCAN-based architectures made up a large proportion of this peak. The results for search space L lacked this peak. RCAN-based models occurred with a depth of up to 28 blocks.

Figure 10 compares search spaces S and L in terms of the model blocks and pre-trained weights. The RCAN model block was sampled more often from S than the WDSR block, while the blocks were sampled evenly from search space L. The selection of pre-trained weights shows a similar pattern: the sampling distribution was uniform for search space L but more unbalanced for search space S.
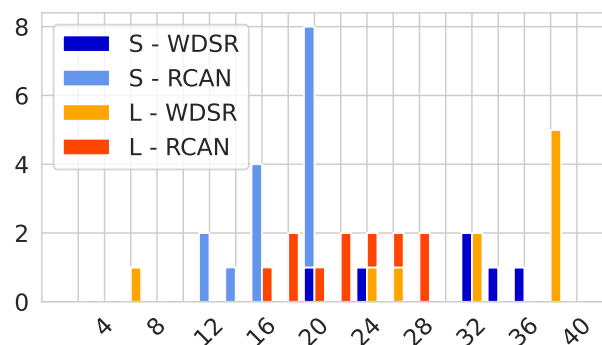

Figure 9. The number of residual blocks in models returned by AutoSR4EO, shown for both search space versions S and L. Each bar shows the proportions of WDSR and RCAN with a colour difference. Search space L was sampled more uniformly than S, showing it is more effective.
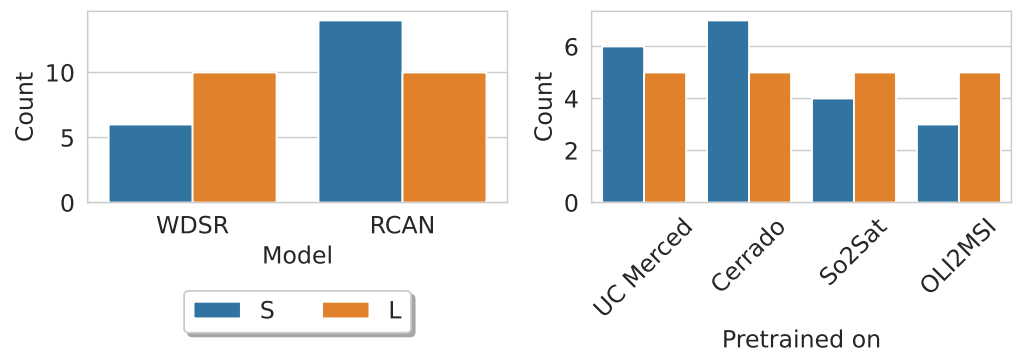
Figure 10. The hyperparameter values chosen in the best networks returned by AutoSR4EO$_S$ and AutoSR4EO$_L$ in each experiment: (**left**) the model blocks and (**right**) the sets of pre-trained weights. For S, some hyperparameters were sampled more than others, while the distribution for L was flat, i.e., each hyperparameter value was chosen with an equal frequency.

5. Discussion

This section covers the interpretation of our results, the limitations of this study and possible future research directions following on from our results.

5.1. Interpretation of the Findings

In this section, we interpret the performance of AutoSR4EO and compare it to that of the baseline methods. Furthermore, we discuss the results of the analysis of the AutoSR4EO search space.

5.1.1. Performance Evaluation

The results from SwinIR and HiNAS (Table 2) showed an interesting pattern: in terms of PSNR, they either outperformed the other methods by a large margin or achieved lower scores. Both SwinIR and HiNAS outperformed the other baselines on the synthetic datasets Cerrado and UC Merced but achieved the lowest scores on the real-world datasets SENT-NICFI and OLI2MSI. These results underline how much the performance of a model can vary when presented with different datasets.

The relatively low PSNR scores of SwinIR and HiNAS on real-world datasets compared to those of the other methods could be explained by the models' failure to model complex real-world data, as both models were trained on synthetic data. These results support the finding of Kohler et al. [14] that evaluation using synthetic datasets can overestimate results.

All methods scored higher on OLI2MSI than on the other datasets. The higher green levels in the OLI2MSI images could explain this. We found by visual inspection that the scenes contained many forests that were quite homogeneous, while UC Merced and SENT-NICFI contained a wider variety of land cover types from larger regions.

AutoSRCNN consistently ranked last (Table 3), suggesting a simple AutoML method is insufficient for the problem of SR for EO images. These results motivate the use of methods with more carefully crafted search spaces, such as AutoSR4EO. The task of SR for EO data cannot be solved with simple CNNs; it requires more sophisticated, and often deeper, network architectures. Deeper networks take longer to train but transfer learning can speed up this process. AutoSR4EO uses both SOTA neural networks and transfer learning.

