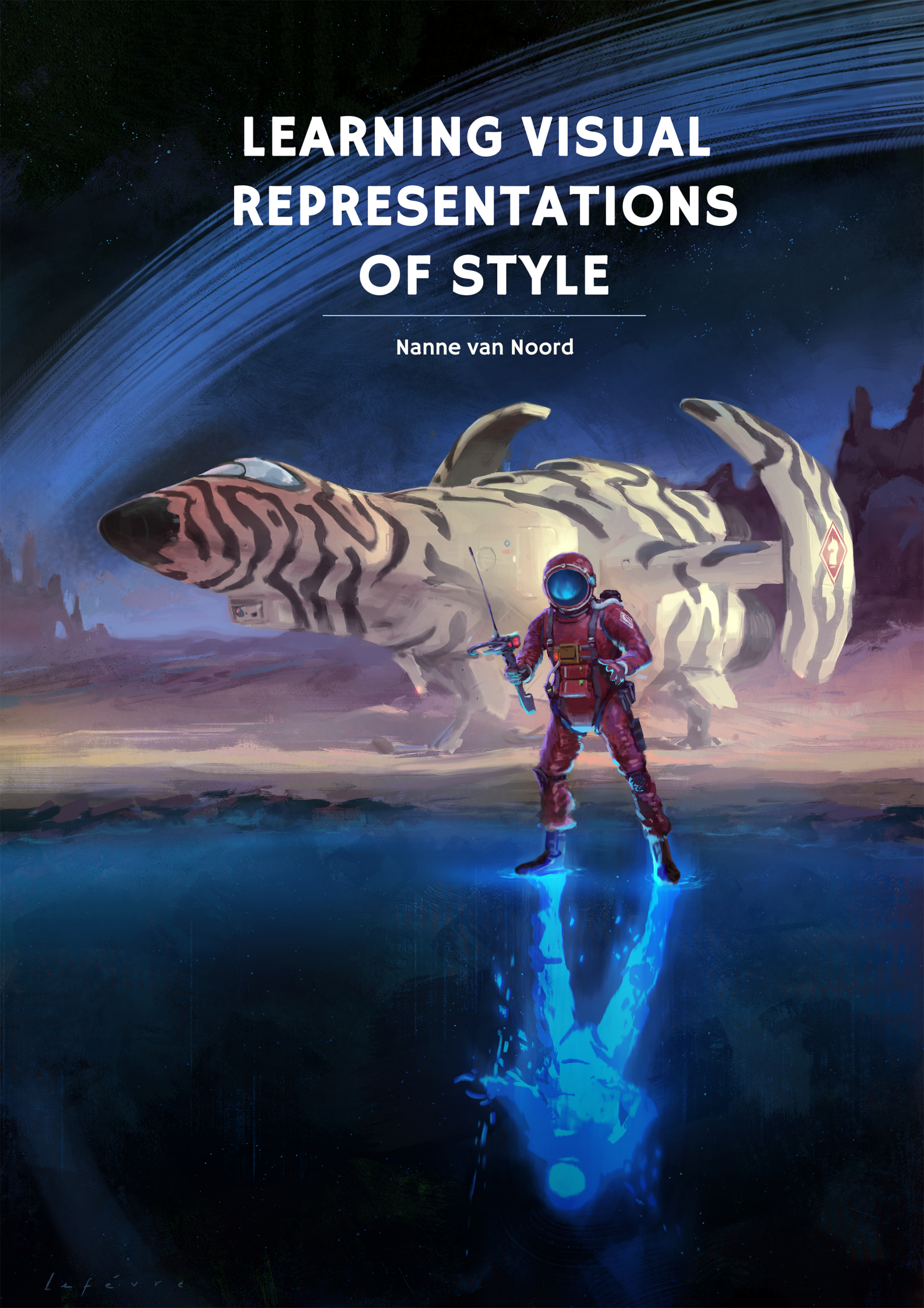
LEARNING VISUAL REPRESENTATIONS OF STYLE

Nanne van Noord



Learning visual representations of style

by Nanne van Noord

Learning visual representations of style

Nanne van Noord

PhD Thesis

Tilburg University, 2018

TICC PhD Series, No. 60

The research reported in this thesis is performed as part of the REVIGO project, supported by the Netherlands Organisation for scientific research (NWO; grant 323.54.004) in the context of the Science4Arts research program.

Cover Design: Joeri Léfevre.

Learning visual representations of style

Proefschrift ter verkrijging van de graad van doctor
aan Tilburg University
op gezag van de rector magnificus,
prof dr. E. H. L. Aarts,
in het openbaar te verdedigen ten overstaan van
een door het college voor promoties aangewezen commissie
in de aula van de Universiteit
op woensdag 16 mei 2018 om 10.00 uur

door

Nanne Jan Eie van Noord

geboren te Goes

Promotores:

Prof. Dr. E.O. Postma

Prof. Dr. M.M. Louwerse

Promotiecommissie:

Prof. Dr. P. Abry

Prof. Dr. R.H. Chan

Prof. Dr. A. Dooms

Prof. Dr. H.J. van den Herik

Prof. Dr. M. Worring

Contents

1

Introduction

1.1 A representation of style

An artist's style is reflected in their artworks, independent from what is depicted [115]. Two artworks created by the same artist that depict two vastly different scenes (e.g., a beach scene and a forest scene) both reflect their style. Stylistic characteristics of an artwork can be used by experts, and sometimes even laymen, to identify the artist that created the artwork. The ability to recognize styles and relate these to artists is associated with *connoisseurship*. Hendriks and Hughes defined connoisseurship as the ability to recognise the artist's style [44]. Connoisseurship is essential in the tasks of authentication and restoration of artworks, because both tasks require detailed knowledge of stylistic characteristics of the artist. Realising connoisseurship in a computer is the main goal of this thesis. The additional goal of the thesis is to give a computer the ability to produce artworks in the style of the artist.

To make it possible for a computer to analyse an artwork it has to be digitised, for instance by means of photography. By photographing the artwork it is converted into a grid of numerical values (i.e., a digital image), where each grid square represents a single pixel with a certain colour specified by the numerical value(s). For image analysis these pixels are considered the raw data, the lowest level of representation. From a mathematical point of view, an image can be thought of as a point (or vector) in *image-space*, a high-dimensional space where each dimension represents the value of a single pixel. The image-space representation of images may be beneficial for image analysis. Images of objects and shapes which are similar in their pixels values, tend to be clustered together in this huge space [113]. The feasibility of image-space representations are hampered by their vulnerability to changes in the appearances of images that do not affect their interpretation. For example, the semantic interpretation of the picture of a cat sitting on the left side of a couch in Figure 1.1(a), is (nearly) identical to the interpretation of the horizontally flipped version in Figure 1.1(b), yet the image-space representations are vastly different. The distance between the image-

space points representing Figure 1.1(a) and Figure 1.1(b) is very large, despite their conceptual similarity. The reason is that the individual pixel values (i.e., the dimensions of the image space) are very different for both images. In computer vision this discrepancy is referred to as the *semantic gap*, the lack of coincidence between the low-level information captured by the pixel values and the high-level interpretation by a human [106].

(a) **(b)**

Figure 1.1: Picture of a cat, and the same picture horizontally flipped. The image-space representations of these pictures are vastly different, yet the semantic interpretation of both pictures is the same.

To overcome the semantic gap it is necessary to perform image analysis using higher-level representations which are abstractions of the raw data, and which describe image features in a manner that is invariant to irrelevant changes in the image [81]. Historically, these features have been obtained by extracting the features with an algorithm (i.e., a feature extractor) which was specifically designed by computer vision researchers for a certain task, through a process called *feature engineering* [6]. This process consists of iteratively designing, creating, and testing feature extractors, often by involving domain experts. Feature engineering has traditionally been the most critical and labour intensive component in the computer vision pipeline [6]. Some of the earliest works on image analysis used representations obtained by specifically modelling the colours, shapes, or texture of images [110, 55, 39], later work focused on more complex and holistic representations [20, 6]. Before turning to an outline of representation learning, what follows is a brief review of five popular feature extractors for image

analysis: colour, shape, texture, local image features, and bag of visual words.

Colour is commonly represented by means of a histogram, as proposed by Swain and Ballard [110]. Colour histograms describe the colour globally, and are by design largely invariant to changes introduced by rotation, translation, and occlusion [31]. These histograms are constructed by quantising the colour values of the individual pixels of an image, irregardless of the spatial arrangement of the pixels. Randomly scrambling the position of all the pixels in an image does not change its colour histogram. This loss of spatial information is the main drawback of colour histograms [46], impeding their applicability to tasks that require understanding of the spatial arrangement of images. For instance, the spatial information that allows us to distinguish the French flag from the Dutch flag is not present in colour histograms, making it impossible to distinguish between these flags based on their colour histograms alone.

Objects in images can be described by their **shape**, in a manner which is invariant to transformations (e.g., rotation and scaling). Shape descriptors are either contour-based or region-based, where contour-based approaches model the boundary of the shape, and region-based approaches model all the pixels within the shape [132]. Shape descriptors have been successfully used in work on object recognition [5, 69], where they were applied to datasets of small and typically simple images, such as the MPEG-7 Shape database which consist of object silhouettes. A limitation of shape descriptors is that invariant descriptions are typically only possible for simple shapes, such as object silhouettes. Therefore, for complex shapes it becomes necessary to construct multiple descriptors at varying sizes, positions, and orientations [55], which is slow and generally infeasible for very large datasets consisting of huge numbers of different objects and shapes.

Texture features aim to capture the spatial arrangement of intensities or colours. Texture features are generally divided into structural and statistical features [39, 119]. Structural texture features are best suited for macro-level spatial configurations, such as brick walls or patterned wallpaper. Structural approaches assume texture consists of a (semi-)regular pattern and the representations of structural texture describe the

basic pattern (texel) being repeated and its positioning rules. Statistical texture features are best suited for micro-level spatial configurations, such as tissue samples or paper grain patterns. Statistical approaches model texture by means of sets of statistics extracted, such as gray-level co-occurrences [40]. Spatial filters are widely used for texture analysis, with Gabor filters being a prominent example of spatial filters [53, 84]. When applied to images, filters give a numerical response to the presence of local image characteristics. Gabor filters respond to gratings with a specific spatial frequency and spatial orientation. A Gabor filter bank contains filters of a range of spatial frequencies and orientations. Filter banks have been successfully applied to a wide variety of image processing tasks, for instance texture classification [84], texture segmentation [53], and object detection [54]. The main disadvantage of texture features, such as filter banks, is that the range of spatial frequencies and orientations should be appropriate for the task at hand. Which orientations and frequencies are appropriate is not always easy to establish.

A number of works have aimed to combine the aforementioned features, such as the combination of colour and shape [55, 31] or of colour and texture [24]. A particularly successful approach was found by describing images using **local image features**, such as Scale-Invariant Feature Transform (SIFT) features [83]. Local image features aim to describe points of interest (keypoints) of images. Keypoints are image patches that stand out or contain interesting information. SIFT features describe keypoints by computing the gradient magnitude and orientation at image sample points and summarising them in a histogram. This allows SIFT features to be largely invariant to rotation, scale, and affine transformations. Local image features were found to be most successful in image matching tasks, in which the challenge is to match two images of the same object or scene from different viewpoints or at different scales [12]. SIFT is able to match the keypoints of such images, in effect solving the matching task. A limitation of local image features is that they only describe the keypoints, and do not characterise the image as a whole.

To overcome the limitations of local image features the **bag of visual words** (BoV)

model was developed [20]. In this model local descriptors extracted from multiple images are clustered to find a limited set of *visual words*, i.e., the building blocks of images. Using these visual words, each image can be characterised as a histogram of occurrences of the visual words. The most successful variants of the BoV model are Fisher Vectors [92], and vector of locally aggregated descriptors (VLAD) [56]. Fisher vectors use Gaussian Mixture Models to construct the visual word dictionary, whereas VLAD uses K-means clustering. Additionally, Fisher vectors encode second-order information about the features, whereas VLAD does not. The BoV model variants differ in the way that the features are engineered by (i) how the visual dictionary is defined, (ii) how local image features are defined, and (iii) what information about features is stored. Feature engineering is a crucial optimisation procedure for each type of application or dataset. The time investment necessary for feature engineering limits the usefulness for BoV based approaches on less researched domains, such as artwork analysis, where there are no developed best practices.

An alternative to feature engineering, which has gained immense popularity in the last decade, is **representation learning**, an approach that learns to capture the information that is useful for the task at hand [6]. Specifically, representation learning has become popular because of an insurgence of deep learning methods, methods which learn representations by stacking multiple (parameterised) nonlinear transformations. Most deep learning methods are types of neural networks (e.g., recurrent or convolutional neural networks), with specific implementations that deal well with the particularities of the data. Deep learning methods have been remarkably successful on a wide range of tasks [75, 98]. In this thesis we explore whether these successes can be generalised to the domain of art investigation. Specifically, we aim to learn representations of the artist's style that enable recognition of the style, and image generation in accordance with the style.

In the remainder of this chapter we will give an overview of the field of image analysis for art investigation in Section 1.2. Followed by the problem statement and the accompanying research questions in Section 1.3. In Section 1.4 we give a primer on the

methodology used for learning representations in this thesis, i.e., deep learning. We conclude this chapter in Section 1.5 with an overview of the structure of the thesis.

1.2 Image analysis for art investigation

Building on a rich history of prior successful programmes, NWO launched the Science4arts programme in 2012. The objective of the programme was to bring together conservators, curators, humanities and science researchers to focus on the changes that artworks and historic objects undergo. Within this programme the Re-assessing Vincent van Gogh (REVIGO) project was funded to study Vincent van Gogh's use of colour by means of digital reconstructions, and to potentially derive lessons for the conservation and interpretation of paintings and drawings of other artists. One aspect of the REVIGO project, as reported in this thesis is to provide methods for the digital analysis and rendering of artworks and their partial reconstructions.

The use of computer algorithms in the study of art works is quite new. Specifically, the characterisation of stylistic features was initiated about a decade ago (see, e.g., [64]). In the past years, there is a surge in the digital analysis of artworks [79, 18, 30, 109]. This change is mainly due to the increasing availability of large datasets of digital representations of cultural heritage (e.g., photographs, X-Ray scans, and 3D scans), which has made it feasible to use data-hungry image analysis algorithms [58].

Although image analysis research for art investigation has focused on a wide variety of tasks, four main tasks are identified and discussed based on their popularity and relevance to learning representations of the artist's style: (1) Artist attribution, (2) Style classification, (3) Neural style transfer, and (4) Inpainting.

Artist attribution. For a number of - often famous - artworks, either the creators are unknown or there is an ongoing debate about the veracity of the attributions. Historically, artworks have been attributed to artists by connoisseurs, where the merit of the attribution depends on their reputation. Although analytical techniques which investigate the chemical composition of an artwork (e.g., inks or pigments [134]), or physical properties of the materials (e.g., thread counting and canvas weave matching

[61, 116]) are commonly used, the attribution often still relies on visual assessments [58]. Although art experts aim to be objective, some level of subjectivity is impossible to avoid. However, a growing body of work has emerged which aims to support art experts by providing analytical tools which perform artist attribution through automatic visual assessment [58, 47, 112, 79, 118, 1, 85, 109].

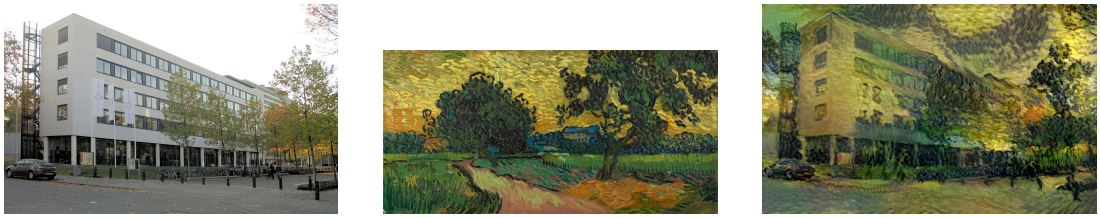
Typically, such automated tools have relied on a predefined notion of how style can be recognised, e.g., by looking at brush strokes [79] or by quantifying material textures [121]. Despite the promise of these tools, their success has only been shown on small datasets, consisting of artworks by only a few artists. For larger datasets these predefined notions and associated engineered features, might not cover the breadth of variation. For example, two artists might be highly similar in material choices and brush stroke style, but only differ in the themes of their artworks. Therefore, the predefined notion of style (i.e., engineered features) limits the artists the tool can recognise. Representation learning circumvents this limitation, because it does not rely on a predefined notion of style. Instead, it learns the style from the raw image data and associated labels. For instance, an image of the painting "The Starry Night" is accompanied by the label "Vincent van Gogh". In Chapters 2 and 3 two studies are presented which show that representation learning can be used to achieve state of the art artist attribution performance.

Style classification. Besides recognising the artist (and their style) a number of works have aimed to recognise the art school or movement (e.g., expressionism, renaissance, popart) to which an artwork belongs [102, 66, 67, 97]. Just like artistry, these are aspects of artworks which are visually recognisable across multiple artworks.

Art movement classification is typically restricted to several dozen art movements, which might differ considerably in appearance. Although a number of studies [66, 97] used learnt representations for this task, these representations were obtained by training on a large dataset of natural images, therefore it is unclear how well these representations capture artwork style. The focus of this thesis is on the style of the artist.

Neural style transfer. The seminal work by Gatys et al. [30] demonstrated a

method for disentangling the representations of the content and the style of an image. Furthermore, the disentangled parts can be recombined with the content and style part of arbitrary images, effectively transferring the style from one image onto another. An example of this so-called neural style transfer is shown in Figure 1.2.



(a) content **(b)** style **(c)** content+style

Figure 1.2: Example of neural style transfer. (a) The picture of the Dante building on the Tilburg university campus forms the content. (b) The artwork '*Landscape at Twilight*' by Vincent van Gogh provides the style. (c) The combination of the content and the style yields the Dante building as a Van Gogh painting.

A key aspect of neural style transfer is that "style" is defined on the basis of a single image, i.e., images (b) and (c) in Figure 1.2 are in the style of '*Landscape at Twilight*', rather than in the style of Vincent van Gogh. For this thesis we define style as a property which goes beyond a single artwork, and one that is present in all of the works by the artist. Therefore, our definition and the definition used for style transfer [30] do not align. Nonetheless, what makes neural style transfer so exciting is that it makes visual what style is, by generating images with the chosen style. To explore this further in a manner matching our definition of style, we present a study in Chapter 4 where we aim to colour greyscale paintings in a manner consistent with the artist's style.

Inpainting. Repairing damaged parts of artworks through retouching or inpainting is common practice for conservators and restorers. Yet, due to the potential of changing the interpretation and appearance of the artwork, restorers often have to show restraint. As a consequence, smaller progressive changes (e.g., discolourations and cracking) might go untreated. Specifically for cracks, a reason to not remove them is because they are *accidental features* of paintings [13], they provide a record of the deterioration of paintings [19].

Nonetheless to make it possible to view the appearance of a painting without cracks, a number of works have developed algorithms which perform digital inpainting of cracks [32, 107, 18]. Additionally, a number of works on inpainting have shown it is possible to inpaint regions with a spatial extent that is much larger than that of cracks [91, 49]. Inpainting larger regions requires understanding the effective modelling of the painting style of the artist and the extrapolation of the context surrounding the region. Representation learning seems to be well-suited to meet this requirement. In Chapter 5 we investigate inpainting of large regions in paintings and natural images using learnt representations.

1.3 Problem statement

Despite the rich computer vision history of engineering task-specific features, recent empirical results have shown that learning representations is a much more promising avenue. Moreover, due to the inherent difficulty in defining the artist's style we opt for a data-driven approach, where the data dictates what characterises the artist's style. Specifically, we focus on representing the artist's style in a manner that enables the digital analysis of artworks for a wide array of art investigation tasks. To this end we formulate the following problem statement (PS).

PS: *To what extent can the artist's style be represented in a digital manner?*

To address the problem statement we identity two requirements for a useful representation of style: (1) the artist's style can be recognised across multiple artworks, and (2) the representation can be used to generate novel content that has the stylistic characteristics of the artist. To guide our attempts at answering the problem statement we rephrase these requirements as the following two research questions.

- **Research question 1 (RQ1):** Is it possible to learn a representation of the artist's style, which can be used to recognise the style of the artist across multiple artworks?

- **Research question 2 (RQ2):** Can we generate novel image content in the style of the artist?

The two main contributions of this thesis are: (1) the development and evaluation of two techniques enabling the recognition of the artist's style across artworks, and (2) the development and evaluation of two new image generation techniques, tailored to the art domain.

1.4 Representation learning

In this section we provide a brief introduction on representation learning, specifically convolutional neural networks which are central in this thesis. We hope to facilitate the reading of the remainder of this thesis by introducing terminology and principles used throughout the thesis.

Training a classifier or predictor directly on raw data is undesirable as certain transformations of the image (e.g., rotational, scale or luminance changes), that do not alter the interpretation of the data, might greatly change the representation of the raw data. For instance, for textual data such changes could be word choices, replacing a word with a synonym should not alter the interpretation, despite the different symbol that is used. Similarly, for speech data, shifting the pitch (uniformly) should not change the meaning of a sentence spoken (in most languages), yet the numerical representation of the raw signal might change greatly. For images, such changes could be rotations or positional changes, which greatly change the pixels, but do not affect the interpretation.

A method for overcoming this limitation of raw data is representation learning; which aims to learn (lower dimensional) representations of the data which capture the useful information [6]. Specifically, in the last decade most representation learning literature has used neural networks to learn useful representations.

For image analysis tasks the most commonly used neural networks are convolutional neural networks (CNNs) [74]. Each neuron in a CNN incorporates a (small) adaptive

filter which is convolved with the input. Using convolution is beneficial for image analysis, because it allows the network to recognise patterns independently of their spatial position. Figure 1.3 illustrates the convolution of a 3×3 filter with a 3×3 region of the input image yielding a single output value. An incomplete but intuitive understanding of convolution with a filter can be obtained by likening the filter coefficient to a template that is compared with the input values. The output value represents the degree to which the input contains the pattern represented by the template. In the neural metaphor, the filter coefficients are the weights of the neurons and the output value is called the "activation" of the neuron.

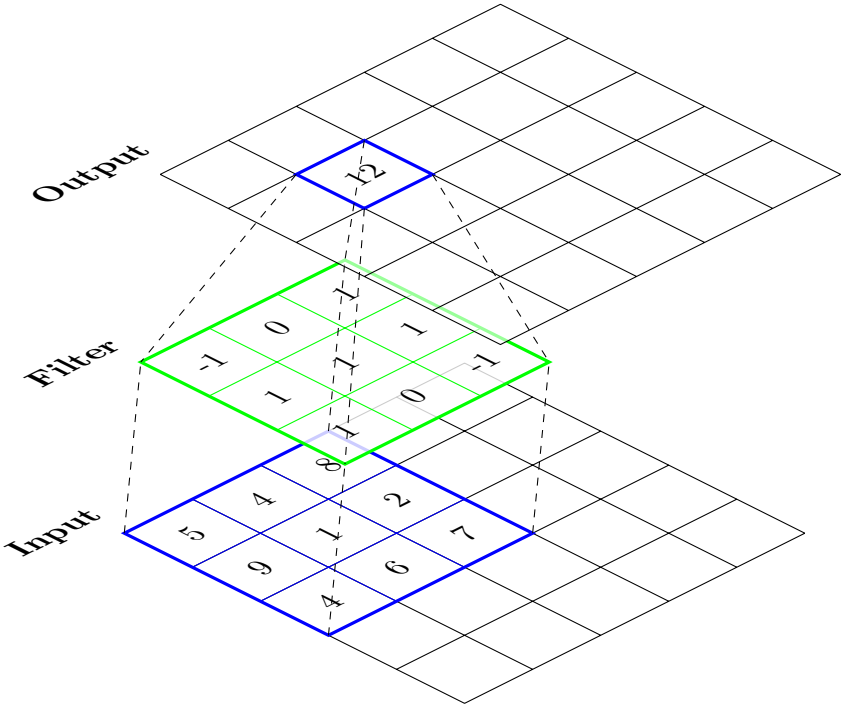


Figure 1.3: Illustration of the convolution of a filter at a single location of an input (i.e., image). The filter coefficients are multiplied element-wise with the input values and summed, yielding the output value 12.

A deep CNN consists of multiple convolutional layers, stacked on top of each other, where the first layer is applied directly to the input, and subsequent layers are applied to the preceding layer. The last layer is the output layer that contains the information relevant to the task it was trained for (e.g., a class label for classification, a scalar

value for regression, or an output image for image generation). Typically, a non-linear transfer function is applied to the neurons in each layer, examples of such transfer functions are: hyperbolic tangent function (tanh), sigmoid function, and rectified linear units (ReLU). By stacking layers of non-linear transformations of the input, neural networks are able to learn increasingly abstract representations of the input.

Moreover, by stacking layers of convolutions (and potentially pooling layers), a CNN is able to learn transformations of larger regions in the input space (the input image). The input to a CNN is typically an image with a spatial resolution greater than spatial dimensions filters of the first layer, which means that this first layer can only transform small regions in the input. The *receptive field* of a neuron or filter is the input region that can affect its output. A CNN applies filters recursively by treating the outputs of the first layer of filters as inputs for the second layer of filters, and so forth. As a consequence, the spatial extent of the neurons' receptive fields grows with each subsequent layer. Figure 1.4 illustrates this for the recursive application of a 3×3 filter in the first two layers of a CNN.

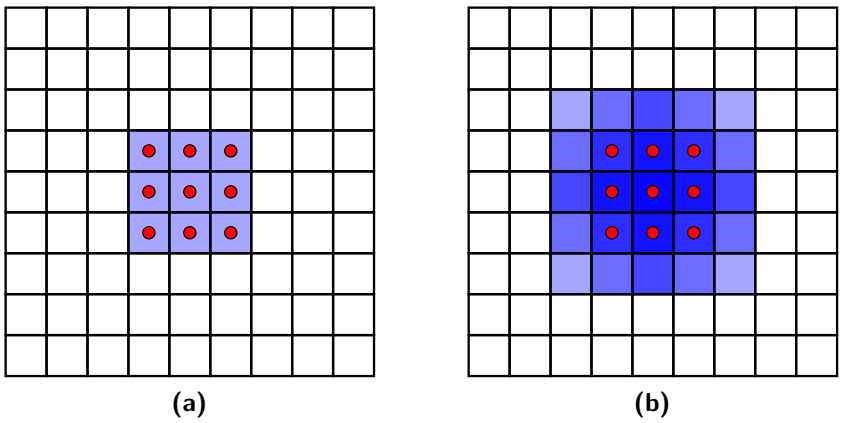


Figure 1.4: Visualisation of the receptive fields of the first two layers of a convolutional neural network. (a) shows the receptive field of a 3×3 filter applied directly to the input. (b) shows the 5×5 receptive field of the same 3×3 filter applied to the output produced by (a).

In theory, we could expand the receptive field in this manner until it covers the entire input, in practice it is more common to introduce downsampling, to obtain a large receptive field with fewer layers. Downsampling in CNN is often achieved through

pooling layers. Pooling layers are inserted between convolutional layers and reduce the spatial dimension of the layer output by applying a pooling operation to a single depth slice region in the output. Common pooling operations are max and average pooling.

Figure 1.5 illustrates max and average pooling. A 4×4 image is shown on the left part of the figure. In max pooling, the maximum value of each 2×2 quarter of the image defines the corresponding output of the pooling layer (top right). In avg pooling, the average value of each 2×2 quarter defines the output.

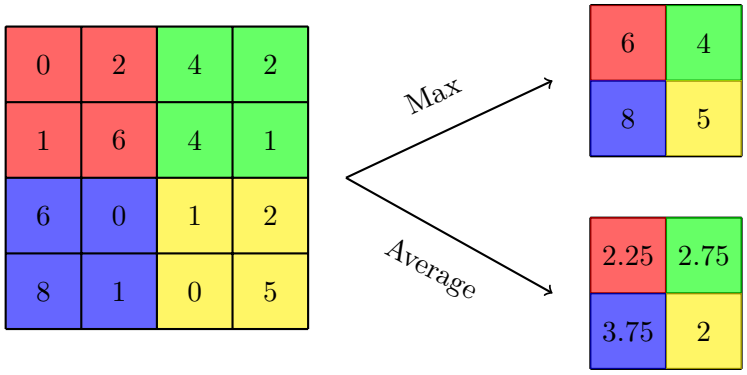


Figure 1.5: Visualisation of max and average Pooling with non-overlapping 2×2 filters. On the left the input region, on the right the output. Above is max pooling, below is average pooling.

By increasing the size of the receptive field, the representations in the successive layers of a CNN become more abstract and describe a larger spatial region. A common example to describe the workings of a CNN is to imagine the analysis of a face, where the initial layers look at low-level features such as edges and contours, and later layers look at higher-level features such as parts of the face (e.g., eyes, nose, mouth).

The models used in the studies reported on in this thesis are based on variants of the CNN as outlined above.

1.5 Structure of the thesis

In Chapter 1 we introduce the reader to the topic of the thesis, and we formulate the problem statement and two research questions. In Chapters 2 to 5 of this thesis we present four studies which investigate representation learning of the artist's style on a

variety of tasks. The four chapters present studies which have been accepted for publication in peer-reviewed journals, or have been submitted for publication. Specifically, Chapter 2 has been published in IEEE Signal Processing Magazine, and Chapter 3 has been published in Pattern Recognition. As a consequence the chapters have been written as self-contained studies, and may contain a certain amount of overlap.

Based on the research questions we can identity two parts to this thesis. In the first part (Chapters 2 and 3) we aim to answer RQ1: "Is it possible to learn a representation of the artist's style, which can be used to recognise the style of the artist across multiple artworks?". In the second part (Chapters 4 and 5) we focus on RQ2: "Can we generate novel image content in the style of the artist?". What follows is a brief description of each of the remaining chapters of the thesis.

Chapter 2 presents a study on artist attribution, exploring the influence of different dataset parameters. In Chapter 3 we extend the artist attribution framework presented in Chapter 2 to deal with scale variations and scale-specific information. Chapter 4 presents a study on image colourisation in a manner which is consistent with the artist's style. In Chapter 5 we present a study on semantic inpainting, reconstructing large missing parts from natural images and paintings. Finally, Chapter 6 concludes the thesis, and provides answers to the research questions and problem statement, as well as directions for future work.

2

Toward discovery of the artist's style

This chapter has been previously published as: N. van Noord, E. Hendriks, & E. Postma (2015). Toward discovery of the artist's style: Learning to recognize artists by their artworks. *IEEE Signal Processing Magazine* vol. 32, no. 4, pp. 46–54, July 2015.

Abstract

Author attribution through the recognition of visual characteristics is a commonly used approach by art experts. By studying a vast number of artworks, art experts acquire the ability to recognise the unique characteristics of artists. In this chapter we present an approach that uses the same principles in order to discover the characteristic features that determine an artist's touch. By training a Convolutional Neural Network (PigeoNET) on a large collection of digitised artworks to perform the task of automatic artist attribution, the network is encouraged to discover artist-specific visual features. The trained network is shown to be capable of attributing previously unseen artworks to the actual artists with an accuracy of more than 70%. In addition, the trained network provides fine-grained information about the artist specific characteristics of spatial regions within the artworks. We demonstrate this ability by means of a single artwork that combines characteristics of two closely collaborating artists. PigeoNET generates a visualisation that indicates for each location on the artwork who is the most likely artist to have contributed to the visual characteristics at that location. We conclude that PigeoNET represents a fruitful approach for the future of computer supported examination of artworks.

2.1 Introduction

Identifying the artist of an artwork is a crucial step in establishing its value from a cultural, historical, and economic perspective. Typically, the attribution is performed by an experienced art expert with a longstanding reputation and an extensive knowledge of the features characteristic of the alleged artist and contemporaries.

Art experts acquire their knowledge by studying a vast number of artworks accompanied by descriptions of the relevant characteristics (features) [95]. For instance, the characteristic features of Vincent van Gogh during his later French period include the outlines painted around objects, complementary colours [8], and rhythmic brush strokes [79]. As Van Dantzig [115] claimed in the context of his *Pictology* approach, describing works by an artist in terms of visual features enables the attribution of works to artists (see also [82]).

