Learned image representations for visual recognition

Larsen, Anders Boesen Lindbo; Larsen, Rasmus; Dahl, Anders Bjorholm

Publication date:
2016

Document Version
Publisher's PDF, also known as Version of record

Learned image representations for visual recognition

Anders Boesen Lindbo Larsen

Advisers: Rasmus Larsen & Anders Bjorholm Dahl

Department of Applied Mathematics and Computer Science
Technical University of Denmark

Technical University of Denmark
Department of Applied Mathematics and Computer Science
Richard Petersens Plads, Building 324
2800 Kongens Lyngby, Denmark

DTU Compute, PHD-2016-418, ISSN: 0909-3192

Abstract

This thesis addresses the problem of extracting image structures for representing images effectively in order to solve visual recognition tasks. Problems from diverse research areas (medical imaging, material science and food processing) have motivated large parts of the methodological development. The solutions are inspired by and extend state-of-the-art techniques for describing and learning image content.

More specifically, the thesis explores two approaches to constructing image representations, namely feature engineering and feature learning. In the feature engineering approach, we devise a new image representation for texture-like patterns based on count statistics of second-order image structure. We demonstrate the discriminative capabilities of this representation on medical images and perform both cell classification and mitosis detection. Moreover, we develop an object identification method based on vector quantized local image descriptors allowing us to distinguish individual meat cuts along a production line and trace them in a non-intrusive manner. In the feature learning approach, we propose to solve the task of segmenting scanning electron microscopy images of calcite crystals by learning a meaningful pixel description to facilitate the actual segmentation. Finally, we present a new unsupervised generative image model addressing the problem of pixel-based similarity measures for images. We propose a scheme for employing feature-based similarity measures and demonstrate how this improves the ability to learn high-level concepts in images of faces.

The thesis argues in favor of learning features and presents new methods for domains with limited amounts of labeled data allowing feature learning to be applied more broadly.

Abstract (Danish)

Denne afhandling beskæftiger sig med at udtrække billedstrukturer for at repræsentere billeder effektivt med henblik på at kunne løse billedgenkendelsesproblemer. Metodeudviklingen er inspirereret af problemer fra så forskellige forskningsområder som medicinsk billedbehandling, fødevarebehandling og materialevidenskab. Løsningerne er inspirerede af og udvider teknikkens standpunkt inden for beskrivelse og læring af billedindhold.

Mere specifikt undersøger denne afhandling to tilgange til at konstruere billedrepræsentationer, nemlig ved manuelt design af billedmønstre samt ved automatisk læring af billedmønstre fra data. I den manuelle tilgang foreslår vi en ny billedrepræsentation til at beskrive teksturmønstre baseret på histogrammer af anden-ordens billedstruktur. De diskriminative egenskaber ved denne repræsentation demonstreres på medicinske billeddata til klassifikation af celler samt til detektion af mitosisforekomster. Derudover udvikler vi en en identifikationsmetode baseret på vektorkvantificerede lokale billedbeskrivelser, hvilket tillader ikke-intrusiv genkendelse af individuelle kødstykker langs en produktionslinje. I læringstilgangen foreslår vi at segmentere scanning-elektronmikroskop-billeder af kalcit-krystaller ved at lære en meningsfuld pixelbeskrivelse der muliggør segmenteringsopgaven. Endelig præsenterer vi en ny ikke-superviseret generativ billedmodel der addreserer problematiske pixel-baserede similaritetsmål til billeder. Vi foreslår at benytte lærte billedmønstre som basis for similaritetsmålet og demonstrerer hvordan dette forbedrer evnen til at fange højniveau strukturer i billeder af ansigter.

Denne afhandling argumenterer for at lære billedmønstre og præsenterer nye metoder for applikationsområder med begrænsede mængder superviserede data, hvilket tillader mønsterlæring at blive anvendt i bredere omfang.

Preface

This thesis deals with methods for characterizing the contents of images using hand-made as well as learned image patterns. I have tried making the text accessible to fellow researchers by favoring the broader perspective of visual recognition rather than diving into technical details. The thesis comprises an introduction to existing state-of-the-art methodology, an overview of my scientific contributions in relation to the methodology as well as a discussion of the contributions. Four articles (listed below) are attached as appendix.

The thesis was prepared at the *Department of Applied Mathematics and Computer Science* at the *Technical University of Denmark* in fulfillment of the PhD degree requirements. Professor *Rasmus Larsen* has acted as main adviser with associate professor *Anders Bjorholm Dahl* as co-adviser. The thesis is made possible with funding from the *inSPIRe* project by the *Strategic Platforms for Innovation and Research* under the *Danish Council for Strategic Research*.

Kongens Lyngby, June 10, 2016

Included papers

The following papers are included in their full form.

- Larsen, A. B. L., Vestergaard, J. S., and Larsen, R. (2014). "HEp-2 Cell Classification Using Shape Index Histograms With Donut-Shaped Spatial Pooling". In: *Medical Imaging, IEEE Transactions on* 33.7, pp. 1573–1580.
- Larsen, A. B. L., Hviid, M. S., Jørgensen, M. E., et al. (2014). "Vision-based method for tracking meat cuts in slaughterhouses". In: *Meat Science* 96.1, pp. 366–

372.