5.1.2. Search Space Analysis

The peak at 20 residual blocks (Figure 9) coincided with the maximum number of residual blocks possible for the RCAN block in search space S. This peak disappeared in the results of search space L as deeper RCAN models performed better on the evaluated datasets. A further increase in the maximum number of blocks was unnecessary as the maximum number of 40 blocks was never chosen.

The results presented in Tables 2 and 3, as well as Figure 9, indicate that search space *L* was more effective than *S*. The distribution of the hyperparameters chosen from this search space was more balanced due to the change in the N_res hyperparameter. No single hyperparameter value dominated. The performance of different methods varies based on data distribution [90]. In these terms, search space *L* better reflected the purpose of AutoML than search space *S* because search space *L* was larger and thus offered a higher number of possible models.

While AutoSR4EO ranked the highest on average, it did not achieve the highest score on every dataset. This issue is not unique to AutoSR4EO. Manually designed SR networks still outperform NAS-based approaches [24] on standard natural image benchmark datasets, despite the potential of AutoML.

Nevertheless, successful AutoML systems are not required to achieve the highest score in every case. The strength of our approach lies in its ability to generalise, as the high ranking of AutoSR4EO shows. AutoSR4EO presents a new approach to the development of SR methods: an approach that is directly applicable to different use cases. This considerable benefit reduces the time spent on selecting and designing pre-processing pipelines for various applications and datasets. Additionally, AutoML techniques have the capacity to make state-of-the-art (SOTA) techniques accessible to practitioners who are less familiar with SOTA machine learning techniques.

Even though manually designed approaches still outperform NAS systems, we believe that a generic and automatic methodology can be useful for three main reasons. Firstly, our proposed methodology is inherently adaptable: AutoSR4EO can produce a good starting point for highly adaptable model design because the same methods can be re-used without any adaptations for different datasets.

Secondly, this starting point supports further improvements using hand-crafted solutions: automated and hand-crafted methods do not have to be mutually exclusive but can rather complement each other. Furthermore, automation can significantly shorten the time required to obtain an effective model because only the manual fine-tuning needs to be repeated when solving a new problem.

Thirdly, automated methods are valuable for practitioners who want to use machine learning techniques but have no prior experience with designing and configuring machine learning models. Automatic model design and configuration make these techniques more accessible to this group of users.

5.2. Limitations

In this section, we discuss possible improvements and changes for the search space, evaluation procedure and SENT-NICFI dataset. We consider the challenges and benefits of creating real-world datasets, like SENT-NICFI, to better evaluate SR methods, as well as the metrics used for evaluation.

5.2.1. Search Space

The AutoSR4EO search space, which contains model blocks based on two SOTA SR networks, shows the potential of our approach. AutoSR4EO achieved the highest average ranking, showing its ability to generalise regardless of the fact that two model blocks may seem like a small number for an AutoML approach. Moreover, the possibility of extending the search space makes our approach more robust to future developments in the field of SR.

A wider array of model blocks could accommodate a larger variety of datasets, possibly also extending beyond optical images. It is possible to add extra blocks to the search space without changing the search strategy. However, the search strategy may need more trials to consistently reach high-performing solutions if the search space is larger.

We expect that the largest gain would be achieved by adding models that differ significantly from WDSR and RCAN in terms of architecture. Intuitively, the more diverse the search space, the more types of datasets for which AutoSR4EO could produce high-performing models.

The number of runs of AutoSR4EO (five per configuration) limited our interpretation of the analysis of the search space. The results did not show evident patterns in the effect of the number of residual blocks. An analysis with a significantly larger sample size may provide a deeper understanding of the effect of model depth in this case. In general, more runs resulting in more final configurations are necessary for more robust statistical comparisons. However, running more experiments would incur considerable computational costs, which were infeasible within the scope of this study.

5.2.2. Real-World Datasets and SENT-NICFI

We evaluated AutoSR4EO on two real-world datasets. The lack of availability of more real-world datasets, to the best of our knowledge, prevented us from further comparing training on synthetic data to training on real-world data. We created SENT-NICFI, containing images of a variety of real-world landscapes, to alleviate this problem, but future research is needed to create more of these real-world multi-sensor datasets and study the impacts of using real-world data compared to synthetic data.

The difference in satellite overpass times is a challenge in creating real-world multi-sensor datasets for supervised SR because it complicates finding matching images that are sufficiently close to each other in terms of time. Some applications, like change detection, require training images that are as close in time as possible. Other factors, like cloud cover, can also interfere with the retrieval of image pairs for training.

Furthermore, it is important to be aware of the target use cases of the datasets used for evaluation. SENT-NICFI was designed without a specific downstream application in mind. The purpose of the dataset was to increase the number of real-world datasets available for the evaluation of SR methodologies. It is yet unclear how performance on downstream tasks could be affected by training SR models on SENT-NICFI.