The advent of computers and high-resolution digital reproductions of artworks gave rise to attempts to partially automate the attribution of artworks [63, 47, 112]. Given the appropriate visual features, machine learning algorithms may automatically attribute artworks to their artists. As was (and still is) common practice in traditional machine learning, feature engineering, i.e., finding or defining the appropriate features, is critical to the success of the task of automatically attributing paintings to the correct painter. Close cooperation with art historians and conservators has facilitated the feature engineering for artist attribution, which led to promising results in the automatic attribution of artworks by van Gogh and his contemporaries [79, 63, 118, 1], highlighting the value of automatic approaches as a tool for art experts.

Despite the success of feature engineering, these early attempts were hampered by the difficulty to acquire explicit knowledge about all the features associated with the artists of artworks. Understandably, the explicit identification of characteristic features posed a challenge to art experts, because (as is true for most experts) their expertise is based on tacit knowledge, which is difficult to verbalise [26]. By adopting a

method capable of automatically recognising the characteristics that are known to be important for the task at hand, the tacit knowledge of art experts may be made explicit [7].

Feature learning is an alternative to feature engineering that learns features directly from the data [7]. Feature learning is much more data intensive than feature engineering, because it requires a large number of examples to discover the characteristic features. In recent years, feature learning has shown great promise by taking advantage of deep architectures, machine learning methods inspired by biological neural networks. A typical example of a deep architecture is a convolutional neural network, which, when combined with a powerful learning algorithm, is capable of discovering (visual) features. Convolutional neural networks outperform all existing learning algorithms on a variety of very challenging image classification tasks [71]. To our knowledge, convolutional neural networks have not yet been applied for automated artist attribution. The objective of this chapter is to present a novel and transparent way of performing automatic artist attribution of artworks by means of convolutional neural networks.

The question may be raised whether automatic artist attribution is possible at all, when using visual information only. It has been frequently argued by scholars working in the art domain that semantic or historical knowledge, as well as technical and analytical information are pivotal in the attribution of artworks. The feasibility of image-based automatic artist attribution is supported by biological studies. Pigeons [122] and honeybees [125] can be successfully trained to discriminate between artists, with pigeons correctly attributing an art work in 90% of the cases in a binary Monet-Picasso attribution task. This shows that a visual system without higher cognitive functions is capable of learning the visual characteristics present in artworks. While it is unlikely that a perfect result can be achieved without incorporating additional information, these findings do pave the way for an attribution approach that learns to recognise visual features from data rather than from prior knowledge.

In this chapter, we present PigeoNET, a convolutional neural network corresponding to the AlexNET architecture described in [71] to which we added a visualisation

component due to [131]. PigeoNET is applied to an artist attribution task by training it on artworks. As such, PigeoNET performs a task similar to the pigeons in [122], by performing artist attribution based solely on visual characteristics. This implies that, in addition to authorship, PigeoNET may also take visual characteristics into consideration that relate indirectly to the artist (e.g., the choice of materials or tools used by the artist) or that are completely unrelated to the artist (e.g., reproduction characteristics such as lighting and digitization procedure). To ensure that the visual characteristics on which the task is solved by PigeoNET make sense, human experts are needed to assess the relevance of the acquired mapping from images of artworks to artists. Our visualisation method allows for the visual assessment by experts of the characteristic regions of artworks.

In our artist attribution experiments, we consider three sources of variation in the training set and assess their effects on attribution performance: (1) heterogeneity versus homogeneity of classes (types of artworks, e.g., paintings, prints, or drawings), (2) number of artists, and (3) number of artworks per artist.

After training, the performance of PigeoNET will be assessed in two ways: (1) by determining how well it attributes previously unseen artworks, and (2) by generating visualisations that reveal artwork regions characteristic of the artist, or in case of artworks that are likely created by two or more artists, generating visualisation that reveal which regions belong to which artist, and could aid in answering outstanding art historical questions.

The remainder of the chapter is organised as follows. In Section 2.2 we describe the PigeoNET model. In Section 2.3 the experimental setup is outlined and the results of the artist attribution task are presented. In Section 2.4 we explore the features acquired by PigeoNET by visualising authorship for specific artworks. We discuss the implications of feature learning for the interdisciplinary domain of automatic artist attribution in Section 2.5. Finally, Section 2.6 concludes by stating that PigeoNET represents a fruitful approach for the future of computer-supported examination of artworks, capable of attributing artists to unseen artworks and generating visualisations

of the authorship per region of an artwork.

2.2 PigeoNET

A convolutional neural network can learn to recognise the visually characteristic features of an artist by adapting filters to respond to the presence of these features in an image [74]. The filters are adapted to respond to a feature by adjusting the parameters, or weights, of the filters until a suitable configuration is found. The proper weights for this configuration are obtained by means of a learning algorithm called back-propagation [76], which requires no prior knowledge, other than the input images and some label (e.g., the artist who created it). In the case of artist attribution, the network will learn to recognise features that are regarded as characteristic of a certain artist, allowing us to discover these characteristics. PigeoNET is a convolutional neural network designed to learn the characteristics of artists and their artworks, so as to recognise and identify their authorship.

The filters in a convolutional neural network are grouped into layers, where the first layer is directly applied to images, and subsequent layers to the responses generated by previous layers. By stacking layers to create a multilayer architecture the filters can respond to increasingly complex features with each subsequent layer. The filters in the initial layers respond to low level visual patterns, akin to Gabor filters [54], whereas the final layers of filters respond to visual characteristic features specific to artists.

Because convolution is used to apply the filters to an image, or the response of a previous layer, the layers of filters are referred to as convolutional layers. The advantage of a convolutional layer, over a traditional neural network layer, is that the weights are shared, allowing the adaptive filters to respond to characteristic features irrespective of their position or location in the input [76]. In order to learn a mapping from the filter responses to a certain artist the convolutional layers are, typically, followed by a number of fully-connected layers which translate the presence and intensity of the filter responses to a single certainty score per artist. The certainty score for an artist is high whenever the responses for filters corresponding to that artist are strong, con-

versely, the certainty score is low when the filter responses are weak or nonexistent. Thus, an unseen artwork can be attributed to an artist for whom the certainty score is the highest.

2.2.1 Visualisation of artist-characteristic regions

While PigeoNET's attribution of an artwork is based on the entire artwork, regions containing visual elements characteristic for an artist will be assigned more weight than others, to achieve a correct attribution [89]. In order to increase our understanding of the attribution performed by PigeoNET, we aim to visualise such artist-characteristic regions. Several methods have been proposed for visualising trained convolutional neural networks [131, 104] and other layered algorithms [27]. We adopt the occlusion sensitivity testing method proposed by [131] for obtaining visualisations of artist-characteristic regions, which can be considered a weakly supervised localisation method. By systematically occluding a small image region of an artwork, the importance of the occluded region is determined by observing the change in the certainty score for the correct artist. When an occluded region is important (or highly characteristic) for correctly identifying the artist, there will be a significant drop in the certainty score generated by PigeoNET. Inversely, occluding a region that is atypical for the correct artist will result in an increase in the certainty score. A region for which occlusion results in a drop of the certainty score is considered characteristic for the artist under consideration. This approach to creating visualisations allows us to show the approximate areas of an artwork which are representative of an artist.

As an illustration, Figure 2.1 depicts *The feast of Achelous* by Peter Paul Rubens and Jan Brueghel the Elder. It is an artwork created by two artists; Rubens painted the persons and Brueghel the scenery [123]. Although there is no single correct artist, the certainty score for Brueghel would decrease if the scenery were to be occluded, whereas the certainty score for Rubens would drop if the figures were occluded. Even when only part of the figures or part of the scenery were to be occluded, we would see a drop in confidence scores. In a similar vein, when even smaller regions of the paint-

Figure 2.1: "Peter Paul Rubens and Jan Brueghel the Elder: The Feast of Achelous" (45.141) In Heilbrunn Timeline of Art History. New York: The Metropolitan Museum of Art, 2000–. http://www.metmuseum.org/toah/works-of-art/45.141. (October 2006)

ing have been occluded, it becomes possible to identify important regions on a much more detailed scale.

2.3 Author attribution experiment

The goal of an artist attribution task is to attribute an unseen artwork to the artist who created it. To be able to perform this task adequately, PigeoNET needs to discover features that distinguish an artist from other artists, but especially to discover features that are characteristic of each artist. In the rest of this section we will discuss the dataset (2.3.1), network architecture (2.3.2), training procedure (2.3.3), evaluation procedure (2.3.3), and the quantitative (2.3.4) and qualitative results (2.3.5).

2.3.1 Dataset

The characteristic features of an artist can be discovered by studying artworks which are representative of that artist. Yet, obtaining a sufficiently large sample of such images is problematic, given the lack of (automatic) methods and criteria to determine whether an artwork is representative. A commonly taken approach to circumvent the need for a representative sample is to take a very large sample. As such, a dataset that contains a large number of images, and a large number of images per artist, is required.

The Rijksmuseum Challenge dataset [85] consists of 112,039 digital photographic reproductions of artworks by 6,629 artists exhibited in the Rijksmuseum in Amsterdam, the Netherlands. All artworks were digitised under controlled settings. Within the set there are 1,824 different types of artworks (e.g., drawings, paintings, and vases) and 406 annotated materials, such as paper, canvas, porcelain, iron, and wood. To our knowledge, this is the largest available image dataset of artworks, and the only dataset that meets our requirements.

We divided the Rijksmuseum Challenge dataset into a training, validation, and test set (cf. [85]). In this chapter these sets are used to train PigeoNET, to optimise the hyper-parameters, and to evaluate the performance of PigeoNET on unseen examples, respectively. The dataset contains a number of artworks which lack a clear attribution, these are labelled as either 'Anonymous' or 'Unknown', 16,686 and 685 respectively. We chose to exclude these artworks, because our objective is to relate visual features to specific artists.

Whilst the Rijksmuseum Challenge dataset contains a large number of images of many different types of artworks by a large number of artists, there are many artists for whom only a few artworks are available or artists who have created many different types of artworks. As stated in the Introduction, these variations might influence the performance of PigeoNET in non-obvious ways. To this end we consider the following three sources of variation: (1) heterogeneity versus homogeneity of classes (types of

artworks), (2) number of artists, and (3) number of artworks per artist.

Two main types of subsets were defined to assess the effect of heterogeneity versus homogeneity of artworks: type A (for "All") and type P (for "Prints"), respectively. As is evident from Table 2.2 on page 28, prints form the majority of artworks in the Rijksmuseum Challenge dataset. The homogeneous type of subsets (P) has three forms: P1, P2 and P3. Subsets of type P1 have varying numbers of artists and artworks per artist (as is the case for A). Subsets of type P2 have a fixed number of artworks per artist. Finally, subsets of type P3 have a fixed number of artists. We remark that the number of examples per artist for the subsets in A and P1 are minimum values. For very productive artists these subsets may include more artworks. For subsets of types P2 and P3, the number of examples is exact and constitutes a random sample of the available works per artist. A detailed overview of the resulting 15 subsets is listed in Table 2.1*. For the heterogeneous subset of at least 256 artworks of type A, Table 2.2 on page 28 provides a more detailed listing which specifies the three most prominent categories: Prints, Drawings, and Other. The Other category includes a variety of different artwork types, including 35 paintings.

All images were down-sampled to 256×256 pixels following the procedure described in [71], to adhere to the fixed input size requirement of the network architecture, and are normalised at runtime by subtracting the mean image as calculated on the training set.

2.3.2 Network Architecture

The architecture of PigeoNET is based on the Caffe [57] implementation† of the network described in [71], and consists of 5 convolutional layers and 3 fully connected layers. The number of output nodes of the last fully-connected layer is equal to the number of artists in the dataset, ranging from 26 to 958 artists.

*The largest subsets for P2 and P3 are identical, but are reported twice in Table 2.1 for clarity.
†Available at: http://caffe.berkeleyvision.org/model_zoo.html.

Table 2.1: Overview of subsets and the number of training, validation, and test images per subset. The subsets are labelled by their types. Type A ("All") are subsets containing varying artworks, examples and, examples per artist. Type P ("Prints") refers to subsets of prints only. P1: varying numbers of artworks, examples and, examples per artist. P2: number of examples constant (128). P3: number of artists constant (78). For A and P1, the numbers of examples per artists represent the minimum numbers, while for P2 and P3, these numbers represent the exact number of artworks per artist.

Subsets	# Examples per artist	# Artists (classes)	# Training images	# Validation images	# Test images
A	10	958	56,024	7,915	15,860
	64	197	37,549	5,323	10,699
	128	97	28,336	4,063	8,058
	256	34	17,029	2,489	4,838
P1	10	673	44,539	6,259	12,613
	64	165	31,655	4,484	8,983
	128	78	23,750	3,408	6,761
	256	29	14,734	2,171	4,200
P2	128	26	3,328	1,209	2,277
	128	39	4,992	1,521	2,970
	128	52	6,656	2,160	4,341
	128	78	9,984	3,408	6,761
P3	10	78	780	3,408	6,761
	64	78	4,992	3,408	6,761
	128	78	9,984	3,408	6,761

2.3.3 Training procedure

An effective training procedure was used (cf. [71]), in that the learning rate, momentum, and weight decay hyperparameters were assigned the values of 10^{-2}, 0.9, and $5 \cdot 10^{-4}$, respectively. The learning rate was decreased by a factor 10 whenever the error on the validation set stopped decreasing. The data augmentation procedure consisted of random crops and horizontal reflections. While orientation is an important feature to detect authorship, the horizontal reflections were used to create a larger sample size, as it effectively doubles the amount of available training data. It thus provides PigeoNET with sufficient data to learn from, although this may negatively impact PigeoNET's ability to pick up on orientation clues to perform classification. In contrast to [71], only a single crop per image was used during training, with crops of size 227×227 pixels, and the batch size was set to 256 images per batch.

Table 2.2: List of the 34 artists with at least 256 artworks and the distribution of artworks over main types (Prints, Drawings, and Other).

#	Name	Prints	Drawings	Other
1	Heinrich Aldegrever	347	27	
2	Ernst Willem Jan Bagelaar	400	27	
3	Boëtius Adamsz. Bolswert	592		
4	Schelte Adamsz. Bolswert	398		
5	Anthonie Van Den Bos	531	3	
6	Nicolaes De Bruyn	515	2	
7	Jacques Callot	1,008	4	1
8	Adriaen Collaert	648	1	
9	Albrecht Dürer	480	9	2
10	Simon Fokke	1,177	90	
11	Jacob Folkema	437	4	3
12	Simon Frisius	396		
13	Cornelis Galle (i)	421		
14	Philips Galle	838		
15	Jacob De Gheyn II	808	75	10
16	Hendrick Goltzius	763	43	4
17	Frans Hogenberg	636		4
18	Romeyn De Hooghe	1,109	5	5
19	Jacob Houbraken	1,105	42	1
20	Pieter De Jode II	409	1	
21	Jean Lepautre	559		1
22	Caspar Luyken	359	18	
23	Jan Luyken	1,895	33	
24	Jacob Ernst Marcus	372	23	2
25	Jacob Matham	546	4	
26	Meissener Porzellan Manufaktur			1,003
27	Pieter Nolpe	344	2	
28	Crispijn Van De Passe I	841	15	
29	Jan Caspar Philips	401	17	
30	Bernard Picart	1,369	132	3
31	Marcantonio Raimondi	448	2	
32	Rembrandt Harmensz. Van Rijn	1,236	119	29
33	Johann Sadeler I	578	1	
34	Reinier Vinkeles	573	50	

All training was performed using the Caffe framework [57] on a NVIDIA Tesla K20m card and took between several hours and several days, depending on the size of the subset.

Evaluation procedure

The objective of the artist attribution task is to identify the correct artist for each unseen artwork in the test set. To this end the performance is measured using the mean class accuracy (MCA), which is the average of the accuracies for all artists. This makes sure that the overall performance is not heavily biased by the performance on a single artist.

During testing the final prediction is averaged over the output of the final softmax layer of the network for 10 crops per image. These crops are the four corner patches and the central patch plus their horizontal reflections.

2.3.4 Results

The results of the artist attribution task are listed in Table 2.3 on page 30. The results on the artist attribution task show that the three sources of variation, (heterogeneity versus homogeneity of classes (types of artworks), number of artists, and number of artworks per artist.) affect the performance in different ways. The effect of heterogeneity versus homogeneity can be assessed by comparing the results for A and P1. The results obtained with P1 are slightly better than those obtained with A (except for 128 examples per artist). However, A and P1 differ also in number of artists, which as shown by the results on P2 and P3 affects the performance.

The total number of artists (P2) and the number of examples per artist (P3) have a more prominent effect on the attribution performance of PigeoNET. Increasing the number of artists while keeping the number of examples per artist constant (as done for P2) leads to a decrease in performance. With more examples per artist (P3) the performance increases tremendously, indicating that PigeoNET is unable to generalise when presented with a small number of examples.

Our results suggest that the effects of the number of artists and the number of examples per artist are closely related. This agrees with the findings reported in [71] and leads to the observation that by considering more examples per artist the number of

Table 2.3: Mean Class Accuracies (MCA) for the artist attribution task on the 15 data subsets. Bold values indicate the best result per type, the overall best result is underlined.

Subsets	# Examples per artist	# Artists (classes)	MCA
A	10	958	52.5
	64	197	68.2
	128	97	74.5
	256	34	**78.3**
P1	10	673	60.0
	64	165	70.2
	128	78	73.3
	256	29	**78.8**
P2	128	26	**63.9**
	128	39	55.6
	128	52	52.7
	128	78	52.0
P3	10	78	13.1
	64	78	38.0
	128	78	**52.0**

artists to be modeled can be increased.

The subsets of type A are comparable to the subsets used by Mensink et al. [85], who obtain a comparable MCA of 76.3 on a dataset containing 100 artists using SIFT features, Fisher vectors, and 1-vs-Rest classification.

Figure 2.2 shows a visualisation of the confusion matrix for the subset with at least 256 examples of all artwork types. The rows and columns correspond to the artists in Table 2.2. The rows represent the artist estimates by PigeoNET, the columns the actual artists. The diagonal entries represent correct attributions which are colour coded.

Upon further analysis of the results for the 256 example subset (A) of all artwork types we observe that the best artist-specific classification accuracy (97.5%) is obtained for Meissener Porzellan Manufaktur, a German porcelain manufacturer (class 26). Among the different types of artworks in the dataset, these porcelain artworks are visually the most distinctive as determined by our model. Given that the visual characteristics of porcelain differ considerably from all other artworks in the dataset, it is

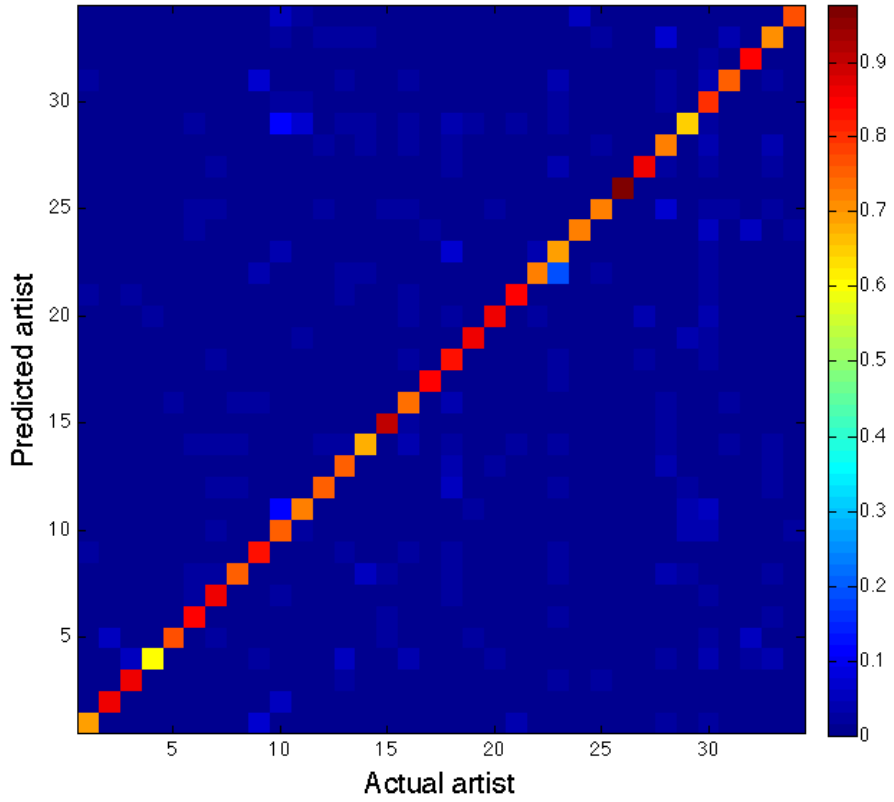


Figure 2.2: Confusion matrix for all artists with at least 256 training examples of all artwork types. The rows represent the artist estimates and the columns the actual artists. Row and column numbers (from left to right and from bottom to top) correspond to those as listed in Table 2.2.

not surprising that the highest classification accuracy is achieved for this class.

The worst artist-specific classification accuracy (60.6%) is achieved for Schelte Bolswert (class 4), as indicated by the yellow square on the diagonal in the confusion matrix (fourth row from below, fourth column from left). The low accuracy may be partially explained by the confusion between Schelte Bolswert and his older brother and instructor Boëtius Bolswert (class 3). Yet, because the classification accuracy for Boëtius Bolswert (86.3) seems much less affected by the confusion, an alternative possibility is that PigeoNET is more inclined to assign visual characteristics that are present in their works to Boëtius Bolswert because his works appear more frequently in the dataset.

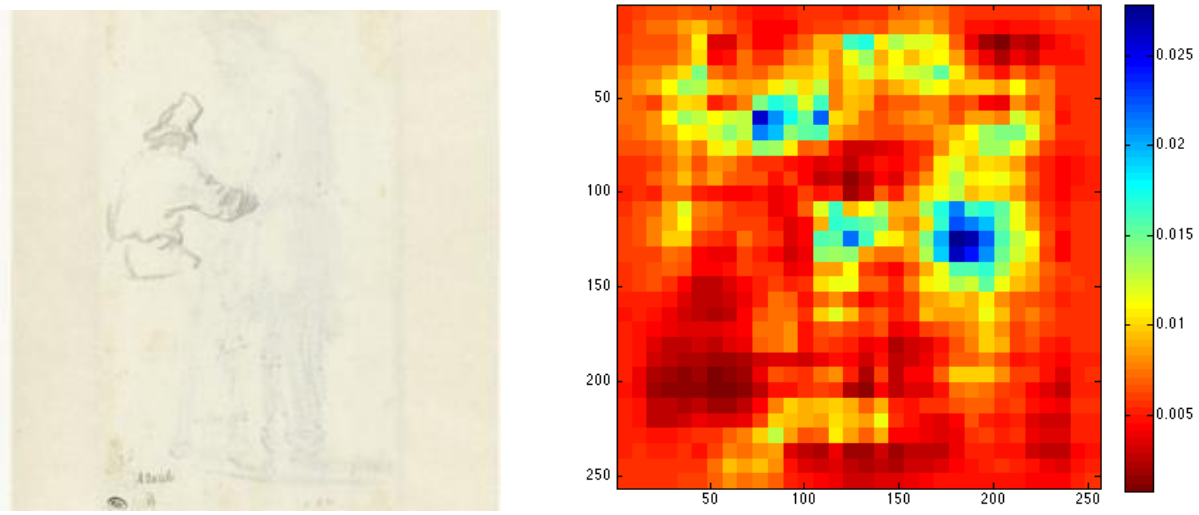
In a similar vein, the misclassifications that occur between Fokke Simon (10) and

Jan Caspar Philips (29), and between Jan Luyken (23) and Caspar Luyken (22) are notable. Fokke Simon was a student of Jan Caspar Philips, and Jan and Caspar Luyken were father and son. Both pairs of artists have worked together on several artworks in the Rijksmuseum Challenge dataset, despite the label in the dataset indicating that these artworks belong to only one of these artists. We became aware of these potential dual-authorship cases after having performed our main experiment. Dual-authorship cases will be examined in more detail in Section 2.4.

2.3.5 Visualisation and assessment

Visualisations of the importance of each region in an artwork can be generated using the regions of importance detection method described in Section 2.2.1, where the occlusions are performed with a grey block of 8×8 pixels, to indicate approximate regions which are characteristic of the artist. The regions of importance can be visualised using heatmap colour coding, as shown in Figure 2.3b. The value of a region in the heatmap corresponds to the certainty score of PigeoNET for the artwork with that region occluded. In other words, a region with a lower value is of greater importance in correctly attributing the artwork, with (dark) red regions being highly characteristic of the artist, and (dark) blue regions being the least characteristic.

When comparing the artwork and heatmap in Figure 2.3 of the drawing by Rembrandt, it is very noticeable that PigeoNET assigns much weight to seemingly empty areas. The texture of the material on which an artwork is created can be indicative of the artist who created the artwork [61]. When taking a closer look at Figure 2.4, with enhanced contrast, it becomes apparent that the areas are not empty and that a distinctive visual texture is present. The visual pattern is sufficiently distinctive and artist-specific for PigeoNET to assign it a larger weight. The pattern is an example of a visual characteristic which is indirectly related to the artist. It illustrates the importance of the transparency of automatic attribution to allow human experts to interpret and evaluate the visual characteristic.



(a) Art work **(b)** Heatmap

Figure 2.3: Image (a) and heatmap (b) of *'Study of a man, seen from behind'* by Rembrandt Harmensz. van Rijn (1629-1630). Lower (red) values in the heatmap correspond to greater importance in correctly identifying Rembrandt.

2.4 Deciding between two artists

In the previous section we used PigeoNET to attribute an artwork to a single artist. Yet, as illustrated by the work of Peter Paul Rubens and Jan Brueghel in Figure 2.1, in many cases two (or more) artists have worked on the same artwork (see also [11]).

As evident from our results, PigeoNET had difficulty in correctly attributing artworks of closely collaborating artists. An intriguing explanation for PigeoNET's failure to assign the 'correct one' of two potential artists to artworks is that the artworks are created by both artists. In that case, it would not be a failure at all and indicates that PigeoNET discovered that the two artists are similar, and it recognises the characteristic features of both artists, even if the work is attributed to only one artist. In the remainder of this section we demonstrate the possibility of using PigeoNET to perform a fine-grained analysis of an artwork, attributing individual image regions to an artist.

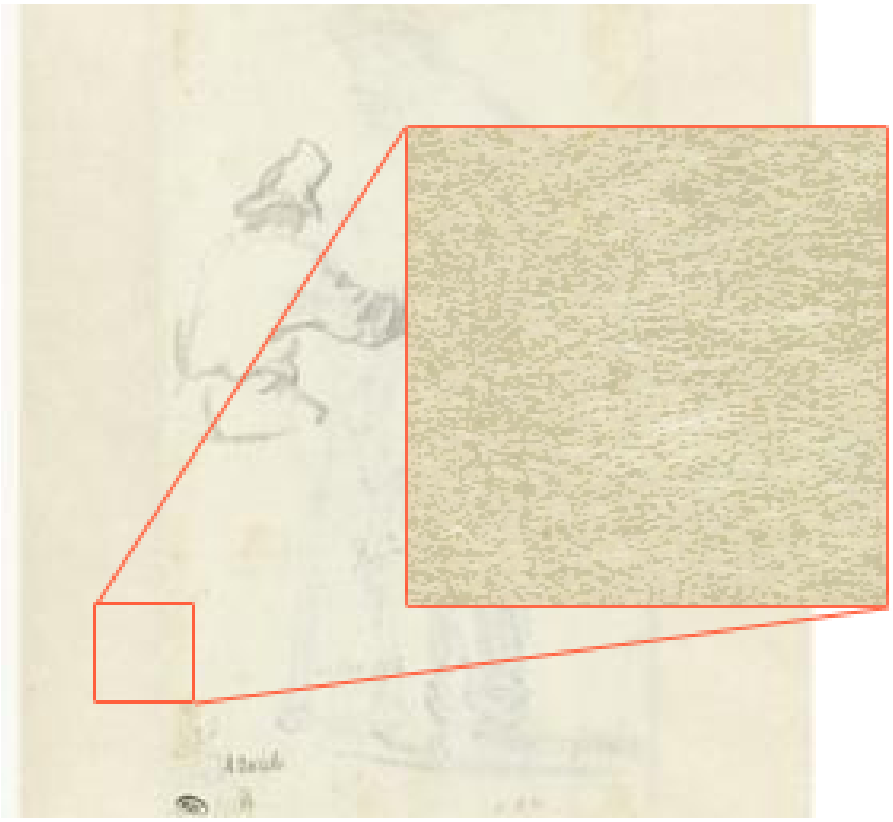


Figure 2.4: Contrast enhanced detail view of a highly textured region of the artwork shown in Figure 2.3a.

2.4.1 Discovering dual-authorship

PigeoNET had difficulty in distinguishing between the works by Jan and Caspar Luyken, father and son who worked together and created many prints. Throughout their careers Jan Luyken chose to depict pious and biblical subjects, whereas Caspar Luyken mostly depicted worldly scenes [3]. As an example, we consider the artwork shown in Figure 2.5, *Transfer of the Spanish Netherlands by Philip II to Isabella Clara Eugenia, Infanta of Spain, 1597*. The work depicts the transfer of the Spanish Netherlands by Filips II to Isabella Clara Eugenia. Although, arguably it is a very worldly scene, it is nevertheless attributed to Jan Luyken. Could it be possible that the artwork is incorrectly attributed to Jan Luyken? Obviously, this is a question that has to be answered by experts of their works.

Figure 2.5: Image of the *Transfer of the Spanish Netherlands by Philip II to Isabella Clara Eugenia, Infanta of Spain, 1597* by Jan Luyken, 1697 - 1699.

Our findings may support them in their assessment. Although, PigeoNET correctly attributed the artwork to Jan Luyken, the reported certainty score for Caspar Luyken is very high. Apparently, PigeoNET responds to visual features that are characteristic of Caspar Luyken. Using PigeoNET's visualisation, we are able to determine for each region how characteristic it is for each of the two artists. We created a visualisation based on the certainty scores for Jan Luyken and Caspar Luyken. Figure 2.6 shows the visualisation using colour coding on a yellow to blue scale. The yellow regions are characteristic for Jan Luyken, whereas the blue regions are characteristic for Caspar Luyken, the green regions are indeterminate and show characteristics of either artists in equal amounts.

This example demonstrates the potential use of PigeoNET to support the study of

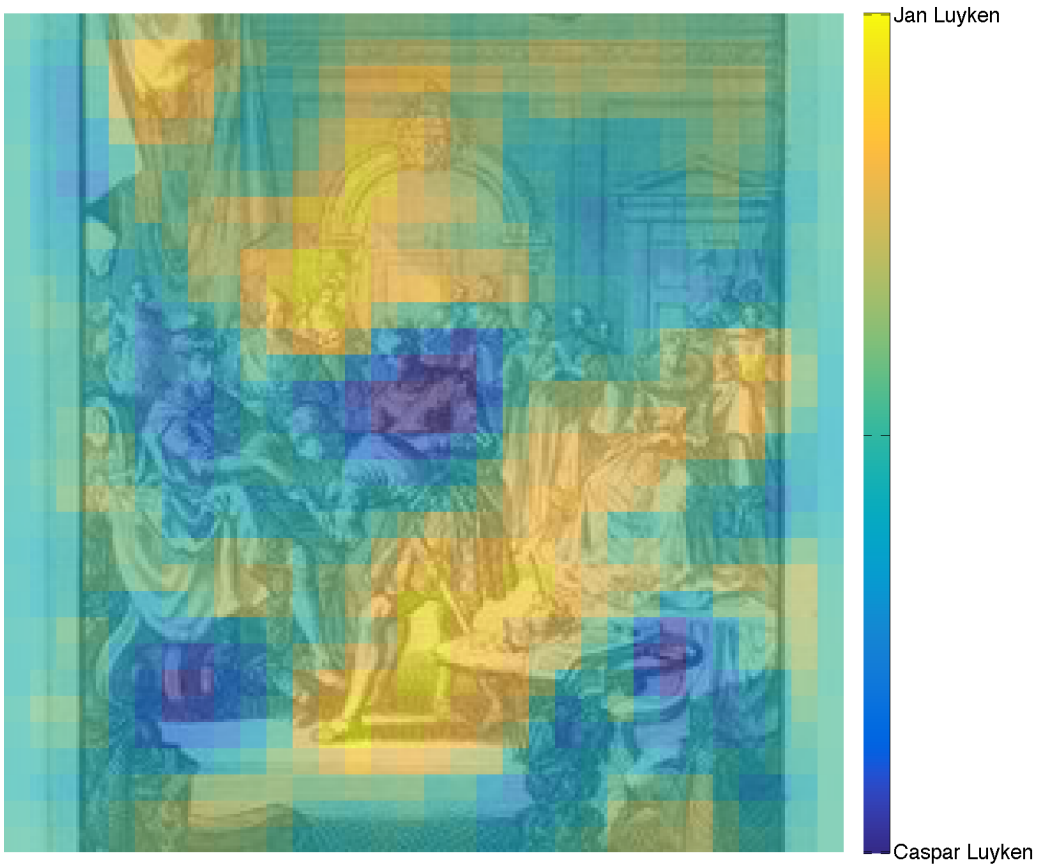


Figure 2.6: Visualisation of how characteristic each image region is for the artists Jan and Caspar Luyken. The yellow regions are characteristic of Jan Luyken, whereas the blue regions are characteristic of Caspar Luyken.

dual-authorship artworks.

