Bachelor DSAI

Quantifying the Effect of Super-Resolution on Aerial Scene Classification

Sharanda Suttorp s2630575

Supervisors:
Julia Wąsala & Dr. Mitra Baratchi

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl

30/06/2023

Abstract

Remote sensing imagery plays a vital role in Earth observation tasks. However, remote sensing data collection platforms often have limitations resulting in low-resolution remote sensing data. Super-resolution techniques offer a solution to improve the resolution of these images. This thesis explores the application of deep learning-based super-resolution frameworks as a pre-processing step to enhance the resolution of remote sensing data. We evaluate the subsequent effect of super-resolution on aerial scene classification. We leveraged a diverse range of pre-trained super-resolution frameworks including ESPCN, Real-ESRGAN, SwinIR, and Swin2SR. The ESPCN model was further fine-tuned with domain-specific data. The performance of these models was assessed by calculating the PSNR and SSIM scores, and by performing statistical tests. PSNR and SSIM are commonly used metrics to evaluate image quality. We incorporated transfer learning with a fine-tuning approach using the ResNet50 and ViT models for the aerial scene classification task. The performance of the classification task was evaluated by computing various metrics and carrying out statistical tests. The findings of this thesis show that the ESPCN model significantly outperforms the other super-resolution models assessed in this study. Additionally, our results indicate that a shallower model yields higher PSNR and SSIM scores for the AID dataset. For the classification tasks, the ESPCN and Swin models achieved higher evaluation scores compared to the native-resolution images. However, the statistical tests revealed that the super-resolved datasets did not yield significant results for the classification task.

Contents

1 Introduction

Remote sensing imagery is used in various fields including agriculture, meteorology, geography, and the military [1]. The applications in these fields include weather prediction, climate change prediction, agricultural crop type identification, land use mapping, disaster management, military target identification, and more [1, 2]. These applications require high-resolution imagery to achieve optimal performance. However, the remote sensing imagery often does not have the desired level of resolution. Data collection platforms, including satellites and air crafts, have limitations such as optical distortion, sensor noise, environmental interference, and cost constraints [1, 2]. Super-resolution techniques offer a solution to this challenge. Super-resolution is the process of enhancing low-resolution images to high-resolution counterparts [3]. Super-resolution has been applied in other fields such as video surveillance [4] and medical imaging [5]. In this thesis, we focus on deep learning-based techniques for super-resolution. Deep learning-based super-resolution aims to learn a mapping between low-resolution and high-resolution image pairs using neural networks. The rapid advancement of deep learning has led many researchers to explore deep learning approaches for super-resolution tasks [6, 7, 8, 9, 10, 11]. They showed significant results along with the development of benchmark datasets for super-resolution. Super-resolution has the potential to improve the spatial and spectral quality of remote sensing imagery. As a result, super-resolution methods have gained popularity within the remote sensing community.

In this thesis, we focus on the specific problem of evaluating the effect of super-resolution on the downstream task of aerial scene classification. To address this problem, we investigate the application of super-resolution techniques as a pre-processing step to improve the resolution and quality of aerial imagery. We aim to evaluate the subsequent effect on the performance of aerial scene classification.

Aerial scene classification is the task of assigning labels to overhead imagery to identify land use. According to Cheng et al., [12] several researchers have investigated applications of scene classification such as environmental monitoring [13], urban planning [14], vegetation mapping [15], and natural hazards detection [16]. For this study, transfer learning is incorporated for the classification task. Marmanis et al. [17] showed that using ImageNet [18] pre-trained networks benefits Earth Observation classification. The aim of transfer learning is to transfer knowledge across different domains by leveraging pre-trained models for different but related tasks [19]. These pre-trained models are typically trained on large datasets, so they have already learned useful feature representations which can be reused for other vision tasks. This is especially useful in scenarios where there is limited training data available for a specific task [19]. Another advantage of transfer learning is that it is able to save training time and computational resources compared to starting a task from scratch [20]. Additionally, a study conducted by Yosinski et al. [21] revealed that using feature weights, even from distant tasks or domains, displayed improved performance compared to using random initialization. They also showed that adopting transfer learning can improve the generalization ability of a model.

The main contributions of this thesis are:

- We investigate the application of deep learning-based super-resolution techniques as a pre-processing step to improve the resolution of aerial imagery.

- We used diverse state-of-the-art pre-trained super-resolution models, including ESPCN, Real-ESRGAN, SwinIR, and Swin2SR, and quantified their performance using commonly used super-resolution evaluation metrics and a statistical test. Additionally, we fine-tuned the ESPCN model using domain-specific data and also quantified its performance.

- We evaluate the effect of super-resolution on the performance of 2 aerial scene classification models, including ResNet50 and ViT. We incorporated transfer learning by using pre-trained models and fine-tuned them using domain-specific data. We also performed a statistical test.

The code repository of this thesis can be accessed via the following link: https://github.com/sharandaa/thesis. This thesis is organized according to the following structure. This chapter contains the introduction; Section 2 discusses related work; Section 3 describes the methodology. Section 4 describes the experiments; Section 5 demonstrates the results; Section 6 concludes and suggests potential directions for further research.

2 Related Work

In this section, we discuss prior research that has been done on super-resolution and its application to remote sensing image classification.

2.1 Single Image Super-Resolution

Super-resolution is typically divided into single-image super-resolution and multiple-image super-resolution. Multiple-image super-resolution is defined as producing a high-resolution image from multiple low-resolution images from the same scene [22]. In this thesis, we focus on single-image super-resolution. A super-resolution technique that was used before deep learning approaches, is interpolation-based methods, such as bicubic interpolation [23, 24]. Despite their simplicity and efficiency, they often lack the ability to generate fine details.

The first deep learning-based approach for super-resolution was proposed by Dong et al. who introduced SRCNN [6]. This network contains 3 convolutional layers and exhibited improved performance compared to bicubic interpolation [2]. Subsequently, multiple CNN-based models have been proposed that aim to learn better representations using deeper and more complex architectures [25, 26, 27]. Additionally, GAN methods have been employed in super-resolution models [28, 8, 29]. Residual connections are another approach to improve the learned representations [30, 31, 32]. They allow information to be passed on to subsequent layers. Furthermore, attention modules have shown significant benefits in super-resolution architectures because of their ability to attend to important information and their ability to model long-range dependencies [33, 34, 35]. Transformer-based architectures use a self-attention technique [36].

In addition to the aforementioned frameworks, there has been research conducted on the application of super-resolution in remote sensing imagery. Xu et al. proposed a super-resolution framework using a deep memory-connected network. Tingting [37] introduced a light super-resolution model for remote sensing imagery. Wąsala et al. [38] proposed AutoSR-RS, which is an automated machine learning framework that aims at building optimal neural networks for diverse remote sensing datasets.

2.2 Super-Resolution applied to Remote Sensing Image Classification

Remote sensing image classification is typically divided into 3 levels: pixel-level, object-level, and scene-level classification [12]. Pixel-level classification is sometimes referred to as semantic segmentation. A visual representation of these 3 levels is illustrated in Figure 1. This study focuses on scene-level classification.

Several studies have explored the application of super-resolution on object detection tasks. Shermeyer et al. [39] used the Very Deep Super-Resolution (VDSR) framework and a custom Random Forest Super-Resolution (RFSR) framework to evaluate the effect of super-resolution on the identification of objects like vehicles, boats, and cars in remote sensing imagery. Courtrai et al. [40] proposed a GAN-based super-resolution network to improve object detection performance in satellite imagery. Musunuri et al. [41] introduced SRODNet which combines super-resolution and object detection for aerial imagery and imagery from a car driver's perspective. There have also been studies that explored the application of super-resolution methods in the domain of semantic segmentation. Wang et al. [42] proposed the Dual Super-Resolution Learning (DSRL) framework to improve segmentation tasks while minimizing additional computation costs. Zhang et al. [43] introduced the FSRSS-Net framework that maps buildings in satellite images using a super-resolution semantic segmentation network.

For the scene classification task, Palacios Salinas et al. [44] conducted a study that leveraged automated machine learning for satellite data classification, while incorporating transfer learning. Other works that leverage transfer learning for scene-level classification are [45, 17, 46]. Dimitrovski et al. [47] introduced an artificial intelligence toolbox for Earth Observation (AiTLAS). This open-source toolbox includes a wide range of remote sensing datasets and deep learning models for evaluating image classification in Earth Observation.

This thesis contributes to the existing literature by conducting a comparative analysis of multiple super-resolution frameworks applied to a remote sensing image dataset. In contrast to previous research, this study aims to quantify the effect of these techniques on scene-level classification specifically, in the context of remote sensing image analysis. There has been extensive research carried out on super-resolution applied to object detection. However, the categories of object detection slightly differ from the categories of scene-level classification. Remote sensing object detection categories typically encompass individual objects such as cars, airplanes, and boats. Scene classification categories include broader remote sensing categories such as urban areas and water bodies.

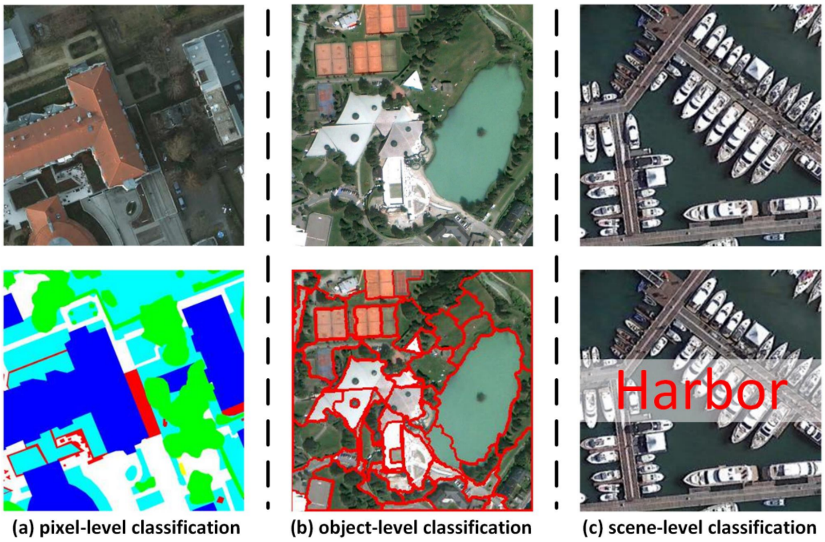
Harbor
(a) pixel-level classification
(b) object-level classification
(c) scene-level classification

Figure 1: The three levels of remote sensing image classification. Source: [12].