Finally, it is important to discuss the role of blind SR models, which do not make assumptions about the degradation kernels of images. This property allows this type of model to overcome some of the problems associated with synthetic datasets. Diffusion models, like that of Wu et al. [51], that learn degradation kernels could reduce the need for real-world datasets in the future. However, real-world datasets are still important for the development of non-blind SR because supervised methods still rely on realistic information on the degradation kernels, which is not provided by synthetic datasets that use simple downsampling procedures, like bicubic interpolation.

5.2.3. Evaluation Metrics and Baselines

The PSNR and SSIM metrics may offer insufficient information for selecting SR models for specific EO pipelines. For instance, images intended for building segmentation could benefit from enhanced edges, while it could be important to preserve the edges from original scenes for applications like land cover classification. Research on the correlations between the SR PSNR and SSIM metrics and downstream task performance is needed to understand which metrics most strongly predict downstream performance and determine whether better performance metrics need to be designed. For instance, future work could include perception-based metrics, like learned perceptual image patch similarity (LPIPS) [91] and Frechet inception distance (FID) [92], and assess whether these metrics are better predictors of downstream performance.

5.3. Future Work

The rapid expansion of the field of deep learning for remote sensing has made increasing numbers of architecture types and training techniques available to researchers. We believe that the main aim of our proposed methods, automated model design, is an important strategy for making effective use of novel techniques. There are still many possible areas of improvement and open challenges to explore, which could improve the usability and adoption of such automated techniques. We discuss four future research topics that could build towards this goal.

Firstly, future work could extend our approach with new model blocks to include the most recent advances in SR; for example, multi-stage residual networks (e.g., BTSRN [93]), progressive reconstruction networks (e.g., LapSRN [94]), multi-branch networks (e.g., IDN [95]), multi-stage vision transformers (e.g., SwinIR [9]) and graph neural networks (GNNs, e.g., DLGNN [96]). Very recently, diffusion-based models ([50,51]) have shown very promising results and overcome some of the challenges posed by GANs. It was not feasible to include these in this work because it would have required many more experiments to validate a larger search space. Aside from model blocks, there is also a need for more research on why manually designed architectures tend to outperform automatically generated architectures. Advances in this area could further improve AutoSR4EO.

Finally, future work should focus on evaluating SR models, including AutoML models such as those evaluated here, within the context of EO pipelines. This is a challenging task because of the multitude of pipeline design choices and interactions between pipeline components, for example, the choice of SR model, downstream task model, training data and training procedure (independent or stacked, where the downstream task loss influences SR model training). Recent work has focused mainly on steps in pipelines as independent units instead of studying them as part of a whole. We need a better understanding of the interactions between pre-processing steps, like SR, and downstream tasks, as well as which design choices have the largest impacts on pipeline results rather than intermediate results.

6. Conclusions

We introduced AutoSR4EO, the first AutoML super-resolution approach for Earth observation images that automatically designs neural networks based on training data. We designed a specialised search space for SR tasks, consisting of SR blocks based on state-of-the-art SR methods. Further, we used pre-trained weights generated from EO datasets to increase training efficiency while better adapting models to EO data. AutoSR4EO provides a good basis for further research on the use of AutoML techniques for EO data because it is easily extendable using new model blocks and pre-trained weights. Additionally, we constructed SENT-NICFI, a novel dataset for SISR for EO images, thus adding to the small number of real-world datasets available for SR for EO images. We evaluated AutoSR4EO on four EO datasets and compared the results to four SOTA baselines and an additional AutoML baseline that we introduced: AutoSRCNN. We compared two search spaces: S and L. AutoSR4EO$_L$ outperformed the baselines on two of the datasets and achieved the highest average ranking among all baselines in terms of both PSNR and SSIM. Models that were pre-trained on synthetic data performed poorly on real-world datasets compared to those that were trained on real-world datasets. From these analyses, we have shown that AutoML is a very promising method for improving SR techniques for EO images. This introduces many opportunities to improve EO-based machine learning tasks.

Data Availability Statement: The following publicly available datasets were used for this study: Brazilian Cerrado Savanna (http://patreo.dcc.ufmg.br/2017/11/12/brazilian-cerrado-savanna-scenes-dataset/, accessed on 28 June 2022); UC Merced (http://weegee.vision.ucmerced.edu/

Abbreviations

The following abbreviations are used in this manuscript:

AutoML	Automated machine learning
EO	Earth observation
ML	Machine learning
NAS	Neural architecture search
PSNR	Peak signal-to-noise ratio
SISR	Single-image super-resolution
SOTA	State-of-the-art
SSIM	Structural similarity index measure
SR	Super-resolution