- Larsen, A. B. L., Schultz, L. N., Dahl, A. B., et al. (2016). "Automatic particle size measurements from a learned image segmentation". In: *submitted*.
- Larsen, A. B. L., Sønderby, S. K., Larochelle, H., et al. (2016). "Autoencoding beyond pixels using a learned similarity metric". In: *Proceedings of the 33rd International Conference on Machine Learning (ICML)*.

Excluded papers

The following papers are excluded from the thesis because they do not fit in with the overall story or because I'm not acting as main author.

- Larsen, A. B. L. (2014). *CUDArray: CUDA-based NumPy*. Tech. rep. DTU Compute 2014-21. Department of Applied Mathematics and Computer Science, Technical University of Denmark.
- Larsen, A. B. L., Dahl, A. B., and Larsen, R. (2015). "Oriented Shape Index Histograms for Cell Classification". In: *Proceedings of the 19th Scandinavian Conference Image Analysis (SCIA)*. Springer International Publishing, pp. 16–25.
- Larsen, A. B. L. (2016). *DeepPy: Pythonic deep learning*. Tech. rep. DTU Compute 2016-6. Department of Applied Mathematics and Computer Science, Technical University of Denmark.
- Nobel-Jørgensen, M., Nielsen, J. B., Larsen, A. B. L., et al. (2013). "Pond of illusion: interacting through mixed reality". In: *SIGGRAPH Asia 2013 Posters*. ACM, p. 26.
- Veta, M., Diest, P. J. van, Willems, S. M., et al. (2015). "Assessment of algorithms for mitosis detection in breast cancer histopathology images". In: *Medical Image Analysis* 20.1, pp. 237–248.
- Hauberg, S., Freifeld, O., Larsen, A. B. L., et al. (2016). "Dreaming More Data: Class-dependent Distributions over Diffeomorphisms for Learned Data Augmentation". In: *Proceedings of the 19th international Conference on Artificial Intelligence and Statistics (AISTATS)*. vol. 51, pp. 342–350.

Acknowledgments

First, I would like to thank my advisers Rasmus Larsen and Anders Bjorholm Dahl for all their patience, guidance and valuable input throughout my PhD studies. I also want to thank Søren Hauberg for supplemental guidance and for his inspiring scientific curiosity. Of course, no stay in our group would be complete without the need to thank Knut Conradsen for his many larger-than-academia insights.

Large parts of this thesis are motivated by problems from other research disciplines and made possible by the valuable data provided by my collaborators. I wish to thank Lars B. Christensen and Marchen S. Hviid for their ideas and all their help creating datasets. I also wish to thank Logan N. Schultz and Henning O. Sørensen for introducing me to the world in micro-scale.

Many thanks to Aaron Courville and Hugo Larochelle for inviting me to Université de Montréal for a visit. I really wish I could have been there longer!

I am grateful towards my colleagues and friends for making my time at *Dutten* both fun and rewarding: Michael Andersen, Anders N. Christensen, Stine Harder, H. Martin Kjær, Lars Maaløe, Jannik B. Nielsen, Oula Puonti, Kasper H. Steenstrup, Casper K. Sønderby, Søren K. Sønderby, Camilla H. Trinderup and Jacob S. Vestergaard, thanks!

Above all, I wish to thank my family and especially my girlfriend Ingrid for enthusiastically supporting and compassionately forgiving my digital diversions.

Contents

1 Introduction

The goal of visual recognition is to extract information from images in order to solve a given task. Here, the meaning of *extract information* is broad since visual recognition is applied across different image types (natural, biomedical, depth maps, 2D/3D, etc.) and spans methods for both processing, interpreting and learning the image input.

Among the earliest examples of successful visual recognition systems is for transistor assembly in the early 70's (Kashioka et al. 1976). Since then, by facilitating the mass production and verification of entire wafers, visual recognition has played an important role in the development of the semiconductor industry and thereby also in the digital revolution (Andreopoulos et al. 2013; Ejiri 2007). Today, visual recognition is applied in many settings including factories, offices, research and surveillance. Examples of higher-level problems where visual recognition systems can compete with human performance are mitosis detection (Veta et al. 2015), traffic sign classification (Stallkamp et al. 2011) and Chinese handwriting recognition (Yin et al. 2013). Looking forward, visual recognition at a human level in complex natural environments is a crucial step towards reaching the long-standing goal of *artificial intelligence* (AI).