3 Methods

This section describes the methodology used to evaluate the performance of super-resolution techniques on aerial scene classification. The main objective of our study is to assess the effectiveness of various super-resolution techniques in accurately classifying a wide range of remote sensing images. To achieve this, we undertook the following steps:

1. We selected an aerial scene dataset, pre-processed the images, and made a train, validation, and test split.
2. The pre-processed images of the test split were super-resolved using various super-resolution methods, including ESPCN, Real-ESRGAN, SwinIR, and Swin2SR. The performance of these models was assessed using evaluation metrics and a statistical test.
3. The pre-trained classification models, ResNet50 and ViT, were fine-tuned using the training part of the dataset.
4. The super-resolved datasets were used to assess the classification performance using evaluation metrics and a statistical test.

This section provides an overview of the key elements of our methodology, including descriptions of the dataset, super-resolution models, classification models, and evaluation metrics.

3.1 Datasets

Different types of datasets are needed for the super-resolution task and the classification task. For the super-resolution task, we require a dataset consisting of pairs of low-resolution images along with their corresponding high-resolution images. Our selected dataset did not contain this low and high-resolution pairing. Therefore, we synthetically created the low-resolution counterparts in order to evaluate the super-resolution performance. There is room for improvement in our method regarding the degradation model we used for down-sampling the images. Our approach included bicubic interpolation, but there exist many other models such as nearest-neighbor interpolation and bilinear interpolation that could be explored. Additionally, some studies used a Gaussian kernel to add blur to an image before down-sampling it [7, 9]. This addition is able to better simulate a camera's down-sampling process.

A labeled image dataset is needed for the classification task. In the context of aerial scene classification, each image is labeled with a specific land use or scene category. It is also worth taking into consideration the differences between "regular" computer vision tasks and satellite vision tasks. "Regular" computer vision tasks refer to the task of classifying natural images. Liu et al. [48] argue that aerial images are taken from the vertical view perspective, while natural images are typically captured from the horizontal view. Furthermore, aerial images may contain more complex backgrounds and the scales of objects may vary depending on the data collection platform such as satellites or air crafts. Besides, the scale of satellite imagery is typically much larger than natural images. They can cover large geographic regions. Figure 2 illustrates the difference between natural and aerial images.

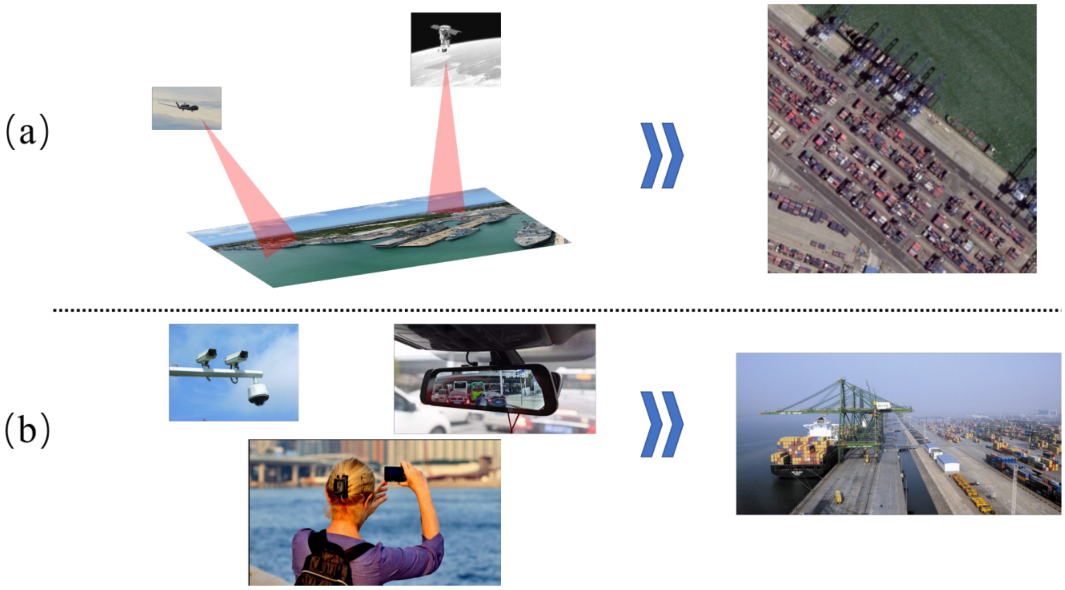

Figure 2: The distinction between an aerial image and a natural image of a port. (a) Aerial image of a port captured by a remote sensing platform. (b) Natural image of a port captured by a surveillance camera. Source: [48].

For the classification task, we applied data augmentation to the images that were used for fine-tuning the classification models. Data augmentation is the process of adding new synthetic data to the dataset by applying modifications to your original data [49]. Examples of modifications include rotating, flipping, or cropping the images. This increases the dataset size and diversity. Studies have shown that data augmentation can reduce overfitting and improve the accuracy and generalization of classification models [50, 51, 52].

3.1.1 AID: Aerial Image Dataset

The AID (Aerial Image Dataset) dataset [53] was selected for the experiments for a number of reasons. The AID dataset is an established benchmark dataset for remote sensing scene classification. Moreover, this dataset offers a realistic representation of a wide range of land cover scenes. The AID datasets also provide ground truth labels for each image. This was required for the classification task. We split the dataset into a training, validation, and test set. The test set was used for the super-resolution tasks and the training and validation sets were used for the classification tasks and for the fine-tuning of the ESPCN model. Figure 3 illustrates the split.

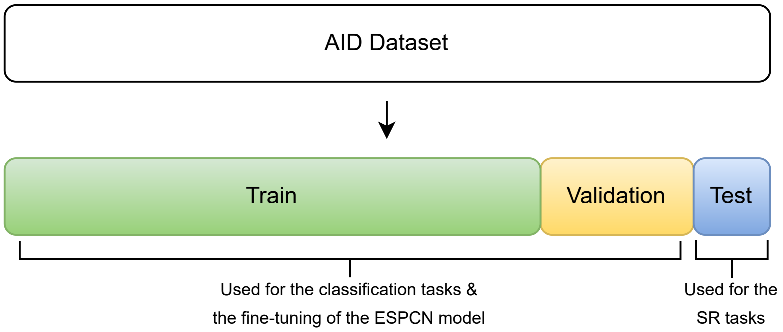
AID Dataset
Train
Validation
Test
Used for the classification tasks &
the fine-tuning of the ESPCN model
Used for the
SR tasks

Figure 3: The split of the AID dataset into a training, validation, and test set.

The AID dataset was developed for aerial scene classification and consists of 10.000 images across 30 categories and was collected through Google Search imagery. Each category contains 220 to 420 samples of 600x600 pixels. The following categories are included: airport, bare land, baseball field, beach, bridge, center, church, commercial, dense residential, desert, farmland, forest, industrial, meadow, medium residential, mountain, park, parking, playground, pond, port, railway station, resort, river, school, sparse residential, square, stadium, storage tanks, and viaduct. Figure 4 shows an example of each category. Additionally, the samples were selected from different countries including China, the United States, England, France, Italy, Japan, Germany, and more.

The AID dataset has multiple improvements compared to previous remote sensing datasets. Firstly, this dataset has higher intra-class variations since the samples are extracted at different seasons, scales, and orientations, and in different regions of the world. For example, the category 'mountain' contains mountains with and without snow. Secondly, AID contains smaller inter-class dissimilarity. The differences between various scene categories are often minimal. For instance, the categories "playground" and "stadium" may both include a sports field, but the primary distinguishing factor is the presence of seating arrangements. To address this issue, AID incorporated more classes compared to previous datasets. Lastly, AID is a relatively large dataset that is typically better for assessing classification methods [54, 55].

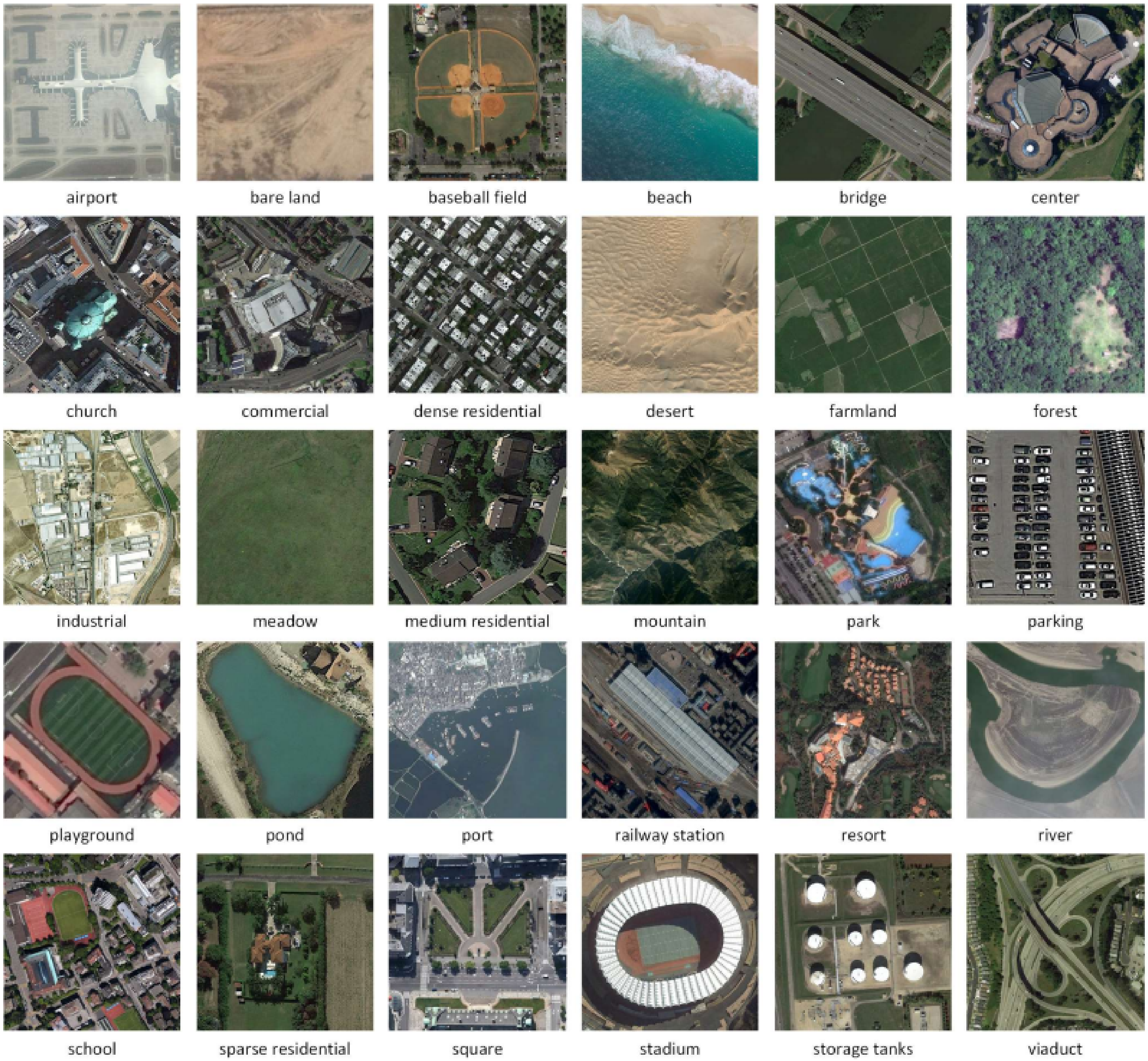

Figure 4: Examples of images from the AID dataset illustrating the 30 Categories. Source: [12].