2.5 Discussion

Previous work on automatic artist attribution has shown that prior knowledge can be leveraged in order to engineer features for automatic artist attribution. In this chapter, we presented a novel approach that does not rely on prior knowledge, and is capable of discovering characteristic features automatically enabling a successful artist attribution. Additionally, we demonstrated that PigeoNET visualisations reveal artwork regions most characteristic of the artist and that PigeoNET can aid in answering out-

standing questions regarding dual-authorship.

In what follows, we discuss considerations regarding the dataset used and address how the selection of subsets may affect the nature of visual characteristics discovered.

Although the Rijksmuseum challenge dataset is the largest available dataset containing digital reproductions of artworks acquired under controlled conditions [85], it does suffer from two main limitations. First, given the wide variety of artwork types, it is unclear how the "controlled conditions" were defined for different artworks. Any variation in the reproduction setting (e.g., illumination, perspective, camera type) may be picked up by PigeoNET. Presumably, our P (prints only) datasets suffer less from this problem. Still, even in these datasets subtle differences in digitization (e.g., artifacts introduced by the scanner or photocamera) may leave visual marks that are picked up by PigeoNET. An ideal dataset for attribution would be one in which no such visual marks are present. However, such datasets do not exist and are hard (if not impossible) to create on this scale. Therefore, transparency of the acquired features by PigeoNET and proper visualisations are essential to aid art experts in their assessment of the feasibility of classifications.

The second limitation concerns the labelling of artworks. After having performed our main experiments, we discovered that for some artworks, the Rijksmuseum catalog lists multiple contributions, whereas the Rijksmuseum challenge dataset only lists a single artist [85]. The contributions listed in the Rijksmuseum catalog vary greatly (from inspiration to dual-authorship) and do not always influence the actual attribution, but do create uncertainty about the attribution of artworks in the Rijksmuseum challenge dataset. Although this significantly limits the possibility of learning stylistic features from such artworks, it does not prohibit PigeoNET from learning visual characteristics that are associated with the primary artist as such characteristics remain present in the artwork. Still, the validity and consistency of attributions is of major concern to safeguard the validity of methods such as PigeoNET. Also in the creation of such databases, involvement of human art experts is required.

The results obtained in this work on the automatic artist attribution task show that

PigeoNET is capable of accurately attributing unseen works to the correct artist. The increase of performance for the sets with a higher number of examples shows that including more examples per artist leads to a better performance. Moreover, the complete Rijksmuseum Challenge dataset is a highly diverse dataset with many different types of art. For some cases (e.g., the porcelain of the Meissener Porzellan Manufaktur) this results in a class that is visually very distinctive from the rest of the dataset, which could make it easier to identify the correct artist. However, when comparing the performances obtained on the homogeneous P1 subsets (prints only) with those on the more heterogeneous A subsets (all artwork types), the difference in performance is quite small. This demonstrates that PigeoNET is capable of learning a rich representation of multiple artwork types without a major impact on its predictive power. Part of the types of features discovered in the A subsets are likely to distinguish between art types (e.g., a porcelain object versus a painting), rather than between author styles. In the P subsets, features will be more tuned to stylistic differences, because these subsets are confined to a single type of artwork.

Our findings indicate that the number of artists and the number of examples per artist have a very strong influence on the performance, which suggests that a further improvement of the performance is possible by expanding the dataset. In future research we will determine to what extent this is the case.

2.6 Conclusion

In this chapter we have evaluated a feature learning system to assess to what extent it is possible to discover an artist's visually characteristic features. The results on the automatic attribution task demonstrate that the system is capable, up to a high degree of accuracy, of using visual characteristics to assign unseen artworks to the correct artist. Moreover, we demonstrated the possibility of using the visual characteristics to reveal the artist of a specific region within an artwork, which in the case of multiple artists could lead to new discoveries about the origin and creation of important works of cultural heritage. Hence, we may conclude that PigeoNET represents a fruitful ap-

proach for future computer-supported examination of artworks.

3

Scale-variant and scale-invariant features

This chapter has been previously published as: N. van Noord, E. Postma (2017). Learning scale-variant and scale-invariant features for deep image classification. *Pattern Recognition*, 61, pp. 583–592.

Abstract

Convolutional Neural Networks (CNNs) require large image corpora to be trained on classification tasks. The variation in image resolutions, sizes of objects and patterns depicted, and image scales, hampers CNN training and performance, because the task-relevant information varies over spatial scales. Previous work attempting to deal with scale variations focused on encouraging scale-invariaatnt CNN representations. However, scale-invariant representations are incomplete representations of images, because images contain scale-variant information as well. This chapter addresses the combined development of scale-invariant and scale-variant representations. We propose a multi-scale CNN method to encourage the recognition of both types of features and evaluate it on a challenging image classification task involving task-relevant characteristics at multiple scales. The results show that our multi-scale CNN outperforms single-scale CNN. This leads to the conclusion that encouraging the combined development of a scale-invariant and scale-variant representation in CNNs is beneficial to image recognition performance.

3.1 Introduction

Convolutional Neural Networks (CNN) have drastically changed the computer vision landscape by considerably improving the performance on most image benchmarks [71, 43]. A key characteristic of CNNs is that the deep(-based) representation, used to perform the classification, is generated from the data, rather than being engineered. The deep representation determines the type of visual features that are used for classification. In the initial layers of the CNN, the visual features correspond to oriented edges or colour transitions. In higher layers, the visual features are typically more complex (e.g., conjunctions of edges or shapes). Finding the appropriate representation for the task at hand requires presenting the CNN with many instances of a visual entity (object or pattern) in all its natural variations, so that the deep representation captures most naturally occurring appearances of the entity.

Three main sources of natural variation are the location, the viewpoint, and the size of an object or pattern. Variations in location are dealt with very well by a CNN [36], which follows naturally from the weight sharing employed in the convolution layers [74]. CNNs can also handle variations in viewpoint by creating filters that respond to viewpoint-invariant features [68]. Size variations pose a particular challenge in CNNs [126], especially when dealing with image corpora containing images of varying resolutions and depicting objects and patterns at different sizes and scales, as a result of varying distances from the camera and blurring by optical imperfections, respectively. This leads to variations in image resolution, object size, and image scale, which are three different properties of images. The relations between image resolution, object size, and image scale is formalized in digital image analysis using Fourier theory [37]. Spatial frequencies are a central concept in the Fourier approach to image processing. Spatial frequencies are the two-dimensional analog of frequencies in signal processing. The fine details of an image are captured by high spatial frequencies, whereas the coarse visual structures are captured by low spatial frequencies. In what follows, we

provide a brief intuitive discussion of the relation between resolution and scale, without resorting to mathematical formulations.

3.1.1 Image resolution, object size, and image scale

Given an image, its resolution can be expressed in terms of the number of pixels (i.e., the number of samples taken from the visual source); low resolution images have fewer pixels than high resolution images. The scale of an image refers to its spatial frequency content. Fine scale images contain the range from high spatial frequencies (associated with small visual structures) down to low spatial frequencies (with large visual structures). Coarse scale images contain low spatial frequencies only. The operation of spatial smoothing (or blurring) of an image corresponds to the operation of a low-pass filter: high spatial frequencies are removed and low spatial frequencies are retained. So, spatial smoothing a fine scale image yields a coarser scale image.

The relation between the resolution and the scale of an image follows from the observation that in order to represent visual details, an image should have a resolution that is sufficiently high to accommodate the representation of the details. For instance, we consider the chessboard pattern shown in Figure 3.1a. Figure 3.1b shows a 6×6 pixel reproduction of the chessboard pattern. The resolution of the reproduction is insufficient to represent the fine structure of the chessboard pattern. The distortion of an original image due to insufficient resolution (or sampling) is called *aliasing* [37].

As this example illustrates, image resolution imposes a limit to the scale at which visual structure can be represented. Figure 3.1c displays the space spanned by resolution (horizontal axis) and scale (vertical axis). The limit is represented by separation of the shaded and unshaded regions. Any image combining a scale and resolution in the shaded area suffers from aliasing. The sharpest images are located at the shaded-unshaded boundary. Blurring an image corresponds to a vertical downward movement into the unshaded region

Having discussed the relation between resolution and scale, we now turn to the discussion of the relation of object size to resolution and scale. Real-world images with a

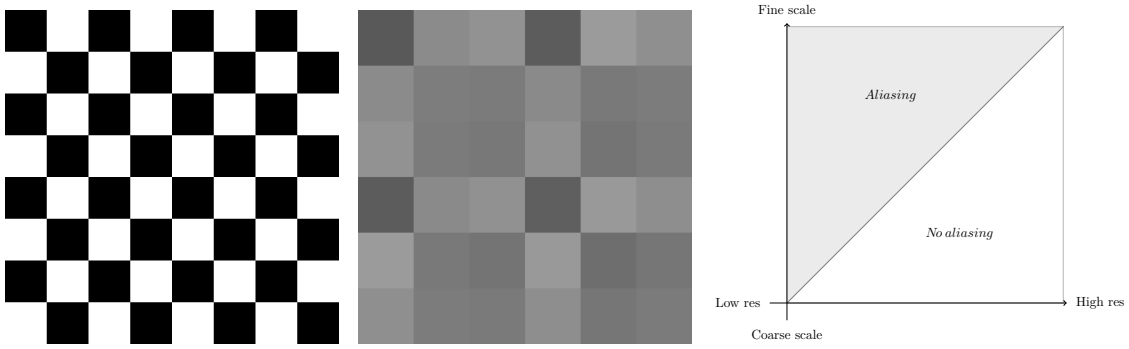


Figure 3.1: Illustration of aliasing. (a) Image of a chessboard. (b) Reproductions of the chessboard with an image of insufficient resolution (6×6 pixels). The reproduction is obtained by applying bicubic interpolation. (c) The space spanned by image resolution and image scale. Images defined by resolution-scale combinations in the shaded area suffer from aliasing. See text for details.

Figure 3.2: Artwork *'Horse-smith with a donkey'* ('Hoefsmid bij een ezel') by Jan de Visscher.

given scale and resolution contain objects and structures at a range of sizes [80], For example, the image of the artwork shown in Figure 3.2, depicts large-sized objects

(people and trees) and small-sized objects (hairs and branches). In addition, it may contain visual texture associated with the paper it was printed on and with the tools that were used to create the artwork. Importantly, the same object may appear at different sizes. For instance, in the artwork shown, there are persons depicted at different sizes. The three persons in the middle are much larger in size than the one at the lower right corner. The relation between image resolution and object size is that the resolution puts a lower bound on the size of objects that can be represented in the image. If the resolution is too low, the smaller objects cannot be distinguished anymore. Similarly, the relation between image scale and object size is that if the scale becomes too coarse, the smaller objects cannot be distinguished anymore. Image smoothing removes the high-spatial frequencies associated with the visual characteristics of small objects.

3.1.2 Scale-variant and scale-invariant image representations

Training CNNs on large image collections that often exhibit variations in image resolution, depicted object sizes, and image scale, is a challenge. The convolutional filters, which are automatically tuned during the CNN training procedure, have to deal with these variations. Supported by the acquired filters the CNN should ignore task-irrelevant variations in image resolution, object size, and image scale and take into account task-relevant features at a specific scale. The filters providing such support are referred to as scale-invariant and scale-variant filters, respectively [35].

The importance of scale-variance was previously highlighted by Gluckman [35] and Park et al. [90], albeit for two different reasons. The first reason put forward by Gluckman arises from the observation that images are only partially described by scale invariance [35]. When decomposing an image into its scale-invariant components, by means of a scale-invariant pyramid, and subsequently reconstructing the image based on the scale-invariant components the result does not fully match the initial image, and the statistics of the resulting image do not match those of natural images. For training a CNN this means that when forcing the filters to be scale-invariant we might

miss image structure which is relevant to the task. By means of space-invariant image pyramids, which separate scale-specific from scale-invariant information, proposed in [35], Gluckman et al. demonstrated that object recognition benefitted from scale-variant information.

The second reason was presented by Park et al. in [90], where they argue that the need for scale-variance emerges from the limit imposed by image resolution, stating that "Recognizing a 3-pixel tall object is fundamentally harder than recognizing a 300-pixel object or a 3000-pixel object." [90, p. 2]. While recognising very large objects comes with it own challenges, it is obvious that the recognition task can be very different depending on the resolution of the image. Moreover, the observation that recognition changes based on the resolution ties in with the previously observed interaction between resolution and scale: as a reduction in resolution also changes the scale. Park et al. [90] identify that most multi-scale models ignore that most naturally occurring variation in scale, within images, occurs jointly with variation in resolution, i.e. objects further away from the camera are represented at a lower scale and at a lower resolution. As such they implement a multi-resolution model and demonstrate that explicitly incorporating scale-variance boosts performance.

Inspired by the earlier studies of Gluckman [35] and Park et al. [90], we propose a multi-scale CNN which explicitly deals with variation in resolution, object size and image scale, by encouraging the development of filters which are scale-variant, whilst constructing a representation that is scale-invariant.

The remainder of this chapter is organised as follows. Section 3.2 contains an overview of previous work that deals with scale variation for learning deep image representations. In Section 3.3 we provide a detailed presentation of our multi-scale CNN for scale-invariant and scale-variant filters. Section 3.4 outlines the task used for evaluating the performance of the multi-scale CNN. In Section 3.5 the experimental setup is described, including the dataset and the experimental method. In Section 3.6 the results of the experiments are presented. We discuss the implications of using multi-scale CNNs in Section 3.7. Finally, Section 3.8 concludes by stating that combining

scale-variant and scale-invariant features contributes to image classification performance.

3.2 Previous work

In this section, we examine learning deep image representations that incorporate scale-variant and/or scale-invariant visual features by means of CNNs. Scale variation in images and its impact on computer vision algorithms is a widely studied problem [80, 83], where invariance is often regarded as a key property of a representation [78]. It has been shown that under certain conditions CNN will develop scale-invariant filters [73]. Additionally, various authors have investigated explicitly incorporating scale-invariance in deep representations learnt by CNN [100, 126, 36, 65, 52]. While these approaches successfully deal with scale-invariance they forgo the problem of recognising scale-variant features at multiple scales [90].

Standard CNN trained without any data augmentation will develop representations which are scale-variant. As such it is only capable of recognising the features it was trained on, at the scale it was trained on, such a CNN cannot deal with scale-variant features at different scales. A straightforward solution to this limitation is to expose the CNN to multiple scales during training, this approach is typically referred to as scale jittering [111, 105, 33]. It is commonly used as a data augmentation approach to increase the amount of training dataset, and as a consequence reduce overfitting. Additionally, it has been shown that scale jittering improves classification performance [105]. While part of the improved performance is due to the increase in training data and reduced overfitting, scale jittering also allows the CNN to learn to recognise more scale-variant features, and potentially develop scale-invariant filters. Scale-invariant filters might emerge from the CNN being exposed to scale variants of the same feature. However, *standard* CNN typically do not develop scale-invariant filters [73], and instead will require more filters to deal with the scaled variants of the same feature [126], in addition to the filters needed to capture scale-variant features. A consequence of this increase in parameters, which increases further when more scale variation is in-

troduced, is that the CNN becomes more prone to overfit and training the network becomes more difficult in general. In practice, this limits scale jittering to small scale variations. Moreover, scale jittering is typically implemented as jittering the resolution, rather than explicitly changing the scale, which potentially means that jittered versions are actually of the same scale.

One approach that is able to deal with larger scale variations, whilst offering many of the same benefits as scale jittering is multi-scale training [124]. Multi-scale training consists of training separate CNN on fixed size crops of resized versions of the same image. At test time the softmax class posteriors of these CNN are averaged into a single prediction, taking advantage of the information from different scales and model averaging [16], resulting in improved performance over single scale classification. However, because the work by Wu et al. [124] is applied to datasets with a limited image resolution, they only explore the setting in which multi-scale training is applied for a relatively small variation in scales, and only two scales. Moreover, as dealing with scale variation is not an explicit aim of their work they do not analyse the impact of dealing with multiple scales, beyond that it increases their performance. Finally, because of the limited range of scales they explored they do not deal with aliasing due to resizing. Aliasing is harmful for any multi-scale approach as it produces visual artifacts which would not occur in natural images of the reduced scale, whilst potentially obfuscating relevant visual structure at that scale.

In this work we aim to explicitly learn scale-variant features for large variations in scale, and make the following three contributions: (1) We present a modified version of multi-scale training that explicitly creates multiple scales, reducing aliasing due to resizing, allowing us to compare larger scale differences whilst reducing redundancy between scales. (2) We introduce a novel dataset of high resolution images that allows us to explore the effects of larger scale variations. (3) We perform an in-depth analysis of the results and compare different scale combinations in order to increase our understanding of the influence of scale-variation on the classification performance.

3.3 Multi-scale Convolutional Neural Network

In this section we present the multi-scale CNN by explaining how a standard (single-scale) CNN performs a spatial decomposition of images. Subsequently, we motivate the architecture of the multi-scale CNN in terms of the scale-dependency of the decomposition.

CNNs perform a stage-wise spatial decomposition of the input, for an image of a face this is typically described in terms of pixels which combine into edges, which combine into contours, into simple parts of faces, and finally into entire faces. This is achieved by repeating alternating convolution and pooling operations across stages. At the first stage, in the convolution operation, the image is transformed by a set of several (learned) filters with a limited spatial extent (typically a small sub-region of the image). After which the pooling operation reduces the dimensionality of the convolution. At each subsequent convolution-pooling stage, the output of the previous stage is convolved by another set of (learned) filters and subsequently pooled [74]. As a consequence, both the complexity of the composite transformation and the image area covered increases with each stage [29]. Therefore, relatively simple visual patterns with a small spatial extent are processed at the early stages, whereas more complex visual patterns with a large spatial extent are processed at the later stages [74, 126]. This dependency closely ties the representation and recognition of a visual pattern to its spatial extent, and thus to a specific stage in the network [101, 41].

The strength of this dependency is determined by the network architecture in which the amount of subsampling (e.g., via strided operations or pooling) is specified, this also determines the size of the spatial output of the network. In the case of a simple two layer network with 2×2 filters as in Figure 3.3, the network produces a single spatial output per 4×4 region in the input. Whereas in a deeper network (containing strided and pooling operations such as in [71]) a single output can describe a 64×64 pixel region of the input. Because the amount of subsampling is determined by the network architecture, the size of the output, or spatial output map, scales with

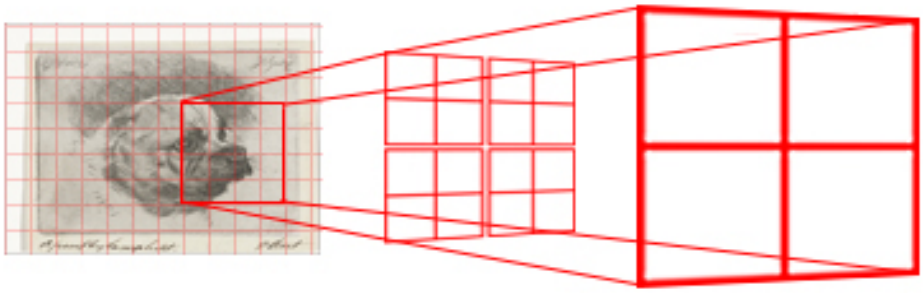


Figure 3.3: CNN perform a stage-wise spatial decomposition. A first layer of 2×2 filters is applied to the input image, followed by second layer of strided 2×2 filters, which spatially subsample the output of the previus layer. This results in a 2×2 output, which describes a 4×4 input region.

the size of the input. Due to the scaling the relative portion of the input described by a single output node decreases: a 4×4 pixels image can be described with 4 non-overlapping 2×2 filters, where each filter describes one-fourth of the image. Yet for an 8×8 image it would require 16 identically sized filters to cover the input, reducing the portion of the image described by each filter to one-sixteenth. The reduction in relative proportion described by a single output strongly influences the characteristics of the filters in the network. Filters that describe one-sixteenth of a portrait picture might only correspond to a part of a nose, or an ear, whereas filters that cover one-fourth of the picture might correspond to an entire cheek, chin, or forehead. For artist attribution this means that a network with filters that cover relatively small parts of the input are suitable to describe the fine characteristics but cannot describe the composition or iconography of the artwork. As such the network architecture should be chosen in concurrence with the resolution of the input.

Because training CNNs on an image dataset results in a hierarchy of feature representations with increasing spatial extent, a network capable of analysing the entire range from fine to coarse visual characteristics in an image requires many stages in order to capture all the intermediate scales. Moreover, as to not discard information by subsampling between stages, the subsampling has to be performed gradually. Gradual subsampling is performed by having a very deep network with many stages, each subsampling a little. The complexity and the number of parameters in a network is

determined by the number of layers and the number of parameters per layer, as such, increasing the number of layers increases the complexity of the network. A more complex network requires more training data, which despite the increasing availability of images of artworks is still lacking. Moreover, the computational demand of the network increases strongly with the complexity of the network, making it infeasible to train a sufficiently complex network [45]. An alternative to increasing the complexity of an individual CNN is to distribute the task over specialised CNNs and combining the resulting predictions into a single one. The biologically motivated multi-column CNN architecture [16] is an example of such an approach.

The multi-scale CNN presented in this chapter is based on a multi-scale image representation, whereby a separate CNN is associated with each scale. This allows the scale-specific CNNs to develop both scale-variant and scale-invariant features. The multi-scale representation is created using a Gaussian pyramid [2]. The bottom level of the pyramid corresponds to the input image, subsequent levels contain smoothed (and down-sampled) versions of the previous levels. A visual representation of the model architecture is shown in Figure 3.4. Note that down-sampling is not necessary to create the higher pyramid levels, and that it is possible to fix the resolution and only change the scale. However, smoothing results in a redundancy between neighbouring pixels, as they convey the same information.

3.4 Image classification task

The proposed multi-scale CNN will be evaluated on a task involving a large data set of images of artworks that are heterogeneous in scale and resolution. In our previous work, we have applied a single CNN to a comparable dataset to study computational artist attribution (where the task was to determine who authored a given artwork) [120]. For artist attribution there is often insufficient information on a single scale to distinguish between very similar artists. For instance, the works of two different artists who use very similar materials to create artworks depicting different scenes might be indistinguishable when considering the very fine details only. Alternatively, when

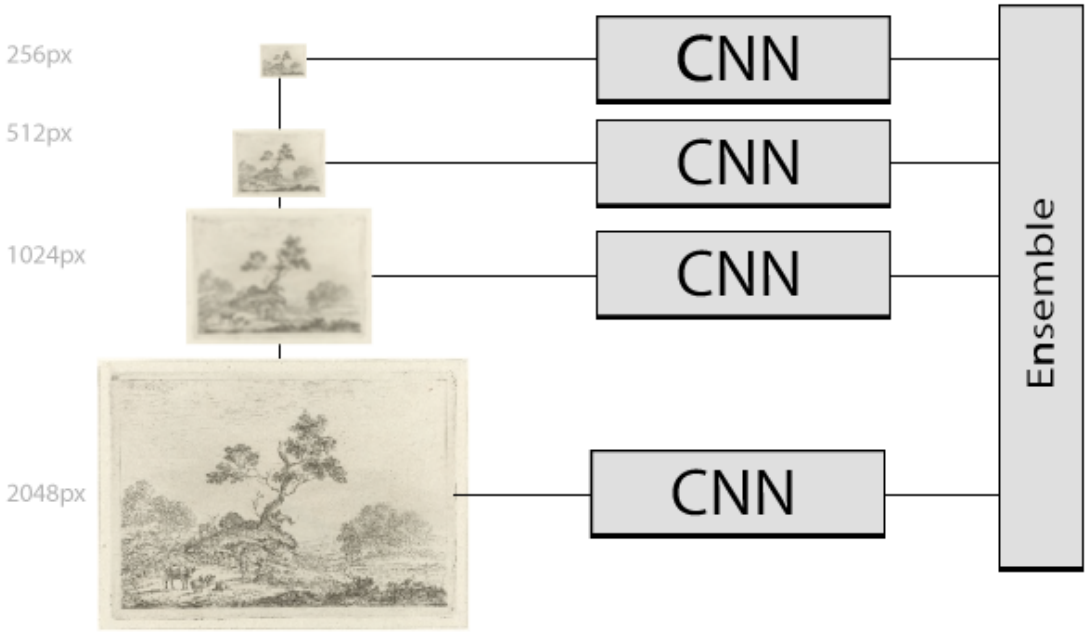


Figure 3.4: Visual representation of the model architecture. A scale-specific network is applied to a different scale, and combined in the ensemble.

artists create artworks depicting a similar scene using different materials, these may be indistinguishable at a coarse spatial scale. Hence, successful artist attribution requires scale-variant features in addition to scale-invariant features.

Artist attribution is typically performed by combining current knowledge on the artist's practices, technical data, and a visual assessment of the artwork as to establish its origin and value from an economical and historical perspective [63]. In recent years it has been shown that this visual assessment can be performed computationally and can lead to promising results on artist attribution image classification tasks [112, 63, 47, 118, 79, 1]. The increased availability of visual data from the vast digital libraries of museums and the challenges associated with the unique nature of artworks has led to an interest in this domain by researchers from a large diversity of fields. This diversity has resulted in a great many different approaches and techniques aimed at tackling the problem of visual artist attribution. The visual assessment of artworks by art experts generally focuses on textural characteristics of the surface (e.g., the can-

vas) or on the application method (e.g., brushstrokes) [44], this in turn has shaped many of the computational approaches to visual artwork assessment (e.g., [63, 60, 79]).

More recently, it has been shown that general purpose computer vision approaches can be used for the visual assessment of artworks, specifically SIFT features [85] and deep-based representations as learned by a CNN for a general object recognition task (i.e., ImageNet) [66, 97] can be used to perform image classification tasks on artworks. This development is a deviation from the practice as performed by art experts, with the focus shifted from small datasets of a few artists with high resolution images (5 to 10 pixels per mm) to large datasets with many artists and lower resolution images (0.5 to 2 pixels per mm). By using images of a lower resolution the amount of details related to the artist's specific style in terms of application method (e.g., brushstrokes) and material choices (e.g., type of canvas or paper) become less apparent, which shifts the focus to coarser image structures and shapes. However, using a multi-scale approach to artist attribution it is possible to use information from different scales, learning features appropriate from both coarse and fine details.

3.5 Experimental setup

This section describes the setup of the artist attribution experiment. The setup consists of a specification of the CNN architecture, the dataset, the evaluation, and the training parameters.

3.5.1 multi-scale CNN architecture

The multi-scale CNN architecture used in this work is essentially an ensemble of single-scale CNN, where the single-scale CNN matches the architecture of the previously proven ImageNet model described in [108]. We made two minor modifications to the architecture described in [108] to account for our larger image size, and different classification task, in that we (1) replaced the final 6×6 average pooling layer by a global average pooling layer which averages the final feature map regardless of its spatial size, and (2) reduce the number of ouputs of the softmax layer to 210 to match the number

of classes in our dataset. A detailed specification of the single-scale CNN architecture can be found in Table 3.1, where conv-n denotes a convolutional layer with f filters with a size ranging from 11×11 to 1×1. The stride indicates the step size of the convolution in pixels, and the padding indicates how much zero padding is performed before the convolution is applied. The ReLU activation function is an element-wise operation on the layer output, discarding any negative filter activations, i.e., $max(0, x)$, where x is a filter activation.

The single-scale CNN architecture used is fully-convolutional, which means that except for the final global average pooling layer it consists solely of convolutional layers. Rather than having max or average pooling layers in the network a convolutional layer with a stride greater than 1 (typically 2) is used. This convolutional layer effectively performs the pooling, but combines it with an additional (learnt) non-linear transformation. A fully convolutional architecture has two main benefits for the work described in this chapter (1) unlike traditional CNN, a fully-convolutional CNN places no restrictions on the input in terms of resolution; the same architecture can be used for varying resolutions, and (2) it can be trained on patches and evaluated on whole images, which makes training more efficient and evaluation more accurate.

Additionally, this architecture has been shown to work well with Guided Backpropagation (GB) [108]. GB is an approach (akin to 'deconvolution' [131]) that makes it possible to visualise what the network has learnt, or which parts of an input image are most characteristic of a certain artist. GB consists of performing a backward pass through the network and computing the gradient with respect to an input image. In order to visualise which parts of an image are characteristic of an artist, the activations of the softmax class posterior layer are all set to zero, except the activation for the artist of interest, and subsequently the gradient with respect to an input image will activate strongest in the areas characteristic of that artist.

Our multi-scale is constructed as an ensemble, or multi-column [16], architecture, in which the softmax class-posteriors of the single-scale CNN are averaged and used as the final predictions for evaluation, the evaluation procedure is further described in

Table 3.1: CNN architecture of single-scale networks as used in this chapter. conv$n$ denote convolutional layers. During training a 224×224 pixels crop is used, the testing is performed on the entire input image (which shortest side is in the range of 256 up to 2048 pixels).

Layer	Filters	Size, stride, pad	Description
Training Data	-	224×224, -, -	RGB image crop
Testing Data	-	Entire image, -, -	Full RGB image
conv1.1	96	11×11, 4, 0	ReLU
conv1.2	96	1×1, 1, 0	ReLU
conv1.3	96	3×3, 2, 1	ReLU
conv2.1	256	5×5, 1, 2	ReLU
conv2.2	256	1×1, 1, 0	ReLU
conv2.3	256	3×3, 2, 0	ReLU
conv3.1	384	3×3, 1, 1	ReLU
conv3.2	384	1×1, 1, 0	ReLU
conv3.3	384	3×3, 2, 0	ReLU + Dropout (50%)
conv4	1024	1×1, 1, 0	ReLU
conv5	1024	1×1, 1, 0	ReLU
conv6	210	1×1, 1, 0	ReLU
global-pool	-	-	Global average
softmax	-	-	Softmax layer

Subsection 3.5.4.

3.5.2 Dataset

The dataset* consists of $58,630$ digital photographic reproductions of print artworks by 210 artists retrieved from the collection of the Rijksmuseum, the Netherlands State Museum. These artworks were chosen based on the following four criteria: (1) Only printworks made on paper, (2) by a single artist, (3) public domain, and (4) at least 96 images by the same artist match these criteria. This ensured that there were sufficient images available from each artist to learn to recognise their work, and excluded any artworks which are visually distinctive due to the material choices (e.g., porcelain). An example of a print from the Rijksmuseum collection is shown in Figure 3.5.

For many types of artworks there is a large degree of variation in their physical size: there are paintings of several meters in width or height, and paintings which are only

*The dataset is available at `https://auburn.uvt.nl/`.

Figure 3.5: Digital photographic reproduction of *'Head of a cow with rope around the horns'* by Jacobus Cornelis Gaal.

tens of centimeters in width or height. Moreover, for such artworks there is a large degree of variation in the ratio of pixels per mm and as such the dimension of the reproductions in pixels. Yet, this makes it very appealing to work with print artworks, as they are much more uniform in terms of physical size as for example paintings. While there is still some variation in physical size for print artworks, as shown in Figure 3.6. Previous approaches have dealt with such variations by resizing all images to a single size, which confounds image resolution with physical resolution.

Normalising the images to obtain fixed pixel to mm ratios would result in a loss of visual detail. Given that our aim is to have our multi-scale CNN develop both scale-invariant and scale-variant filters, we take the variation in scales and resolutions for granted.

A four-level Gaussian (low-pass) pyramid is created following the standard procedure for creating Gaussian Pyramids described in [2, 88]. Initially all images are resized so that the shortest side (height or width) is 2048 pixels, as to preserve the aspect ratio, creating the first pyramid level. From this first level the subsequent pyramid level is created by smoothing the previous level, and down-sampling by removing every other pixel column and row (effectively reducing the image size by a factor two).