References

1. He, D.; Shi, Q.; Liu, X.; Zhong, Y.; Xia, G.; Zhang, L. Generating annual high resolution land cover products for 28 metropolises in China based on a deep super-resolution mapping network using Landsat imagery. *GIScience Remote Sens.* **2022**, *59*, 2036–2067. [CrossRef]
2. Synthiya Vinothini, D.; Sathyabama, B.; Karthikeyan, S. Super Resolution Mapping of Trees for Urban Forest Monitoring in Madurai City Using Remote Sensing. In Proceedings of the Computer Vision, Graphics, and Image Processing, Guwahati, India, 19 December 2016; Mukherjee, S., Mukherjee, S., Mukherjee, D.P., Sivaswamy, J., Awate, S., Setlur, S., Namboodiri, A.M., Chaudhury, S., Eds.; Springer: Cham, Switzerland, 2017; pp. 88–96. [CrossRef]
3. Garcia-Pedrero, A.; Gonzalo-Martín, C.; Lillo-Saavedra, M.; Rodríguez-Esparragón, D. The Outlining of Agricultural Plots Based on Spatiotemporal Consensus Segmentation. *Remote Sens.* **2018**, *10*, 1991. [CrossRef]
4. Wang, P.; Zhang, G.; Leung, H. Improving Super-Resolution Flood Inundation Mapping for Multispectral Remote Sensing Image by Supplying More Spectral Information. *IEEE Geosci. Remote Sens. Lett.* **2019**, *16*, 771–775. [CrossRef]
5. Shermeyer, J.; Van Etten, A. 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, Long Beach, CA, USA, 15–20 June 2019. [CrossRef]
6. Zou, F.; Xiao, W.; Ji, W.; He, K.; Yang, Z.; Song, J.; Zhou, H.; Li, K. Arbitrary-oriented object detection via dense feature fusion and attention model for remote sensing super-resolution image. *Neural Comput. Appl.* **2020**, *32*, 14549–14562. [CrossRef]
7. Haris, M.; Shakhnarovich, G.; Ukita, N. Task-driven super resolution: Object detection in low-resolution images. In Proceedings of the International Conference on Neural Information Processing, Killarney, Ireland, 12–17 July 2015; Springer: Berlin/Heidelberg, Germany, 2021; pp. 387–395. [CrossRef]
8. Michel, J.; Vinasco-Salinas, J.; Inglada, J.; Hagolle, O. SEN2VENUS, a Dataset for the Training of Sentinel-2 Super-Resolution Algorithms. *Data* **2022**, *7*, 96. [CrossRef]
9. Liang, J.; Cao, J.; Sun, G.; Zhang, K.; Gool, L.V.; Timofte, R. SwinIR: Image Restoration Using Swin Transformer. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Los Alamitos, CA, USA, 17–21 October 2021; pp. 1833–1844. [CrossRef]
10. Ahn, J.Y.; Cho, N.I. Neural Architecture Search for Image Super-Resolution Using Densely Constructed Search Space: DeCoNAS. In Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021; pp. 4829–4836. [CrossRef]
11. Ahn, N.; Kang, B.; Sohn, K.A. Fast, accurate, and lightweight super-resolution with cascading residual network. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 252–268. [CrossRef]
12. Ran, Q.; Xu, X.; Zhao, S.; Li, W.; Du, Q. Remote sensing images super-resolution with deep convolution networks. *Multimed. Tools Appl.* **2020**, *79*, 8985–9001. [CrossRef]
13. Xu, W.; XU, G.; Wang, Y.; Sun, X.; Lin, D.; WU, Y. High Quality Remote Sensing Image Super-Resolution Using Deep Memory Connected Network. In Proceedings of the IGARSS 2018–2018 IEEE International Geoscience and Remote Sensing Symposium, Valencia, Spain, 22–27 July 2018; pp. 8889–8892. [CrossRef]

14. Köhler, T.; Bätz, M.; Naderi, F.; Kaup, A.; Maier, A.; Riess, C. Toward Bridging the Simulated-to-Real Gap: Benchmarking Super-Resolution on Real Data. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 2944–2959. [CrossRef]

15. Elsken, T.; Metzen, J.H.; Hutter, F. Neural Architecture Search: A Survey. *J. Mach. Learn. Res.* **2019**, *20*, 1997–2017.

16. Yang, Y.; Newsam, S. 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 (GIS '10), San Jose, CA, USA, 2–5 November 2010; pp. 270–279. [CrossRef]

17. Wang, J.; Gao, K.; Zhang, Z.; Ni, C.; Hu, Z.; Chen, D.; Wu, Q. Multisensor Remote Sensing Imagery Super-Resolution with Conditional GAN. *J. Remote Sens.* **2021**, *2021*. [CrossRef]

18. European Space Agency. Orbit—Sentinel 2-Mission-Sentinel Online-Sentinel Online. Available online: https://sentinel.esa.int/web/sentinel/missions/sentinel-2/satellite-description/orbit. (accessed on 16 May 2022).