3.2 Super-Resolution Models

We have used a number of pre-trained super-resolution models and we fine-tuned one super-resolution model with the AID dataset. Pre-trained models offer time-saving advantages as well as reduce computational efforts [56]. A disadvantage of pre-trained models includes domain mismatch. Domain mismatch occurs when the data of the pre-trained model differs significantly from the data that is used for another task. This is also the case for our experiments: the pre-trained models used for the experiments are mainly trained on natural images. Consequently, they might fail to capture the domain-specific nuances of remote sensing data. Fine-tuning allows a model to adapt to a specific domain [21].

The following pre-trained models have been selected to assess different super-resolution performances: ESPCN [7], Real-ESRGAN [9], SwinIR [10], and Swin2SR [11]. ESPCN is CNN-based, Real-ESRGAN is GAN-based and SwinIR and Swin2SR are both transformer-based. ESPCN was selected because it is a relatively efficient model compared to other CNN-based models such as EDSR [57]. This might be advantageous in various remote sensing tasks such as natural hazard detection, where time is of the essence. Real-ESRGAN was chosen because they leveraged an extensive degradation model during their training process. This enables the model to imitate real-world degradation more accurately. We selected SwinIR and Swin2SR because they exhibited improved performance compared to state-of-the-art models while having fewer parameters. Our objective was to compare various super-resolution architectures. Additionally, we chose models that were publicly available.

3.2.1 ESPCN: Efficient Sub-Pixel Convolutional Neural Network

ESPCN [7] is a 3-layer CNN architecture where the up-scaling process is performed by the last layer of the network. The previous layers learn the feature maps from the low-resolution input. The architecture of the network is illustrated in Figure 5. As a result, ESPCN is computationally faster than previous models such as the SRCNN [6] model, which upscales the image at the start of the network.

The OpenCV library includes a super-resolution class that can be used to load the ESPCN model. This class was used for the experiments alongside the trained ESPCN models `ESPCN_x2.pb`, `ESPCN_x3.pb` and `ESPCN_x4.pb` from this repository[1]. These models were trained on the DIV2K [58] dataset.

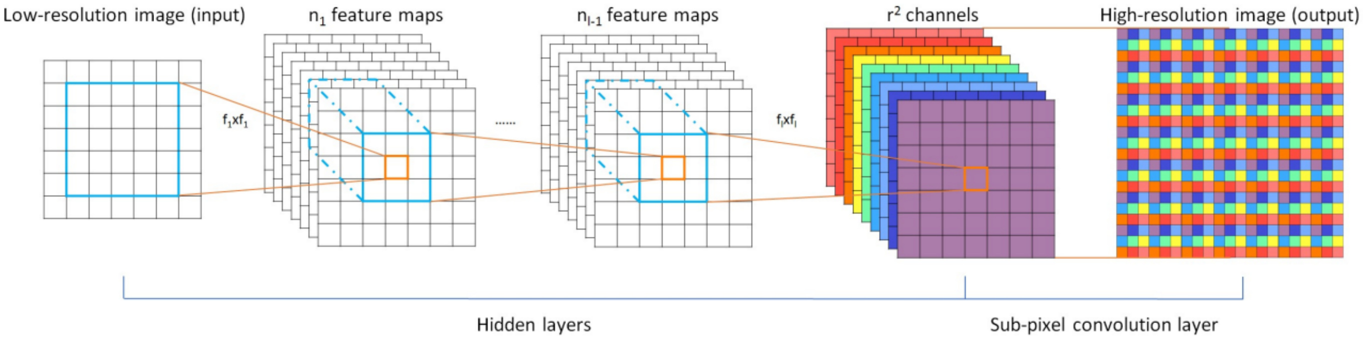

Figure 5: The architecture of the ESPCN model. The network contains 2 convolutional layers for constructing feature maps and a sub-pixel convolution layer for reconstructing the high-resolution result. Source: [7]

The model we fine-tuned was the ESPCN model because this model exhibited superior performance. We used the model and code from Keras [2] to fine-tune the ESPCN model. This model was pre-

[1] https://github.com/fannymonori/TF-ESPCN/tree/master/export

[2] https://keras.io/examples/vision/super_resolution_sub_pixel/

trained on the BSDS500 dataset [59] and we fine-tuned this model with the training and validation set of the AID dataset.

3.2.2 Real-ESRGAN: Real Enhanced Super-Resolution Generative Adversarial Networks

Real-ESRGAN [9] is a GAN-based model and contains two networks: a generator network and a discriminator network. The generator network is trained using low-resolution/high-resolution pairs to ultimately learn a mapping to super-resolve an image from a low-resolution input. The discriminator network attempts to distinguish the ground truth images from the super-resolved images created by the generator and updates both models accordingly. The Real-ESRGAN model aims to extend the ESRGAN [8] model by training it with a large synthetic dataset. This synthetic dataset was generated by applying a random degradation model, such as applying blur, adding noise, and using different down-sampling methods (bilinear, bicubic, etc.).

For the Real-ESRGAN experiment the provided repository[3] was used with the following pre-trained models: `RealESRGAN_x2plus` and `RealESRGAN_x4plus`. These models were trained on the datasets: DIV2K, Flickr2K, and OutdoorSceneTraining. There is no model provided for super-resolving with a scale factor of x3, hence there are no results for this scale.

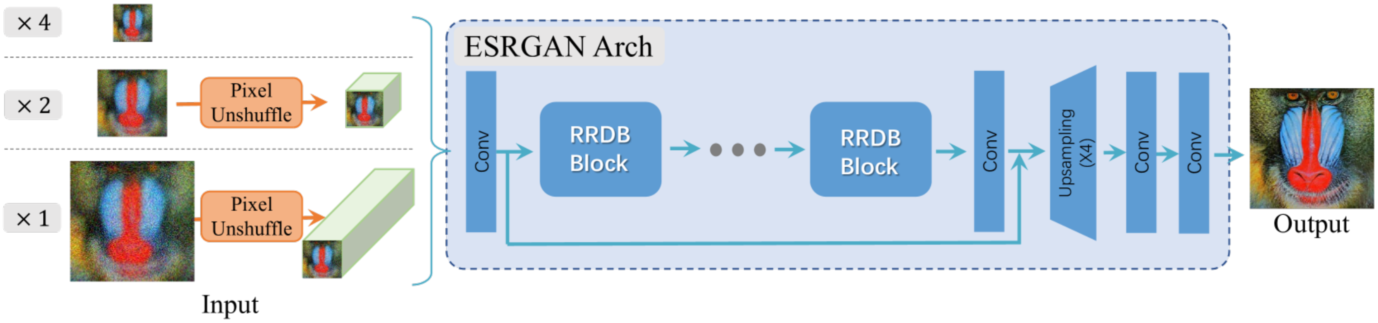

Figure 6: The architecture of the Real-ESRGAN model. The network uses a pixel unshuffle procedure to reduce the spatial size of the images. The RRDBs (residual-in-residual dense blocks) are blocks that contain multiple residual connections between the convolution layers. Source: [9].

3.2.3 SwinIR: Swin Image Restoration

The **SwinIR** [10] model incorporates the Swin Transformer [60] in its architecture. An advantage of using a Transformer [61] is that it learns global features across images using a multi-head self-attention mechanism. The SwinIR model consists of three parts: shallow feature extraction, deep feature extraction, and high-quality image reconstruction modules. The shallow feature extraction contains 1 convolutional layer and the deep feature extraction includes multiple residual Swin Transformer blocks (RSTBs). Each RSTB contains several Swin Transformer layers and a convolution layer. The Swin Transformer layers incorporate multi-headed self-attention. Residual

[3] https://github.com/xinntao/Real-ESRGAN

connections can preserve information from earlier layers [62]. Liang et al. [10] showed that SwinIR achieved higher PSNR with fewer parameters than previous super-resolution models.

For the SwinIR super-resolution the provided repository[4] was used to perform the super-resolution tasks. We used the classical version and the lightweight version. The classical version includes 6 RSTBs for deep feature reconstruction, while the lightweight model contains 4 RSTBs. For the classical super-resolution, we used a patch size of `64` and the following pre-trained models: `001_classicalSR_DF2K_s64w8_SwinIR-M_x2.pth`, `001_classicalSR_DF2K_s64w8_SwinIR-M_x3.pth`, and `001_classicalSR_DF2K_s64w8_SwinIR-M_x4.pth`. For the lightweight super-resolution, we used the following pre-trained models: `002_lightweightSR_DIV2K_s64w8_SwinIR-S_x2.pth`, `002_lightweightSR_DIV2K_s64w8_SwinIR-S_x3.pth`, and `002_lightweightSR_DIV2K_s64w8_SwinIR-S_x4.pth`. We chose the models that are trained on the DIV2K and Flickr2K image datasets since Real-ESRGAN and Swin2SR are also trained on said datasets. This ensures a fair comparison between the models. To enhance the images to 1200x1200 pictures, this model was used: `003_realSR_BSRGAN_DFO_s64w8_SwinIR-M_x4_GAN.pth`.