This smoothing and down-sampling step is repeated, every time taking the previous level as the starting point, to create the remaining two pyramid levels. The smoothing steps were performed by recursively convolving the images with the Gaussian kernel G, which is defined as:

$$G = \frac{1}{256}\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

.

The resulting Gaussian pyramid consists of four levels of images with the shortest side being 256, 512, 1024, and 2048 pixels for each level respectively.

The dataset is divided into a training (70%), validation (10%), and test set (20%). The training set is used to train the network, the validation set is used to optimise the hyperparameters, and the evaluation set is used to estimate the prediction performance. All results reported in this chapter are based on the test set.

3.5.3 Training parameters

All networks were trained using an effective training procedure (cf. [71]), with the values of the learning rate, momentum, and weight decay hyperparameters being 10^{-2}, 0.9, and $5 \cdot 10^{-4}$, respectively. Whenever the error on the validation set stopped decreasing the learning rate was decreased by a factor 10.

3.5.4 Evaluation

The evaluation is performed on entire images. The fully-convolutional nature of the multi-scale CNN makes it unnecessary to perform cropping. The scale-specific prediction for an image is the average over the spatial output map, resulting in a single scale-specific prediction for the entire image. The performance on all experiments is reported using the Mean Class Accuracy (MCA), which is the average of the accu-

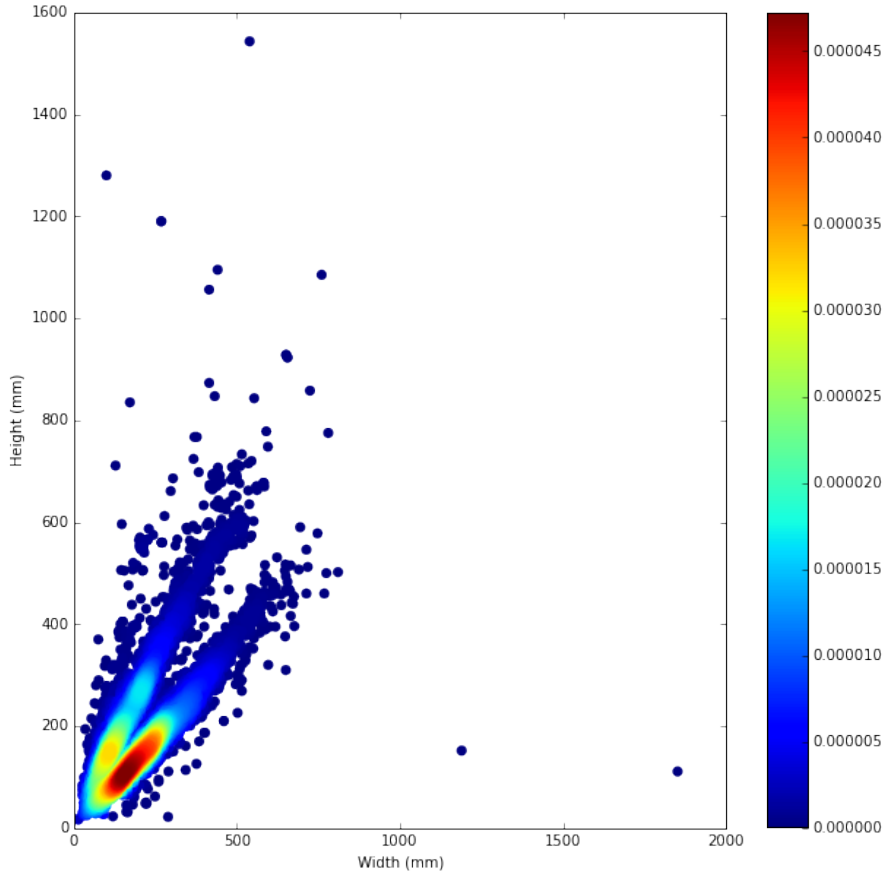


Figure 3.6: Scatter plot of physical dimensions of the artworks in the test set in millimeters; each point represents an artwork, its colour indicating the density in the area around it. The scatter plot shows that the rectangular is the predominant shape of artworks with either a landscape or a portrait orientation. The majority of the artworks cluster around a size of 250×150 mm.

racy scores obtained per artist. We report the MCA because it is not sensitive to unbalanced classes and it allows for a comparison of the results with those reported in [85, 120]. The MCA is equal to the mean of the precision per class, as such we also report the mean of the recall per class, and the harmonic mean of these mean precision and mean recall measures, also known as the F-score.

Additionally, we compare our results to those obtained by performing multi-scale training as described in [124]. We implemented multi-scale training using the same CNN architecture as used previously, and only varied the input data. Rather than blurring the images before subsampling the images, we follow [124] and directly sub-

Table 3.2: Mean Class Accuracies, Mean recall and F-score for the four individual scales and the ensemble of four scales for our approach.

Scale	MCA	Mean recall	F-Score
256	70.36	65.03	67.59
512	75.69	69.70	72.57
1024	67.69	44.08	53.48
2048	62.03	38.54	47.55
Ensemble	**82.11**	**72.50**	**77.01**

sample the images, as such the scales do not form a Gaussian Pyramid. Because the highest scale is not blurred in either case these results are identical, and are produced by the same network.

Furthermore, we report the pair-wise correlations between the Class Accuracy (CA) for each artist for the four different scales for both approaches. The pair-wise correlations between scales indicates the similarity of the performance for individual artists at those two scales. A high correlation indicates that the attributions of an artist are largely the same at both scales, whereas a low correlation indicates that the artworks of an artists are classified differently at the two scales which suggests the relevance of scale-specific information.

3.6 Results

The results of each individual scale-specific CNN of the multi-scale CNN and the ensemble averages are reported in Table 3.2. The best-performing single scale is 512. The ensemble-averaged score of the multi-scale CNN outperforms each individual scale by far. As is evident from Table 3.3, no combination of three or fewer scales outperforms the multi-scale (four-scale) CNN. We report the results obtained by multi-scale training [124] in Table 3.4.

The MCA and mean recall obtained for the resolutions greater than 512 decrease, this suggests that there is a ceiling in performance and that further increasing the resolution would not help to improve the performance. Yet, combining the predictions from each scale in an ensemble results in a boost in performance. The pair-wise cor-

Table 3.3: Mean Class Accuracies for all possible scale combinations obtained with our approach, a '+' indicates inclusion of the scale. In bold are the combinations which lead to the best combined performance in each block. The best overall score is underlined.

256	512	1024	2048	MCA	Mean recall	F-Score
+				70.36	65.03	67.59
	+			**75.69**	**69.7**	**72.57**
		+		67.96	44.08	53.48
			+	62.03	38.54	47.55
+	+			78.06	**71.61**	**74.69**
+		+		75.92	67.65	71.54
+			+	76.24	67.92	71.84
	+	+		79.15	67.71	72.98
	+		+	**80.21**	68.11	73.66
		+	+	71.41	45.4	55.51
+	+	+		80.15	72.14	75.94
+	+		+	80.87	**72.47**	**76.44**
+		+	+	79.27	68.89	73.72
	+	+	+	**80.95**	65.9	72.66
+	+	+	+	**82.12**	**72.5**	**77.01**

Table 3.4: Mean Class Accuracies for all possible scale combinations using the Mean-scale training procedure described in [124], a '+' indicates inclusion of the scale. In bold are the combinations which lead to the best combined performance in each block. The best overall score is underlined.

256	512	1024	2048	MCA	Mean recall	F-Score
+				70.56	65.74	68.07
	+			**73.5**	**68.36**	**70.84**
		+		65.63	57.96	61.56
			+	62.03	38.54	47.55
+	+			75.93	**71.02**	**73.4**
+		+		75.13	70.2	72.58
+			+	75.68	68.08	71.68
	+	+		74.8	68.51	71.51
	+		+	**77.8**	66.2	71.53
		+	+	68.7	54.69	60.9
+	+	+		78.21	**72.94**	**75.48**
+	+		+	**79.16**	71.95	75.38
+		+	+	77.72	70.54	73.95
	+	+	+	77.04	66.65	71.47
+	+	+	+	**79.98**	**73.02**	**76.34**

relations between scales as reported in Table 3.5 show larger correlations for adjacent scales than for non-adjacent scales. This pattern of correlations agrees with the causal

Table 3.5: Correlations between results per artist for each image scale

	256	512	1024	2048
256	1.00	0.56	0.27	0.18
512	0.56	1.00	0.44	0.29
1024	0.27	0.44	1.00	0.54
2048	0.18	0.29	0.54	1.00

Table 3.6: Correlations between results per artist for each image scale using multi-scale training [124].

	256	512	1024	2048
256	1.00	0.60	0.33	0.26
512	0.60	1.00	0.52	0.35
1024	0.33	0.52	1.00	0.40
2048	0.26	0.35	0.40	1.00

connection of adjacent scales (i.e., a lower scale image is directly derived from the adjacent higher scale image. Additionally, we also report the correlations between the scales using multi-scale training (cf. [124]) in Table 3.6. We note that in general the correlations in the latter case are stronger than the former, which shows that there is a greater performance difference across artists between scales for our approach, which indicates that the single-scale CNN for our approach learn a greater variety of scale-variant features.

To provide some insight into the artist-specific relevance of the four different scales, Table 3.7 lists the top five artists with the least and most variation between scales as determined by the standard deviation of their MCA across scales. From this table it can be observed that there is a large variation between artists in terms of which scales work well, where for some artists performance is highly scale-specific (a perfect performance is achieved on one scale and a completely flawed performance on another), and for others performance does not depend on scale (the performance is stable across scales).

To illustrate the effect of resolution on the automatic detection of artist-specific features, Guided Backpropagation [108] was used to create visualisations of the artwork *'Horse-smith with a donkey'* by Jan de Visscher at the four scales. Figure 3.7 shows the results of applying Guided Backpropagation to the artwork. The visualisations

Table 3.7: Overview of artists with the least and most variation between scales, and their MCA per scale.

Top five artists with least variation between scales.				
Artist	256	512	1024	2048
Johannes Janson	66.67	65.12	67.74	65.22
Pieter de Mare	80.0	82.67	86.0	81.25
Jacobus Ludovicus Cornet	73.53	76.47	73.33	79.17
Cornelis van Dalen (II)	100.0	94.44	100.0	100.0
Lucas Vorsterman (I)	85.42	89.8	83.67	88.57
Top five artists with most variation between scales.				
Artist	256	512	1024	2048
Joannes van Doetechum (I)	100.0	100.0	0.0	0.0
Totoya Hokkei	100.0	0.0	0.0	0.0
Gerrit Groenewegen	88.89	100.0	100.0	0.0
Abraham Genoels	86.67	64.29	0.0	0.0
Charles Meryon	64.0	86.67	100.0	0.0

(a) Artwork at 256 **(b)** Activation **(c)** Artwork at 512 **(d)** Activation

(e) Artwork at 1024 **(f)** Activation **(g)** Artwork at 2048 **(h)** Activation

Figure 3.7: Visualisations of the activations for the artwork *'Horse-smith with a donkey'* by Jan de Visscher at four scales. The activation shows the importance of the highlighted regions for correctly identifying the artist, the colours have been contrast enhanced for increased visibility. Best viewed in colour.

show the areas in the input image that the network considers characteristic of Jan de Visscher for that scale. A clear shift to finer details is observed when moving to higher resolutions.

As the multi-scale CNN produces a prediction vector for each image we are able to calculate the similarity of the artworks in terms of the distance in a high-dimensional

space. Using t-SNE [117] we visualise these distances in a two-dimensional space in Figure 3.8, the spatial distance indicates the similarity between images at determined by the ensemble. The t-SNE visualisation of the distances shows a clear clustering of similar artworks, in terms of shape, colour, and content.

From these visualisations we can observe that the multi-scale representation is able to express the similarities between artworks in terms of both fine and coarse characteristics. Moreover, multi-scale representation makes it possible to express the similarity between artworks which are only similar on some scales (i.e., if only the fine, or only the coarse characteristics are similar), as shown in Figure 3.8.

3.7 Discussion

In this work we explored the effect of incorporating scale-variance, as put forward by Gluckman [35], in CNN and how it can be used to learn deep image representations that deal well with variations in image resolution, object size, and image scale. The main idea behind scale-variance is that decomposing an image in scale-invariant components results in an incomplete representation of the image, as a part of the image structure is not scale-invariant. As stated in Section 3.1 Gluckman showed that image classification performance can be improved by using the scale variant image structure. This means that a good multi-scale image representations is capable of capturing both the task-relevant scale-variant and scale-invariant image structure. To this end we presented an approach for learning scale-variant and scale-invariant representations by means of an ensemble of scale-specific CNN. By allowing each scale-specific CNN to learn the features which are relevant for the task at that scale, regardless whether they are scale-invariant or not, we are able to construct a multi-scale representation that captures both scale-variant and scale-invariant image features.

We demonstrated the effectiveness of our multi-scale CNN approach on an artist attribution task, on which it outperformed a single-scale CNN and was superior to the state-of-the-art performance on the attribution task. Furthermore, we show that the best performance is achieved by combining all scales, exploiting the fact that scale-

specific attribution performance varies greatly for different artists.

Is a multi-scale approach really necessary? Our approach requires multiple scale-specific CNNs, which may be combined into a single more sophisticated CNN which acquires coarse- to fine-grained features, using high resolution images. However, such a network would have to be significantly deeper and more complex than the network used in this chapter. This would increase the computational cost for training and the amount of training data that is needed beyond what is practically feasible at this time. Therefore, we cannot rule out that a single sophisticated CNN may obtain a similar performance as our multi-scale CNN. Moreover, we suspect that such a single-scale network will struggle with coarse characteristics which are very dissimilar when observed at a fine scale, but very similar on a coarse scale, as the coarse scale analysis is conditioned on the fine scale analysis. Therefore, we expect that a single very complex CNN will not work as well as our multi-scale CNN.

Additionally, we compared our approach to Multi-scale training [124] and showed that construction a Gaussian Pyramid of the input increases performance and decreases the correlations between scales. While constructing the Gaussian Pyramid increases the computational load slightly, we believe that the reduced correlations between scales implies that our approach is better at capturing the scale variant characteristics, and is subsequently able to leverage these for increased performance.

Compared to previously proposed CNN architectures that deal with scale-variation, our approach requires many more model parameters, as the parameters are not shared between the single-scale CNN. However, we consider this a key attribute of the approach as it enables the model to learn scale-variant features, and moreover, because the parameters are not shared the models can be trained independently and in parallel. Despite this, a potential downside of our approach is that we do not explicitly learn scale-invariant features, while they might implicitly emerge from the training procedure, future work on how to explicitly learn scale-variant and scale-invariant features is needed.

We expect that the use of multi-scale CNNs will improve performances on image

recognition tasks that involve images with both fine and coarse-grained task-relevant details. Examples of such tasks are scene classification, aerial image analysis, and biomedical image analysis.

3.8 Conclusion

There is a vast amount visual information to be gleaned from multi-scale images in which both the coarse and the fine grained details are represented. However, capturing all of this visual information in a deep image representation is non trivial. In this chapter we proposed an approach for learning scale-variant and scale-invariant representations from high-resolution images. By means of a multi-scale CNN architecture consisting of multiple single-scale CNN, we exploit the strength of CNN in learning scale-variant representations, and combine these over multiple scales to encourage scale-invariance and improve performance. We demonstrated this by analysing the large amount of available details in multi-scale images for a computational artist attribution task, improving on the current state-of-the-art.

Moreover, we found that the representations at the various scales differ both in performance and in image structure learnt, and that they are complementary: averaging the class posteriors across all scales leads to optimal performance. From these findings we may conclude by stating that encouraging the combined development of scale-invariant and scale-variant representations in CNNs is beneficial to image recognition performance for tasks involving image structure at varying scales and resolutions and merits further exploration.

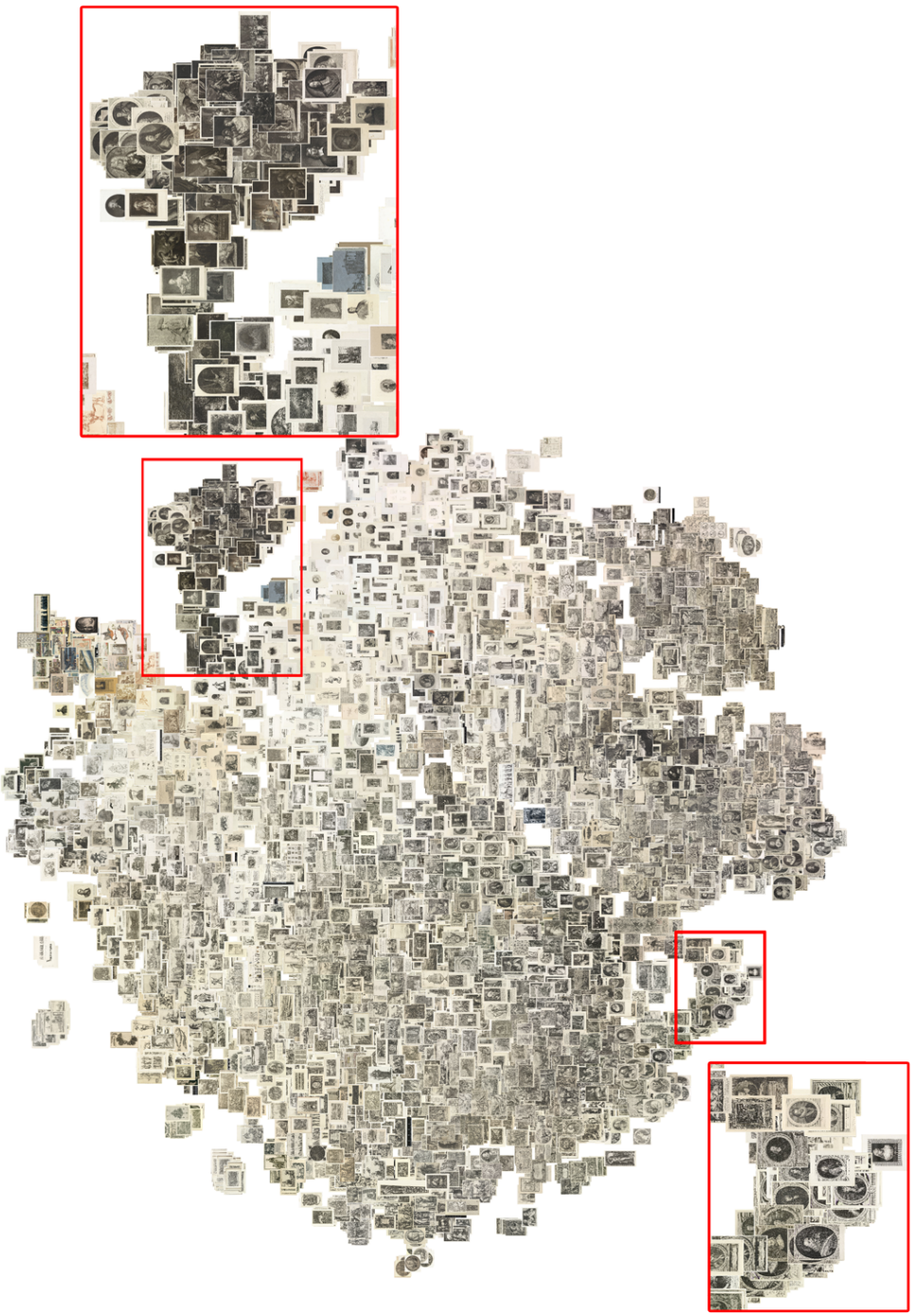


Figure 3.8: t-SNE plot of all artworks in the test set where spatial distance indicates the similarity as observed by the network. Zoomed excerpts shown of outlined areas, illustrating examples of highly similar clusters.

4

A learned representation of artist specific colourisation

This chapter has been previously published as: N. van Noord, E. Postma (2017). A learned representation of artist specific colourisation. *ICCV Workshop on e-Heritage*.

Abstract

The colours used in a painting are determined by artists and the pigments they had at their disposal. Therefore, knowing who made the painting should help in determining which colours to hallucinate when given a colourless version of the painting. The main aim of this chapter is to determine if we can create a colourisation model for paintings which generates artist-specific colourisations. Building on earlier work on natural-image colourisation, we propose a model capable of producing colourisations of paintings by incorporating a conditional normalisation scheme, i.e., conditional instance normalisation. The results indicate that a conditional normalisation scheme is beneficial to the performance. In addition, we compare the colourisations of our model that is trained on a large dataset of paintings, with those of competitive models trained on natural images and find that the painting-specific training is beneficial to the colourisation performance. Finally, we show the results of stylistic colour transfer experiments in which artist-specific colourisations are applied to the artworks of other artists. From our findings we may conclude that painting colourisation is feasible and benefits from being trained on a dataset of paintings and from applying a conditional normalisation scheme.

4.1 Introduction

Image colourisation is the task of hallucinating a colour image given a greyscale image. This task is clearly underconstrained in that a pixel with a given greyscale value can be assigned a number of different colours. Nonetheless, for most natural images there are colours which are much more likely than others, e.g., given a tropical beach scene we can all imagine that the sky and water are blue, the sand a light tan, and the palm leaves green. In other words, the semantics of the image region impose constraints on what would be plausible colours. If we are able to recognise *what* is depicted, we may be able to suggest a plausible colourisation. Recent work has shown that Convolutional Neural Networks (CNN) can obtain sufficient visual understanding to perform automatic image colourisation [72, 133, 23, 48, 51].

Depending on the type of image other factors than the image semantics might play a role in determining the likelihood of colours. For paintings the idiosyncratic use of colours by the artist greatly influences the likelihood of colours. While (realistic) paintings are often intended as realistic representations of natural scenes, the geographical, historical, and economical availability of colourants might have restricted the artist's use of colour. Additionally, and maybe more important to painters, their choice of colours is often guided by aesthetic considerations [86]. As such we pose that due to the inherent complexity of colouring paintings it is necessary to take into account both the image semantics, and the artist's palette. An example of the influence the artist's palette has on the used colours can be seen in Figure 4.1, showing two similar scenes, one with realistic colours and the other with seemingly unrealistic colours.

An image colourisation model might learn to take the artist's palette into account in the following two ways. The first way of taking the artist's palette into account is by acquiring a model of the artist's style. Previous work has shown that CNNs are capable of acquiring a model of the artist's style [120]. Therefore, the model could learn to recognise which visual content is artist-specific, and use this to facilitate artist-specific colourisation. The second way of taking the artist's palette into account, is

(a) "*View in the Woods*" **(b)** "*Evening; Red Tree*"

Figure 4.1: Examples of two paintings depicting a similar scene, but with very different colour usage. Left is "*View in the Woods*" ("*Bosgezicht*") by Jan van Kessel (courtesty of the Rijksmuseum) and right "*Evening; Red Tree*" ("*Rode boom*") by Piet Mondrian (courtesy of the Gemeentemuseum Den Haag).

to condition (part of) the CNN on the artist, and explicitly enforce that it acquires an artist-specific mapping.

In this chapter, we compare these two approaches for producing artist-specific colourisations of paintings. Our results indicate that explicitly conditioning the network makes it possible to influence the colourisation, but that surprisingly even without this explicit signal the network is able to hallucinate plausible colours.

The remainder of this chapter is organised as follows. Section 4.2 reviews previous work on image colourisation, normalisation, and computational art analysis. In Section 4.3 we describe the details of our approach. Followed by Section 4.4 in which the results are presented, as well as a number of qualitative comparisons of the colourisation results for various models. In Section 4.5 we discuss several questions which arose during this work. Finally, in Section 4.6 we conclude by stating that the approach presented is capable of producing highly diverse visually appealing colourisations of paintings.

4.2 Previous work

This section reviews earlier work pertaining to our colourisation approach: image colourisation, normalisation, and computational art analysis. Finally, our contributions are explained.

4.2.1 Image Colourisation

Work on image colourisation can be divided into user-based approaches and fully automatic approaches. User-based approaches rely on interaction (e.g., provide *scribbles* or reference images) with the user, whereas fully automatic approaches aim to provide a colourised image without user interaction, see [15] for a comprehensive overview.

Recent work on fully automatic image colourisation has shown that Convolutional Neural Networks (CNN) are capable of producing visually appealing colourisation results [72, 133, 23, 48, 51]. CNN-based fully automatic approaches can be categorised into two groups: (1) per-pixel descriptor approaches [21, 72] and (2) encoder-decoder type architectures [48, 133, 23, 51]. The per-pixel descriptor approach consists of passing the input image through a (pretrained) CNN and extracting a hypercolumn descriptor [41] for each pixel. The per-pixel descriptors are subsequently fed to a classifier that predicts the colour based on the descriptor. Hypercolumns describe the region around the pixel at different scales, incorporating a large amount of context, which results in accurate predictions. However, densely extracting hypercolumns from an image is very memory intensive, making it expensive to train an end-to-end system. Larsson et al. [72] propose to extract the hypercolumns from a subset of randomly chosen locations, but only show that this works for fine-tuning a network, not for training a network from scratch.

In contrast, so called encoder-decoder architectures have shown very promising results when trained from scratch [51]. Typically, this type of architecture consists of an encoder which follows a traditional CNN layout, i.e., several layers which have an increasing number of filters and a decreasing spatial resolution. Followed by a decoder

which either upsamples using interpolation (e.g., nearest-neighbour, bilinear, or bicubic), or *deconvolution* (i.e., fractional strided convolution) [131]. Encoder-decoder architectures are trained in either a Generative Adversial setting [51], or with a pixel-wise loss [48, 133, 23].

4.2.2 Normalisation

Most modern CNNs make use of Batch Normalisation (BN) for each nonlinear unit in the network. BN reduces *internal covariate shift* (changes in the distribution of the inputs for a layer, due to weight updates in preceding layers) and accelerates training [50]. Given a batch of size T, BN normalises each channel c of its input $x \in R^{T \times C \times W \times H}$ such that it has zero-mean and unit-variance. Formally, BN is defined as:

$$y_{tijk} = \gamma_i \left(\frac{x_{tijk} - \mu_i}{\sigma_i} \right) + \beta_i. \tag{4.1}$$

where μ_i and σ_i describe the mean and standard deviation for channel C_i across the spatial axes W and H, and the batch of size T. Additionally, for each channel there is a pair of learned parameters γ and β, that scale and shift the normalised value such that they may potentially recover the original activations if needed [50]. BN is applied in a different way during training and testing. Ideally we would calculate μ_i and σ_i on the whole dataset prior to training, but as they depend on the incrementally learned weight values of preceding layers this is not possible. Instead, during training μ_i and σ_i are calculated on the actual batch and added to moving averages. The resulting averages are used during testing.

In recent work on style transfer, it was shown that accounting for instance-specific contrast improves generation results [114]. The approach, called Instance Normalisation (IN), modifies BN in the following two ways: (1) IN calculates μ_i and σ_i for each specific instance rather than for the entire batch as in BN. (2) IN does not maintain moving averages, and is applied identically during training and testing. We expect that IN might also be beneficial for painting colourisation, or even image colourisa-

tion in general, because uniform contrast changes should not alter the colourisation substantially. Moreover, a dataset of paintings consists of samples generated from different distributions (i.e., painters), as such we expect it is very unlikely that a single mean and variance are sufficient to adequately normalise the activations without introducing artifacts.

More recently, there has been work on extending feedforward style transfer [62] to deal with multiple styles by conditioning the shifting and scaling parameters on the style [25]. Conditional Instance Normalisation (CIN) modify IN such that the γ and β parameters are $N \times C$ matrices rather than length C vectors, where N is equal to the number of styles being modelled. In this work we will use CIN to modify the colour use of different artists, by conditioning the shifting and scaling parameters on the artist.

4.2.3 Computational art analysis

There is large and growing body of work on the computational analysis of artworks [59, 66, 120]. While a large portion of this work is concerned with learning characteristics of artists for classification, an increasing body of work is emerging which tries to capture artist-specific characteristics for generative purposes [30, 114, 25]. This latter type of work, is generally concerned with *style transfer* (i.e., given a style image S and a content image C produce a single image with style S_{style} and content $C_{content}$). In this work we are only concerned with the colour aspects of the style.

4.2.4 Our Contributions

In this work we make the following three contributions: (1) We present an image colourisation model* building on components from previous works, which we apply and evaluate on a dataset of paintings. (2) We compare various normalisation schemes, investigating the influence of batch versus instance normalisation, and conditional versus unconditional normalisation. (3) We show that the models using conditional and 'un-

*https://github.com/Nanne/conditional-colour

conditional' instance normalisation utilise their visual understanding of image regions in an artist-specific way, resulting in visually appealing and diverse colourisations of paintings.

4.3 Method

In this work we use a *'encoder-decoder'*-style convolutional neural network to perform end-to-end colourisation of paintings, with the additional goal of learning the artist's unique palette. To explicitly learn the artist's palette, or colour use, we add Conditional Instance Normalisation (CIN) to the network, where the γ and β parameters are conditioned on the artist.

4.3.1 Loss

For image colourisation the goal is to learn a mapping $\hat{Y} = F(X)$ from a greyscale image $X \in \mathbb{R}^{H \times W}$ to a colour image Y, where the pixel lightness values are taken to represent the greyscale image, and H, W are the image width and height respectively. Typically, colour images are represented in RGB colour space that combines colour information with luminance (intensity) information, luminance is encoded in the mean of the R, G, and B channels.

For image colourisation the CIE Lab colour space is more appropriate, because it represents luminance (L) as a channel separate from the two colour channels **a** and **b**. Colourisation in **Lab** colour space means mapping the **L** channel of an image to the **Lab** channels. In CIE Lab, **a** represents colours along the red-green axis and **b** along the blue-yellow axis. Both CIE Lab colour values are continuous valued. Hence, colourisation could be formulated as a regression task. However, previous work has shown that formulating colourisation as a regression task tends to result in desaturated colours [72, 133]. This is most likely due to the tendency of regression to favour the mean when dealing with a multimodal distribution across colours, i.e., if a colour regression model is trained on a database of t-shirts, where half of the t-shirts are completely white, and the other half are completely black it will probably favour grey at

test time.

A common solution to deal with this limitation of regression is to reformulate the task as a classification task, by discretising the target, and effectively predicting a histogram across colour bins for each pixel. We discretise the **a** and **b** channels separately by binning the axes with Q equal-width bins, where we set $Q = 32$ following [72]. Therefore, Y becomes a four dimensional matrix $Y \in [0,1]^{H \times W \times Q \times 2}$, and the loss effectively becomes the sum of the cross entropy loss for both the **a** and the **b** channel.

4.3.2 Class rebalancing

Zhang et al. [133] show that during training it is possible to re-weight the loss at each pixel, following an approach akin to sample weighting. The loss at each pixel is re-weighted based on a weighting factor determined by the rarity of the target colour. This approach prevents the loss function from being dominated by highly common colours and is similar to the approach described in [23].

Following the procedure described in [133] we estimate the empirical probability distribution of colours in the discretised space $p \in \Delta^Q$ on the training set, which is smoothed with a Gaussian kernel G_σ. Subsequently, the contribution of the probability-weighted distribution is parameterised by $\lambda \in [0,1]$. More formally, Zhang et al. [133] define the weighting factor $w \in \mathbb{R}^Q$ as:

$$w \propto \big((1-\lambda)(G_\sigma \circ p) + \lambda\big)^{-1} \tag{4.2}$$

Unlike [133] we have discretised the **a** and **b** channels separately, therefore we also have separate losses for the **a** and **b** channels. Subsequently, we weight the channels independently using weighting factors w_A and w_B respectively. We used the values of $\lambda = \frac{1}{2}$ and $\sigma = 5$ following [133].

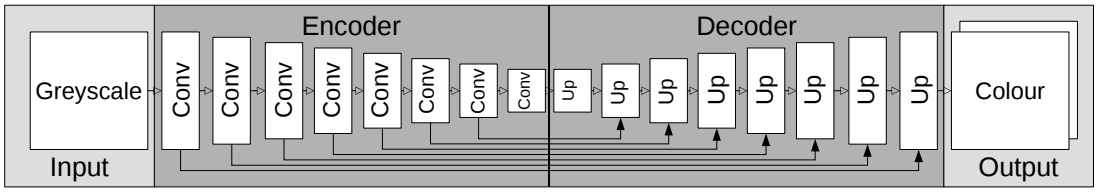


Figure 4.2: Visualisation of the network architecture. *Conv* refers to a convolution layer, and *Up* to an upsampling layer. The network input is 224×224 and the output is $224 \times 224 \times 2Q$. The bottom arrows between matching layers in the encoder and decoder indicate skip connections. Skip connections differ from regular connections in that they are concatenated to the output of the matching layer, integrating lower-level features at a higher spatial resolution to upsampled higher-level features.