On the face of it, many recognition tasks seem easy and well-defined to humans. E.g. the task of locating birds in images. Even if a person have never seen a bird, we can relatively easily explain the components that make up a bird (beak, wings, feathers, etc.) and the person should then be able to learn a generalized bird representation from seeing only a few examples. In contrast, such tasks are inherently hard for computers to solve. Simply just explaining to a computer the concept of a feather and how it appears in a plumage requires a preexisting understanding of material properties in a 3D world. As a consequence, visual recognition problems are often underestimated by overly optimistic beginners and even practitioners. Most famous is the example from Massachusetts Institute of Technology in 1966 where a summer project proposal begins with the sentence: "*The summer vision project is an attempt to use our summer workers effectively in the*

Figure 1: Computer science in a nutshell for the uninitiated. Visual recognition problems are often surprisingly harder than what they may seem. Source: `http://xkcd.com/1425/`

construction of a significant part of a visual system." (Papert 1966). Half a century later, we still have not finished what the summer project set out to. The difficulty of judging problem sizes in computer science is comically illustrated in Figure 1. Thus, the discrepancy between human capabilities of visual perception and our valuation of its complexity is remarkable. Humans are given a visual system (eyes and neurons) from nature and learn visual perception during infancy along with a 3D understanding of the world. What we consider a trivial and autonomous task builds on years of training and learning about our environment.

Taking a more philosophical perspective, one can compare a computer learning visual recognition to the setup from Plato's *Allegory of the cave* (Jowett 1941). In this story, a group of persons have been confined their entire lives in an underground cave chained and fixated towards the wall of the cave such that the wall is all they have ever seen. Behind them burns a fire, and between them and the fire objects are moving around casting shadows on the wall. Thus, the persons see only shadow projections of the real objects as they have never been able to turn their heads. Similarly, a computer program starts from scratch and typically see the world as projections in form of 2D pixel arrays. Now in the allegory, one of the persons (the philosopher) escapes the chains and discovers the truth by exploring the cave as well as the nature outside. He then returns to the chained persons and tries to explain to them the true world whose shadow appearance they are mistaking for reality. The allegory is an example of Plato's theory of the perfect world of *Ideas* versus the imperfect world as

it is perceived by humans. Comparably, human visual perception is the perfect world relative to the computer's imperfect vision full of restraints (slow data processing, poor sensor signal and little understanding about the world). The job of a computer vision engineer then becomes that of the philosopher; to teach the program about the perfect world in terms of the crude projections the program is given as input. Curiously, the allegory ends fatally when the chained people violently resists the philosopher's nonsense as he appears blinded after having been exposed to the bright light sources!

1.1 The effectiveness of good image representations

This thesis focuses on the problem of representing images on a form allowing for efficient processing and higher-level abstractions. A good representation is crucial for the performance of a visual recognition system since it can greatly simplify the decision-making part on top of the image representation.

The representation of RGB pixel arrays is the standard format for storing, distributing and displaying images. RGB pixels are designed for visual fidelity to humans but are poorly suited for computer vision methods. Pixel arrays have a large memory footprint requiring many CPU clock cycles for operations over the data. Moreover, the abstraction level of pixels is as low as possible since pixels are intended to be the atoms of the image.

For many years, the *feature description* subfield of computer vision has focused on devising clever image representations (also called features) to improve on the poor efficiency and low abstraction level of pixels. While these alternative representations lack the ability to be visually interpretable like pixels, they make it possible to build efficient computer vision systems to solve a given task. Many popular image representations capture simple structures like edge orientation and summarize them with count statistics as this has shown widely useful (Dalal et al. 2005; Felzenszwalb et al. 2010; Lowe 2004).

Until recently, feature description schemes have mostly been designed manually by computer vision researchers using methods from image/signal processing. The task of finding a good feature representation for a given problem is cumbersome and typically requires many iterations of trial and error. In fact, the famous SIFT descriptor paper (Lowe 2004) is the result of many years of parameter tuning. Even today, the method is still being tweaked for numerous new applications which explains why the paper ranks among the top cited computer science articles (CiteSeerX 2016). Therefore, when the seminal work of Krizhevsky et al. (2012) successfully demonstrated a data-driven approach to learning image features for object

recognition, the implications were profound throughout the computer vision community. Using a *convolutional neural networks* (convnet) (LeCun et al. 1998) they were able to learn image features far superior to existing hand-crafted features for visual recognition in natural images (Russakovsky et al. 2015). Thus, the challenging job of devising feature descriptors had been automated and even surpassed by a machine learning-based approach.

This thesis is written during the radical changes in the field of visual recognition. Originally setting out to explore traditional feature description, the thesis has changed focus towards the learning-based approach. Thus, this work strives to provide a view on image representations from both angles.

1.2 Contributions

In one sentence, this thesis explores a range of visual recognition problems and contributes to the field with methods for solving the problems as well as new models for both representing and learning to represent image content. In more detail, the contributions of the thesis are:

- A feature descriptor for texture-like image structures based on histograms of second-order image structure.
- A method for classifying cell images based on the new image descriptor.
- A method for identifying images of meat cuts along a production line using histograms of local image features.
- A method for segmenting images of calcite particles by learning pixel classification as preprocessing.
- A generative model for unsupervised learning using learned representations to better measure similarities in image space.

1.3 Outline

The thesis comprises three sections not including this introduction. In Section 2, we introduce feature description from the ground up to popular image features. At the end of the section, we introduce two of the included papers that employ these feature description methods. In Section 3, we describe the learning approach to image description before introducing two papers on learned image representation. Section 4 discusses the contributions of the thesis and a more general outlook on image representation.