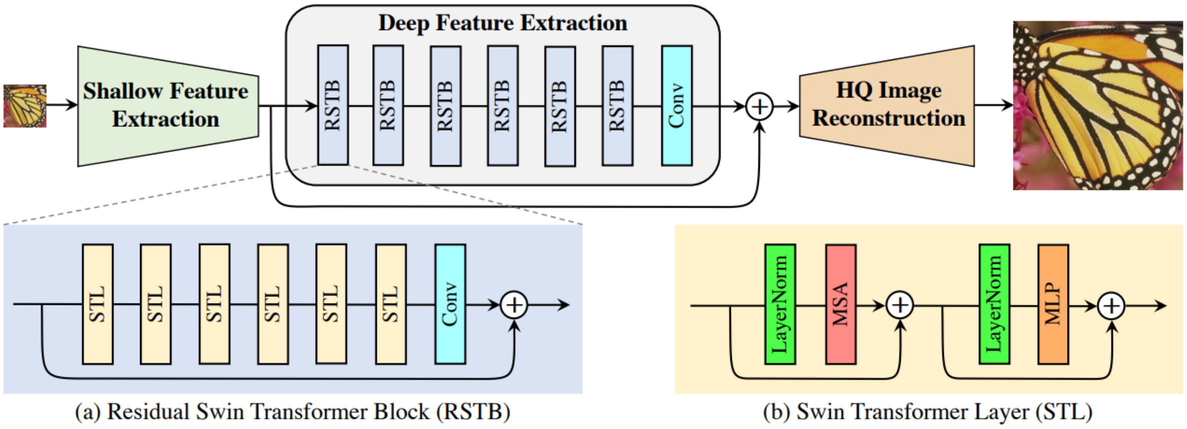

Figure 7: The architecture of the SwinIR model. The shallow feature extraction consists of a convolution layer. The deep feature extraction contains multiple RSTBs. Each RSTB contains multiple STLs. The high-quality image reconstruction layer contains a sub-pix convolution layer. Source: [10].

3.2.4 Swin2SR: Swin to Super-Resolution

he **Swin2SR** [11] model is based on the SwinIR model. One of the key differences is that it employs the new Swin Transformer V2 [63] in the RSTBs. The V2 model presents an improvement over the V1 model by exhibiting improved stability during the training phase [63]. The study conducted by Conde et al. [11] showed that Swin2SR achieved the same results as SwinIR for some tasks but with 33% fewer training iterations.

For the Swin2SR super-resolution the provided repository[5] was used to perform the classical

[4] https://github.com/JingyunLiang/SwinIR
[5] https://github.com/mv-lab/swin2sr

image super-resolution task with a training patch size of 64 and the following pre-trained models: `Swin2SR_ClassicalSR_X2_64.pth` and `Swin2SR_ClassicalSR_X4_64.pth`. These models are trained on the DIV2K and Flickr2K datasets. Swin2SR has no results for the x3 scale since there is not a pre-trained model available that can handle this scale. For the classification task, the images could not be upscaled to 1200x1200 images since there was no 'Real-World Image Super-Resolution' model for this scale. However, there was a model available to handle a scaling factor of 4: `Swin2SR_RealworldSR_X4_64_BSRGAN_PSNR.pth`.

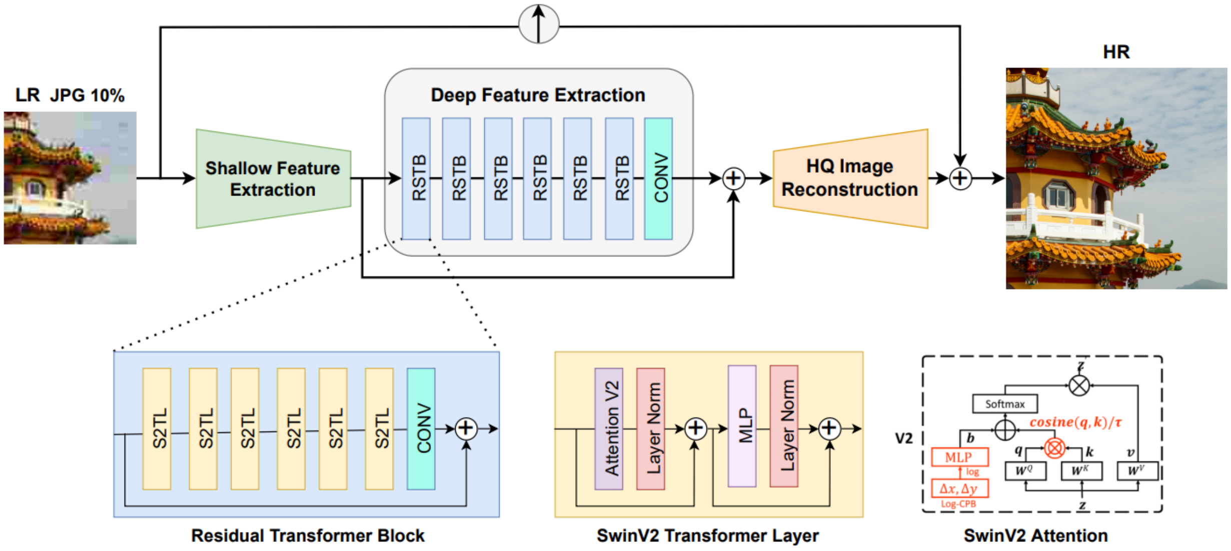

Figure 8: The architecture of the Swin2SR model. This architecture is similar to the SwinIR architecture. Swin2SR modified the residual transformer blocks by swapping the Swin Transformer layers with the new Swin Transformer V2 module. Source: [11].

3.3 Classification Models

For the classification task, we leveraged transfer learning and a fine-tuning approach. Fine-tuning is a common practice to improve the pre-trained model for a specific task [21]. It involves training the pre-trained model further with task-specific data to adapt to the new task. The training and validation sets of the AID dataset were used for this purpose. There are different approaches for fine-tuning a pre-trained network. Nogueira et al. [45] employed 2 different approaches: (1) fine-tuning all layers and (2) fine-tuning only the final layers while keeping the weights of earlier layers fixed. For this thesis, we focused on fine-tuning all the layers because this approach appeared to exhibit superior performance as opposed to only fine-tuning the higher-level layers. A possible explanation for this could be that the data used for the pre-trained model is different compared to the AID dataset. The pre-trained models were trained using the ImageNet [18] dataset which consists of natural images. Liu et al [48] state that it is not feasible to use the common object-detection algorithms trained on the domain of natural images for aerial image object detection. We believe aerial scene classification encounters a similar challenge as the learned representations from ImageNet did not transfer well to the AID dataset.

The following models have been used for the classification task: ResNet50 [62] and ViT [64]. ResNet50 was pre-trained on ImageNet-1K [18] and ViT was pre-trained on ImageNet-21K [65]. The ImageNet-1K dataset consists of more than one million natural images and the ImageNet-21K dataset consists of more than 14 million images. We chose ResNet50 because it is a commonly used baseline model in the computer vision community [66]. Additionally, Helber et al. [46] compared a range of CNNs on the EuroSAT dataset and obtained the best results for the fine-tuned ResNet50 model pre-trained on ImageNet. We selected the ViT model because Dimitrovski et al. [47] showed that this model achieved the highest top 1 accuracy for the AID dataset.

3.3.1 ResNet50: Residual Network 50

ResNet50 [62] is a convolutional neural network consisting of 50 layers and is part of the residual network family. Convolutional layers, pooling layers, and fully connected layers make up the model. The main advancement is the use of residual blocks. Residual blocks contain shortcut connections that allow the output from one layer to be taken to a layer deeper in the model, to preserve information from earlier layers. The ResNet50 model has been applied in several computer vision tasks, such as image classification [67], object detection [68], and semantic segmentation [69].

3.3.2 ViT: Visual Transformer

ViT [64] is a transformer-based [61] network consisting of patch and position embeddings and transformer encoder blocks. We employed the ViT-Base model which consists of 12 transformer encoder blocks. The ViT model divides an image into patches. Patches are equivalent to tokens (words) in natural language processing tasks that use transformer architectures. These patches have a fixed size and do not overlap with each other. Subsequently, the patches are flattened, and then lower dimensional linear embeddings are created of these flattened patches. The results are called patch embeddings. To preserve positional information, positional embeddings are added to every patch embedding. The patch and positional embeddings are then used as input for the transformer encoder blocks. The ViT model uses the same transformer encoder structure as proposed by Vaswani et al. [61]. Each block consists of 2 layers: a multi-headed self-attention layer and a multi-layer perceptron (MLP) layer (feed-forward neural network). MLP layers capture local information while multi-headed self-attention layers capture global information. Additionally, there are residual connections [62] after every layer. The ViT model has been applied in several computer vision tasks, such as image classification [70], object detection [71], and instance segmentation [72].

3.4 Evaluation Metrics

3.4.1 Super-Resolution Evaluation

The evaluation metrics used to assess the performance of the super-resolution models are peak signal-to-noise ratio (PSNR) [73] and structural similarity index (SSIM) [74]. The PSNR, expressed in decibels (dB), indicates how much noise is present between the original image and the modified image. It describes the ratio between the maximum possible value of a signal and the value of noise in the modified image. The formula to calculate the PSNR is shown in Equation 1. f represents the original image and g is the modified image and they are both of size $m * n$. MAX is the maximum possible pixel value. For this study, we used the 3-channel RGB (red, green, and blue) system where the maximum possible pixel value is 255. When multi-spectral satellite images are used, the number and type of channels may differ. The maximum possible pixel value should be adjusted accordingly [44]. The mean squared error (MSE) is computed between the original and modified picture according to Equation 2. Assuming that f and g are 2D arrays of the images, m represents the number of rows and i is the index of the row, while n represents the number of columns, and j is the index of the column. $f(i,j)$ and $g(i,j)$ represent pixel values with indices i and j. The MSE compares the true pixel value to the pixel value of the modified image. In other words, this is called the error. The error is computed for every pixel and after being squared and averaged, this results in the MSE. If the MSE approaches 0, the PSNR approaches infinity. Therefore, a higher PSNR indicates a higher-quality image.

$$PSNR(f,g) = 10 * log_{10}(\frac{MAX^2}{MSE(f,g)}) \tag{1}$$

$$MSE(f,g) = \frac{1}{mn}\sum_{i=1}^{m-1}\sum_{j=1}^{n-1}\|f(i,j) - g(i,j)\|^2 \tag{2}$$

The SSIM is another commonly used metric for assessing image quality [10, 11]. The main difference between SSIM and PSNR is that SSIM takes into account the human visual system for evaluation. SSIM considers 3 components for calculating its index: luminance, contrast, and structure. In Equations 3, 4, 5, and 6, x and y refer to the original and modified images.