19. Norway's International Climate and Forest Initiative (NICFI). 2022. Available online: https://www.nicfi.no/ (accessed on 28 June 2022).

20. Gu, S.; Sang, N.; Ma, F. Fast image super resolution via local regression. In Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012), Tskuba Science City, Japan, 11–15 November 2012; pp. 3128–3131.

21. Timofte, R.; De, V.; Gool, L.V. Anchored Neighborhood Regression for Fast Example-Based Super-Resolution. In Proceedings of the 2013 IEEE International Conference on Computer Vision, Sydney, Australia, 1–8 December 2013; pp. 1920–1927. [CrossRef]

22. Timofte, R.; De Smet, V.; Van Gool, L. A+: Adjusted anchored neighborhood regression for fast super-resolution. In *Computer Vision–ACCV 2014, Revised Selected Papers, Part IV 12, Proceedings of the 12th Asian Conference on Computer Vision, Singapore, 1–5 November 2014*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 111–126. [CrossRef]

23. Michaeli, T.; Irani, M. Nonparametric Blind Super-resolution. In Proceedings of the 2013 IEEE International Conference on Computer Vision, Sydney, Australia, 1–8 December 2013; pp. 945–952. [CrossRef]

24. Moser, B.B.; Raue, F.; Frolov, S.; Palacio, S.; Hees, J.; Dengel, A. Hitchhiker's Guide to Super-Resolution: Introduction and Recent Advances. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 1–21. [CrossRef]

25. Dong, C.; Loy, C.C.; He, K.; Tang, X. Learning a Deep Convolutional Network for Image Super-Resolution. In Proceedings of the Lecture Notes in Computer Science (including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Zurich, Switzerland, 6–12 September 2014; 8692 LNCS, pp. 184–199. [CrossRef]

26. Kim, J.; Lee, J.K.; Lee, K.M. Accurate image super-resolution using very deep convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 1646–1654. [CrossRef]

27. Lim, B.; Son, S.; Kim, H.; Nah, S.; Mu Lee, K. Enhanced deep residual networks for single image super-resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 21–26 July 2017; pp. 136–144. [CrossRef]

28. Fan, Y.; Yu, J.; Huang, T.S. Wide-activated deep residual networks based restoration for bpg-compressed images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 18–22 June 2018; pp. 2621–2624.

29. Li, J. Multi-scale Residual Network for Image Super-Resolution. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018. [CrossRef]

30. Zhang, Y.; Li, K.; Li, K.; Wang, L.; Zhong, B.; Fu, Y. Image super-resolution using very deep residual channel attention networks. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 286–301. [CrossRef]

31. Wang, C.; Li, Z.; Shi, J. Lightweight Image Super-Resolution with Adaptive Weighted Learning Network. *arXiv* **2019** [CrossRef]

32. Liu, Z.S.; Wang, L.W.; Li, C.T.; Siu, W.C. Hierarchical Back Projection Network for Image Super-Resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Long Beach, CA, USA, 16–20 June 2019. [CrossRef]

33. Ledig, C.; Theis, L.; Huszár, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z.; et al. Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4681–4690. [CrossRef]

34. Jiang, K.; Wang, Z.; Yi, P.; Wang, G.; Lu, T.; Jiang, J. Edge-enhanced GAN for remote sensing image superresolution. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 5799–5812. [CrossRef]

35. Wang, X.; Yu, K.; Wu, S.; Gu, J.; Liu, Y.; Dong, C.; Qiao, Y.; Loy, C.C. ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks. In *ECCV 2018 Workshops, Proceedings of the Computer Vision, Munich, Germany, 8–14 September 2018*; Leal-Taixé, L., Roth, S., Eds.; Springer: Cham, Switzerland, 2019; pp. 63–79.

36. Rakotonirina, N.C.; Rasoanaivo, A. ESRGAN+: Further Improving Enhanced Super-Resolution Generative Adversarial Network. In Proceedings of the ICASSP 2020, 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 3637–3641. [CrossRef]

37. Sajjadi, M.S.M.; Schölkopf, B.; Hirsch, M. EnhanceNet: Single Image Super-Resolution Through Automated Texture Synthesis. In Proceedings of the IEEE International Conference on Computer Vision (ICCV) Venice, Italy, 22–29 October 2017. [CrossRef]

38. Tao, Y.; Muller, J.P.; Hamedianfar, A.; Shafri, H. Super-Resolution Restoration of Spaceborne Ultra-High-Resolution Images Using the UCL OpTiGAN System. *Remote Sens.* **2021**, *13*, 2269. [CrossRef]