The thesis is structured linearly as it first covers relevant methodological concepts before introducing the related papers. The reader can also choose to skip directly to the paper introductions and use the methodological sections as reference.

2 Manual image representations

This part of the thesis introduces the classical approach to feature description for computer vision. Here, the mapping from pixels to feature representation is designed wrt. the image input as well as the decision making (machine learning) on top of the features. The literature on feature description is abundant and in the following we will try to cover the popular approaches. In general, feature descriptor pipelines comprise three following steps which we will each describe in separate sections: 1) Extraction of low-level features, e.g. edges. 2) Summarization of the low-level features to represent them with a more convenient vector-form. 3) Aggregation of summarized features with the purpose of learning archetypes among the features.

The concepts of *invariance* and *robustness* are central to feature description. When a feature representation $\text{feat}(x)$ of an input x is invariant towards a transformation $\text{transf}(\cdot)$, the features are not affected by these transformations of the input, i.e.

$$\text{feat}(x) = \text{feat}(\text{transf}(x)) \quad . \tag{2.1}$$

Invariance is important to account for the different sources of variability that affect object appearances (e.g. illumination, orientation, size and position). Ideally, if the illumination of an object is irrelevant for the given task, the feature representation should discard information about illumination since this may complicate later steps of the vision system. Robustness is reminiscent of invariance and can be thought of as handling visual variability gracefully, e.g. by not being too sensitive to image rotations. Robustness is also important in anomaly situations like occlusions or extreme transformations that may destroy parts of the image content.

2.1 Low-level feature extraction

Low-level image structures are the atomic building blocks for visual recognition. These can be anything from individual pixels (intensities) to edges and higher-order structure (e.g. saddle points). Many of these structures

are designed from intuition and implemented with crude but fast approximations in the form of small convolutional filters (Dalal et al. 2005; Laws 1980; Leung et al. 2001). Usually, the filters can also be derived using the sound mathematical framework of *scale space theory* which we will use in the following. In practice, though, approximations with small convolutional filters work fine; Dalal et al. (2005) even report the best performance with the vector $[-1, 0, 1]$ as first-order derivative filter.

2.1.1 Differential image structure

We describe low-level image features as the differential structure of a 2D image $I(\boldsymbol{x})\colon \Omega \to \mathbb{R},\ \Omega \subseteq \mathbb{R}^2$. When working with images, the notion of *scale* is fundamental as the structure we are interested in has a certain size that we should try to match. The *Gaussian scale space* representation computes an image at scale σ as

$$L(\boldsymbol{x}; \sigma) = (G(\sigma) * I)(\boldsymbol{x}) \quad , \tag{2.2}$$

where $*$ denotes convolution and G is the Gaussian kernel and σ controls the width of the kernel,

$$G(\boldsymbol{x}; \sigma) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{\boldsymbol{x} \cdot \boldsymbol{x}}{2\sigma^2}\right) \quad , \quad \sigma \geq 0 \quad , \quad G(\boldsymbol{x}; 0) \equiv \delta \quad . \tag{2.3}$$

For $\sigma = 0$, we have the original image. By increasing σ we can blur the image thereby removing structure at smaller scales while preserving larger image structures.

We seek to describe image structure by computing derivatives. However, pixel images are discrete and therefore non-differentiable. Luckily, the commutativity property of convolution allows us to compute image differentials by convolving derivatives of the Gaussian with the image:

$$L_{x^n y^m}(\boldsymbol{x}; \sigma) = \sigma^{n+m} \frac{\partial^{n+m}}{\partial x^n \partial y^m} \left(G(\sigma) * I\right)(\boldsymbol{x}) \tag{2.4}$$

$$= \sigma^{n+m} \left(\left(\frac{\partial^{n+m}}{\partial x^n \partial y^m} G(\sigma)\right) * I\right)(\boldsymbol{x}) \quad . \tag{2.5}$$

Here, n and m indicate the differentiation order along the x and y axis respectively. The factor σ^{n+m} scale-normalizes the image derivatives to account for the amplitude of Gaussian that decreases as the scale increases (Lindeberg 1998). For future notational convenience we may omit the function arguments and substitute $L_{x^n y^m}(\boldsymbol{x}; \sigma)$ with simply $L_{x^n y^m}$. Thus, $L_{x^n y^m}$ is implicitly assumed to be computed at some scale σ and location $\boldsymbol{x}$.

2.1.2 Gradient orientation

Edge structures described as the orientation of image gradients is perhaps the most widely used computer vision feature as it forms the basis for SIFT (Lowe 2004) and its many derivatives. Using first-order derivatives, we calculate gradient orientations o from

$$o(\boldsymbol{x};\sigma) = \arctan2\left(L_y, L_x\right) \quad , \quad o_{\text{unsigned}}(\boldsymbol{x};\sigma) = \arctan\left(\frac{L_y}{L_x}\right) \quad . \tag{2.6}$$

Depending on the image characteristics, unsigned orientations spanning 180° may be preferable over signed orientations spanning 360°. E.g. unsigned orientations are used for pedestrian detection since the brightness level of a person relative to the background is uninformative in natural settings. To measure the strength of the orientation, we compute the gradient magnitude m from

$$m(\boldsymbol{x};\sigma) = \sqrt{L_x^2 + L_y^2} \quad . \tag{2.7}$$

2.1.3 The shape index

Beyond first-order image structure, the literature becomes more varied as there is no dominating feature extractor. Higher-order structures are often used for texture recognition problems (Crosier et al. 2010; Ojala et al. 2002), but can also complement first-order structure to improve recognition performance (Larsen et al. 2015; Wang et al. 2009). In the following we present the *shape index* measure for describing second-order image curvature (Koenderink et al. 1992).