Luminance considers the brightness of a picture. Equation 3 is used to compare the luminances of two pictures. μ_x and μ_y take the average value of the pixels in the image to measure the average luminance. The metric $l(x,y)$ ranges between 0 and 1, where 1 indicates that the luminance of the two pictures is exactly the same. C_1 is a stabilizing constant.

$$l(x,y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \tag{3}$$

Contrast measures how the pixel intensities are spread in an image. A high contrast value indicates that there are dark as well as light regions in an image. Equation 4 is used to compare the contrast between two images, where C_2 is also a stabilizing constant. The contrast value is measured using the standard deviation of the pixel values, depicted as σ_x and σ_y in Equation 4.

$$c(x,y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \tag{4}$$

To compare the structures of images, the covariance and correlation are computed between the images. To illustrate, if a pixel in image x is above the mean pixel value and if the corresponding pixel in image y is also above the mean pixel value, then the covariance will increase. The covariance is depicted as σ_{xy} in Equation 5. The structure metric is computed using Equation 5, where C_3 is a stabilizing constant.

$$s(x,y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \tag{5}$$

Ultimately, these three components combined form the SSIM equation depicted in Equation 6. α, β and γ are parameters to adjust the importance of each component. The SSIM ranges from 0 to 1 where a higher score indicates greater resemblance.

$$SSIM(x,y) = [l(x,y)]^{\alpha} \cdot [c(x,y)]^{\beta} \cdot [s(x,y)]^{\gamma} \tag{6}$$

3.4.2 Classification Evaluation

We used top 1 accuracy, top 5 accuracy, precision, recall, and the F1-score for the classification evaluation [75]. The top 5 accuracy metric examines if the target label is included among the 5 predictions with the highest probability. The formulas for the metrics are depicted below. TP, FP, TN, and FN describe the number of true positives, false positives, true negatives, and false negatives respectively. The macro metrics were used because every class is equally important.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{7}$$

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{10}$$

4 Experiments

This section outlines an overview of the experimental setup used in this study. We will discuss both the super-resolution and classification experimental setups, along with the computational resources utilized.

4.1 Super-Resolution Experimental Setup

The AID dataset was used for the experiments. This dataset consists of 10.000 images. These images were first split into a train, validation, and test set. The split was created using the sci-kit-learn library with a random seed of 58. The test size was 10%, thus the test set contains 1000 images. The test set was used for the super-resolution tasks. The training and validation sets were used to fine-tune the ESPCN model.

The test images have been down-sampled using bicubic interpolation from the OpenCV library and are down-sampled with scaling factors of x2, x3, and x4 resulting in image dimensions of 300x300, 200x200, and 150x150 pixels, respectively. The down-scaled images are subsequently super-resolved to their original dimensions of 600x600 pixels using the super-resolution models discussed in Section 3.2. The super-resolution models are compared by calculating the PSNR and SSIM scores of the super-resolved datasets of each scale. In addition, we performed the Wilcoxon signed-rank test to examine the significance of the results between the models [76, 77]. To assess the classification performance of super-resolved datasets, the images were up-scaled beyond their original dimensions using the super-resolution models. The images were enhanced to 1200x1200 pixels, corresponding to a scaling factor of x2.

The ESPCN model underwent fine-tuning for three trials for each scaling factor of x2, x3, and x4. The trials were carried out using random seeds of 35, 36, and 37. The images for fine-tuning were also down-sampled using bicubic interpolation. The Adam optimizer and mean squared error loss were used to compile the model. We used the default learning rate of 0.001 and a batch size of 32. The model was trained for 50 epochs while using model checkpoints and early stopping to prevent overfitting. For the model checkpoints and early stopping, we used a value of 10 for the patience parameter and monitored the loss. To assess the results, we calculated the PSNR and SSIM scores.

4.2 Transfer Learning Experimental Setup

Both the ResNet50 model and ViT model were trained 3 times. We used the training split of the AID dataset for the fine-tuning process. This training split was further split into a training and validation split using the sci-kit-learn library with a random seed of 35, 36, and 37 for the 3 trials. All pixel values of the images were normalized to the range [0, 1] by dividing each value by 255. Furthermore, data augmentation was applied to the training images. Table 1 depicts the parameter configuration for the data augmentation. A data frame was used for loading the labels for the corresponding images.

We used Tensorflow version 2.12.0 for the implementation of the ResNet50 classification model. ResNet50 takes input images with resolutions of 224x224 pixels, therefore all images were resized to these dimensions. Subsequently, an AveragePooling2D layer, a Flatten layer, a Dense layer, a Dropout layer, and another Dense layer were added to the base model.
We employed the Keras implementation for the ViT classification model. We used the ViT-B32 model so with a patch size of 32. We used input image sizes of 256x256 considering the patch size of 32. We added a Flatten layer, a BatchNormalization layer, a Dense layer, and another BatchNormalization layer to the base model.

The last Dense layers of both models have a softmax activation function since it is a multi-class classification task. We did not freeze any layers during the fine-tuning process. The Adam optimizer and categorical cross-entropy loss were used to compile the models. We used the default learning rate of 0.001 and a batch size of 64. The models were trained for 50 epochs while using model checkpoints and early stopping. For the model checkpoints and early stopping, we used a value of 10 for the patience parameter and monitored the loss. We computed the evaluation metrics top 1 accuracy, top 5 accuracy, precision, recall, and the F-1 score for both the original images and the super-resolved images. We performed the Wilcoxon signed-rank test to examine the significance of the results among the models and datasets [76, 77].

Parameter	Setting
rotation_range	20
zoom_range	0.15
width_shift_range	0.2
height_shift_range	0.2
shear_range	0.15
horizontal_flip	True

Table 1: Data augmentation configurations for the ResNet50 and ViT implementation.

4.3 Computational Resources

All the experiments were conducted using the GPU from the Grace cluster of the ADA research group from Leiden University. The duration of each experiment varied depending on the specific task, ranging from a few minutes to 10 hours. The fine-tuning experiments of the classification models required a training duration of approximately 10 hours.

5 Results

This section presents our results of the super-resolution tasks and the classification tasks of this study.

5.1 Super-Resolution Results

Table 2 and Figure 9 present the results of the super-resolution part of this study. The ESPCN model demonstrates superior performance for the PSNR and SSIM scores compared to the other models across all scaling factors. This finding is unexpected considering previous research by Liang et al. [10] and Conde et al. [11], which indicated that both SwinIR and Swin2SR models outperformed several CNN-based super-resolution models.

There is a considerable difference between the architecture of the ESPCN model and the architecture of the Swin models. The SwinIR and Swin2SR frameworks are deeper in terms of layers compared to the ESPCN framework, and they contain residual Swin transformer blocks. The exact details of the architectures of these models are discussed in Section 3.2. Comparing the architectures of the ESPCN model and the Swin models suggest that the use of the residual Swin transformer blocks does not result in higher PSNR and SSIM for the AID dataset. Furthermore, the lightweight SwinIR model yields higher PSNR and SSIM scores than the classical SwinIR version. The lightweight version has fewer layers than the classical version. This observation shows that a shallower model produces higher PSNR and SSIM scores for the AID dataset. It is noteworthy that the ESPCN model also contains fewer layers than the Swin models while yielding better results. It is a possibility that a deep network with more layers is susceptible to overfitting. The increased complexity can cause the model to learn the training data too well and is not able to generalize for unseen data [78].

The fine-tuned ESPCN model resulted in lower PSNR and SSIM scores than the non-fine-tuned ESPCN model. An explanation for this could be that we used an ESPCN model that was pre-trained on a different dataset than the non-fine-tuned ESPCN model. The non-fine-tuned ESPCN model was trained on the DIV2K dataset, whereas the fine-tuned ESPCN model was pre-trained on the BSDS500 dataset. There are no results for the scaling factors of x2 and x4 since there was no pre-trained model available for these tasks.

	x2		x3		x4	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
ESPCN	**32.21**	**0.9152**	**27.94**	**0.7948**	**26.27**	**0.7216**
Fine-tuned ESPCN	-	-	27.76 ± 0.22	0.7947 ± 0.00	-	-
Real-ESRGAN	27.20	0.8114	-	-	23.95	0.6419
SwinIR Classical	28.25	0.8384	23.68	0.6835	21.70	0.5978
SwinIR Lightweight	28.52	0.8427	23.98	0.6894	23.98	0.6894
Swin2SR	28.27	0.8386	-	-	21.88	0.6017

Table 2: The mean PSNR and SSIM results of the super-resolution models on the test split of the AID dataset for different scales. The best and second best results are shown in red and blue respectively. The results that are significantly the best are shown in **bold** text. The ESPCN model outperforms the other models for every scale. Some scales were not available, these are marked with "-".