39. Jia, S.; Wang, Z.; Li, Q.; Jia, X.; Xu, M. Multiattention Generative Adversarial Network for Remote Sensing Image Super-Resolution. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–15. [CrossRef]

40. Xu, Y.; Luo, W.; Hu, A.; Xie, Z.; Xie, X.; Tao, L. TE-SAGAN: An Improved Generative Adversarial Network for Remote Sensing Super-Resolution Images. *Remote Sens.* **2022**, *14*, 2425. [CrossRef]

41. Guo, M.; Zhang, Z.; Liu, H.; Huang, Y. NDSRGAN: A Novel Dense Generative Adversarial Network for Real Aerial Imagery Super-Resolution Reconstruction. *Remote Sens.* **2022**, *14*, 1574. [CrossRef]

42. Singla, K.; Pandey, R.; Ghanekar, U. A review on Single Image Super Resolution techniques using generative adversarial network. *Optik* **2022**, *266*, 169607. [CrossRef]

43. Gao, C.; Chen, Y.; Liu, S.; Tan, Z.; Yan, S. AdversarialNAS: Adversarial Neural Architecture Search for GANs. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020. [CrossRef]

44. Gong, X.; Chang, S.; Jiang, Y.; Wang, Z. AutoGAN: Neural Architecture Search for Generative Adversarial Networks. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Los Alamitos, CA, USA, 27 October–2 November 2019; pp. 3223–3233. [CrossRef]

45. Ganepola, V.V.V.; Wirasingha, T. Automating generative adversarial networks using neural architecture search: A review. In Proceedings of the 2021 International Conference on Emerging Smart Computing and Informatics, ESCI 2021, Pune, India, 5–7 March 2021; pp. 577–582. [CrossRef]

46. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv* **2020**, arXiv:2010.11929.

47. Wu, H.; Xiao, B.; Codella, N.; Liu, M.; Dai, X.; Yuan, L.; Zhang, L. CvT: Introducing Convolutions to Vision Transformers. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Los Alamitos, CA, USA, 11–17 October 2021; pp. 22–31. [CrossRef]

48. Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; Jegou, H. Training data-efficient image transformers; distillation through attention. In *Proceedings of Machine Learning, Proceedings of the 38th International Conference on Machine Learning, Online, 18–24 July 2021*; Meila, M., Zhang, T., Eds.; Springer: Berlin/Heidelberg, Germany, 2018; Volume 139, pp. 10347–10357.

49. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Los Alamitos, CA, USA, 11–17 October 2021; pp. 9992–10002. [CrossRef]

50. Han, L.; Zhao, Y.; Lv, H.; Zhang, Y.; Liu, H.; Bi, G.; Han, Q. Enhancing Remote Sensing Image Super-Resolution with Efficient Hybrid Conditional Diffusion Model. *Remote Sens.* **2023**, *15*, 3452. [CrossRef]

51. Wu, H.; Ni, N.; Wang, S.; Zhang, L. Conditional Stochastic Normalizing Flows for Blind Super-Resolution of Remote Sensing Images. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 1–16. [CrossRef]

52. Ali, A.M.; Benjdira, B.; Koubaa, A.; Boulila, W.; El-Shafai, W. TESR: Two-Stage Approach for Enhancement and Super-Resolution of Remote Sensing Images. *Remote Sens.* **2023**, *15*, 2346. [CrossRef]

53. Feurer, M.; Klein, A.; Eggensperger, K.; Springenberg, J.T.; Blum, M.; Hutter, F. Efficient and Robust Automated Machine Learning. In Proceedings of the 28th International Conference on Neural Information Processing Systems—Volume 2 (NIPS'15), Cambridge, MA, USA, Montreal, QC, Canada, 7–12 December 2015; pp. 2755–2763.

54. Feurer, M.; Eggensperger, K.; Falkner, S.; Lindauer, M.; Hutter, F. Auto-Sklearn 2.0: Hands-Free AutoML via Meta-Learning. *J. Mach. Learn. Res.* **2022**, *23*, 1–61.

55. Erickson, N.; Mueller, J.; Shirkov, A.; Zhang, H.; Larroy, P.; Li, M.; Smola, A. AutoGluon-Tabular: Robust and Accurate AutoML for Structured Data. *arXiv* **2020**, arXiv:2003.06505.

56. Wang, C.; Wu, Q.; Weimer, M.; Zhu, E. Flaml: A fast and lightweight automl library. *Proc. Mach. Learn. Syst.* **2021**, *3*, 434–447.