4.3.3 Network architecture

The network architecture used for our colourisation model is based on the "U-Net" architecture[94] used in [51], and is shown in Figure 4.2. The U-Net architecture is an encoder-decoder architecture with skip connections between matching layers in the encoder and the decoder. The skip connections enable a direct mapping between layers at the same spatial scale. This allows the encoder-decoder path of the network to model the mapping from the grey values to colours, without being responsible for a reconstruction of all image details. We modified U-Net by replacing the upsampling (de)convolution layers with upsampling by means of nearest-neighbour interpolation, followed by a convolutional layer, as described in [25]. This upsampling method helps to avoid high spatial frequency noise [25] and 'Checkerboard' artifacts [87]. The filter size for all convolutional layers was set to 4×4, and all convolutional layers in the encoder use a stride of 2. All layers use a ReLU nonlinearity, except the last layer which is followed by a softmax activation function.

The network outputs a colour histogram for each pixel, to convert this to an actual colour we take the 'expectation' over the histogram, i.e., the weighted sum of the colour bins [72]. This results in smooth colour transitions and avoids the discontinuities obtained when taking the colour of the highest bin.

4.3.4 Training details

For training we use ADAM [70] ($\alpha = 0.001, \beta_1 = 0.9, \beta_2 = 0.999$), and all the weights are initialised using Xavier weight initialisation [34]. In terms of data augmentation we perform random horizontal flips, take 224×224 pixel crops, and introduce a random uniform brightness shift on the **L** channel in the interval $[-d, d)$. The value of d was chosen to be smaller than noticeable to human observers i.e., the colour difference (ΔE) was smaller than 1 [103].

4.4 Experiment

To evaluate our colourisation model we compare the performances of the following seven approaches on a painting dataset.

1. **Greyscale** - Baseline using greyscale versions of images (i.e., original L channel and ab channels set to 0).
2. **Larsson et al.** [72] - A CNN based approach using sparse hypercolumns trained on natural images.
3. **Zhang et al.** [133] - An encoder-decoder style network trained on natural images and paintings.
4. **BN** - Our model using Batch Normalisation trained on paintings.
5. **IN** - Our model using Instance Normalisation trained on paintings.
6. **CIN** - Our model using Conditional Instance Normalisation trained on paintings, conditioned on 1.678 artists.
7. **Randomised-CIN** - Our model with Conditional Instance Normalisation, using a random artist rather than the actual. If conditioning on the artist works then we would expect this to perform worse than our CIN model.

For each of the seven approaches, we compute the root mean square error (RMSE) across all pixels in **ab** space, and the peak signal to noise ratio (PSNR) in RGB space per image, following [72]. The greyscale approach functions as a baseline by providing no colourisation, i.e., all zero **ab** values.

The second and third approach (by Larsson et al. and Zhang et al.), are originally trained on a dataset of natural image (the ImageNet dataset) [96], and not on paintings. Both approaches incorporate copies of the first layers from a trained VGG-16 model [105], and are state-of-the-art (natural) image colourisation models. To compare the influence of the training data we fine-tune model[†] by Zhang et al. [133] on our painting dataset. There are two motivations for fine-tuning, (1) the performance of the models trained on natural images show how well such models generalise to paintings. (2) Fine-tuning the model allows us to compare the benefits of training on paintings, and how our model compares to this model in a comparable setting. For the four variations of our model the scores reveal the effectiveness of the different normalisation schemes, where the randomised-CIN is used as an extra validation of the CIN model. If the randomised-CIN model performs worse than the CIN model we can infer that the conditioning is effective. In addition, we perform qualitative evaluations of the best performing colourisation approach and demonstrate the transfer of the colour style of one artist onto an artwork of another artist.

In the remainder of this section we will introduce the dataset used for the experiment, and present the results the different approaches obtain.

4.4.1 Painting colourisation dataset

The painting colourisation performances is evaluated on the "*Painters by Numbers*" dataset as published on Kaggle[‡]. This dataset is a collection of images collected from different sources, though the majority was retrieved from "*Wikiart*" a repository which was used in a number of previous publications involving computational artwork analysis [97, 99]. Compared to the Rijksmuseum Challenge dataset used in Chapters 1 and 2 this dataset contains mostly paintings, and is subsequently much more colourful, making it more suited for colourisation.

A portion of the images included in this dataset are colourless or contain very lit-

[†]We were unable to perform any type of training with the model by Larsson et al.
[‡]https://www.kaggle.com/c/painter-by-numbers

tle colour. For most of these images this is because they are drawings on paper, and while the paper might not be purely white, a greyscale prediction would often be very close to the ground truth. Nevertheless, we chose to keep these images in the dataset as we feel they are inherent to the task, and fine-tuning the cut-off point for how much colour is desirable might arbitrarily influence the task.

From the "*Painters by Numbers*" dataset we select the subset of artists who have at least 5 artworks in the dataset, which results in a dataset consisting of $101,580$ photographic reproductions of artworks produced by a total of $1,678$ artists. Subsequently we divide the dataset into a training, validation, and test set used for training the model, evaluating stopping criteria, and reporting evaluation performances, respectively. Both the test and validation set consist of 5000 images obtained by stratified random sampling.

4.4.2 Painting colourisation

In this subsection the results on the main image colourisation task in this work are described. All results are measured using the root mean square error (RMSE) averaged across all pixels in **ab** space, and the peak signal-to-noise ratio (PSNR) per image in RGB space, following [72].

Results of the comparison between the seven approaches described in Section 4.4 in Table 4.1 show that our model achieves the highest performance according to RMSE. On this dataset all models score below the PSNR baseline, despite our model achieving the highest performance of all models. We suspect that the high PSNR for the baseline is an artifact of the colourless images in the dataset, and the calculation of this metric in RGB space. Nevertheless, we pose that the metric remains useful to compare performance between approaches.

Our model outperforms the baseline regardless of the normalisation scheme, and it outperforms the two previous colourisation approaches (by Larsson et al. and Zhang et al.) regardless of whether they were trained on natural images or fine-tuned on paintings. Nonetheless, there are differences in performance between the normalisa-

Table 4.1: Painting colourisation results measured using RMSE across all pixels, and PSNR in RGB space. A good reconstruction has a low RMSE, and a high PSNR. "Greyscale" is a baseline which provides no colourisation.

Method	RMSE	PSNR
Greyscale	0.175	**24.66**
Trained on natural images		
Larsson et al. [72]	0.168	22.18
Zhang et al. [133]	0.163	22.29
Fine-tuned on paintings		
Zhang et al. [133]	0.175	21.65
Trained on paintings		
BN	0.146	23.26
IN	0.149	23.31
CIN	**0.145**	**23.34**
Randomised CIN	0.164	22.31

tion schemes. With CIN performing slightly better than IN and BN, both in terms of RMSE and PSNR. Moreover, from the comparison between CIN and randomised-CIN we can learn that conditioning on the correct artist is important, in that using a random artist results in a deteriorated performance, which demonstrates that the CIN model learns to colourise in an artist-specific manner.

For a qualitative comparison between our models we show three sets of the colourisation results, the first set in Figure 4.3 shows the best case performance, the second set in Figure 4.4 the worst case, and the third set in Figure 4.5 the expected performance. These sets were created based on the RMSE obtained by the best performing model (Ours CIN). In Figure 4.3 we show the colour paintings in the best case. The best performances were obtained for a few natively greyscale paintings/drawings contained in the dataset. These will be discussed separately. The presence of these paintings/drawings is presumably also the cause for the high PSNR for the greyscale baseline.

When comparing the colourisations in Figure 4.3 we can observe that all three normalisation schemes produce plausible colourisations, despite not always exactly matching the ground truth. It appears that the IN and BN model produce colours which are more typical for the entire dataset, whereas CIN produces colours which closer match

Figure 4.3: Example colourisation results on Painters by Numbers. Colour images with lowest RMSE according to our CIN model.

the original: a more saturated red in the first row, greys/silvers instead of browns in the third row, and a yellow sky rather than a blue sky in the last row. These results are in line with what we would expect as differences between these models.

The cases for which we obtain the highest RMSE are those shown in Figure 4.4. For these (abstract) artworks there appears to be little to no visual semantics that provide clues about the colours used. The experimental use of colour by abstract artists such as Mark Rothko (in the second row) makes colourisation virtually impossible.

In order to see the expected performance of the CIN model we present the images

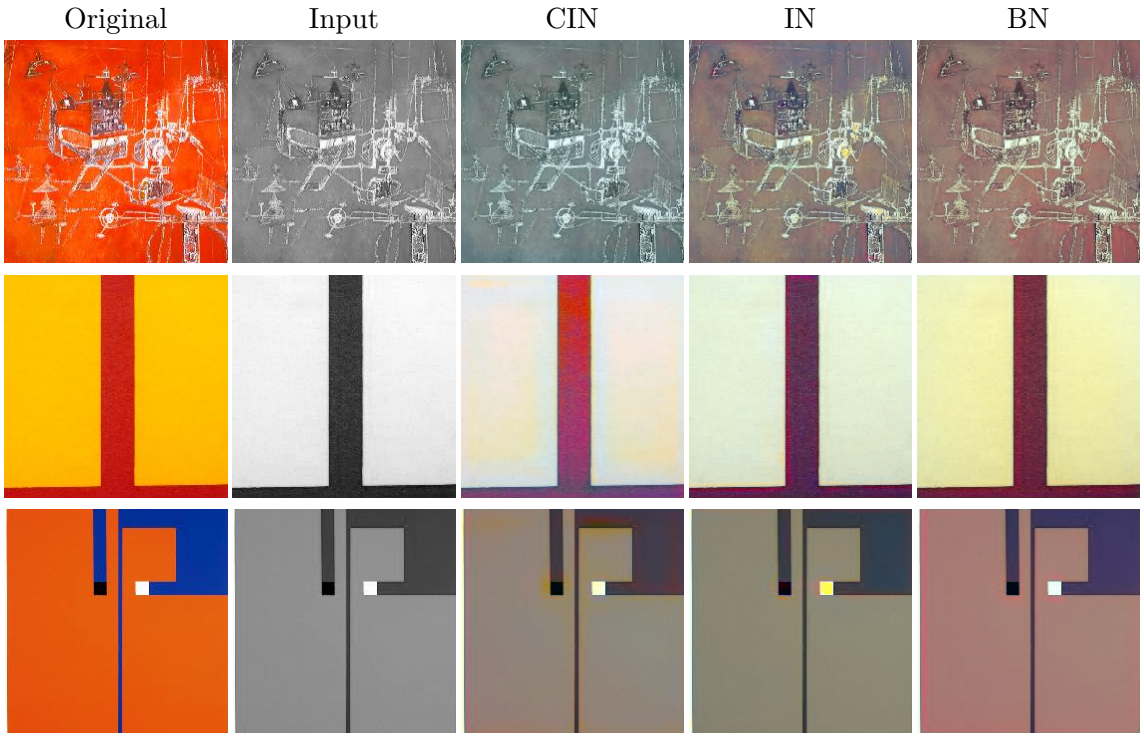


Figure 4.4: Example colourisation results on Painters by Numbers. Shown examples have the highest RMSE according to our CIN model.

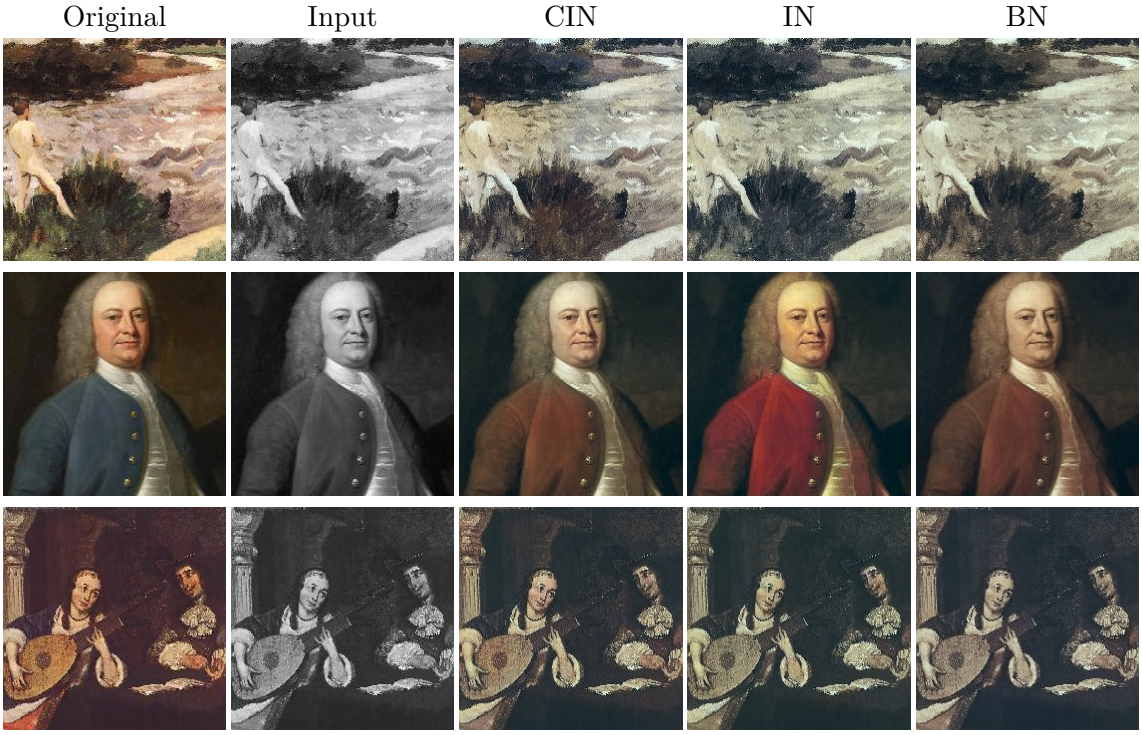


Figure 4.5: Example colourisation results on Painters by Numbers. Shown examples were randomly sampled from around the median RMSE for our CIN model.

shown in Figure 4.5, which were randomly sampled from around the median RMSE. These images show that the colourisations for both CIN and IN are very consistent with the original, although all models predict the jacket in the artwork on the second row to be red rather than blue. However, given that there is no indication in the input which colour it should be, and either colour is equally plausible we would consider this a good colourisation. The colourisations produced by BN are not far behind, though they seem to be less spatially consistent.

In Figure 4.3 we showed the colour images for which the CIN model obtained the lowest RMSE. As stated, the lowest RMSE scores were obtained for the natively greyscale images shown in Figure 4.6. The best hallucination for natively greyscale paintings and drawings, is reproduction of the input input (with potentially a slight uniform hue change). It appears all models are able to learn to generate a greyscale reproduction, though with slight differences in hue. In hindsight, we could have removed the na-

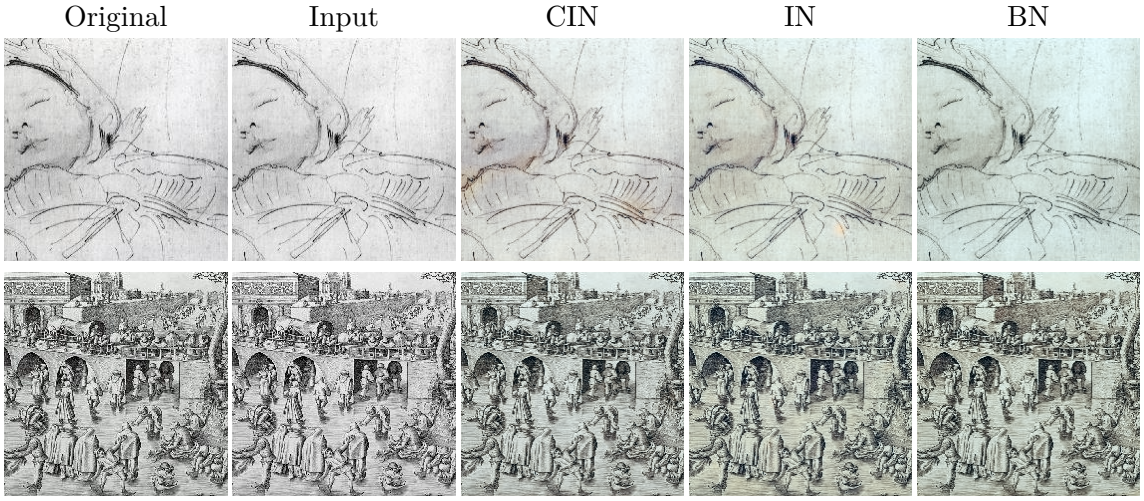


Figure 4.6: Example colourisation results on Painters by Numbers. Images with lowest RMSE according to our CIN model.

tively colourless or almost colourless artworks from the Painters by Numbers dataset to make the colourisation task more consistent. However, deciding where to place the threshold on how colourful an artwork needs to be is rather subjective and beyond the scope of this research.

For qualitative comparison between our best performing model (CIN) and the models by Larsson et al. [72] and Zhang et al. [133], we present two comparisons of colourisations. First, in Figure 4.7 we show three images for which the absolute difference in RMSE between our CIN model and the Larsson et al. [72] model is the largest. From these images we can observe that this mainly concerns abstract artworks for which a human observer would have difficulty picking the most plausible colourisation. Since, our CIN model has artist-specific information, it can produce a reasonable colour, despite the lack of semantic information in the image.

For the second comparison between our model and previous work, we show three images randomly sampled from around the median RMSE as obtained by the model by Larsson et al. [72] in Figure 4.8. The first row shows an example of a scene for which is easily recognised by humans, yet the approaches produce widely varying results. The approaches of Larsson [72] and Zhang [133] seem to generate desaturated

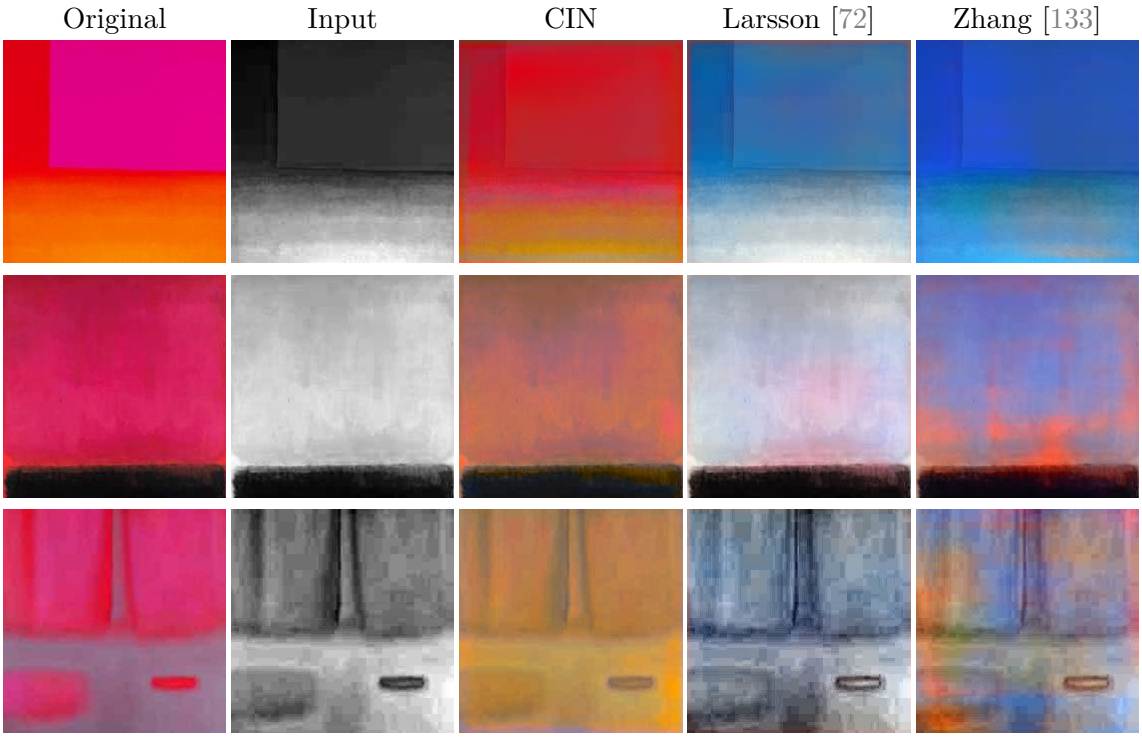


Figure 4.7: Example colourisation results on Painters by Numbers. Images where our CIN model outperforms [72] with the biggest RMSE difference.

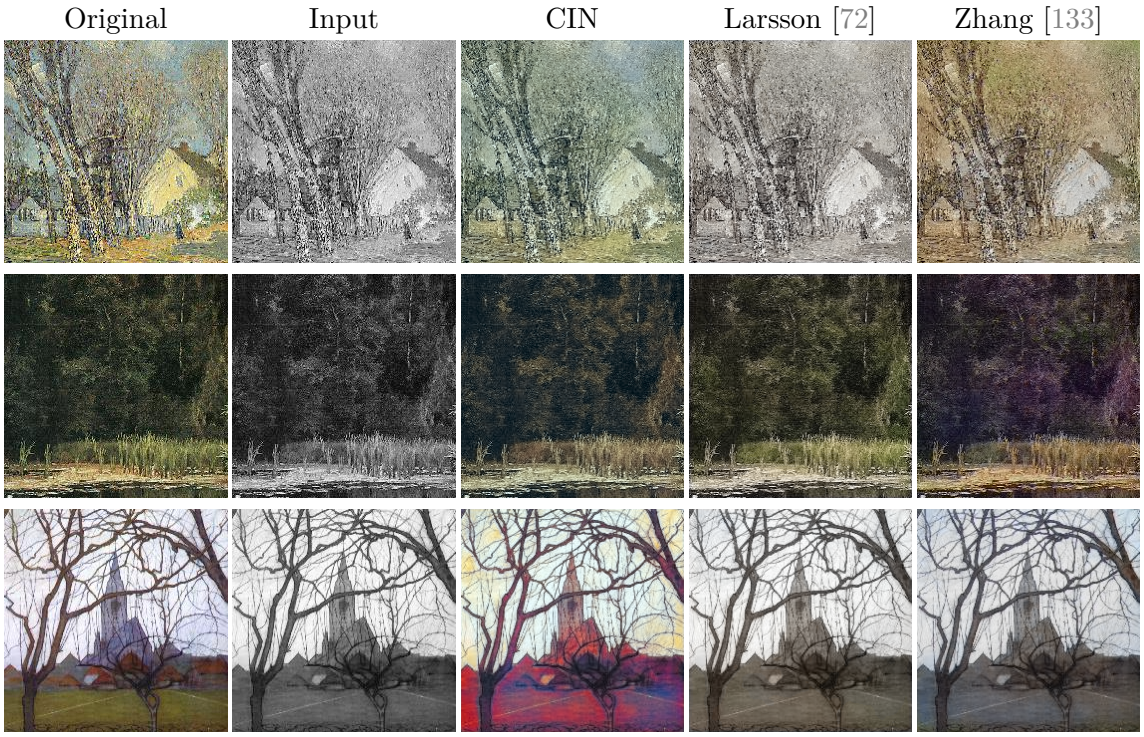


Figure 4.8: Example colourisation results on Painters by Numbers. Shown examples were randomly sampled from around the median RMSE for the model by Larsson et al.[72]

colourisations due to an imperfect recognition of the visual contents. Whereas the CIN model produces a fairly realistic colourisation. The second row shows an example of successful recognition of the visual contents, which result in more plausible colourisations by all three models. For the last row all three approaches produce results which differ strongly from the ground truth, the approaches by Larsson and Zhang produce very desaturated images, yet our model produces a highly saturated image. We suspect this is due to the strong influence of the conditioning on the artist, as the colourisation resembles the artist's style (Piet Mondrian, see Figure 4.1b).

4.4.3 Stylistic colour transfer

In the previous section we have shown that the performances of normalisation schemes are very similar. For generative purposes, the CIN model has an additional advantage in that we can choose in which colour style to render the artwork. As a result, we can

transfer the colour style of one artist onto an artwork of another artist. In this section we perform a qualitative comparison of a number of artworks on which we applied stylistic colour transfer. As the sources for our colour transfer experiments, we selected the colour styles of Maria Primachenko and Mark Rothko, because of their prominent use of colour. Note that this approach differs from what is commonly referred to as colour transfer, in that we learn the style of an artist from a database of images, rather than from a single reference image [93].

The stylistic colour transfer visualisations can be found in Figure 4.9. These columns (from left to right) show the original artwork in colour, the greyscale input to the model, a colourisation produced conditioned on the actual artist, a colourisation conditioned on Maria Primachenko, and a colourisation conditioned on Mark Rothko.

The first row shows an artwork by Roy Lichtenstein. The colourisation conditioned on his colour style is not very close to the original. Still, it does match the colour palette of many of his other artworks. The colourisation conditioned on Maria Primachenko is much more yellow, with some purple highlights. The colourisation conditioned on Mark Rothko is mainly in shades of red and orange. A similar pattern can be observed in the next row for the colourisations of an artwork by Marc Chagall, where again the colourisation differs from the original but matches that of the colour palette seen in other artworks by Marc Chagall. In the last row an artwork by Louisa Matthiasdottir is shown, here the colourisation closer matches the original, but similarly to the other examples the stylistic colour transfer results differ greatly from the original and the colourisation.

For all artworks we can observe that the three colourisations differ strongly, illustrating the artist-specific effect of the CIN model.

4.5 Discussion

The main aim of this chapter was to determine if we can create a colourisation model for paintings which can deal with the inherent complexity of the task due to the influence of both image semantics and the artist's palette. Our results indicate that auto-

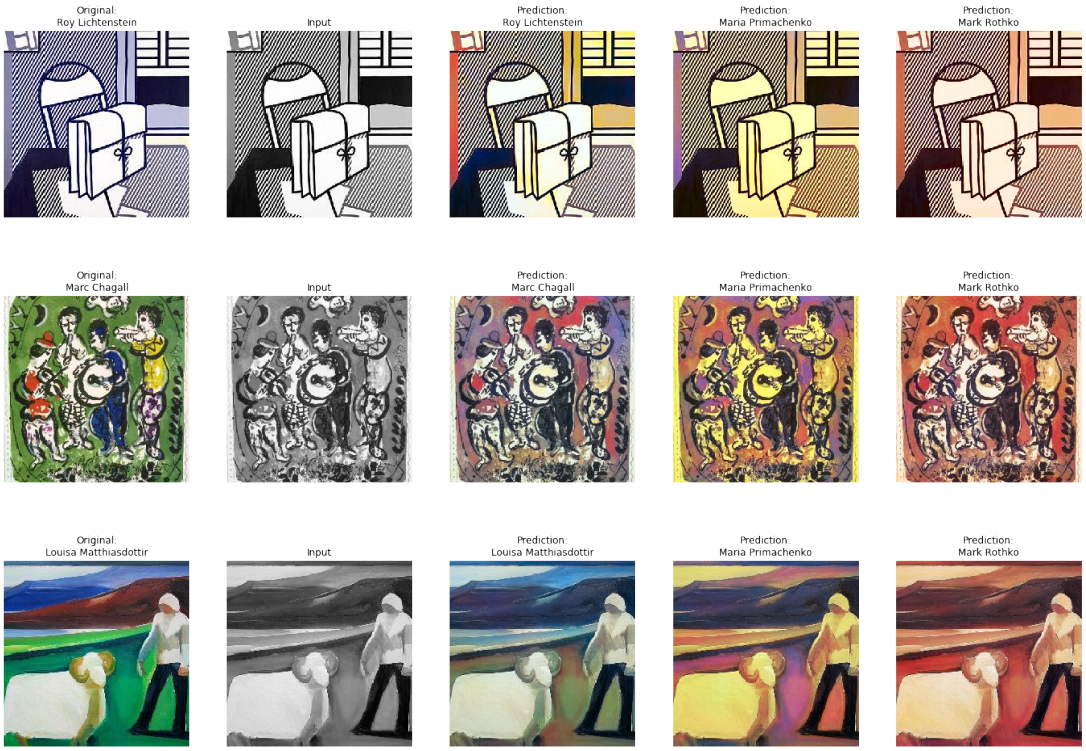


Figure 4.9: Stylistic colour transfer results. For three greyscale images we show the colourisation results of conditioning on the actual artist (third column) on Maria Primachenko (fourth column), and on Mark Rothko (last column).

matic colourisation models can produce plausible colourisations for paintings, and that performing the colourisation in an artist-specific manner appears beneficial. In what follows, we discuss (1) artist-specific colourisation, (2) normalisation schemes, (3) the use of paintings (rather than natural images) for training a colourisation model, and (4) evaluation of painting colourisation models.

(1) Artist-specific colourisation. We aimed to learn a representation of the artists colour usage such that we could do artist-specific colourisation. We compared an approach to do this explicitly (CIN) with two approaches which might be able to do this implicitly (BN and IN). Our results show that while the CIN approach can be used to explicitly alter the colourisation, the IN (and to a lesser extent the BN) approach appear to recognise the artist and use this as an information source for the colourisation. Therefore, we pose that the minor difference in performance between CIN and IN is due to the ability of the IN approach to recognise the artist or the art style to a sufficient extent, such that it is not necessary to explicitly pass this as a signal to the network.

(2) Normalisation Schemes. We found the difference in performance between the normalisation schemes to be very small. CIN offers some additional functionality in that we can influence the colourisation, at the cost of extra (conditional) parameters. Moreover, while in the work of [25] CIN is used to achieve impressive style transfer results, we pose that the representational power of the scale and shift parameters in CIN is insufficient to capture the full complexity of an artist's palette. Therefore, the main difference between the normalisation schemes seem to come down to saturation levels and small colour variations. Still, the benefits of CIN are very clear and give a definite improvement in performance for painting colourisation. It would be worthwhile to investigate whether this is also the case for other image colourisation tasks.

(3) Use of paintings for training. It could be argued that a painting specific colourisation model is not necessary, as applying realistic colours learned from natural scenes should be sufficient to produce satisfactory results. Our results indicate that the visual structure in paintings is different to such a large extent that image colourisation mod-

els trained on natural scenes only generalise to paintings which are (hyper)realistic, and do not recognise the structure in more abstract paintings. Our results indicate that fine-tuning such a network does not help to overcome this, rather that it appears to worsen the results. Additionally, besides differences in image structure for abstract paintings, these paintings also tend to use a different palette than found in nature, making it necessary to train a model specifically for this task. Although the model used for either task could be a generically applicable model.

(4) Evaluation of painting colourisation. A notable problem for image colourisation is how to do the evaluation. While quantitative measures, such as the ones used in this work, given an indication of the performance of the model, they have a number of pitfalls. These pitfalls mainly concern the bias of these measures to prefer greyscale over a wrong colour, even when the saturation levels match the ground truth (i.e., greyscale is preferred over blue when the ground truth is green). To overcome this, a number of works have employed user studies [133, 51], or external evaluation by means of a classification task [133]. For painting colourisation the former is hindered by the presence of abstract paintings for which naive users have difficulty judging the plausibility. The latter approach leads to incomparable results when applied to our work as our conditional model receives information about who the artist is, which might give it an unfair advantage. How to accurately evaluate colourisation models remains an open question.

4.6 Conclusion

In this work we proposed an image colourisation model capable of producing colourisations of paintings specific to the colour style of an artist. While the model's performance was demonstrated on paintings and artists, we pose that it is a general approach which could be applied to a wide variety of image colourisation tasks, as none of the components are specific to the painting domain. However, we pose that for cultural heritage applications the conditional aspect is most applicable, as there is often a creative human component which determines the image appearance. In conclusion,

our model is capable of producing plausible colourisations of paintings, and is highly diverse when varying the artist on which the colourisation is conditioned.

5

Light-weight pixel context encoders for image inpainting

This chapter is based on the article: N. van Noord, E.O. Postma. Light-weight pixel context encoders for image inpainting. *Submitted for publication.*

Abstract

In this work we propose Pixel Content Encoders (PCE), a light-weight image inpainting model, capable of generating novel content for large missing regions in images. Unlike previously presented convolutional neural network based models, our PCE model has an order of magnitude fewer trainable parameters. Moreover, by incorporating dilated convolutions we are able to preserve fine grained spatial information, achieving state-of-the-art performance on benchmark datasets of natural images and paintings. Besides image inpainting, we show that without changing the architecture, PCE can be used for image extrapolation, generating novel content beyond existing image boundaries.