The shape index is computed from the partial derivatives contained in the Hessian matrix

$$\nabla^2 L(\boldsymbol{x};\sigma) = \begin{bmatrix} L_{x^2} & L_{xy} \\ L_{xy} & L_{y^2} \end{bmatrix} \quad . \tag{2.8}$$

The Hessian matrix is square and symmetric allowing us to compute the pair of real eigenvalues κ_1 and κ_2:

$$\begin{aligned} \kappa_1(\boldsymbol{x};\sigma) &= \frac{1}{2}\left(L_{x^2} + L_{y^2} - \sqrt{(L_{x^2} - L_{y^2})^2 + 4L_{xy}^2}\right) \\ \kappa_2(\boldsymbol{x};\sigma) &= \frac{1}{2}\left(L_{x^2} + L_{y^2} + \sqrt{(L_{x^2} - L_{y^2})^2 + 4L_{xy}^2}\right) \end{aligned} \tag{2.9}$$

κ_1 and κ_2 are the *principal curvatures*. They describe the strength of the curvature along the *extremal directions* where the curvatures are minimal and

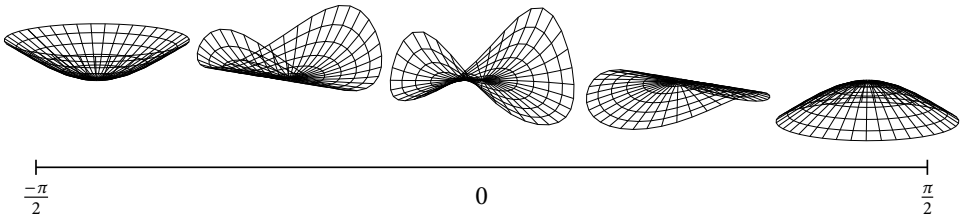


Figure 2: Second-order curvatures along the shape index interval $]-\pi/2, \pi/2[$. The shapes are aligned vertically and have the same curvedness.

maximal, respectively. The shape index $s \in \left]\frac{-\pi}{2}, \frac{\pi}{2}\right[$ is defined as

$$s(\boldsymbol{x}; \sigma) = \arctan\left(\frac{\kappa_1 + \kappa_2}{\kappa_1 - \kappa_2}\right) \quad . \tag{2.10}$$

Note that κ_1 and κ_2 are invariant to the orientation of the curvature which is captured by the eigenvectors of $\nabla^2 L$ (the extremal directions). By discarding the eigenvectors, the shape index becomes invariant to image rotations. Moreover, the shape index has the attractive property of describing all second-order shapes onto a continuous interval providing a smooth and intuitive transition between the shapes, see Figure 2. In addition to the shape index a measure of *curvedness* c is defined,

$$c(\boldsymbol{x}; \sigma) = \sqrt{\kappa_1^2 + \kappa_2^2} \quad . \tag{2.11}$$

The curvedness indicates the strength of the shape described by the shape index such that we differentiate between flat and indistinct vs. prominent shapes. See Figure 3 for examples. We can think of the shape index measures s and c similar to the gradient orientation measures o and m, as the first represents the structure while the second represents its strength. Alternatively, we can skip the explicit calculation of the Hessian eigenvalues and express s and c as functions of the second-order derivatives:

$$s(\boldsymbol{x}; \sigma) = \arctan\left(\frac{-L_{x^2} - L_{y^2}}{\sqrt{(L_{x^2} - L_{y^2})^2 + 4L_{xy}^2}}\right) \tag{2.12}$$

$$c(\boldsymbol{x}; \sigma) = \sqrt{L_{x^2}^2 + 2L_{xy}^2 + L_{y^2}^2} \quad . \tag{2.13}$$

2.2 Low-level feature summarization

After having computed low-level features on top of the image pixels, we want to bring them on a form more convenient than the pixel grid. This

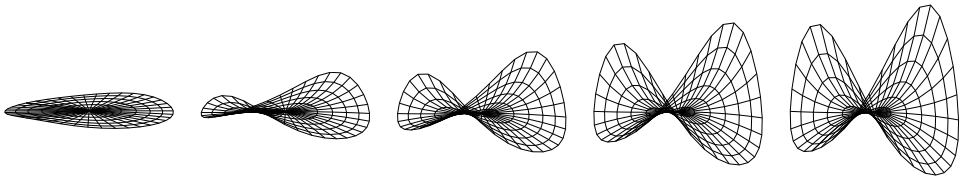


Figure 3: The saddle shape $s = 0$ with increasing curvedness.

is typically done by summarizing feature statistics in histograms. By construction, the histogram representation have two very practical properties for visual recognition: 1) The histogram contributions are collected over a local region, this means that the histogram representation becomes robust to small image translations. 2) Bin contributions are usually smoothed with neighboring bins such that the histogram representation is also robust to small changes of the feature (e.g. image rotation for gradient orientations). For these reasons and thanks to the widespread adoption of SIFT-like feature description, local feature histograms have become ubiquitous in computer vision.