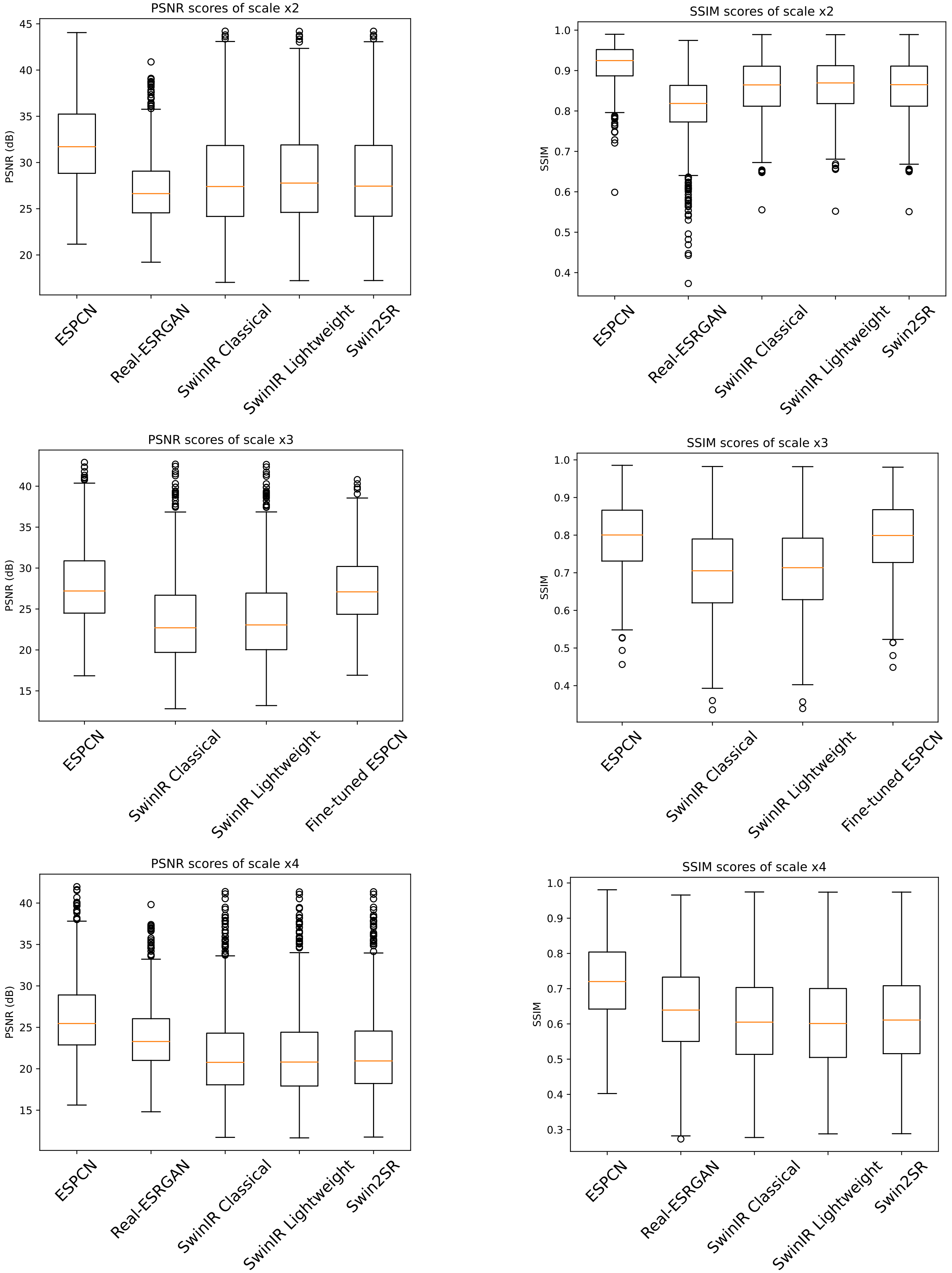

Figure 9: The PSNR and SSIM results of the super-resolved images as box plots for every model and every scaling factor. The ESPCN model outperforms the other models for every scale for both PSNR and SSIM.

We performed the Wilcoxon signed rank test for the difference in results between the best model and the second-best model. These are the ESPCN model and the SwinIR Lightweight model respectively. The Wilcoxon signed rank test was selected because the results are not normally distributed. The results are presented in Table 3. The results for the best model, ESPCN, are statistically significantly better than the results of the second-best model, SwinIR Lightweight, using $\alpha = 0.05$. This is the case for all the scaling factors and for both the PSNR and SSIM scores.

	ESPCN vs. SwinIR lightweight			
	PSNR		SSIM	
	P-value	Significant results?	P-value	Significant results?
Scale x2	3.59e-165	Yes	3.33e-165	Yes
Scale x3	3.34e-165	Yes	3.34e-165	Yes
Scale x4	3.33e-165	Yes	3.33e-165	Yes

Table 3: Results of the Wilcoxon signed-rank test for the difference of results between the ESPCN model and the SwinIR lightweight model. The results show that the ESPCN model is significantly better than the SwinIR lightweight model for all scales using $\alpha = 0.05$.

5.2 Classification Results

Table 4, Figure 10, and Figure 11 present the results of the classification part of this study. The baseline consists of the original pictures. For the fine-tuned ResNet50 model, the ESPCN model was the only model that improved classification for all the metrics. The other models showed no improvement compared to the baseline. This is in line with the super-resolution results. The pre-trained ResNet50 model without fine-tuning was also tested. However, without any fine-tuning, the model exhibits very poor performance with evaluation metrics approaching a value close to 0.0. For the fine-tuned ViT model, the ESPCN model showed superior performance for the top 1 accuracy, precision, recall, and the F1-score. SwinIR and Swin2SR displayed superior performance for the top 5 accuracy. When comparing the two classification models, it is notable that the variance (error bars) of ViT is smaller than the variance of ResNet50. This indicates that the results of ViT are more consistent. Figure 12 shows the confusion matrix of the ResNet50 model with the ESPCN super-resolved dataset.

	SR model	Top 1 Acc.	Precision	Recall	F1-score	Top 5 Acc.
ResNet50	Baseline	0.864±0.023	0.885±0.006	0.866±0.026	0.862±0.021	0.989±0.004
	ESPCN	0.867±0.020	0.886±0.006	0.868±0.024	0.864±0.019	0.990±0.004
	FT ESPCN	0.866 ±0.026	0.886 ±0.009	0.867 ±0.029	0.863 ±0.025	0.989±0.004
	Real-ESRGAN	0.823±0.038	0.866±0.012	0.825±0.043	0.819±0.038	0.978±0.007
	SwinIR	0.816±0.045	0.861±0.019	0.818±0.050	0.812±0.045	0.977±0.009
	Swin2SR	0.836±0.037	0.873±0.012	0.838±0.041	0.835±0.037	0.981±0.008
ViT	Baseline	0.835±0.000	0.845±0.004	0.833±0.000	0.832±0.002	0.972±0.007
	ESPCN	0.838±0.002	0.846±0.006	0.837±0.003	0.833±0.005	0.977±0.007
	FT ESPCN	0.835 ±0.006	0.846 ±0.010	0.834 ±0.005	0.830 ±0.008	0.977±0.008
	Real-ESRGAN	0.814±0.010	0.827±0.011	0.812±0.008	0.809±0.012	0.976±0.007
	SwinIR	0.831±0.008	0.838±0.008	0.830±0.006	0.826±0.009	0.978±0.006
	Swin2SR	0.835±0.004	0.845±0.006	0.834±0.002	0.831±0.006	0.978±0.007

Table 4: Top 1 accuracy, precision, recall, F1-score, and top 5 accuracy results for ResNet50 and ViT. The best results are shown in red. For ResNet50, ESPCN outperforms every other model for all metrics. For ViT, ESPCN outperforms every metric except the top 5 accuracy. SwinIR and Swin2SR show superior results for the top 5 accuracy.

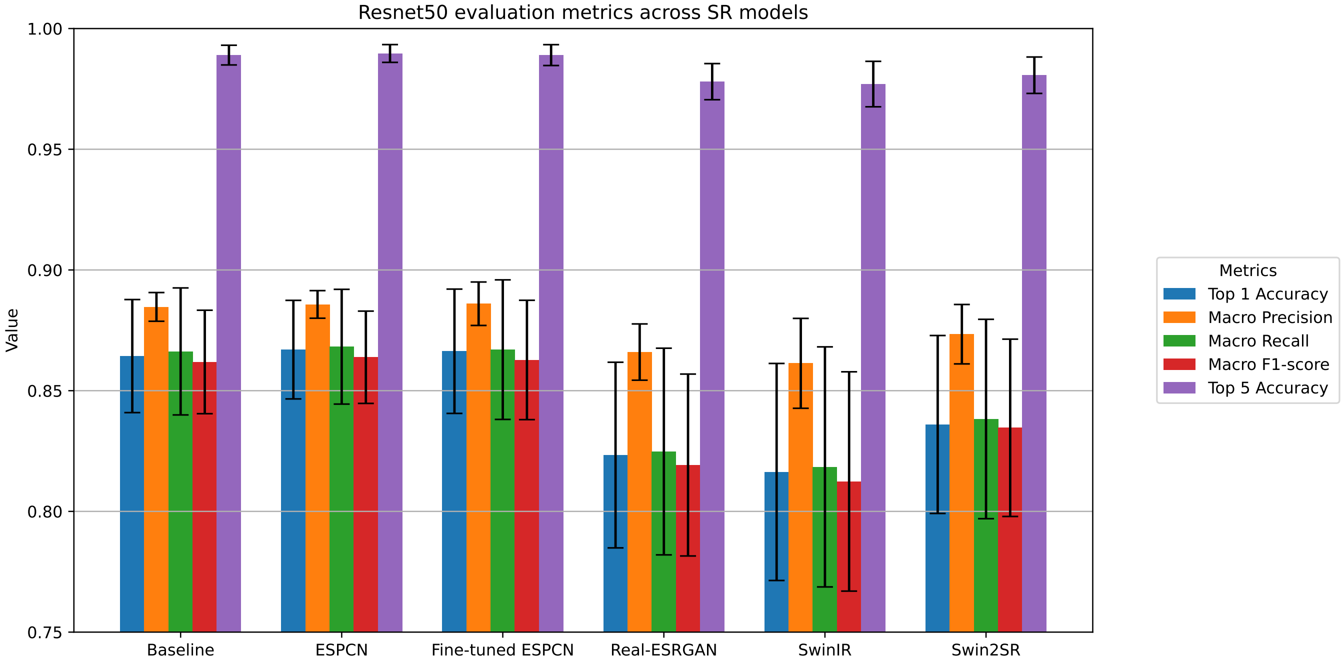

Figure 10: The results of the finetuned ResNet50 model for the different super-resolved datasets using the selected super-resolution models. The baseline is the original test images without any super-resolution. The (fine-tuned) ESPCN model demonstrated improvements in all evaluation metrics compared to the baseline. The other models did not display enhancements in any evaluation metric when compared to the baseline.