5.1 Introduction

Reconstructing missing or damaged regions of paintings has long required a skilled conservator or artist. Retouching or inpainting is typically only done for small regions, for instance to hide small defects [9]. Inpainting a larger region requires connoisseurship and imagination: the context provides clues as to how the missing region might have looked, but generally there is no definitive evidence. Therefore, sometimes the choice is made to inpaint in a conservative manner. Take for example the painting in Figure 5.1, the left bottom corner was filled with a 'neutral' colour as to not change the interpretation of the artwork. However, with the emergence of powerful computer vision methods specialising in inpainting [14, 18, 91, 49], it has become possible to explore what a potential inpainting result might look like, without physically changing the painting.

Although image inpainting algorithms are not a novel development [9, 4], recent work has shown that approaches based on Convolutional Neural Networks (CNN) are capable of inpainting large missing image regions in a manner which is consistent with the context [91, 128, 127, 49]. Unlike, scene-completion approaches [42], which search for similar patches in a large database of images, CNN-based approaches are capable of *generating* meaningful content [91].

A key aspect of CNN-based inpainting approaches and of many CNN architectures in general [101], is that an image is described at multiple scales by an encoder that reduces the spatial resolution through pooling and downsampling. Each layer (or block of layers) of the network processes the image at a certain scale, and passes this scale-specific information on to the next layer. This encoding process continues until a single low dimensional representation of the image is found, which describes the entire image. Because this architecture resembles a funnel, the final representation is sometimes referred to as *the bottleneck*. Figure 5.2 shows a visualisation of two CNN architectures; one for classification, and one for image generation (similar to an autoen-

Figure 5.1: "*An old woman of Arles*" by Vincent van Gogh (courtesy of the Van Gogh Museum). The left bottom corner was manually inpainted with a 'neutral' colour.

coder). Both architectures encode the image into a bottleneck representation, after which the classification network processes it with a classifier, typically a softmax regression layer [71], and the image generation network feeds it to a decoder [51]. The decoder subsequently performs a number of upsampling steps to generate the output image.

A downside of downsampling in the encoder is the loss of spatial detail - detail which might be crucial to the task [130]. For inpainting this is most prevalent when considering the local consistency [49]; the consistency between the edge of the inpainted region and the edge of the context. A lack of local consistency will result in an obvious transition from the context to the inpainted region. Although increasing the size of the bottleneck, i.e., making it wider, appears to alleviate this to some extent [91], it comes at the cost of a tremendous increase in model parameters. Luckily, recent work has shown that it is possible to encode an image while preserving the spatial resolu-

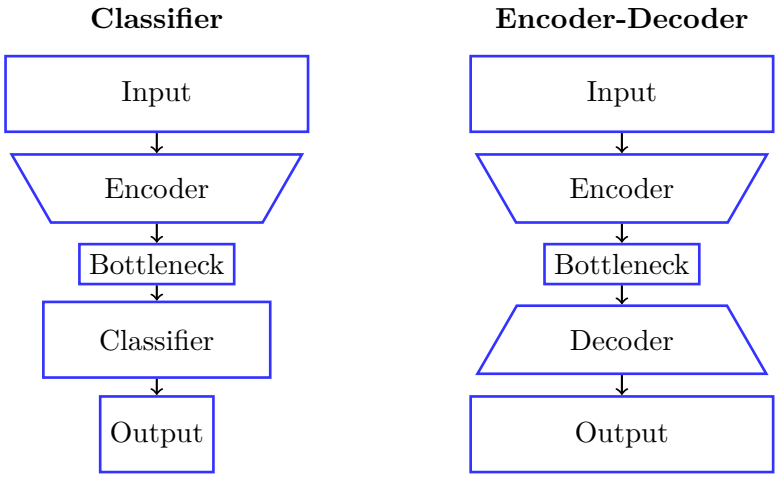


Figure 5.2: Visualisation of a classification CNN architecture (left), and an image generation architecture (right). In both architectures the encoder downsamples the input into a low(er) dimensional representation: *the bottleneck*.

tion [129, 130]. *Dilated convolutions* make it possible to expand the receptive field of a CNN, without downsampling or increasing the number of model parameters. We define the receptive field of a CNN as the size of the region in the input space that affect the output neurons of the encoder. For instance, a single layer CNN with 3×3 filters would have a receptive field of 3×3, adding identical layers on top would increase the receptive field to 5×5, 7×7, etc. We refer to Subsection 5.2.2 for an explanation of how the receptive field of a CNN grows when using dilated convolutions.

Many of the shown results obtained with CNN-based inpainting models, have been achieved using complex architectures with many parameters, resulting in a necessity of large amounts of data, and often long training times [91, 49]. Although simpler architectures have been proposed [128], these are typically only demonstrated on small datasets with relatively little variation (i.e., only faces or only facades of buildings). Therefore, we aim to produce a light-weight inpainting model, which can be applied to large and complex datasets. In this chapter, we demonstrate that using dilated convolutions we can construct a simple model that is able to obtain state-of-the-art performance on various inpainting tasks.

The remainder of this chapter is organised as follows. In Section 5.2 we discuss

related work on inpainting and dilated convolutions. In 5.3 we describe in particular Pixel Context Encoders, the architecture of our model and how it is trained. Section 5.4 describes the experiments and the results we obtain on a variety of inpainting tasks. Lastly, in Section 5.5 we conclude that our model is much less complex than existing models, while outperforming them on benchmark datasets of natural images and paintings.

5.2 Related work

In this section we will discuss work related to image inpainting, dilated convolutions and their application to inpainting, and finally our contributions.

5.2.1 Image inpainting

When a single pixel is missing from an image we can look at the adjacent pixels and average their colour values to produce a reasonable reconstruction of the missing pixel. When a larger region formed by directly adjacent pixels is missing, it is necessary to take into account a larger neighbourhood surrounding the missing region. Moreover, it may become insufficient to only smooth out the colour, to reconstruct the region in a plausible manner. Additionally, for smaller regions it can be sufficient to only incorporate textural or structural information [10], however inpainting larger regions requires understanding of the entire scene [91]. For example, given a picture of a face, if part of the nose is missing it can be reconstructed by looking at the local context and textures. But once the entire nose is missing it requires understanding of the entire face to be able to reconstruct the nose, rather than smooth skin [128].

The challenge of inferring (large) missing parts of images is at the core of image inpainting, the process of reconstructing missing or damaged parts of images [9].

Classical inpainting approaches typically focus on using the local context to reconstruct smaller regions, in this chapter we will focus on recent work using Convolutional Neural Networks (CNN) to encode the information in the entire and inpaint large regions [91, 128, 127, 28, 49]. From these recent works we will focus on two works, first

the work by Pathak et al. [91] who designed the (until now) 'canonical' way of performing inpainting with CNN. Second, we will focus on the work by Iizuka et al. [49], who very recently proposed several extensions of the work by Pathak et al., including incorporating dilated convolutions.

Pathak et al. [91] present Context Encoders (CEs), a CNN trained to inpaint while conditioned on the context of the missing region. CE describe the context of the missing region by encoding the entire image into a bottleneck representation. Specifically, the spatial resolution of the input image is reduced with a factor 128; from 128×128, to the bottleneck representation - a single vector. To compensate for the loss of spatial resolution they increase the width of the bottleneck to be 4000 dimensional. Notably, this increases the total number of model parameters tremendously, as compared to a narrower bottleneck.

CEs are trained by means of a reconstruction loss (L2), and an adversarial loss. The adversarial loss is based on Generative Adversarial Networks (GAN) [38], which involves training a discriminator D to distinguish between real and fake examples. The real examples are samples from the data x, whereas the fake examples are produced by the generator G. Formally the GAN loss is defined as:

$$\min_G \max_D \mathbb{E}_{x \sim p_{data}(x)}[log(D(x) + log(1 - D(G(x))] \tag{5.1}$$

by minimising this loss the generator can be optimised to produce examples which are indistinguishable from real examples. In [91] the generator is defined as the CE, and the discriminator is a CNN trained to distinguish original images from inpainted images.

In a more recent paper, Iizuka et al. [49] propose two extensions to the work by Pathak et al. [91]: (1) They reduce the amount of downsampling by incorporating dilated convolutions, and only downsample by a factor 4, in contrast to Pathak et al. who downsample by a factor 128. (2) They argue that in order to obtain globally and locally coherent inpainting results, it is necessary to extend the GAN framework used

in [91] by using two discriminators. A 'local' discriminator which focuses on a small region centred around the inpainted region, and a 'global' discriminator which is trained on the entire image. Although the qualitative results presented in [49] appear intuitive and convincing, the introduction of a second discriminator results in a large increase in the number of trainable parameters.

Ablation studies presented in a number of works on inpainting have shown that the structural (e.g., L1 or L2) loss results in blurry images [91, 127, 49]. Nonetheless, these blurry images do accurately capture the coarse structure, i.e., the low spatial frequencies. This matches an observation by Isola et al. [51], who stated that if the structural loss captures the low spatial frequencies, the GAN loss can be tailored to focus on the high spatial frequencies (the details). Specifically, Isola et al. introduced PatchGAN, a GAN which focuses on the structure in local patches, relying on the structural loss to ensure correctness of the global structure. PatchGAN, produces a judgement for $N \times N$ patches, where N can be much smaller than the whole image. When N is smaller than the image, PatchGAN is applied convolutionally and the judgements are averaged to produce a single outcome.

Because the PatchGAN operates on patches it has to downsample less, reducing the number of parameters as compared to typical GAN architectures, this fits well with our aim to produce a light-weight inpainting model. Therefore, in our work we choose to use the PatchGAN for all experiments.

Before turning to explanation of the complete model in section 5.3, we first describe dilated convolutions in more detail.

5.2.2 Dilated convolutions

The convolutional layers of most CNN architectures use discrete convolutions. In discrete convolutions a pixel in the output is the sum of the elementwise multiplication between the weights in the filter and a region of adjacent pixels in the input. Dilated or l-dilated convolutions offer a generalisation of discrete convolutions [129] by introducing a dilation factor l which determines the 'sampling' distance between pixels in

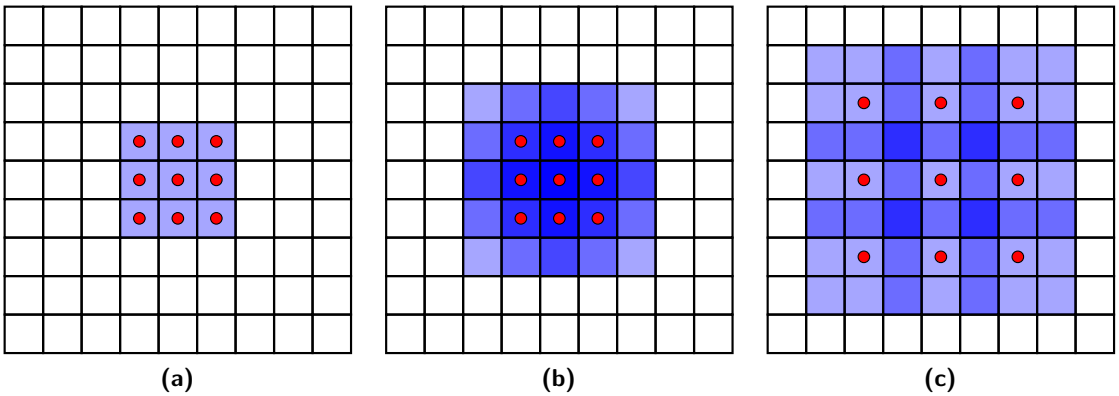


Figure 5.3: Comparison of 1-dilated versus 2-dilated filter. (a) shows the receptive field of a 3×3 1-dilated filter directly on the input. (b) shows the 5×5 receptive field of a 1-dilated 3×3 filter applied to (a). (c) shows the 7×7 receptive field of a 2-dilated 3×3 filter applied to (a). (c) has a larger receptive field than (b), with the same number of parameters.

the input. For $l = 1$, dilated convolutions correspond to discrete convolutions. By increasing the dilation factor, the distance between pixels sampled from the input becomes larger. This results in an increase in the size of the receptive field, without increasing the number of weights in the filter. Figure 5.3 provides a visual illustration of dilated convolution filters.

Recent work has demonstrated that architectures using dilated convolutions are especially promising for image analysis tasks requiring detailed understanding of the scene [129, 130, 49]. For inpainting the aim is to fill the missing region in a manner which is both globally and locally coherent, therefore it relies strongly on a detailed scene understanding. In this work we incorporate lessons learnt from the work by Yu et al. [129, 130] and Iizuka et al. [49] and present a lightweight and flexible inpainting model with minimal downsampling.

5.2.3 Our contributions

In this work we make the following four contributions. (1) We present a light-weight and flexible inpainting model, with an order of magnitude fewer parameters than used in previous work. (2) We show state-of-the-art inpainting performance on datasets of natural images and paintings. (3) While acknowledging that a number of works have

explored inpainting of cracks in paintings [32, 107, 18], we pose that we are the first to explore inpaintings of large regions of paintings. (4) We demonstrate that our model is capable of extending images (i.e., image extrapolation), by generating novel content which extends beyond the edges of the current image.

5.3 Pixel Context Encoders

In this section we will describe our inpainting model: Pixel Context Encoders (PCE). Firstly, we will describe the PCE architecture, followed by details on the loss function used for training.

5.3.1 Architecture

Typically, Convolutional Neural Networks which are used for image generation follow an encoder-decoder type architecture [51]. The encoder compresses the input, and the decoder uses the compressed representation (i.e., the bottleneck) to generate the output. Our PCE does not have a bottleneck, nevertheless we do distinguish between a block of layers which encodes the context (the encoder), and a block of layers which take the encoding of the image and produces the output image, with the missing region filled in (the decoder).

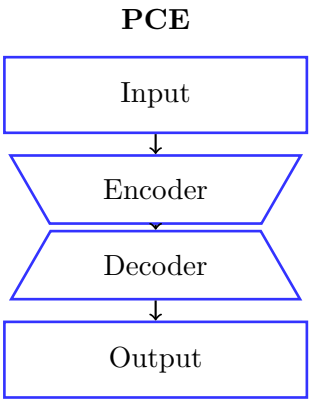


Figure 5.4: Visualisation of the PCE architecture. Unlike traditional encoder-decoder architectures, the PCE architecture does not have a bottleneck. The encoder describes the context (i.e., the input), and the decoder generates the output image.

Table 5.1: Growth of the PCE receptive field (RF) and dilation rate d as a function of the number of layers (depth), with a filter size of 3×3. The first two layers are discrete convolutions with a stride of 2.

Depth	d	RF size
1	1	3×3
2	1	7×7
3	2	23×23
4	4	55×55
5	8	119×119
6	16	247×247

The encoder consists of two downsampling layers, followed by a block of n dilated convolutional layers. The downsampling layers of the encoder are discrete convolutions with a stride of 2. For the subsequent dilated convolution layers, the dilation rate d increases exponentially. The depth of the encoder is chosen such that the receptive field of the encoder is (at least) larger than the missing region, for all of our experiments $n = 4$, resulting in a receptive field of 247×247. Table 5.1 shows how the size of the receptive field grows as more layers are added to the encoder.

By incorporating strided convolutions in the first two layers we follow Iizuka et al. [49] and downsample the images by a factor 4, our empirical results showed that this improves inpainting performance and drastically reduces (5 to 6 times) memory requirements as compared to no downsampling. We pose that the increased performance stems from the larger receptive field, and the local redundancy of images, i.e., neighbouring pixels tend to be very similar. Nonetheless, we expect that stronger downsampling will results in too great of a loss of spatial resolution, lowering the inpainting performance.

The decoder consists of a block of 3 discrete convolutional layers which take as input the image encoding produced by the encoder. The last two layers of the decoder are preceded by a nearest-neighbour interpolation layer, which upsamples by a factor 2, restoring the image to the original resolution. Additionally, the last layer maps the image encoding back to RGB space (i.e., 3 colour channels), after which all pixels which were not missing are restored to the ground-truth values.

All convolutional layers in the encoder and decoder, except for the last decoder

layer, are followed by a Batch Normalisation layer [50]. The activation functions for all convolutional layers in the encoder and decoder are Exponential Linear Units (ELU) [17].

5.3.2 Loss

PCEs are trained through self-supervision; an image is artificially corrupted, and the model is trained to regress back the uncorrupted ground-truth content. The PCE F takes an image x and a binary mask M (the binary mask M is one for masked pixels, and zero for the pixels which are provided) and aims to generate plausible content for the masked content $F(x, M)$. During training we rely on two loss functions to optimise the network: a L1 loss and a GAN loss. For the GAN loss we specifically use the PatchGAN discriminator introduced by Isola et al. [51].

The **L1 loss** is masked such that the loss is only non-zero inside the corrupted region:

$$\mathcal{L}_{L1} = \|M \odot (F(x, M) - x)\|_1 \tag{5.2}$$

where $\odot$ is the element-wise multiplication operation.

Generally, the **PatchGAN loss** is defined as follows:

$$\min_G \max_D \mathbb{E}_{x \sim p_{data}(x)}[log(D(x) + log(1 - D(G(x))] \tag{5.3}$$

where the discriminator D aims to distinguish real from fake samples, and the generator G aims to fool the discriminator. For our task we adapt the loss to use our PCE as the generator:

$$\begin{aligned} \mathcal{L}_{GAN} = \min_F \max_D \mathbb{E}_{x \sim p_{data}(x)}[log(D(x) + \\ log(1 - D(F(x, M))], \end{aligned} \tag{5.4}$$

our discriminator is similar to the global discriminator used in [49], except that we restore the ground-truth pixels before processing the generated image with the discrim-

inator. This allows the discriminator to focus on ensuring that the generated region is consistent with the context.

The overal loss used for training thus becomes:

$$\mathcal{L} = \lambda \mathcal{L}_{L1} + (1 - \lambda)\mathcal{L}_{GAN} \tag{5.5}$$

where λ is fixed at 0.999 for all experiments, following [91].

5.4 Experiments

To evaluate the performance of our PCE we test it on a number of datasets and variations of the inpainting task. In this section we will describe the datasets and the experimental setting used for training, the results of image inpainting on 128×128 and 256×256 images, and lastly the image extrapolation results.

All results are reported using the Root Mean Square Error (RMSE) and Peak Signal Noise Ratio (PSNR) between the uncorrupted ground truth and the output produced by the models.

5.4.1 Datasets

ImageNet. As a set of natural images we use the subset of $100,000$ images that Pathak et al. [91] selected from the ImageNet dataset [96]. The performance is reported on the complete ImageNet validation set consisting of $50,000$ images.

PaintersN. The "*Painters by Numbers*" dataset (PaintersN) as published on Kaggle* consists of $103,250$ photographic reproductions of artworks by well over a thousand different artists. The dataset is divided into a training set ($93,250$ images), validation set (5000 images), and test set (5000 images) used for training the model, optimising hyper-parameters, and reporting performances, respectively.

For both datasets all images were scaled such that the shortest side was 256 pixels, and then a randomly located 256×256 crop was extracted.

*https://www.kaggle.com/c/painter-by-numbers

5.4.2 Experimental settings

In this section the details on the settings of the hyperparameters and training procedure are provided. The layers of the encoder and the decoder consist of 128 filters with spatial dimensions of 3×3 for all experiments in this work. All dilated layers were initialised using identity initialisation cf. [129], which sets the weights in the filter such that initially each layer simply passes its input to the next. All discrete convolutional layers were initialised using Xavier initialisation [34].

The PatchGAN discriminator we used consists of 5 layers of filters with spatial dimensions of 3×3, using LeakyReLU as the activation function ($\alpha = 0.2$). For the first 4 layers the number of filters increases exponentially (i.e., 64, 128, 256, 512), the 5th layer outputs a single channel, the real/fake judgement for each patch in the input.

The network was optimised using ADAM [70] until the L1 loss on the training set stopped decreasing. We were able to consider the training loss as we found that there was no real risk of overfitting. Probably, this is due to the low number of model parameters. The size of the minibatches varied depending on memory capabilities of the graphics card.

All images were scaled to the target resolution using bilinear interpolation when necessary. During training the data was augmented by randomly horizontally flipping the images.

Using the hyperparameter settings specified above, our PCE model has significantly fewer model parameters than previously presented inpainting models. Table 5.2 gives an overview of the model parameters[†] for the most relevant models. Clearly, the number of parameters of the PCE model is much smaller than those of comparable methods.

[†]At the time of writing the exact implementation by Iizuka et al. was not available, therefore we calculated the number of parameters based on the sizes of the weight matrices given in [49], thus not counting any bias, normalisation, or additional parameters.

Table 5.2: Number of parameters for the generators and discriminators of the inpainting models by Pathak et al. [91], Iizuka et al. [49], and ours.

Model	# Generator	# Discriminator
CE [91]	71,130,531	2,766,529
Iizuka et al. [49]	6,061,600	29,322,624
PCE	1,041,152	1,556,416

5.4.3 Region inpainting

A commonly performed task to evaluate inpainting, is region inpainting [91, 128, 127]. Typically, in region inpainting a quarter of all the pixels are removed by masking the centre of the image (i.e., centre region inpainting). This means that for a 256×256 image the central 128×128 region is removed. A variant of centre region inpainting is random region inpainting where the missing region is not fixed to the centre of the image, but is placed randomly. This requires the model to learn to inpaint the region independently of where the region is, forcing it to be more flexible.

In this section we will first present results of centre region image inpainting on 128×128 images, followed by the results of centre and random region inpainting on 256×256 images.

To evaluate the centre-region inpainting performance of our PCE model we compare it against the performance of the CE model by Pathak et al. [91]. For this reason, we initially adopt the maximum resolution of the model of Pathak et al., i.e., 128×128, subsequent results will be presented on 256×256 images. For the 128×128 ImageNet experiments we use the pretrained model release by Pathak et al. For the 128×128 results on the PaintersN dataset we have trained their model from scratch.

The model by Pathak et al. [91] uses an overlap (of 4 pixels) between the context and the missing region. Their intention with this overlap is to improve consistency with the context, but as a consequence it also makes the task slightly easier, given that the masked region shrinks by 4 pixels on all sides. For all centre region inpainting experiments we also[‡] add a 4 pixel overlap between the context and the missing region,

[‡]PCE do not require this overlap to achieve a smooth transition between the context and the missing region. Nonetheless we incorporate to make it a fair comparison.

Table 5.3: Centre region inpainting results on 128×128 images with a 64×64 masked region. RMSE and PSNR for models trained on the ImageNet and PaintersN datasets (horizontally), and evaluated on both datasets (vertically).

		ImageNet		PaintersN	
Trained on	Model	RMSE	PSNR	RMSE	PSNR
Imagenet	CE [91]	43.12	15.44	40.69	15.94
	PCE	**22.88**	**20.94**	**22.53**	**21.08**
PaintersN	CE [91]	43.69	15.32	40.58	15.96
	PCE	24.35	20.40	23.33	20.77

however unlike Pathak et al. we do not use a higher weight for the loss in the overlapping region, as our model is able to achieve local consistency without this additional encouragement.

In Table 5.3 the results on 128×128 images are shown, all models are trained and evaluated on both the ImageNet and dataset the PaintersN dataset, to explore the generalisability of the models. The performance of our PCE model exceeds that of the model by Pathak for both datasets. Nonetheless, both models perform better on the PaintersN dataset, implying that this might be an easier dataset to inpaint on. Overall, our PCE model trained on the $100,000$ image subset of ImageNet performs best, achieving the lowest RMSE and highest PSNR on both datasets.

Additionally, in Figures 5.5 and 5.6 we show examples of centre region inpainting on the ImageNet and PaintersN datasets, respectively. Qualitatively, our PCE model appears to generate content which is less blurry and more consistent with the context. Obviously, both models struggle to recover content which was not available in the input, but when not considering the ground truth, and only the generated output, we observe that our PCE model produces more plausible images.

As our PCE model is capable of inpainting images larger than 128×128, we show results on 256×256 images, with a 128×128 missing region in Table 5.4. Additionally, in this table we also show random region inpainting results. The random region inpainting models were trained without overlap between the context and the missing region.

The results in Table 5.4 not only show that our model is capable of inpainting images at a higher resolution, they also show that randomising the location of the miss-

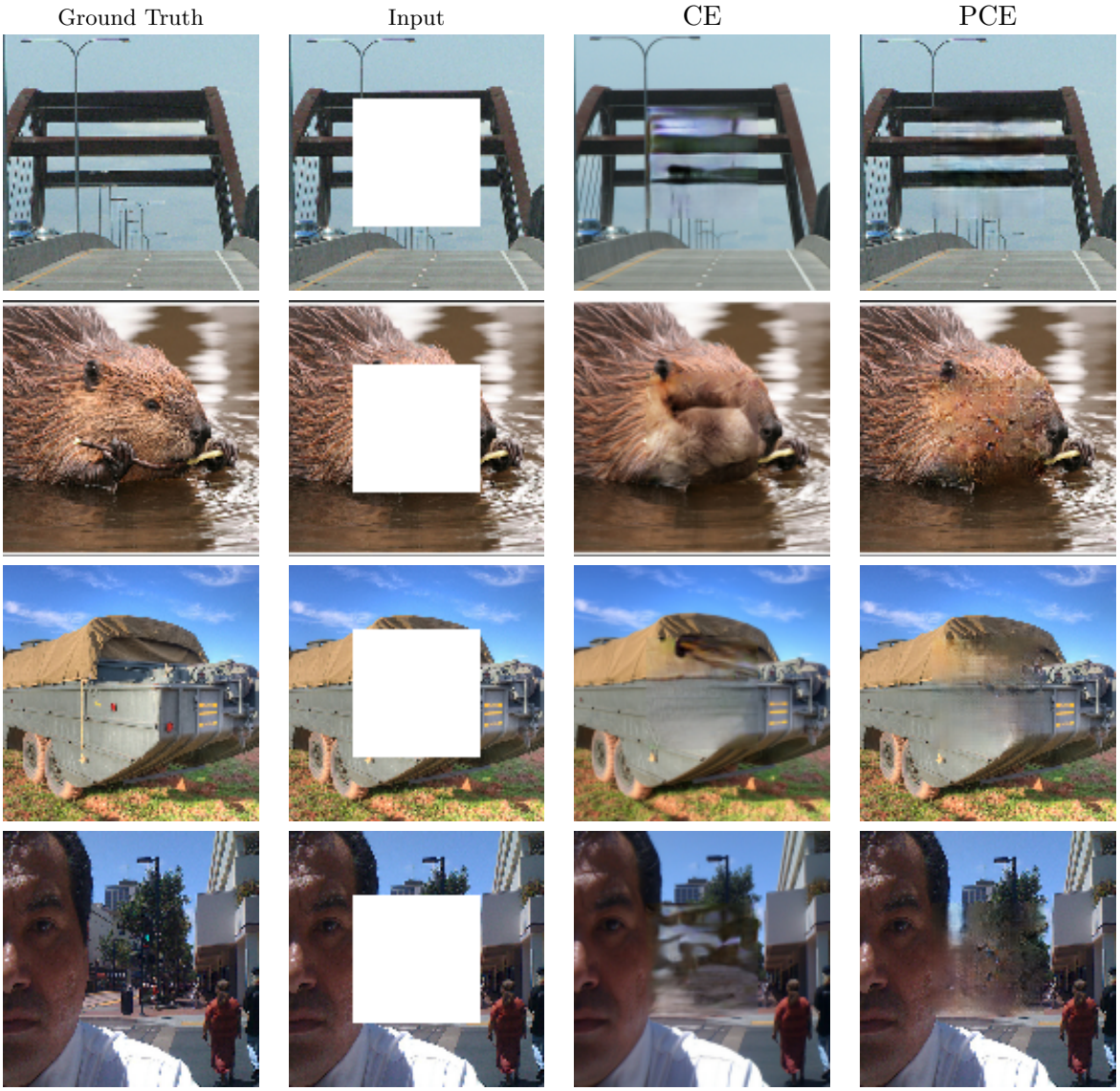


Figure 5.5: Comparison between CE [91] and our PCE model, on inpainting a 64×64 region in 128×128 images taken from the ImageNet validation set.

ing region only has a minimal effect on the performance of our model. Although all results were obtained by training a model specifically for the task, we note that no changes in model configuration were necessary to vary between tasks.

In Figure 5.7 we show several centre region inpainting examples generated by our PCE model on 256×256 images.

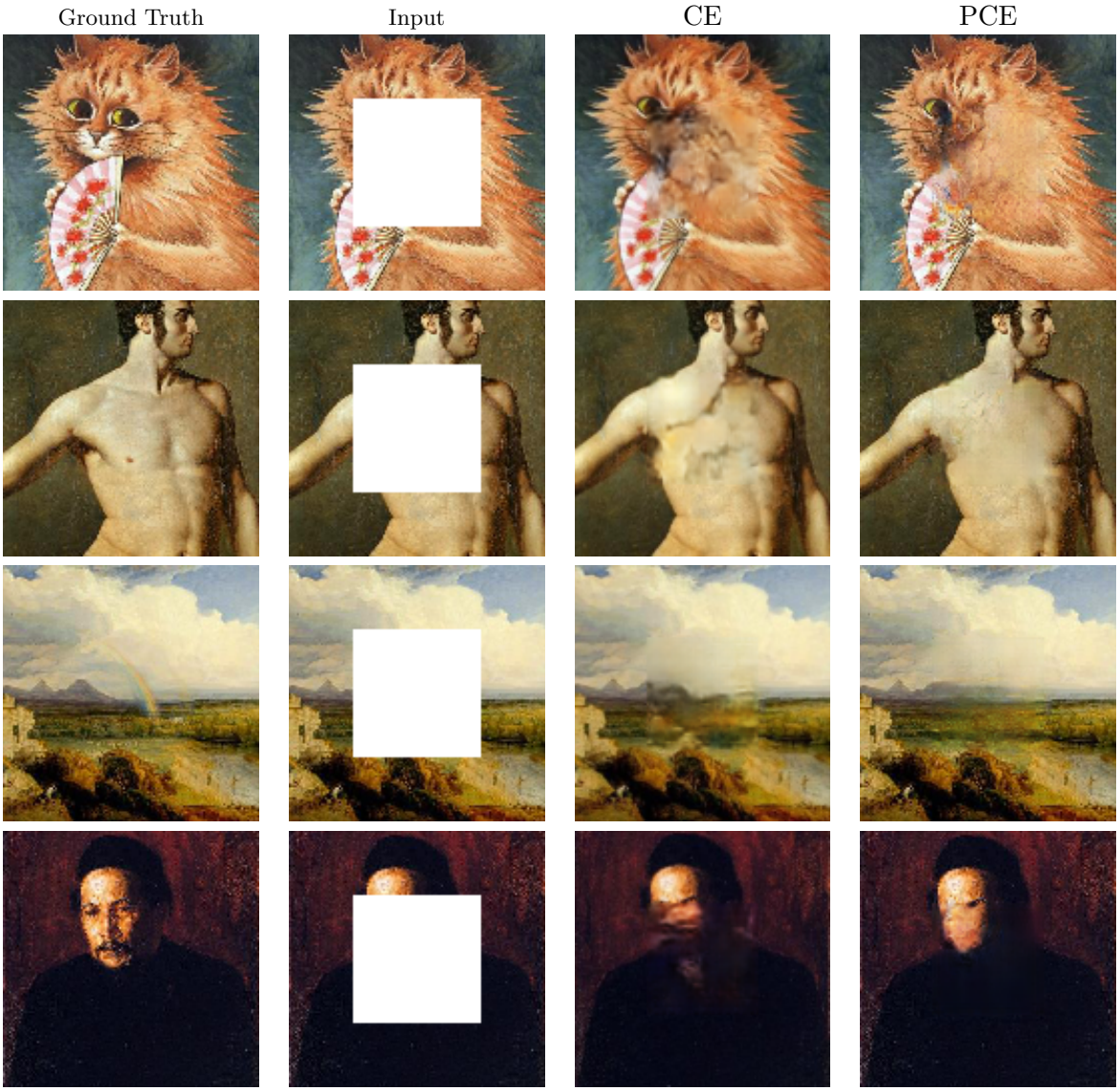


Figure 5.6: Comparison between CE [91] and our PCE model, on inpainting a 64×64 region in 128×128 images taken from the PaintersN test set.