In continuation of the scale space framework, we will describe local histograms using the *locally orderless* image representation (Koenderink et al. 1999). In practice, most descriptors calculate their histograms with alternative faster formulations. In the following, we present histogram construction for gradient orientations, but very similar ideas apply features like the shape index. Feature histograms are constructed by first choosing a set of n bin centers $b_1, \ldots, b_n$ uniformly distributed along the feature value interval, i.e. $]-\pi, \pi[$ for signed gradient orientations. For the SIFT descriptor $n = 8$. We calculate the bin contribution c to the bin centered at b_i by applying a Gaussian window with scale β,

$$c(b_i, \boldsymbol{x}; \beta, \sigma) = \frac{m}{\beta\sqrt{2\pi}} \exp\left(-\frac{(o - b_i)^2}{2\beta^2}\right) \quad . \tag{2.14}$$

Note that the gradient magnitude m is also applied as weight allowing prominent edges to have larger influence than indistinct edges. Because the gradient orientation has a circular range, bin contributions should be wrapped such that the angle 1° contributes to 359°. This is achieved using the circular normal distribution:

$$c_{\text{circular}}(b_i, \boldsymbol{x}; \beta, \sigma) = \frac{m}{I_0(\beta^{-2})2\pi} \exp\left(\beta^{-2}\cos(o - b_i)\right) \quad , \tag{2.15}$$

where $I_0(\cdot)$ is the modified Bessel function of order 0. At this point, c is still pixel-wise bin contributions. To collect bin contributions from the local

neighborhood, we convolve with (yet another) Gaussian kernel with scale parameter α.

$$h(b_i, \boldsymbol{x}; \alpha, \beta, \sigma) = G(\alpha) * c_{\text{circular}}(b_i, \boldsymbol{x}; \beta, \sigma) \tag{2.16}$$

Finally, we build the locally orderless histogram H as the concatenation of bin contributions:

$$H(\boldsymbol{x}; \alpha, \beta, \sigma) = [h(b_1, \boldsymbol{x}; \alpha, \beta, \sigma), \ldots, h(b_n, \boldsymbol{x}; \alpha, \beta, \sigma)] \tag{2.17}$$

To recap the feature extraction so far, there are three scales in play: The *inner* scale σ determines at what level of detail the image feature is extracted. The *tonal* scale β regulates the bin width of the feature histogram and the smoothing of contributions to neighboring bins. The *outer* scale α determines the size of the spatial support of the histogram contributions. The locally orderless representation for feature histograms is arguably a bit involved for something that often gets described in papers as "*we convolve with the kernel* $[-1, 0, 1]$". However, the locally orderless images formulation is capable of expressing the feature extraction in a consistent mathematical framework. Most appealing, this framework explicitly models the different scales/distributions that are implicitly chosen when using off-the-shelf convolutional kernels.

Using Eq. 2.17 we can sample feature histograms from any image position. Though, most feature descriptions sample histograms in a regular $m \times n$ grid:

$$\text{feat}_H(I) = \begin{bmatrix} H(\boldsymbol{x}_{1,1}) & \ldots & H(\boldsymbol{x}_{1,n}) \\ \vdots & \ddots & \vdots \\ H(\boldsymbol{x}_{m,1}) & \ldots & H(\boldsymbol{x}_{m,n}) \end{bmatrix} \tag{2.18}$$

We then concatenate the extracted histograms into a single vector which we normalize to unit sum. The normalization step is crucial because it makes the histogram entries invariant to affine transformation of image intensities.

For many tasks, the multi-histogram representation is used directly for decision-making with some machine learning method on top. Since the histograms are computationally cheap to extract densely from an image, this representation is ideal for sliding window methods where a fast linear classifier is applied on top to determine if the object of interest is present (Dalal et al. 2005; Felzenszwalb et al. 2010).

2.3 Feature aggregation

Feature histograms extracted in a grid are good at capturing rigid objects where the layout of the object parts is limited in variation, e.g. for the

pedestrian detection task where people usually are standing in an upright position. For objects where this is not the case, the histogram templates break down as they cannot capture the variations efficiently. Instead, a common approach is to discard the global object structure and describe the image at many small locations. These locations can be chosen by an *interest point* detector (usually finding blob structures) or simply by sampling densely across the image. The idea behind this approach is that small image patches are representative for the image content even though they do not reveal the entire object. Recently, Ullman et al. (2016) have explored the ability of humans to recognize objects from small image regions. While humans are good at this task compared to computers, their recognition performance is surprisingly sensitive to small changes in the size and the location of the image regions as the image content quickly become ambiguous. Thus, by going from global to local description we avoid the challenging part of modeling global object variations at the risk of increased ambiguity when relying on small image patches.