57. Jin, H.; Song, Q.; Hu, X. Auto-Keras: An Efficient Neural Architecture Search System. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '19), New York, NY, USA, 4–18 August 2019; pp. 1946–1956. [CrossRef]

58. Zimmer, L.; Lindauer, M.; Hutter, F. Auto-Pytorch: Multi-Fidelity MetaLearning for Efficient and Robust AutoDL. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 3079–3090. [CrossRef] [PubMed]

59. de Sá, N.C.; Baratchi, M.; Buitenhuis, V.; Cornelissen, P.; van Bodegom, P.M. AutoML for estimating grass height from ETM+/OLI data from field measurements at a nature reserve. *GIScience Remote Sens.* **2022**, *59*, 2164–2183. [CrossRef]

60. Zheng, Z.; Fiore, A.M.; Westervelt, D.M.; Milly, G.P.; Goldsmith, J.; Karambelas, A.; Curci, G.; Randles, C.A.; Paiva, A.R.; Wang, C.; et al. Automated Machine Learning to Evaluate the Information Content of Tropospheric Trace Gas Columns for Fine Particle Estimates Over India: A Modeling Testbed. *J. Adv. Model. Earth Syst.* **2023**, *15*, e2022MS003099. [CrossRef]

61. Helber, P.; Bischke, B.; Dengel, A.; Borth, D. EuroSAT: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2019**, *12*, 2217–2226. [CrossRef]

62. Zhu, X.X.; Hu, J.; Qiu, C.; Shi, Y.; Kang, J.; Mou, L.; Bagheri, H.; Haberle, M.; Hua, Y.; Huang, R.; et al. So2Sat LCZ42: A Benchmark Data Set for the Classification of Global Local Climate Zones [Software and Data Sets]. *IEEE Geosci. Remote Sens. Mag.* **2020**, *8*, 76–89. [CrossRef]

63. Cheng, G.; Han, J.; Lu, X. Remote Sensing Image Scene Classification: Benchmark and State of the Art. *Proc. IEEE* **2017**, *105*, 1865–1883. [CrossRef]

64. Palacios Salinas, N.R.; Baratchi, M.; van Rijn, J.N.; Vollrath, A. Automated Machine Learning for Satellite Data: Integrating Remote Sensing Pre-Trained Models into AutoML Systems. In Proceedings of the Machine Learning and Knowledge Discovery in Databases. Applied Data Science Track: European Conference, ECML PKDD 2021, Bilbao, Spain, 13–17 September 2021; p. C1. [CrossRef]

65. Polonskaia, I.S.; Aliev, I.R.; Nikitin, N.O. Automated Evolutionary Design of CNN Classifiers for Object Recognition on Satellite Images. *Procedia Comput. Sci.* **2021**, *193*, 210–219. [CrossRef]

66. Chu, X.; Zhang, B.; Xu, R. Multi-Objective Reinforced Evolution in Mobile Neural Architecture Search. In *Proceedings Part IV, Proceedings of the Computer Vision—ECCV 2020 Workshops, Glasgow, UK, 23–28 August 2020*; Springer: Cham, Switzerland, 2020; pp. 99–113. [CrossRef]

67. Chu, X.; Zhang, B.; Ma, H.; Xu, R.; Li, Q. Fast, Accurate and Lightweight Super-Resolution with Neural Architecture Search. In Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), Taichung, Taiwan, 18–21 July 2021; pp. 59–64. [CrossRef]

68. Ahn, J.Y.; Cho, N.I. Multi-Branch Neural Architecture Search for Lightweight Image Super-Resolution. *IEEE Access* **2021**, *9*, 153633–153646. [CrossRef]

69. Zhang, H.; Li, Y.; Chen, H.; Shen, C. Memory-Efficient Hierarchical Neural Architecture Search for Image Denoising. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Los Alamitos, CA, USA, 12 June 2020; pp. 3654–3663. [CrossRef]

70. Chen, Y.C.; Gao, C.; Robb, E.; Huang, J.B. NAS-DIP: Learning Deep Image Prior with Neural Architecture Search. In *Proceedings of the Computer Vision—ECCV 2020*; Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M., Eds.; Springer: Cham, Switzerland, 2020; pp. 442–459. [CrossRef]

71. Agustsson, E.; Timofte, R. NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 21–26 July 2017; pp. 1122–1131. [CrossRef]

72. Bevilacqua, M.; Roumy, A.; Guillemot, C.; Alberi-Morel, M.L. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In Proceedings of the 23rd British Machine Vision Conference (BMVC), Surrey, UK, 3–7 September 2012.