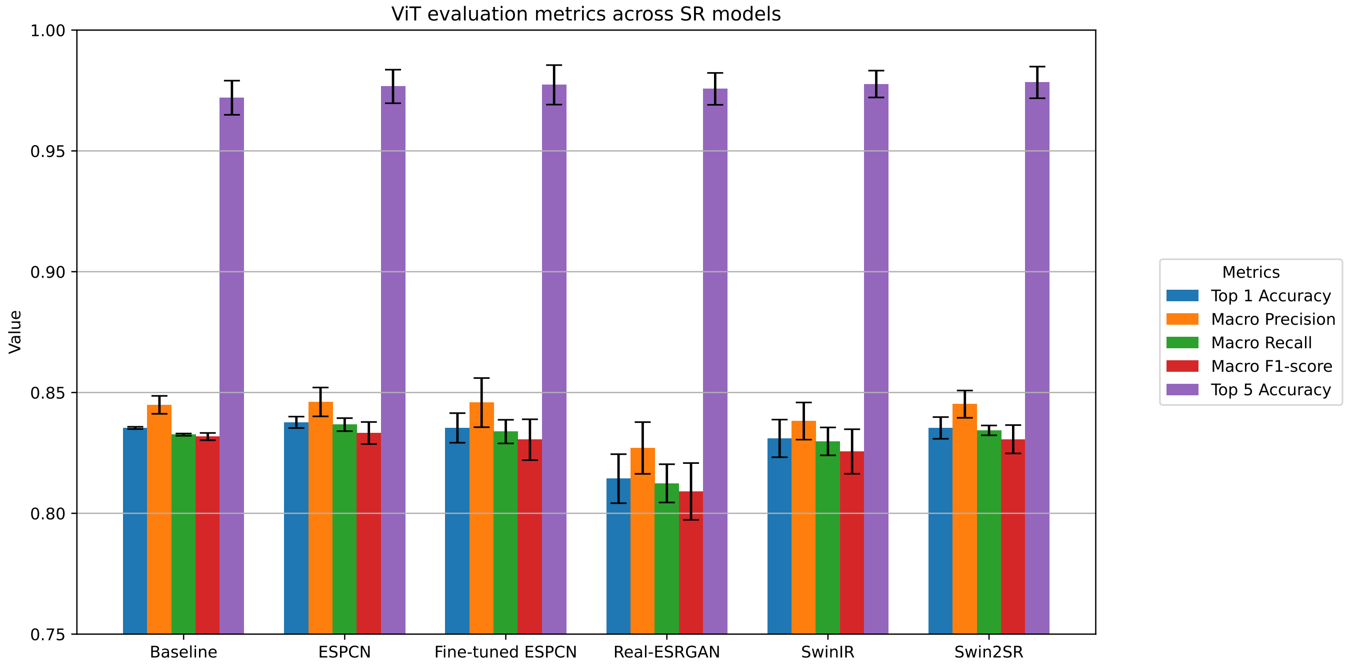

Figure 11: The results of the finetuned ViT model for the different super-resolved datasets using the selected super-resolution models. The baseline is the original test images without any super-resolution. The (fine-tuned) ESPCN model displays improved performance for the top 1 accuracy, precision, recall, and the F1-score. SwinIR and Swin2SR demonstrate improved results for the top 5 accuracy. It is notable that the ViT model contains smaller error bars, indicating more consistent performance.

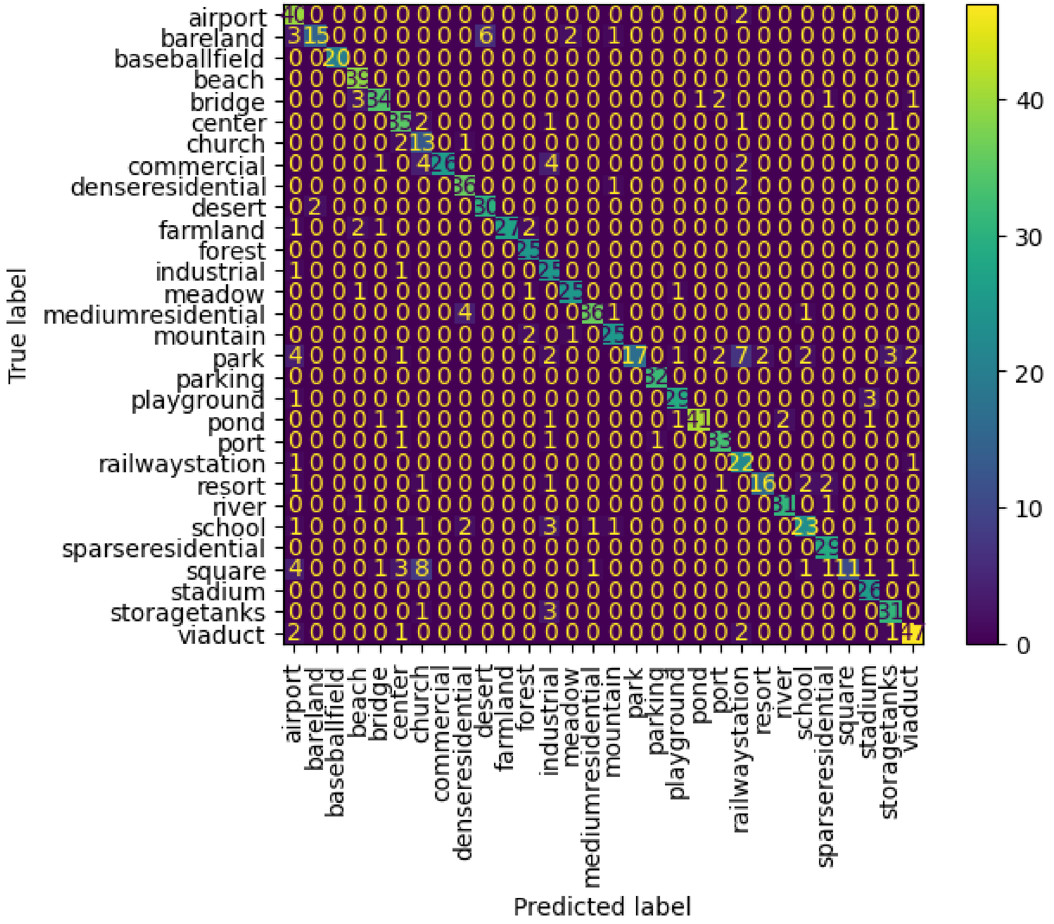

Figure 12: Confusion matrix using the pre-trained and fine-tuned ResNet50 model with ESPCN super-resolved dataset.

We performed the Wilcoxon signed rank test for the difference in results between the ResNet50 baseline and the ResNet50 ESPCN, and for the difference in results between the ViT baseline and the best results of the super-resolution models for ViT (ESPCN, SwinIR, and Swin2SR). The results are presented in Table 5 and Table 6. The results of the super-resolution models for both ResNet50 and ViT are not statistically significantly better than the results of the baseline using $\alpha = 0.05$.

	ResNet50 baseline vs. ResNet50 ESPCN	
Metric	P-value	Significant results?
Top 1 accuracy	0.5	No
Precision	0.5	No
Recall	0.75	No
F1-score	0.75	No
Top 5 accuracy	0.16	No

Table 5: The results of the Wilcoxon signed rank test for the difference of results between the ResNet50 baseline and the ResNet50 ESPCN model. The results show that the super-resolved ESPCN dataset is not significantly better than the baseline consisting of the native-resolution images.

	ViT baseline vs. ViT ESPCN, SwinIR, Swin2SR	
Metric	P-value	Significant results?
Top 1 accuracy (ESPCN)	0.25	No
Precision (ESPCN)	0.75	No
Recall (ESPCN)	0.25	No
F1-score (ESPCN)	0.75	No
Top 5 accuracy (SwinIR)	0.25	No
Top 5 accuracy (Swin2SR)	0.25	No

Table 6: The results of the Wilcoxon signed rank test for the difference of results between the ViT baseline and the ViT ESPCN, SwinIR, and Swin2SR models. The results show that the super-resolved datasets are not significantly better than the baseline consisting of the native-resolution images.

6 Conclusions and Further Research

In this thesis, we have explored the application of super-resolution as a pre-processing technique to enhance the resolution of aerial imagery. We evaluated a range of pre-trained super-resolution models, including ESPCN, Real-ESRGAN, SwinIR, and Swin2SR using the evaluation metrics PSNR and SSIM. Additionally, we fine-tuned the ESPCN model using domain-specific data and evaluated its performance. The findings of our super-resolution study indicated that the ESPCN model has significantly the best results compared to the other models. Our results also demonstrate that shallower super-resolution models display higher PSNR and SSIM scores for the AID dataset. We assessed the effect of super-resolution on the downstream task of aerial scene classification. We utilized ResNet50 and ViT with transfer learning for the classification task and performed a statistical test. The ESPCN and Swin models yielded higher classification evaluation scores compared to the native-resolution images. However, the results of the significance test revealed that our super-resolution approach did not yield significant results for the classification tasks. Even the ESPCN model, which was significantly the best-performing super-resolution model, did not demonstrate significant results for the classification task.

In terms of further research, there are multiple angles that could be explored. One could explore fine-tuning all the pre-trained super-resolution models used in this study, as well as performing hyperparameter optimization. In addition, we used a limited selection of super-resolution models. One could investigate other super-resolution frameworks to gain a broader understanding of the available models. Furthermore, we only used the AID dataset in this study. Further research could entail assessing and comparing multiple remote sensing imagery datasets. Lastly, we did not analyze the computational efficiency of the super-resolution frameworks. In real-life applications, detecting a forest fire, for example, it is important that these frameworks are able to operate at high speeds.

References

[1] X. Wang, J. Yi, J. Guo, Y. Song, J. Lyu, J. Xu, W. Yan, J. Zhao, Q. Cai, and H. Min, "A review of image super-resolution approaches based on deep learning and applications in remote sensing," *Remote Sensing*, vol. 14, no. 21, 2022.

[2] P. Wang, B. Bayram, and E. Sertel, "A comprehensive review on deep learning based remote sensing image super-resolution methods," *Earth-Science Reviews*, vol. 232, p. 104110, 2022.

[3] Z. Wang, J. Chen, and S. C. H. Hoi, "Deep learning for image super-resolution: A survey," 2020.

[4] P. Mathur, A. K. Singh, S. Azeemuddin, J. Adoni, and P. Adireddy, "A real-time super-resolution for surveillance thermal cameras using optimized pipeline on embedded edge device," in *2021 17th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 1–7, 2021.

[5] W. Du and H. Tian, "Transformer and gan based super-resolution reconstruction network for medical images," 2022.

[6] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," 2015.

[7] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," 2016.

[8] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, C. C. Loy, Y. Qiao, and X. Tang, "Esrgan: Enhanced super-resolution generative adversarial networks," 2018.

[9] X. Wang, L. Xie, C. Dong, and Y. Shan, "Real-esrgan: Training real-world blind super-resolution with pure synthetic data," 2021.

[10] J. Liang, J. Cao, G. Sun, K. Zhang, L. V. Gool, and R. Timofte, "Swinir: Image restoration using swin transformer," 2021.

[11] M. V. Conde, U.-J. Choi, M. Burchi, and R. Timofte, "Swin2sr: Swinv2 transformer for compressed image super-resolution and restoration," 2022.