After having subdivided the image into many small parts, we need a way to bring the image parts on a manageable form, that is, a fixed-size vector representation. The so-called *bag of visual words* (BOVW) approach provides a popular solution (Csurka et al. 2004) to this problem. Given the n image regions, we extract local image patches $P^{(1)}, \ldots, P^{(n)}$ and compute their feature histogram representations $\text{feat}_H\left(P^{(i)}\right)$. In this feature space, we perform vector quantization to capture the distribution of local image features. This can be achieved offline by performing k-means clustering yielding k cluster centers $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k$. For each local image feature we then determine its nearest cluster center:

$$\boldsymbol{\mu}_{\min}^{(i)} = \underset{\boldsymbol{\mu}' \in \{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k\}}{\operatorname{argmin}} \left\| \boldsymbol{\mu}' - \text{feat}_H\left(P^{(i)}\right) \right\| \tag{2.19}$$

We represent an image by counting in (yet another) histogram the number of feature assignments to each cluster:

$$\text{feat}_{\text{BOVW}}(I) = \left[\sum_{i=1}^{n} \left[\boldsymbol{\mu}_1 = \boldsymbol{\mu}_{\min}^{(i)} \right], \ldots, \sum_{i=1}^{n} \left[\boldsymbol{\mu}_k = \boldsymbol{\mu}_{\min}^{(i)} \right] \right] \quad , \tag{2.20}$$

This histogram is subsequently normalized. Several alternatives/extensions to the vanilla BOVW method exist. For example, histogram contributions can be weighted using posterior probabilities from clustering with a Gaussian mixture model. Moreover, instead of counting occurrences we can sum up residuals between features and the cluster centers (Jégou et al. 2010). We can also reintroduce global spatial structure by decomposing the input image into multiple subregions and computing BOVW vectors for each of them (Lazebnik et al. 2006).

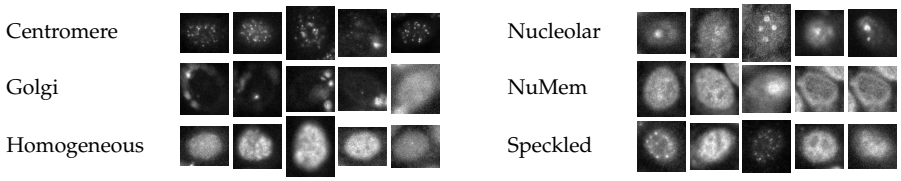


Figure 4: Examples of the 6 different staining pattern classes.

2.4 Contribution: Shape index histograms for medical image analysis

In this section, we approach the problem of characterizing image structure in medical images. The related paper is attached in Appendix A:

– Larsen, A. B. L., Vestergaard, J. S., and Larsen, R. (2014). "HEp-2 Cell Classification Using Shape Index Histograms With Donut-Shaped Spatial Pooling". In: *Medical Imaging, IEEE Transactions on* 33.7, pp. 1573–1580.

In the paper, our goal is to classify images of cells under indirect immunoflourescence illumination in order to diagnose autoimmune diseases. See Figure 4 for examples of the cell images. We approach the problem with a traditional visual recognition setup consisting of feature extraction followed by classification with a kernel-based support vector machine.

Our method is an example of how careful feature engineering to a specific problem can lead to significant performance improvements. We employ shape index histograms to capture the blob-like structures in a rotation invariant manner which is a good fit for many types of medical images. Moreover, to increase distinctiveness of the feature representation, we perform a rotation invariant spatial decomposition suitable for cell images. That is, we sample histograms with concentric donut-shaped spatial support instead of typical grid sampling schemes (Eq. 2.17) which imposes a rotation-sensitive structure in feature description, see Figure 5.

2.4.1 Competition entries

The method was originally developed with the intention of entering two competitions: *Contest on HEp-2 Cells Classification* (Hobson et al. 2015) and *Assessment of Mitosis Detection Algorithms* (Veta et al. 2015). We have already described the first competition as this is the problem our paper is concerned with. The second competition is similar in many aspects as the task is to detect cells undergoing mitosis in whole slide histopathology images, see Figure 6. The level of mitotic activity in a tissue sample is an important prognostic marker for cancer and is currently determined from cumbersome

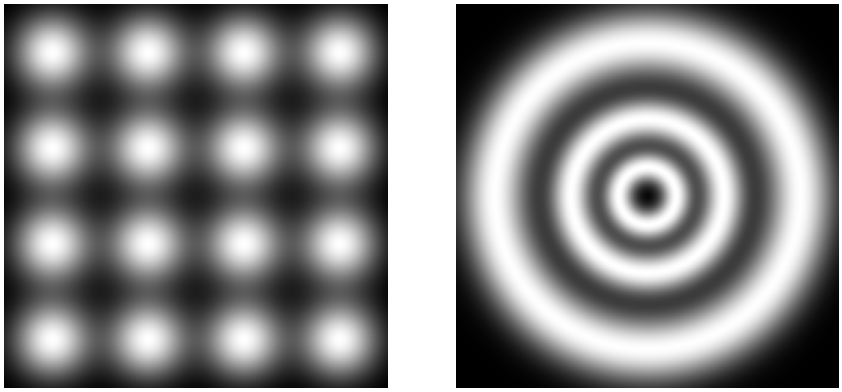


Figure 5: Spatial support of local feature histograms. Left: ordinary grid sampling with Gaussian windows. Right: concentric donut sampling.