73. Zeyde, R.; Elad, M.; Protter, M. On single image scale-up using sparse-representations. In Proceedings of the International Conference on Curves and Surfaces, Avignon, France, 24–30 June 2010; pp. 711–730. [CrossRef]

74. Huang, J.B.; Singh, A.; Ahuja, N. Single image super-resolution from transformed self-exemplars. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 5197–5206. [CrossRef]

75. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255. [CrossRef]

76. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778. [CrossRef]

77. Wang, Z.; Chen, J.; Hoi, S.C. Deep Learning for Image Super-Resolution: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 3365–3387. [CrossRef] [PubMed]

78. Timofte, R.; Gu, S.; Wu, J.; Van Gool, L.; Zhang, L.; Yang, M.H.; Haris, M.; Shakhnarovich, G.; Ukita, N.; Hu, S.; et al. NTIRE 2018 Challenge on Single Image Super-Resolution: Methods and Results. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, USA, 18–22 June 2018; pp. 965–96511. [CrossRef]

79. Ha, V.K.; Ren, J.; Xu, X.; Zhao, S.; Xie, G.; Vargas, V.M. Deep learning based single image super-resolution: A survey. *Int. J. Autom. Comput.* **2019**, *16*, 413–426. [CrossRef]

80. Li, J.; Pei, Z.; Zeng, T. From Beginner to Master: A Survey for Deep Learning-based Single-Image Super-Resolution. *arXiv* **2021**, arxiv:2109.14335.

81. Microsoft. *Neural Network Intelligence*; McGraw-Hill, Inc.: New York, NY, USA, 2021.

82. Nogueira, K.; Dos Santos, J.A.; Fornazari, T.; Freire Silva, T.S.; Morellato, L.P.; Torres, R.D.S. Towards vegetation species discrimination by using data-driven descriptors. In Proceedings of the 2016 9th IAPR Workshop on Pattern Recogniton in Remote Sensing (PRRS), Cancun, Mexico, 4 December 2016; pp. 1–6. [CrossRef]

83. Razzak, M.T.; Mateo-Garcia, G.; Lecuyer, G.; Gomez-Chova, L.; Gal, Y.; Kalaitzis, F. Multi-spectral multi-image super-resolution of Sentinel-2 with radiometric consistency losses and its effect on building delineation. *ISPRS J. Photogramm. Remote Sens.* **2023**, *195*, 1–13. [CrossRef]

84. Matsui, Y.; Ito, K.; Aramaki, Y.; Fujimoto, A.; Ogawa, T.; Yamasaki, T.; Aizawa, K. Sketch-based manga retrieval using manga109 dataset. *Multimed. Tools Appl.* **2017**, *76*, 21811–21838. [CrossRef]

85. Zhao, H.; Gallo, O.; Frosio, I.; Kautz, J. Loss functions for neural networks for image processing. *arXiv* **2015**, arXiv:1511.08861.

86. Liu, H.; Qian, Y.; Zhong, X.; Chen, L.; Yang, G. Research on super-resolution reconstruction of remote sensing images: A comprehensive review. *Opt. Eng.* **2021**, *60*, 100901. [CrossRef]

87. Rohith, G.; Kumar, L.S. Paradigm shifts in super-resolution techniques for remote sensing applications. *Vis. Comput.* **2021**, *37*, 1965–2008. [CrossRef]

88. Tsagkatakis, G.; Aidini, A.; Fotiadou, K.; Giannopoulos, M.; Pentari, A.; Tsakalides, P. Survey of Deep-Learning Approaches for Remote Sensing Observation Enhancement. *Sensors* **2019**, *19*, 3929. [CrossRef] [PubMed]

89. Conover, W.J. *Practical Nonparametric Statistics*, 3rd ed.; Wiley Series in Probability and Statistics; John Wiley & Sons: Nashville, TN, USA, 1998.

90. Yang, C.; Fan, J.; Wu, Z.; Udell, M. AutoML Pipeline Selection: Efficiently Navigating the Combinatorial Space. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, San Diego, CA, USA, 23–17 August 2020; p. 11. [CrossRef]

91. Zhang, R.; Isola, P.; Efros, A.A.; Shechtman, E.; Wang, O. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018.

92. Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; Hochreiter, S. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *Proceedings of the Advances in Neural Information Processing Systems*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates Inc.: Red Hook, NY, USA, 2017; Volume 30.

93. Fan, Y.; Shi, H.; Yu, J.; Liu, D.; Han, W.; Yu, H.; Wang, Z.; Wang, X.; Huang, T.S. Balanced Two-Stage Residual Networks for Image Super-Resolution. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 21–26 July 2017; pp. 1157–1164. [CrossRef]

94. Lai, W.S.; Huang, J.B.; Ahuja, N.; Yang, M.H. Deep Laplacian Pyramid Networks for Fast and Accurate Super-Resolution. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5835–5843. [CrossRef]

95. Hui, Z.; Wang, X.; Gao, X. Fast and accurate single image super-resolution via information distillation network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 723–731. [CrossRef]

96. Liu, Z.; Feng, R.; Wang, L.; Han, W.; Zeng, T. Dual Learning-Based Graph Neural Network for Remote Sensing Image Super-Resolution. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–14. [CrossRef]