[12] G. Cheng, X. Xie, J. Han, L. Guo, and G.-S. Xia, "Remote sensing image scene classification meets deep learning: Challenges, methods, benchmarks, and opportunities," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 3735–3756, 2020.

[13] T. Zhang and X. Huang, "Monitoring of urban impervious surfaces using time series of high-resolution remote sensing images in rapidly urbanized areas: A case study of shenzhen," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. PP, pp. 1–17, 02 2018.

[14] A. Tayyebi, B. C. Pijanowski, and A. H. Tayyebi, "An urban growth boundary model using neural networks, gis and radial parameterization: An application to tehran, iran," *Landscape and Urban Planning*, vol. 100, no. 1, pp. 35–44, 2011.

[15] X. Li and G. Shao, "Object-based urban vegetation mapping with high-resolution aerial photography as a single data source," *International Journal of Remote Sensing*, vol. 34, no. 3, pp. 771–789, 2013.

[16] G. Cheng, K. Li, T. Zhao, J. Han, H. Li, and J. Fang, "Automatic landslide detection from remote-sensing imagery using a scene classification method based on bovw and plsa," *International Journal of Remote Sensing*, vol. 34, pp. 45–59, 01 2013.

[17] D. Marmanis, M. Datcu, T. Esch, and U. Stilla, "Deep learning earth observation classification using imagenet pretrained networks," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 1, pp. 105–109, 2016.

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

[19] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," 2020.

[20] T. Liu, S. Alibhai, J. Wang, Q. Liu, X. He, and C. Wu, "Exploring transfer learning to reduce training overhead of hpc data in machine learning," in *2019 IEEE International Conference on Networking, Architecture and Storage (NAS)*, pp. 1–7, 2019.

[21] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," 2014.

[22] N. Panchal, B. Limbasiya, and A. Prajapati, "Survey on multi-frame image super-resolution," *International Journal of Scientific & Technology Research*, vol. 2, pp. 233–237, 2013.

[23] R. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, no. 6, pp. 1153–1160, 1981.

[24] D. Khaledyan, A. Amirany, K. Jafari, M. H. Moaiyeri, A. Z. Khuzani, and N. Mashhadi, "Low-cost implementation of bilinear and bicubic image interpolation for real-time image super-resolution," 2020.

[25] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," 2016.

[26] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," 2016.

[27] J. Kim, J. K. Lee, and K. M. Lee, "Deeply-recursive convolutional network for image super-resolution," 2016.

[28] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, “Photo-realistic single image super-resolution using a generative adversarial network,” 2017.

[29] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, “Self-attention generative adversarial networks,” 2019.

[30] Y. Tai, J. Yang, and X. Liu, “Image super-resolution via deep recursive residual network,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2790–2798, 2017.

[31] N. Ahn, B. Kang, and K.-A. Sohn, “Fast, accurate, and lightweight super-resolution with cascading residual network,” 2018.

[32] T. Tong, G. Li, X. Liu, and Q. Gao, “Image super-resolution using dense skip connections,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 4809–4817, 2017.

[33] P. Wang and E. Sertel, “Channel–spatial attention-based pan-sharpening of very high-resolution satellite images,” *Knowledge-Based Systems*, vol. 229, p. 107324, 2021.

[34] B. Niu, W. Wen, W. Ren, X. Zhang, L. Yang, S. Wang, K. Zhang, X. Cao, and H. Shen, “Single image super-resolution via a holistic attention network,” 2020.

[35] T. Dai, J. Cai, Y. Zhang, S.-T. Xia, and L. Zhang, “Second-order attention network for single image super-resolution,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11057–11066, 2019.

[36] F. Yang, H. Yang, J. Fu, H. Lu, and B. Guo, “Learning texture transformer network for image super-resolution,” 2020.

[37] S. Tingting, “A light model for super-resolution of remote sensing images,” *Journal of Physics: Conference Series*, vol. 2171, p. 012029, jan 2022.

[38] J. Wasala, M. Baratchi, and S. Marselis, “Autosr-rs: An automl approach to super-resolution for remote sensing,” June 2022.

[39] J. Shermeyer and A. V. Etten, “The effects of super-resolution on object detection performance in satellite imagery,” 2019.

[40] L. Courtrai, M.-T. Pham, and S. Lefèvre, “Small object detection in remote sensing images based on super-resolution with auxiliary generative adversarial networks,” *Remote Sensing*, vol. 12, p. 3152, 09 2020.

[41] Y. R. Musunuri, O.-S. Kwon, and S.-Y. Kung, “Srodnet: Object detection network based on super resolution for autonomous vehicles,” *Remote Sensing*, vol. 14, no. 24, 2022.

[42] L. Wang, D. Li, Y. Zhu, L. Tian, and Y. Shan, “Dual super-resolution learning for semantic segmentation,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3773–3782, 2020.

[43] T. Zhang, H. Tang, Y. Ding, P. Li, C. Ji, and P. Xu, "Fsrss-net: High-resolution mapping of buildings from middle-resolution satellite images using a super-resolution semantic segmentation network," *Remote Sensing*, vol. 13, no. 12, 2021.

[44] N. R. Palacios Salinas, M. Baratchi, J. N. van Rijn, and A. Vollrath, "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*, (Berlin, Heidelberg), p. 447–462, Springer-Verlag, 2021.

[45] K. Nogueira, J. dos Santos, T. Fornazari, T. Silva, P. Morellato, and R. Torres, "Towards vegetation species discrimination by using data-driven descriptors," pp. 1–6, 12 2016.

[46] P. Helber, B. Bischke, A. Dengel, and D. Borth, "Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification," 2019.

[47] I. Dimitrovski, I. Kitanovski, D. Kocev, and N. Simidjievski, "Current trends in deep learning for earth observation: An open-source benchmark arena for image classification," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 197, pp. 18–35, mar 2023.

[48] Z. Liu, X. Zhang, C. Liu, H. Wang, C. Sun, B. Li, W. Sun, P. Huang, Q. Li, Y. Liu, H. Kuang, and J. Xiu, "Relationrs: Relationship representation network for object detection in aerial images," 2021.

[49] S. Yang, W. Xiao, M. Zhang, S. Guo, J. Zhao, and F. Shen, "Image data augmentation for deep learning: A survey," 2022.

[50] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," 2017.

[51] R. Poojary, R. Raina, and A. K. Mondal, "Effect of data-augmentation on fine-tuned cnn model performance," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 10, 12 2020.

[52] A. Mikołajczyk and M. Grochowski, "Data augmentation for improving deep learning in image classification problem," in *2018 International Interdisciplinary PhD Workshop (IIPhDW)*, pp. 117–122, 2018.

[53] G.-S. Xia, J. Hu, F. Hu, B. Shi, X. Bai, Y. Zhong, L. Zhang, and X. Lu, "AID: A benchmark data set for performance evaluation of aerial scene classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, pp. 3965–3981, jul 2017.

[54] M. Sordo and Q. Zeng, "On sample size and classification accuracy: A performance comparison," in *Biological and Medical Data Analysis* (J. L. Oliveira, V. Maojo, F. Martín-Sánchez, and A. S. Pereira, eds.), (Berlin, Heidelberg), pp. 193–201, Springer Berlin Heidelberg, 2005.

[55] A. Alwosheel, S. van Cranenburgh, and C. G. Chorus, "Is your dataset big enough? sample size requirements when using artificial neural networks for discrete choice analysis," *Journal of Choice Modelling*, vol. 28, pp. 167–182, 2018.

[56] L. Shao, F. Zhu, and X. Li, "Transfer learning for visual categorization: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 5, pp. 1019–1034, 2015.

[57]

[58] E. Agustsson and R. Timofte, "Ntire 2017 challenge on single image super-resolution: Dataset and study," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.

[59] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th Int'l Conf. Computer Vision*, vol. 2, pp. 416–423, July 2001.

[60] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," 2021.

[61] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017.

[62] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[63] Z. Liu, H. Hu, Y. Lin, Z. Yao, Z. Xie, Y. Wei, J. Ning, Y. Cao, Z. Zhang, L. Dong, F. Wei, and B. Guo, "Swin transformer v2: Scaling up capacity and resolution," 2022.

[64] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2021.

[65] T. Ridnik, E. Ben-Baruch, A. Noy, and L. Zelnik-Manor, "Imagenet-21k pretraining for the masses," 2021.

[66] B. Koonce, *Convolutional Neural Networks with Swift for Tensorflow: Image Recognition and Dataset Categorization.* 01 2021.

[67] M. A. Kadhim and M. H. Abed, *Convolutional Neural Network for Satellite Image Classification*, pp. 165–178. Cham: Springer International Publishing, 2020.

[68] N. Dvornik, K. Shmelkov, J. Mairal, and C. Schmid, "Blitznet: A real-time deep network for scene understanding," 2017.

[69] D. Kwon and S. Kwak, "Semi-supervised semantic segmentation with error localization network," 2022.

[70] C.-F. Chen, Q. Fan, and R. Panda, "Crossvit: Cross-attention multi-scale vision transformer for image classification," 2021.

[71] Y. Li, H. Mao, R. Girshick, and K. He, "Exploring plain vision transformer backbones for object detection," 2022.

[72] W. Chen, X. Du, F. Yang, L. Beyer, X. Zhai, T.-Y. Lin, H. Chen, J. Li, X. Song, Z. Wang, and D. Zhou, "A simple single-scale vision transformer for object localization and instance segmentation," 2022.

[73] A. Horé and D. Ziou, "Image quality metrics: Psnr vs. ssim," in *2010 20th International Conference on Pattern Recognition*, pp. 2366–2369, 2010.

[74] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[75] S. Thirumaladevi, K. Veera Swamy, and M. Sailaja, "Remote sensing image scene classification by transfer learning to augment the accuracy," *Measurement: Sensors*, vol. 25, p. 100645, 2023.

[76] G.-L. Chen, C.-C. Hsu, and M.-H. Wu, "Adaptive distribution learning with statistical hypothesis testing for covid-19 ct scan classification," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pp. 471–479, October 2021.

[77] M. Lopez-Martin, A. Nevado, and B. Carro, "Detection of early stages of alzheimer's disease based on meg activity with a randomized convolutional neural network," *Artificial Intelligence in Medicine*, vol. 107, p. 101924, 07 2020.

[78] X. Ying, "An overview of overfitting and its solutions," *Journal of Physics: Conference Series*, vol. 1168, p. 022022, feb 2019.