5.4.4 Image extrapolation

In this section, we explore image extrapolation; generating novel content beyond the image boundaries. By training a PCE to reconstruct the content on the boundary of an image (effectively inverting the centre region mask), we are able to teach the model to extrapolate images. In Table 5.5 we show the results of image extrapolation

Table 5.4: Centre and random region inpainting results for PCE on 256×256 images with a 128×128 masked region. RMSE and PSNR for models trained on the ImageNet and PaintersN datasets (horizontally), and evaluated on both datasets (vertically). The first two rows are for centre region inpainting, the last two for random region inpainting.

		ImageNet		PaintersN	
Region	Trained on	RMSE	PSNR	RMSE	PSNR
Centre	Imagenet	24.36	20.40	23.87	20.57
	PaintersN	24.99	20.17	23.41	20.74
Random	Imagenet	24.62	20.30	24.20	20.45
	PaintersN	25.13	20.13	24.06	20.51

Table 5.5: Image extrapolation results for PCE on 256×256 images based on a provided 192×192 centre region. RMSE and PSNR for models trained on the ImageNet and PaintersN datasets (horizontally), and evaluated on both datasets (vertically).

	ImageNet		PaintersN	
Trained on	RMSE	PSNR	RMSE	PSNR
Imagenet	31.81	18.08	32.82	17.81
PaintersN	32.67	17.85	32.39	17.92

obtained by only providing the centre 192×192 region of 256×256 images, aiming to restore the 64 pixel band surrounding it. For our region inpainting experiments we corrupted $\frac{1}{4}$th of the pixels, whereas for this task $\frac{9}{16}$th of the pixels are corrupted. Despite the increase in size of the reconstructed region, the difference in performance is not very large, highlighting the viability of image extrapolation with this approach.

In Figure 5.8 we show four examples obtained through image extrapolation. Based on only the provided input our PCE is able to generate novel content for the 64 pixel band surrounding the input. Although the output does not exactly match the input, the generated output does appear plausible.

Additionally, in Figure 5.9 we show images obtained by applying the PCE trained for image extrapolation to uncorrupted images, resized to 192×192 pixels. By resizing the images to the resolution of the region the model was train on, the model will generate a band of 64 pixels of novel content, for which there is no ground truth.

5.5 Conclusion

In this chapter we presented a novel inpainting model: Pixel Content Encoders (PCE), by incorporating dilated convolutions and PatchGAN we were able to reduce the com-

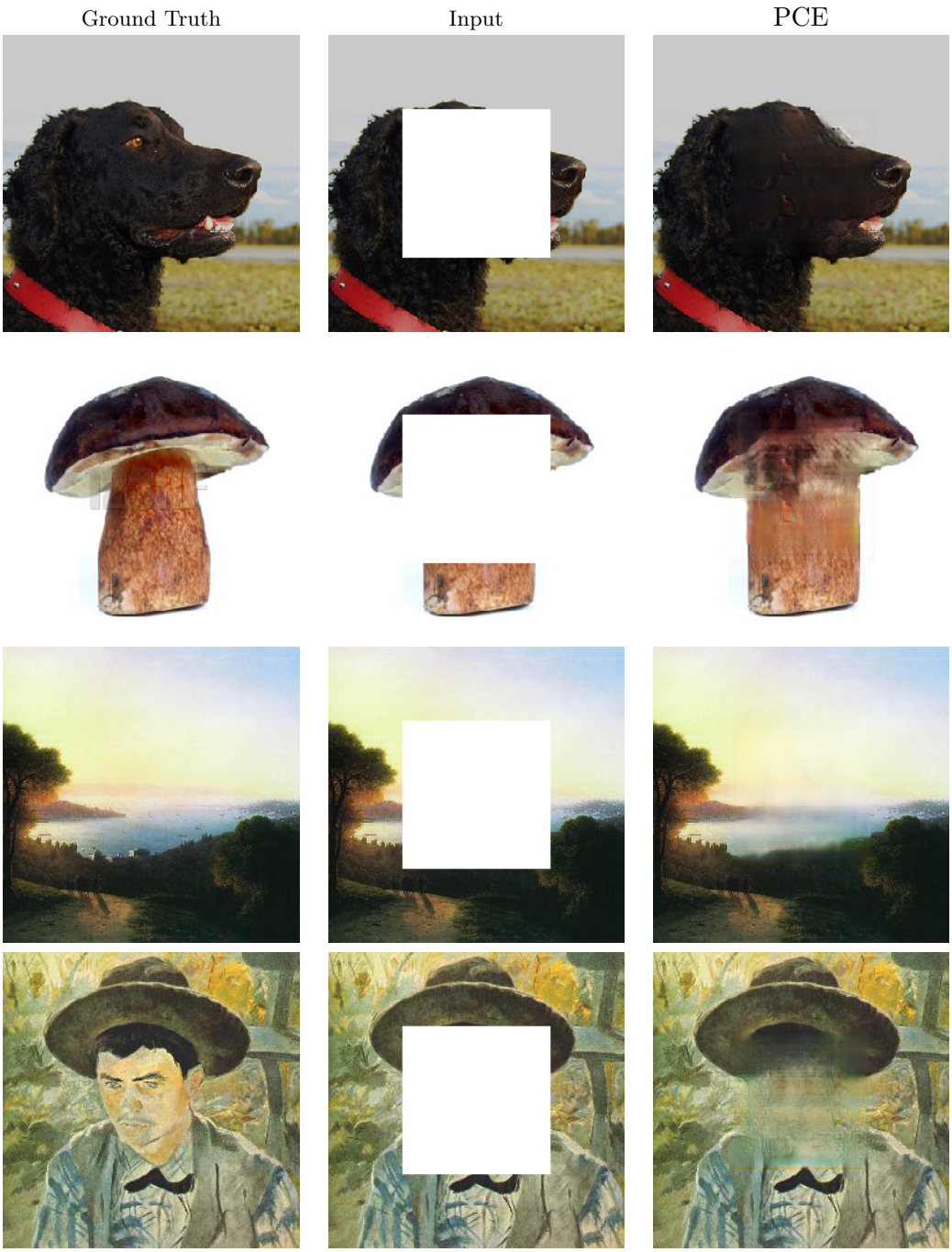


Figure 5.7: Examples produced by our PCE model, on inpainting a 128×128 region in 256×256 images taken from the ImageNet validation set in the first two rows, and PaintersN test set in the last two rows.

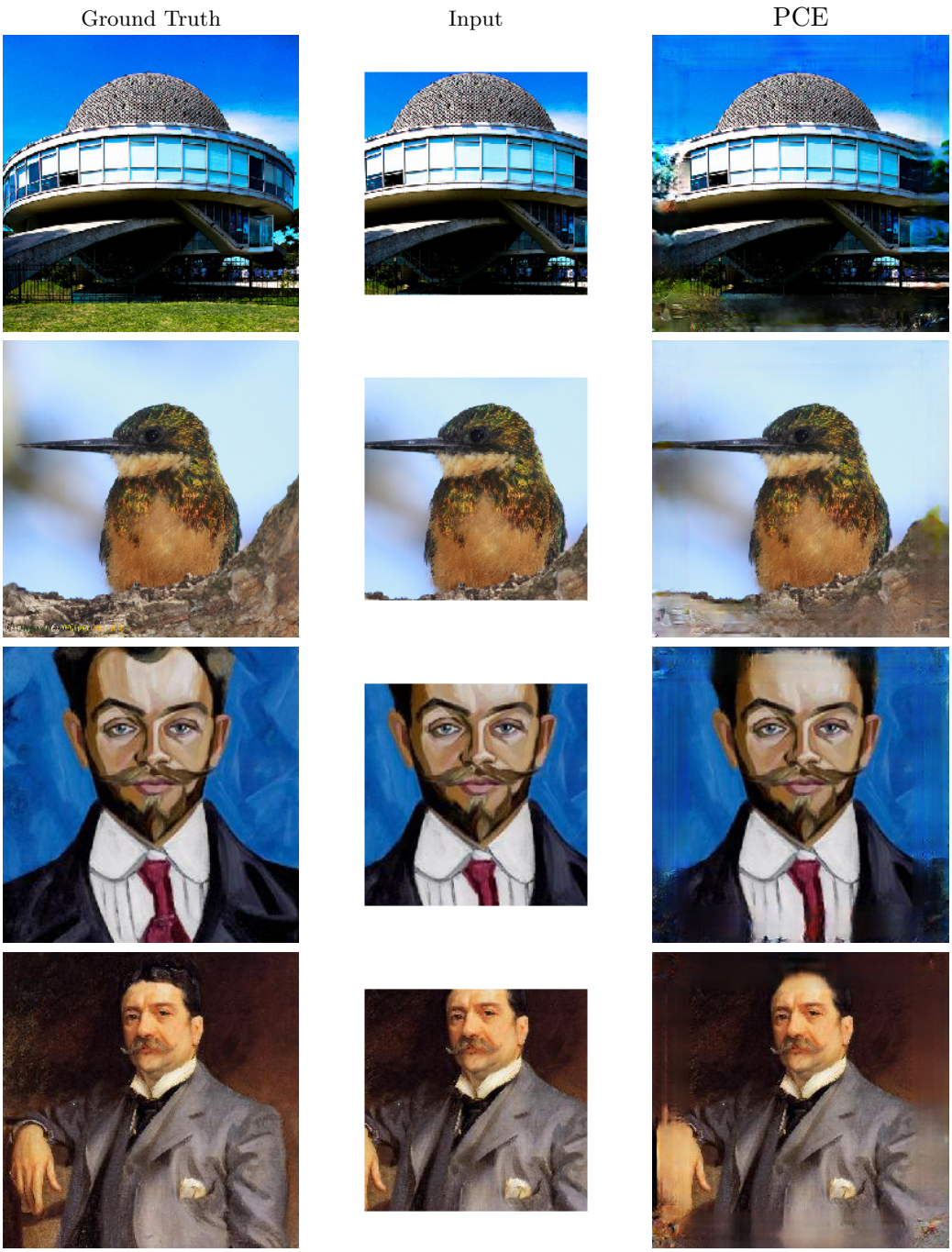


Figure 5.8: Examples produced by our PCE model, on extrapolating 192×192 regions taken from 256×256 images from the ImageNet validation set in the first two rows, and PaintersN test set in the last two rows.

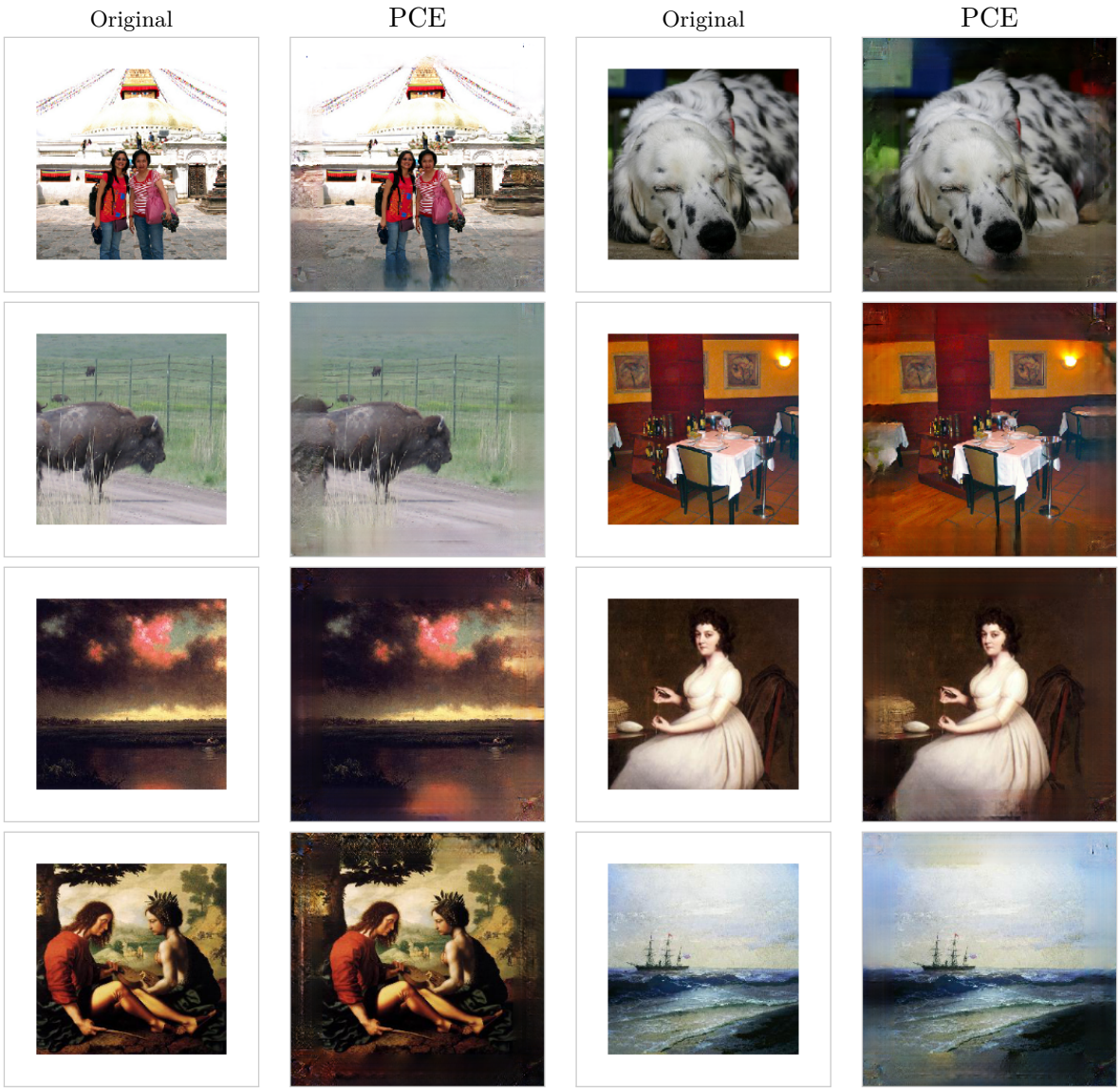


Figure 5.9: Examples produced by our PCE model, on extrapolating 192×192 images beyond their current boundaries. The images in the first two rows are from the ImageNet validation, and in the last two rows from the PaintersN test set.

plexity of the model as compared to previous work. Moreover, by incorporating dilated convolutions PCE are able to preserve the spatial resolution of images, as compared to encoder-decoder style architectures which lose spatial information by compressing the input into 'bottleneck' representations.

We trained and evaluated the inpainting performance of PCE on two datasets of

natural images and paintings, respectively. The results show that regardless of the dataset PCE were trained on they outperform previous work on either dataset, even when considering cross-dataset performance (i.e., training on natural images and evaluating on paintings, and vice versa). Based on the cross-dataset performance we pose that PCE solve the inpainting problem in a largely data-agnostic manner. By encoding the context surrounding the missing region PCE are able to generate plausible content for the missing region in a manner that is coherent with the context.

The approach presented in this chapter does not explicitly take into account the artist's style. However, we would argue that the context reflects the artist's style, and that generated content coherent with the context is therefore also reflects the artist's style. Future research on explicitly incorporating the artist's style is necessary to determine whether it is beneficial for inpainting on artworks to encode the artist's style in addition to the context.

We conclude that PCE offer a promising avenue for image inpainting and image extrapolation. With an order of magnitude fewer model parameters than previous inpainting models, PCE obtain state-of-the-art performance on benchmark datasets of natural images and paintings. Moreover, due to the flexibility of the PCE architecture it can be used for other image generation tasks, such as image extrapolation. We demonstrate the image extrapolation capabilities of our model by restoring boundary content of images, and by generating novel content beyond the existing boundaries.

6

Conclusion

In this chapter we bring together the answers to the two research questions investigated in Chapters 2-3, and 4-5. Second, we address the problem statement, which was the foundation for the research conducted in this thesis. Lastly, we present - what we consider to be - promising directions for future work.

6.1 Answers to the research questions

In the research presented in this thesis, we addressed two research questions. In this section we will answer each of the research questions by referring to the results and conclusions from the associated chapters.

Research question 1 (RQ1): Is it possible to learn a representation of the artist's style, which can be used to recognise the style of the artist across multiple artworks?

To answer RQ1, in Chapter 2 we performed an initial exploration of the feasibility of representation learning for recognising the artist's style across artworks. Subsequently, in Chapter 3 we addressed a limitation of the work presented in Chapter 2, resulting in the definitive answer to RQ1.

In Chapter 3 we presented the first study on representation learning for artist attribution. While previous work aimed to use existing feature extractors to attribute artists to artworks [63, 47, 112], we showed that it is possible to learn representations - directly from artwork data - which can be used to attribute artist, to a high degree of accuracy, to the correct artwork. Specifically, we presented a Convolutional Neural Network (PigeoNET), which is trained in a supervised manner to attribute artist to the correct artwork. We explore the effectiveness of PigeoNET in two different datasets configurations, which differ with respect to the variety of types of artworks (e.g., pocelain, paintings, drawings): a homogeneous dataset and heterogeneous dataset. Our results show that PigeoNET performs best on a homogeneous dataset, with only a single type of artwork.

Additionally, we explored the influence of the size of the data on the performance

of PigeoNET. Unsurprisingly, our results showed that more data is beneficial to the performance, nonetheless, even in very data-poor settings PigeoNET performs well above chance level. To summarise, in Chapter 2 we demonstrated that it is possible to learn representations of the artist's style, and to recognise it across artworks up to a high degree of accuracy.

Although promising, the performance obtained in Chapter 2 disregarded an important source of information, namely: scale. Artworks have highly varying physical dimensions, but not all visual characteristics present in an artwork scale proportionally to these physical dimensions. For instance, by using a larger piece of canvas we do not enlarge the weave patterns we might observe when viewing the artwork from up close. Yet, by resizing images of artworks to a uniform size we lose such scale-specific information [35]. In Chapter 3 we thus expanded on the work presented in Chapter 2 by addressing multi-scale artwork analysis.

The importance of scale is well-known in computer vision [80], yet previous work on deep learning has only aimed to take advantage of it by means of simple data augmentation, to learn features which are (slightly) more robust to scale variations [111, 105, 33, 124]. In doing so, this work does not make use of scale-specific information. In Chapter 3 we presented an approach for learning scale-specific representations, and combining these to obtain a single multi-scale representation. Our results showed that combining multiple scales-specific representations results in a greater artist attribution ability, as compared to single-scale approaches. The model presented in Chapter 3 advanced the state-of-the-art for automatic artist attribution, outperforming both previous work using learnt representations, and work using engineered features.

Based on the studies presented in Chapters 2 and 3 we may conclude that in principle it is possible to learn representations which can be used to recognise the style of the artist across multiple artworks. Learnt representations do not (yet) enable perfect recognition, nonetheless we demonstrated that representation learning is a fruitful approach to recognising the artist's style.

Research question 2 (RQ2): Can we generate novel image content in the style of the artist?

To answer RQ2, we performed two studies which aim to generate novel image content. First, in Chapter 4 we investigated generating novel image colour in the style of the artist. Followed by Chapter 5, where we investigate restoring missing parts of images of artworks through inpainting. Before answering RQ2 we will first briefly summarise the findings of Chapters 4 and 5.

The palette of an artist is an important part of their style. How an artist decides to colour a certain image region or object is a stylistic choice. Being able to replicate these choices, by colouring an image in a manner which matches how the artist would have coloured it, demonstrates understanding of the artist's style. In Chapter 4 we investigated learning representations for artist-specific colourisation.

Previous work on image colourisation has shown that CNNs are capable of automatically colouring images [72, 133, 23, 48, 51]. Specifically, through self-supervision by removing the colours from an image a CNN can be trained to restore the ground-truth colours. By training through self-supervision, the CNN learns to associate the semantics of image regions with the ground truth colours, allowing it to produce plausible colours for unseen images.

Yet, for artworks the colours of an image are also determined by the artist's style, and not just the image semantics. Therefore, in Chapter 4 we presented an approach for learning representations of artist-specific colourisation. Our results showed that we are able to create plausible colourisations of artworks, and that performing the colourisation in an artist-specific manner appears beneficial. Moreover, we showed that our model can also use its understanding of the style from one artist, and use it to colour artworks by other artists, demonstrating the diversity in colourisation our model can produce.

In Chapter 5 we investigated restoring large missing parts of artworks and natural images through inpainting. Retouching, or inpainting, is common practice in the

cultural heritage domain. Typically, the aim is only to inpaint small regions, for instance to hide small defects [9]. Yet, for paintings through accidental or natural causes it might be necessary to inpaint larger regions. Moreover, inpainting larger regions requires a greater understanding of the image and the area surrounding the missing region (i.e., the context) [91].

Previous work on inpainting using CNNs trained with self-supervision has shown that it is possible to inpaint large missing regions in natural images [91, 128, 127, 49]. Yet, this has not been explored for artworks, whereas for demonstrating the potential of generating image content in the style of the artist it is a highly suited task. Therefore, in Chapter 5 we not only explored inpainting on natural images, but also on paintings. Additionally, we presented a novel inpainting model, which has an order of magnitude fewer trainable parameters than previous work. Our results showed that our inpainting model advances the state-of-the-art for image inpainting, and can plausibly inpaint missing image regions.

Based on the studies presented in Chapters 4 and 5 we may conclude that in principle we can generate image content in the style of the artist. Although we have not exhaustively explored all aspects of image content generation, we show that based on a reference image (i.e., a colourless image, or the context surrounding missing area) we are able to add novel content to the reference image. Moreover, recent work on image translation has shown that images in the style of the artist can be produced when using a photograph as a reference image [30, 135]. Therefore, based on our work and recent developments we may conclude that it is possible to generate novel image content based on references images.

6.2 Answer to the problem statement

Based on the answers to our research questions we can now formulate an answer to the problem statement.

PS: *To what extent can the artist's style be represented in a digital manner?*

From our results we may conclude that the artist's style can be represented in a digital manner, to an extent that enables: (1) the recognition of the artist's style across multiple artworks to a higher degree of accuracy than previously possible, and (2) the generation of novel image content in the style of the artist.

Based on the answer to the problem statement we can now reflect on how our research contributed to the main goal mentioned in Chapter 1, namely the goal of realising the ability for connoisseurship in a computer, and giving the computer the ability to produce artworks in the style of the artist. With respect to the former part of the goal: our results in Chapters 2 and 3 show that a computer can attribute an artwork to the correct artist in up to 80% of the cases. Nevertheless, we do not argue that this qualifies as connoisseurship, as our system only uses visual information, and has very limited exposure to potential distractors, such as students in the artist' workshop or forgers. Yet, the work we presented is a step in the direction of connoisseurship, and may eventually contribute to a computational system that truly qualifies as an art connoisseur.

With respect to the latter part of the goal, in Chapters 4 and 5 we demonstrated that given reference material our models can add colour to colourless artworks, and inpaint missing regions of artworks. But, this does not fully cover what we mean by producing an artwork, as this implies creating truly novel artworks from scratch. Nonetheless, the contributions we make might someday become components in a system that can produce a new Van Gogh.

6.3 Future work

We identify three main directions for future work on art analysis: (1) multi-task learning, (2) stylistic changes, and (3) alternative types of imaging data. In what follows we will discuss each of these directions.

First, while learning a representation specifically for a certain task can be - as we have shown - very fruitful, there is not sufficient data available to learn a representation for every task. Specifically, for classification tasks it is necessary to have labelled

data, which might only be available in small amounts. However, if we are able to learn more general representations on tasks for which there is sufficient data, we might be able to re-use these representations for other data-poor tasks. Therefore, future work should address multi-task learning. By learning representations on multiple tasks at the same time, we might learn more general representations. In [109] this is explored for four recognition tasks, but ideally multi-task settings for learning representations of the artist's style would include both recognition and generation tasks.

Second, artists might go through a number of stylistic changes throughout their oeuvre. For example, Vincent van Gogh changed his style repeatedly (e.g., his earlier work made in Nuenen differs strongly from his later work in Paris). These changes are an important component of the artist's style which should be addressed by future work. Specifically, future work might focus on incorporating these stylistic changes into a single representation, or by learning a representation that captures the temporally unfolding stylistic development. Alternatively, future work could focus on disentangling artist characteristics which do not change over time from characteristics that are defining for a certain time-period, enabling more fine-grained attribution.

Third, in this work the original format of the data were always RGB images obtained by photographing artworks. However, there are many more imaging methods beyond just photography. Such methods include X-ray [22], multispectral [134] and hyperspectral [77] imaging, and tomography. While many of these methods are being used for art investigation, there are no public, large-scale datasets available yet. Future work should focus on collecting, sharing, and analysing alternative types of imaging data, such that we might learn representations which enable applications such as pigment identification, digital artificial aging, and more accurate artist attribution.

This thesis has provided foundational aspects of such future work by demonstrating the feasibility of representation learning for art analysis.

References

[1] P. Abry, H. Wendt, and S. Jaffard. When Van Gogh meets Mandelbrot: Multifractal classification of painting's texture. *Signal Processing*, 93(3):554–572, mar 2013.

[2] E. H. Adelson, C. H. Anderson, J. Bergen, P. Burt, and J. M. Ogden. Pyramid methods in image processing. *RCA Engineer*, 29(6):33–41, 1984.

[3] Amsterdams Historical Museum. Jan and Casper Luyken, book illustrators: Father and son Luyken.

[4] C. Barnes, E. Shechtman, A. Finkelstein, D. B. Goldman, and A. Systems. PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing. *ACM Transactions on Graphics*, 28(3):24–1, 2009.

[5] S. Belongie, J. Malik, and J. Puzicha. Shape Matching and Object Recognition Using Shape Contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(24), 2002.

[6] Y. Bengio, A. Courville, and P. Vincent. Representation learning: a review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–828, aug 2013.

[7] Y. Bengio, A. Courville, and P. Vincent. Representation Learning: A Review and New Perspectives. In *International Conference on Learning Representations*, pages 1–30, jun 2014.

[8] I. Berezhnoy, E. Postma, and J. van den Herik. Computer analysis of Van Gogh's complementary colours. *Pattern Recognition Letters*, 28(6):703–709, apr 2007.

[9] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image Inpainting. *ACM SIGGRAPH*, 2000.

[10] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher. Simultaneous structure and texture image inpainting. *IEEE Transactions on Image Processing*, 12(8):882–889, 2003.

[11] H. Bevers, L. Hendrix, W. W. Robinson, and P. Schatborn. *Drawings by Rembrandt and His Pupils: Telling the Difference*. J. Paul Getty Museum, 2009.

[12] M. Brown, G. Hua, and S. Winder. Discriminative learning of local image descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):43–57, jan 2011.

[13] S. L. Bucklow. A Stylometric Analysis of Craquelure. In *Formal Connoisseurship and the Characterisation of Craquelure*, pages 503–521. 1998.

[14] R. H. Chan, J. F. Yang, and X. M. Yuan. Alternating direction method for image inpainting in wavelet domain. *SIAM Journal on Imaging Sciences*, 4(3):807–826, 2011.

[15] G. Charpiat, I. Bezrukov, Y. Altun, and B. Hofmann, Matthias Schölkopf. Machine Learning Methods for Automatic Image Colorization. In R. L. Editor, editor, *Computational Photography: Methods and Applications*, pages 1–27. CRC Press, 2011.

[16] D. Cireşan, U. Meier, J. Masci, and J. Schmidhuber. Multi-column deep neural network for traffic sign classification. *Neural Networks*, 32:333–338, 2012.

[17] D.-A. Clevert, T. Unterthiner, and S. Hochreiter. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). *arXiv preprint*, pages 1–14, 2016.

[18] B. Cornelis, T. Ružić, E. Gezels, A. Dooms, A. Pižurica, L. Platiša, J. Cornelis, M. Martens, M. De Mey, and I. Daubechies. Crack detection and inpainting for virtual restoration of paintings: The case of the Ghent Altarpiece. *Signal Processing*, 93(3):605–619, mar 2013.

[19] B. Cornelis, Y. Yang, J. T. Vogelstein, A. Dooms, I. Daubechies, and D. Dunson. Bayesian crack detection in ultra high resolution multimodal images of paintings. *2013 18th International Conference on Digital Signal Processing, DSP 2013*, pages 1–13, 2013.

[20] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Europe. Visual Categorization with Bags of Keypoints. In *Workshop on Statistical Learning in Computer*

Vision, European Conference on Computer Vision, 2004.

[21] R. Dahl. Automatic colorization. http://tinyclouds.org/colorize/:, 2016.

[22] N. Deligiannis, B. Cornelis, M. R. D. Rodrigues, S. Member, and I. Daubechies. Multi-Modal Dictionary Learning for Image Separation With Application In Art Investigation. *arXiv preprint*, pages 1–13, 2017.

[23] A. Deshpande, J. Lu, M.-c. Yeh, and D. Forsyth. Learning Diverse Image Colorization. *arXiv preprint*, 2016.

[24] A. Drimbarean and P. F. Whelan. Experiments in colour texture analysis. *Pattern Recognition Letters*, 22(10):1161–1167, 2001.

[25] V. Dumoulin, J. Shlens, M. Kudlur, G. Brain, and M. View. A learned representation for artistic style. In *arXiv preprint*, 2016.

[26] M. Eraut. Non-formal learning and tacit knowledge in professional work. *The British Journal of educational Psychology*, 70 (Pt 1):113–136, 2000.

[27] D. Erhan, Y. Bengio, A. Courville, and P. Vincent. Visualizing higher-layer features of a deep network. *Bernoulli*, (1341):1–13, 2009.

[28] R. Gao and K. Grauman. From One-Trick Ponies to All-Rounders: On-Demand Learning for Image Restoration. *arXiv preprint*, 2017.

[29] C. Garcia and M. Delakis. Convolutional face finder: A neural architecture for fast and robust face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11):1408–1423, 2004.

[30] L. A. Gatys, A. S. Ecker, and M. Bethge. A Neural Algorithm of Artistic Style. *arXiv preprint*, pages 3–7, 2015.

[31] T. Gevers and A. W. M. Smeulders. PicToSeek: Combining Color and Shape Invariant Features for Image Retrieval. *IEEE Transactions on Image Processing*, 9(1):102–119, 2000.

[32] I. Giakoumis, N. Nikolaidis, and I. Pitas. Digital image processing techniques for the detection and removal of cracks in digitized paintings. *IEEE Transactions on Image Processing*, 15:178–188, 2006.

[33] R. Girshick. Fast R-CNN. *arXiv preprint*, 2015.

[34] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedfor-

ward neural networks. *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 9:249–256, 2010.

[35] J. Gluckman. Scale variant image pyramids. In *Computer Vision and Pattern Recognition*, 2006.

[36] Y. Gong, L. Wang, R. Guo, and S. Lazebnik. Multi-scale Orderless Pooling of Deep Convolutional Activation Features. *ArXiv*, pages 1–17, mar 2014.

[37] R. C. Gonzalez and R. E. Woods. *Digital Image Processing (3rd Edition).* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006.

[38] I. J. Goodfellow, J. Pouget-abadie, M. Mirza, B. Xu, D. Warde-farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

[39] R. M. Haralick. Statistical and Structural Approaches to Texture. *Proceedings of the IEEE*, 67(5):786–804, 1979.

[40] R. M. Haralick, K. Shanmugan, and I. Dinstein. Textural features for image classifcation. *IEEE transactions on systems, man and cybernetics*, 3(6):610–621, 1973.

[41] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for Object Segmentation and Fine-grained Localization. *Computer Vision and Pattern Recognition (CVPR)*, nov 2015.

[42] J. Hays and A. A. Efros. Scene Completion Using Millions of Photographs. *ACM Transactions on Graphics*, 26(3):1–8, 2007.

[43] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. *ArXiv preprint*, dec 2015.

[44] E. Hendriks and S. Hughes. Van Gogh's brushstrokes: marks of authenticity? *Art, Conservation, and Authenticities: Material, Concept, Context*, (August 1882):57–62, 2008.

[45] L. Hou, D. Samaras, T. Kurc, and Y. Gao. Efficient Multiple Instance Convolutional Neural Networks for Gigapixel Resolution Image Classification. *arXiv preprint*, 2015.

[46] J. Huang, S. R. Kumar, M. Mitra, W.-j. Zhu, and R. Zabih. Image Indexing Using Color Correlograms. In *Computer Vision and Pattern Recognition (CVPR)*,

1997.

[47] J. M. Hughes, D. J. Graham, and D. N. Rockmore. Quantification of artistic style through sparse coding analysis in the drawings of Pieter Bruegel the Elder. *Proceedings of the National Academy of Sciences of the United States of America*, 107(4):1279–1283, jan 2010.

[48] S. Iizuka, E. Simo-serra, and H. Ishikawa. Let there be Color!: Joint End-to-end Learning of Global and Local Image Priors for Automatic Image Colorization with Simultaneous Classification. *ACM Transactions on Graphics*, 35(4):1–11, 2016.

[49] S. Iizuka, E. Simo-serra, and H. Ishikawa. Globally and Locally Consistent Image Completion. *ACM Transactions on Graphics (Proc. of SIGGRAPH 2017)*, 36(4):107:1—-107:14, 2017.

[50] S. Ioffe and C. Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *International Conference on Machine Learning*, pages 448–456. JMLR, 2015.

[51] P. Isola, J.-y. Zhu, T. Zhou, and A. A. Efros. Image-to-Image Translation with Conditional Adversarial Networks. *arXiv preprint*, 2016.

[52] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial Transformer Networks. *Advances in Neural Information Processing Systems*, pages 1–14, 2015.

[53] A. K. Jain and F. Farrokhnia. Unsupervised Texture Segmentation Using Gabor Filters. *Pattern Recognition*, 24:1167–1186, 1991.

[54] A. K. Jain, N. K. Ratha, and S. Lakshmanan. Object detection using Gabor filters. *Pattern Recognition*, 30(2):295–309, 1997.

[55] A. K. Jain and A. Vailaya. Image Retrieval using Color and Shape. *Pattern Recognition*, 29(8):1233–1244, 1995.

[56] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *Computer Vision and Image Understanding*, 2010.

[57] Y. Jia. Caffe: An open source convolutional architecture for fast feature embed-

ding. *http://caffe.berkeleyvision.org*, 2013.

[58] C. Johnson, E. Hendriks, I. Berezhnoy, E. Brevdo, S. Hughes, I. Daubechies, J. Li, E. Postma, and J. Wang. Image processing for artist identification. *IEEE Signal Processing Magazine*, 25(4):37–48, 2008.

[59] C. R. Johnson, E. Hendriks, I. J. Berezhnoy, E. Brevdo, S. M. Hughes, I. Daubechies, J. Li, E. Postma, and J. Z. Wang. Image processing for artist identification. *IEEE Signal Processing Magazine*, 25(4):37 – 48, 2008.

[60] D. Johnson, E. Hendriks, and C. J. Jr. Interpreting canvas weave matches. *Art Matters*, pages 53–61, 2013.

[61] D. Johnson, C. Johnson, A. Klein, W. Sethares, H. Lee, and E. Hendriks. A thread counting algorithm for art forensics. In *2009 IEEE 13th Digital Signal Processing Workshop and 5th IEEE Signal Processing Education Workshop*, pages 679–684, 2009.

[62] J. Johnson, A. Alahi, and L. Fei-fei. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In *European Conference on Computer Vision*, 2016.

[63] C. R. Johnson, Jr., E. Hendriks, I. J. Berezhnoy, E. Brevdo, S. M. Hughes, I. Daubechies, J. Li, E. Postma, and J. Z. Wang. Image processing for artist identification. *IEEE Signal Processing Magazine (Special Section - Signal Processing in Visual Cultural Heritage)*, (25):37–48, 2008.

[64] P. Kammerer, M. Lettner, E. Zolda, and R. Sablatnig. Identification of drawing tools by classification of textural and boundary features of strokes. *Pattern Recognition Letters*, 28(6):710–718, 2007.

[65] A. Kanazawa and D. Jacobs. Locally Scale-Invariant Convolutional Neural Networks. pages 1–11.

[66] S. Karayev, M. Trentacoste, H. Han, A. Agarwala, T. Darrell, A. Hertzmann, and H. Winnemoeller. Recognizing Image Style. In *British Machine Vision Conference (BMVC)*, nov 2014.

[67] F. S. Khan, S. Beigpour, J. van de Weijer, and M. Felsberg. Painting-91: A large scale database for computational painting categorization. *Machine Vision and Ap-*

plications, 25(6):1385–1397, jun 2014.

[68] S. R. Kheradpisheh, M. Ghodrati, M. Ganjtabesh, and T. Masquelier. Deep Networks Resemble Human Feed-forward Vision in Invariant Object Recognition. *arXiv preprint*, 2015.

[69] W.-y. Kim and Y.-s. Kim. A region-based shape descriptor using Zernike moments. *Signal Processing: Image Communication*, 16:95–102, 2000.

[70] D. P. Kingma and J. L. Ba. Adam: a Method for Stochastic Optimization. In *International Conference on Learning Representations*, pages 1–13, 2015.

[71] A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[72] G. Larsson, M. Maire, and G. Shakhnarovich. Learning Representations for Automatic Colorization. In *European Conference on Computer Vision (ECCV)*, 2016.

[73] Q. Le, J. Ngiam, Z. Chen, D. H. Chia, and P. Koh. Tiled convolutional neural networks. *Advances in Neural Information Processing Systems*, pages 1–9, 2010.

[74] Y. LeCun and Y. Bengio. Convolutional Networks for Images, Speech, and Time-Series. *The Handbook of Brain Theory and Neural Networks*, 1995.

[75] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[76] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation Applied to Handwritten Zip Code Recognition, 1989.

[77] S. Legrand, F. Vanmeert, G. van der Snickt, M. Alfeld, W. de Nolf, J. Dik, and K. Janssens. Examination of historical paintings by state-of-the-art hyperspectral imaging methods: from scanning infra-red spectroscopy to computed X-ray laminography. *Heritage Science*, 2(13):1–11, 2014.

[78] K. Lenc and A. Vedaldi. Understanding image representations by measuring their equivariance and equivalence. *Computer Vision and Pattern Recognition*, pages 991–999, 2015.

[79] J. Li, L. Yao, E. Hendriks, and J. Z. Wang. Rhythmic brushstrokes distinguish

van Gogh from his contemporaries: findings via automated brushstroke extraction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(6):1159–76, jun 2012.

[80] T. Lindeberg. Scale-space theory: a basic tool for analyzing structures at different scales. *Journal of Applied Statistics*, 21(1):225–270, 1994.

[81] Y. Liu, D. Zhang, G. Lu, and W.-Y. Ma. A survey of content-based image retrieval with high-level semantics. *Pattern Recognition*, 40(1):262–282, jan 2007.

[82] T. Lombardi. The Classification of Style in Fine-Art Painting. pages 1–9, 2005.

[83] D. G. Lowe. Distinctive image features from scale invariant keypoints. *International Journal of Computer Vision*, 60(2), 2004.

[84] B. Manjunath and W. Ma. Texture Features for Browsing and Retrieval of Image Data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):837–842, 1996.

[85] T. Mensink and J. van Gemert. The Rijksmuseum Challenge: Museum-Centered Visual Recognition. *Proceedings of International Conference on Multimedia Retrieval*, pages 2–5, 2014.

[86] S. M. Nascimento, J. M. Linhares, C. Montagner, C. A. João, K. Amano, C. Alfaro, and A. Bailão. The colors of paintings and viewers' preferences. *Vision Research*, 130:76–84, 2017.

[87] A. Odena, V. Dumoulin, and C. Olah. Deconvolution and Checkerboard Artifacts. http://distill.pub/2016/deconv-checkerboard/, 2016.

[88] M. Ogden, H. Adelson, R. Bergen, and J. Burt. Pyramid-based computer graphics. *RCA Engineer*, 1985.

[89] M. Oquab. Weakly supervised object recognition with convolutional neural networks To cite this version : Weakly supervised object recognition with convolutional neural networks. 2014.

[90] D. Park, D. Ramanan, and C. Fowlkes. Multiresolution models for object detection. In *European Conference on Computer Vision*, pages 1–14, 2010.

[91] D. Pathak, J. Donahue, T. Darrell, and A. A. Efros. Context Encoders: Feature Learning by Inpainting. In *Conference on Computer Vision and Pattern Recogni-*

tion, 2016.

[92] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. *European Conference on Computer Vision*, 6314:143–156, 2010.

[93] E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley. Color Transfer between Images. *IEEE CG&A special issue on Applied Perception*, 21:34–41, 2001.

[94] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. *MIC-CAI*, pages 234–241, 2015.

[95] J. Rush. *Acquiring a concept of Painting style*. PhD thesis, 1974.

[96] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *arXiv preprint*, page 37, 2014.

[97] B. Saleh and A. Elgammal. Large-scale Classification of Fine-Art Paintings: Learning The Right Metric on The Right Feature. *arXiv preprint*, page 21, 2015.

[98] J. Schmidhuber. Deep Learning in Neural Networks: An Overview. Technical report, 2014.

[99] B. Seguin, C. Striolo, and F. Kaplan. Visual Link Retrieval in a Database of Paintings. In *European Conference on Computer Vision (ECCV)*, pages 753–767, 2016.

[100] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. 2013.

[101] P. Sermanet and Y. Lecun. Traffic sign recognition with multi-scale convolutional networks. *Proceedings of the International Joint Conference on Neural Networks*, (SEPTEMBER 2011):2809–2813, 2011.

[102] L. Shamir, T. Macura, and N. Orlov. Impressionism, expressionism, surrealism: Automated recognition of painters and schools of art. *ACM Transactions on Applied Perception*, pages 1–18, 2010.

[103] G. Sharma, W. Wu, and E. N. Dalal. The CIEDE2000 Color-Difference Formula: Implementation Notes, Supplementary Test Data, and Mathematical Observations.

Color Research & Application, 30(1):21–30, 2005.

[104] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. *arXiv preprint*, pages 1–8, dec 2014.

[105] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *ArXiv*, pages 1–14, sep 2015.

[106] A. W. M. Smeulders, S. Member, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-Based Image Retrieval at the End of the Early Years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, 2000.

[107] V. Solanki and A. R. Mahajan. Digital Image Processing Approach for Inspecting and Interpolating Cracks in Digitized Pictures. *International Journal of Recent Trends in Engineering (IJRTE)*, 1:97–99, 2009.

[108] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for Simplicity: The All Convolutional Net. In *Internation Conference on Learning Representations*, pages 1–14, 2015.

[109] G. Strezoski and M. Worring. OmniArt: Multi-task Deep Learning for Artistic Data Analysis. *arXiv preprint*, 2017.

[110] M. J. Swain and D. H. Ballard. Color Indexing. 32:11–32, 1991.

[111] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going Deeper with Convolutions. *arXiv preprint*, pages 1–12, 2014.

[112] R. Taylor, R. Guzman, and T. Martin. Authenticating Pollock paintings using fractal geometry. *Pattern Recognition Letters*, 28(6):695–702, 2007.

[113] M. Turk and A. Pentland. Face Recognition Using Eigenfaces, 1991.

[114] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Instance Normalization: The Missing Ingredient for Fast Stylization. *arXiv preprint*, (2016), 2016.

[115] M. M. van Dantzig. *Pictology. An analytical method for attribution and evaluation of pictures. Ed. by the van Dantzig Foundation.* Brill Leiden, 1973.

[116] L. van der Maaten and R. Erdmann. Automatic Thread-Level Canvas Analysis. *IEEE Signal Processing Magazine*, 38:38–45, 2015.

[117] L. van der Maaten and G. Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[118] L. J. P. van der Maaten and E. O. Postma. Texton-Based Analysis of Paintings. In *SPIE Optical Imaging and Applications*, volume 7798, aug 2010.

[119] L. van Gool, P. Dewaele, and A. Oosterlinck. Texture analysis anno 1983. *Computer Vision, Graphics and Image Processing.*, 29:336–357, 1985.

[120] N. van Noord, E. Hendriks, and E. Postma. Toward Discovery of the Artist's Style: Learning to recognize artists by their artworks. *IEEE Signal Processing Magazine*, 32(4):46–54, 2015.

[121] L. Van Tilborgh, T. Meedendorp, E. Hendriks, D. H. Johnson, R. C. Johnson Jr., and R. G. Erdmann. Weave matching and dating of Van Gogh's paintings: an interdisciplinary approach. *The Burlington Magazine*, 1:112–122, 2012.

[122] S. Watanabe, J. Sakamoto, and M. Wakita. Pigeons' discrimination of paintings by Monet and Picasso. *Journal of the Experimental Analysis of Behavior*, 63(2):165–174, 1995.

[123] A. T. Woollett and A. van Suchtelen. *Rubens and Brueghel: A Working Friendship*. 2006.

[124] R. Wu, S. Yan, Y. Shan, Q. Dang, and G. Sun. Deep Image: Scaling up Image Recognition. *arXiv preprint*, page 12, 2015.

[125] W. Wu, A. M. Moreno, J. M. Tangen, and J. Reinhard. Honeybees can discriminate between Monet and Picasso paintings. *Journal of Comparative Physiology A*, 199(1):45–55, oct 2012.

[126] Y. Xu, T. Xiao, J. Zhang, K. Yang, and Z. Zhang. Scale-Invariant Convolutional Neural Networks. *ArXiv*, 2014.

[127] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and H. Li. High-Resolution Image Inpainting using Multi-Scale Neural Patch Synthesis. *arXiv preprint*, 2017.

[128] R. Yeh, C. Chen, T. Y. Lim, M. Hasegawa-johnson, and M. N. Do. Semantic Image Inpainting with Perceptual and Contextual Losses. *arXiv preprint*, 2016.

[129] F. Yu and V. Koltun. Multi-Scale Context Aggregation by Dilated Convolutions. In *International Conference on Learning Representations*, pages 1–9, 2016.

[130] F. Yu, V. Koltun, I. Labs, and T. Funkhouser. Dilated Residual Networks. *arXiv preprint*, 2017.

[131] M. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. *European Conference on Computer Vision (ECCV)*, 8689:818–833, 2014.

[132] D. Zhang and G. Lu. Review of shape representation and description techniques. *Pattern Recognition*, 37:1–19, 2004.

[133] R. Zhang, P. Isola, and A. A. Efros. Colorful Image Colorization. In *European Conference on Computer Vision (ECCV)*, 2016.

[134] Y. Zhao, R. S. Berns, L. a. Taplin, and J. Coddington. An Investigation of Multispectral Imaging for the Mapping of Pigments in Paintings. In D. G. Stork and J. Coddington, editors, *Proc. of SPIE-IS&T*, volume 6810, feb 2008.

[135] J.-y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

Summary

An artist's style is reflected in their artworks. This style is independent from the content of the artwork, two artworks depicting two vastly different scenes (e.g., a beach scene and a forest scene) both reflect the artist's style. By recognising the style of the artist, experts, and sometimes even laymen, can tell the same artist created both artworks. Replicating this ability for connoisseurship in a computer, and potentially even giving a computer the ability to produce artworks in the style of the artist, is the main goal of this thesis.

To analyse artworks with a computer we can use techniques from the field of computer vision. Traditionally, these techniques rely on handcrafted features to describe the characteristics of an image. However, in recent years the field of computer vision has been revolutionalised by the emergence of deep learning models. These are models which learn from the data a feature representation optimised for a task.

Deep learning models have been shown to be enormously successful on a wide range of computer vision tasks. Therefore, in this work we explore the applications of deep learning models for learning representations of the artist's style. By learning the representation we might discover new visual characteristics of artists, enriching our understanding of the artist's style. Moreover, such a representation might enable new and novel applications. To guide the research in this thesis we formulate the following problem statement:

To what extent can the artist's style be represented in a digital manner?

To address the problem statement we identify two requirements for a useful representation of style: (1) the artist's style can be recognised across multiple artworks, and

(2) the representation can be used to generate novel content that has the stylistic characteristics of the artist. To guide our attempts at answering the problem statement we rephrase these requirements as the following two research questions.

- **Research question 1:** Is it possible to learn a representation of the artist's style, which can be used to recognise the style of the artist across multiple artworks?
- **Research question 2:** Can we generate novel image content in the style of the artist?

The first part of the thesis, Chapters 2 and 3 present two studies which aim to answer the first research question, whereas Chapters 4 and 5 are aimed at the second research question.

Specifically, in Chapter 2 we present a study on artist attribution, and the dataset properties which influence the quality of the learnt representation. For example, the number of artists and the number of artworks per artist might influence how well a model could learn to discriminate between artists. Our findings show that our model is able to learn to recognise artists up to a high degree of accuracy (irrespective of the dataset configuration), and that the discriminating characteristics can be related back to the visual content of the artwork.

In Chapter 3 we expand on the initial work on artist attribution done in Chapter 2 by investigating scale variation and scale-specific information. The distance at which we view an object determines what details are visible. For instance, when viewing a painting from too far away we cannot see the texture of the canvas, however if we are too close we might not see what type of scene is depicted on the painting. Similar limitations apply to a computer, in this chapter we aim to overcome these and present a multi-scale method which is able to integrate information from multiple scales, outperforming single-scale approaches.

Chapter 4 presents a study on image colourisation. Colouring a greyscale image in a plausible manner requires understanding of the contents of the image. For example, if we recognise the sky as being the sky we can colour it blue, and the clouds white.

Similarly, if we recognise the artist we might adapt our palette to match the artist's, which might lead us to colour the sky purple. In this chapter we show that by focusing on the palette of a specific artist we are able to learn a representation of the artist's colour choice, producing artist-specific colourisations of paintings.

In Chapter 5 we investigate inpainting, the task of restoring (large) missing regions in images. For instance, given an image of a damaged painting we might provide a reconstruction of the damaged area. In this chapter we present a model which is capable of restoring missing regions in images of both natural scenes and paintings. By analysing the context surrounding the missing region, the model is capable of generating novel image content, coherent with the context. Our experimental results show that, while simpler, our model outperforms previous inpainting models.

Finally, in Chapter 6 we answer the two research questions and the problem statement. We conclude that the artist's style can be represented by training deep learning models, the artist's style can be represented in a digital manner, empowering applications such as artist attribution, image colourisation, and image inpainting. Following the conclusion, we present three directions of future work.

List of Publications

[1] N. van Noord and E. Postma, A learned representation of artist-specific colourisation, in *2nd ICCV Workshop on e-Heritage*, 2017.

[2] N. van Noord and E. Postma, Learning scale-variant and scale-invariant features for deep image classification, *Pattern Recognition*, vol. 61, pp. 583–592, 2017.

[3] N. van Noord, E. Hendriks, and E. Postma, Toward discovery of the artist's style: Learning to recognize artists by their artworks, *IEEE Signal Processing Magazine*, vol. 32, no. 4, pp. 46–54, 2015.

[4] M. van Zaanen and N. van Noord, Evaluation of selection in context-free grammar learning systems, in *Proceedings of the 12th International Conference on Grammatical Inference*, 2014, pp. 193–206.

[5] N. van Noord and E. Postma, Capturing the gist of colour, in *Netherlands Conference on Computer Vision*, 2014.

[6] C. R. Johnson Jr., P. Messier, W. A. Sethares, A. G. Klein, C. Brown, A. H. Do, P. Klausmeyer, P. Abry, S. Jaffard, H. Wendt, S. Roux, N. Pulstenik, N. van Noord, L. van der Maaten, E. Postma, J. Coddington, L. A. Daffner, H. Murata, H. Wilhelm, S. Wood, and M. Messier, Pursuing automated classification of historic photographic papers from raking light images, *Journal of the American Institute for Conservation*, vol. 53, no. 3, pp. 159–170, 2014.

[7] S. Aussems, B. Goris, V. Lichtenberg, N. van Noord, R. Smetsers, and M. van Zaanen, Unsupervised identification of compounds, in *Annual Belgian-Dutch Conference on Machine Learning (BENELEARN 2013)*, 2013, pp. 18–25.

[8] N. van Noord, E. Postma, and E. Hendriks, Colour-texture analysis of paintings using ICA filter banks, in *Annual Belgian-Dutch Conference on Machine Learning (BENELEARN 2013)*, 2013, pp. 74–81.

[9] P. Messier, R. Johnson, H. Wilhelm, W. A. Sethares, A. G. Klein, P. Abry, S. Jaffard, H. Wendt, S. Roux, N. Pustelnik, N. van Noord, L. van der Maaten, and E. Postma, Automated surface texture classification of inkjet and photographic media, in *NIP & Digital Fabrication*, 2013, pp. 85–92.

[10] T. Avontuur, I. Balemans, L. Elshof, N. van Noord, and M. van Zaanen, Developing a oart-of-speech tagger for Dutch tweets, *Computational Linguistics in the Netherlands Journal 2*, vol. 2, pp. 34–51, 2012.

[11] M. van Zaanen and N. van Noord, Model merging versus model splitting context-free grammar induction, in *Proceedings of the 11th International Conference on Grammatical Inference*, 2012, pp. 224–236.

[12] M. Aminian, T. Avontuur, Z. Azar, I. Balemans, L. Elshof, R. Newell, N. van Noord, A. Ntavelos, and M. van Zaanen, Assigning part-of-speech to Dutch tweets, in *Proceedings of the LREC workshop: @NLP can u tag #user_generated_content?!* s.l.: s.n., 2012, pp. 9–14.

TiCC Ph.D. Series

1. Pashiera Barkhuysen. Audiovisual Prosody in Interaction. Promotores: M.G.J. Swerts, E.J. Krahmer. Tilburg, 3 October 2008.
2. Ben Torben-Nielsen. Dendritic Morphology: Function Shapes Structure. Promotores: H.J. van den Herik, E.O. Postma. Co-promotor: K.P. Tuyls. Tilburg, 3 December 2008.
3. Hans Stol. A Framework for Evidence-based Policy Making Using IT. Promotor: H.J. van den Herik. Tilburg, 21 January 2009.
4. Jeroen Geertzen. Dialogue Act Recognition and Prediction. Promotor: H. Bunt. Co-promotor: J.M.B. Terken. Tilburg, 11 February 2009.
5. Sander Canisius. Structured Prediction for Natural Language Processing. Promotores: A.P.J. van den Bosch, W. Daelemans. Tilburg, 13 February 2009.
6. Fritz Reul. New Architectures in Computer Chess. Promotor: H.J. van den Herik. Co-promotor: J.W.H.M. Uiterwijk. Tilburg, 17 June 2009.
7. Laurens van der Maaten. Feature Extraction from Visual Data. Promotores: E.O. Postma, H.J. van den Herik. Co-promotor: A.G. Lange. Tilburg, 23 June 2009 (cum laude).
8. Stephan Raaijmakers. Multinomial Language Learning. Promotores: W. Daelemans, A.P.J. van den Bosch. Tilburg, 1 December 2009.
9. Igor Berezhnoy. Digital Analysis of Paintings. Promotores: E.O. Postma, H.J. van den Herik. Tilburg, 7 December 2009.
10. Toine Bogers. Recommender Systems for Social Bookmarking. Promotor: A.P.J. van den Bosch. Tilburg, 8 December 2009.
11. Sander Bakkes. Rapid Adaptation of Video Game AI. Promotor: H.J. van den Herik. Co- promotor: P. Spronck. Tilburg, 3 March 2010.
12. Maria Mos. Complex Lexical Items. Promotor: A.P.J. van den Bosch. Co-promotores: A. Vermeer, A. Backus. Tilburg, 12 May 2010 (in collaboration with the Department of Language and Culture Studies).
13. Marieke van Erp. Accessing Natural History. Discoveries in data cleaning, structuring, and retrieval. Promotor: A.P.J. van den Bosch. Co-promotor: P.K. Lendvai. Tilburg, 30 June 2010.
14. Edwin Commandeur. Implicit Causality and Implicit Consequentiality in Language Comprehension. Promotores: L.G.M. Noordman, W. Vonk. Co-promotor: R. Cozijn. Tilburg, 30 June 2010.
15. Bart Bogaert. Cloud Content Contention. Promotores: H.J. van den Herik, E.O. Postma. Tilburg, 30 March 2011.

16. Xiaoyu Mao. Airport under Control. Promotores: H.J. van den Herik, E.O. Postma. Co- promotores: N. Roos, A. Salden. Tilburg, 25 May 2011.
17. Olga Petukhova. Multidimensional Dialogue Modelling. Promotor: H. Bunt. Tilburg, 1 September 2011.
18. Lisette Mol. Language in the Hands. Promotores: E.J. Krahmer, A.A. Maes, M.G.J. Swerts. Tilburg, 7 November 2011 (cum laude).
19. Herman Stehouwer. Statistical Language Models for Alternative Sequence Selection. Promotores: A.P.J. van den Bosch, H.J. van den Herik. Co-promotor: M.M. van Zaanen. Tilburg, 7 December 2011.
20. Terry Kakeeto-Aelen. Relationship Marketing for SMEs in Uganda. Promotores: J. Chr. van Dalen, H.J. van den Herik. Co-promotor: B.A. Van de Walle. Tilburg, 1 February 2012.
21. Suleman Shahid. Fun & Face: Exploring non-verbal expressions of emotion during playful interactions. Promotores: E.J. Krahmer, M.G.J. Swerts. Tilburg, 25 May 2012.
22. Thijs Vis. Intelligence, Politie en Veiligheidsdienst: Verenigbare Grootheden? Promotores: T.A. de Roos, H.J. van den Herik, A.C.M. Spapens. Tilburg, 6 June 2012 (in collaboration with the Tilburg School of Law).
23. Nancy Pascall. Engendering Technology Empowering Women. Promotores: H.J. van den Herik, M. Diocaretz. Tilburg, 19 November 2012.
24. Agus Gunawan. Information Access for SMEs in Indonesia. Promotor: H.J. van den Herik. Co-promotores: M. Wahdan, B.A. Van de Walle. Tilburg, 19 December 2012.
25. Giel van Lankveld. Quantifying Individual Player Differences. Promotores: H.J. van den Herik, A.R. Arntz. Co-promotor: P. Spronck. Tilburg, 27 February 2013.
26. Sander Wubben. Text-to-text Generation Using Monolingual Machine Translation. Promotores: E.J. Krahmer, A.P.J. van den Bosch, H. Bunt. Tilburg, 5 June 2013.
27. Jeroen Janssens. Outlier Selection and One-Class Classification. Promotores: E.O. Postma, H.J. van den Herik. Tilburg, 11 June 2013.
28. Martijn Balsters. Expression and Perception of Emotions: The Case of Depression, Sadness and Fear. Promotores: E.J. Krahmer, M.G.J. Swerts, A.J.J.M. Vingerhoets. Tilburg, 25 June 2013.
29. Lisanne van Weelden. Metaphor in Good Shape. Promotor: A.A. Maes. Co-promotor: J. Schilperoord. Tilburg, 28 June 2013.
30. Ruud Koolen. "Need I say More? On Overspecification in Definite Reference." Promotores: E.J. Krahmer, M.G.J. Swerts. Tilburg, 20 September 2013.
31. J. Douglas Mastin. Exploring Infant Engagement. Language Socialization and Vocabulary. Development: A Study of Rural and Urban Communities in Mozambique. Promotor: A.A. Maes. Co-promotor: P.A. Vogt. Tilburg, 11 October 2013.
32. Philip C. Jackson. Jr. Toward Human-Level Artificial Intelligence – Representation and Computation of Meaning in Natural Language. Promotores: H.C. Bunt, W.P.M. Daelemans. Tilburg, 22 April 2014.
33. Jorrig Vogels. Referential choices in language production: The Role of Accessibility. Promotores: A.A. Maes, E.J. Krahmer. Tilburg, 23 April 2014.

34. Peter de Kock. Anticipating Criminal Behaviour. Promotores: H.J. van den Herik, J.C. Scholtes. Co-promotor: P. Spronck. Tilburg, 10 September 2014.

35. Constantijn Kaland. Prosodic marking of semantic contrasts: do speakers adapt to addressees? Promotores: M.G.J. Swerts, E.J. Krahmer. Tilburg, 1 October 2014.

36. Jasmina Marić. Web Communities, Immigration and Social Capital. Promotor: H.J. van den Herik. Co-promotores: R. Cozijn, M. Spotti. Tilburg, 18 November 2014.

37. Pauline Meesters. Intelligent Blauw. Promotores: H.J. van den Herik, T.A. de Roos. Tilburg, 1 December 2014.

38. Mandy Visser. Better use your head. How people learn to signal emotions in social contexts. Promotores: M.G.J. Swerts, E.J. Krahmer. Tilburg, 10 June 2015.

39. Sterling Hutchinson. How symbolic and embodied representations work in concert. Promotores: M.M. Louwerse, E.O. Postma. Tilburg, 30 June 2015.

40. Marieke Hoetjes. Talking hands. Reference in speech, gesture and sign. Promotores: E.J. Krahmer, M.G.J. Swerts. Tilburg, 7 October 2015.

41. Elisabeth Lubinga. Stop HIV. Start talking? The effects of rhetorical figures in health messages on conversations among South African adolescents. Promotores: A.A. Maes, C.J.M. Jansen. Tilburg, 16 October 2015.

42. Janet Bagorogoza. Knowledge Management and High Performance. The Uganda Financial Institutions Models for HPO. Promotores: H.J. van den Herik, B. van der Walle, Tilburg, 24 November 2015.

43. Hans Westerbeek. Visual realism: Exploring effects on memory, language production, comprehension, and preference. Promotores: A.A. Maes, M.G.J. Swerts. Co-promotor: M.A.A. van Amelsvoort. Tilburg, 10 Februari 2016.

44. Matje van de Camp. A link to the Past: Constructing Historical Social Networks from Unstructured Data. Promotores: A.P.J. van den Bosch, E.O. Postma. Tilburg, 2 Maart 2016.

45. Annemarie Quispel. Data for all: Data for all: How professionals and non-professionals in design use and evaluate information visualizations. Promotor: A.A. Maes. Co-promotor: J. Schilperoord. Tilburg, 15 Juni 2016.

46. Rick Tillman. Language Matters: The Influence of Language and Language Use on Cognition Promotores: M.M. Louwerse, E.O. Postma. Tilburg, 30 Juni 2016.

47. Ruud Mattheij. The Eyes Have It. Promoteres: E.O. Postma, H. J. Van den Herik, and P.H.M. Spronck. Tilburg, 5 October 2016.

48. Marten Pijl, Tracking of human motion over time. Promotores: E. H. L. Aarts, M. M. Louwerse Co-promotor: J. H. M. Korst. Tilburg, 14 December 2016.

49. Yevgen Matusevych, Learning constructions from bilingual exposure: Computational studies of argument structure acquisition. Promotor: A.M. Backus. Co-promotor: A.Alishahi. Tilburg, 19 December 2016.

50. Karin van Nispen. What can people with aphasia communicate with their hands? A study of representation techniques in pantomime and co-speech gesture. Promotor: E.J. Krahmer. Co-promotor: M. van de Sandt-Koenderman. Tilburg, 19 December 2016.

51. Adriana Baltaretu. Speaking of landmarks. How visual information inuences reference in spatial domains. Promotores: A.A. Maes and E.J. Krahmer. Tilburg, 22 December 2016.

52. Mohamed Abbadi. Casanova 2, a domain specific language for general game development. Promotores: A.A. Maes, P.H.M. Spronck and A. Cortesi. Co-promotor: G. Maggiore. Tilburg, 10 March 2017.

53. Shoshannah Tekofsky. You Are Who You Play You Are. Modelling Player Traits from Video Game Behavior. Promotores: E.O. Postma and P.H.M. Spronck. Tilburg, 19 Juni 2017.

54. Adel Alhuraibi, From IT-Business Strategic Alignment to Performance: A Moderated Mediation Model of Social Innovation, and Enterprise Governance of IT. Promotores: H.J. van den Herik and Prof. dr. B.A. Van de Walle. Co-promotor: Dr. S. Ankolekar. Tilburg, 26 September 2017.

55. Wilma Latuny. The Power of Facial Expressions. Promotores: E.O. Postma and H.J. van den Herik. Tilburg, 29 September 2017.

56. Sylvia Huwaë, Different Cultures, Different Selves? Suppression of Emotions and Reactions to Transgressions across Cultures. Promotores: E.J. Krahmer and J. Schaafsma. Tilburg, 11 October, 2017.

57. Mariana Serras Pereira, A Multimodal Approach to Children's Deceptive Behavior. Promotor: M. Swerts. Co-promotor: S. Shahid Tilburg, 10 January, 2018.

58. Emmelyn Croes, Meeting Face-to-Face Online: The Effects of Video-Mediated Communication on Relationship Formation. Promotores: E.J. Krahmer and M. Antheunis. Co-promotor A.P. Schouten. Tilburg, 28 March 2018.

59. Lieke van Maastricht, Second Language Prosody: Intonation and Rhythm in Production and Perception. Promotores: E.J. Krahmer and M. Swerts. Tilburg, 9 May 2018.

60. Nanne van Noord, Learning visual representations of style. Promotores: E.O. Postma and M. Louwerse. Tilburg, 16 May 2018.