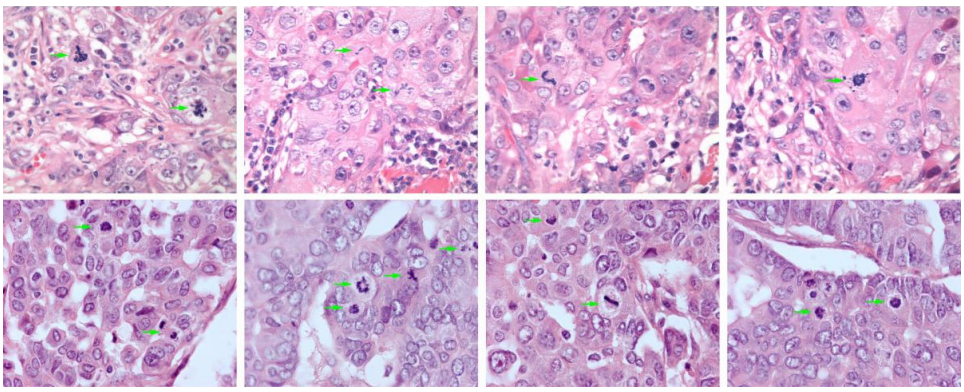


Figure 6: Crops from histopathology images. Examples of mitosis are marked with green arrows. Image adapted from `http://amida13.isi.uu.nl`.

manual inspection. We convert the detection task to a classification task simply by detecting cell location with a purple blob filter (the cell DNA has been stained purple). What remains is the task of learning a binary classifier between mitotic/non-mitotic cells.

In both competitions, shape index histograms have shown competitive performance. For HEp-2 cell classification, our method was ranked 2nd just 0.11% below the winner which was statistically insignificant, see Figure 7. For mitosis detection, our method was also ranked 2nd, this time with a significantly larger margin to the 1st place that relied on deep convolutional neural networks. Compared to conventional visual recognition methods, though, shape index histograms showed superior performance. The results from both competitions are shown in Figure 8. In general, we consider the performance of shape index histograms impressive compared to the simplicity of the method compared to the other competition entries that

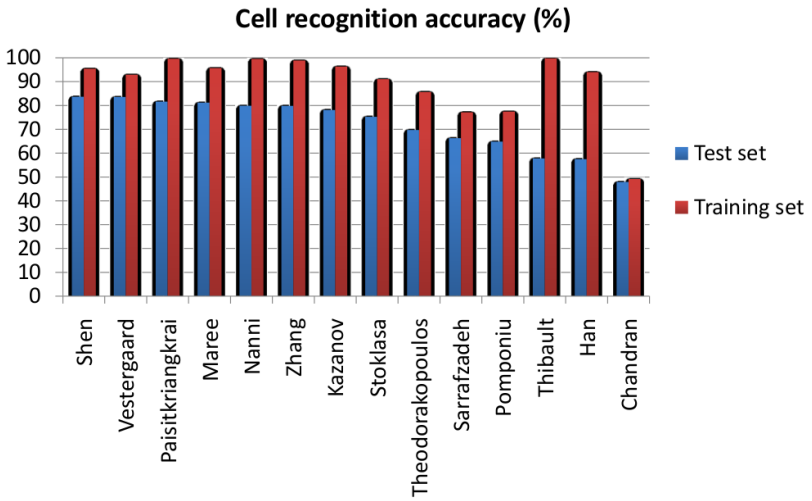


Figure 7: Results from the HEp-2 cell classification competition. Our method is named *Vestergaard*. Image source: `http://nerone.diem.unisa.it/contest-icip-2013-test`.

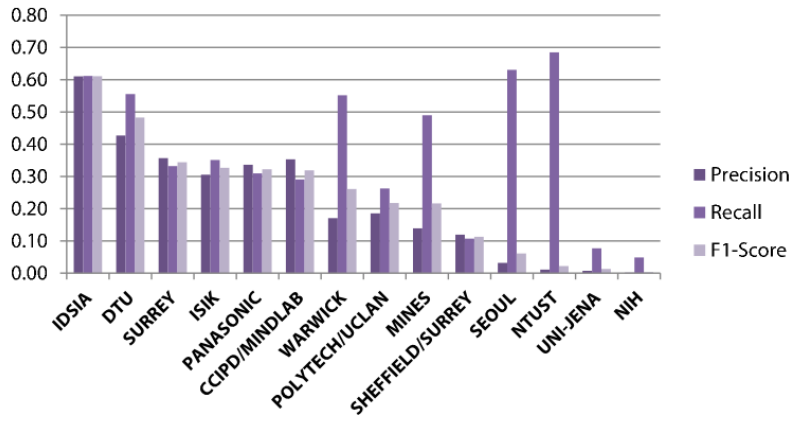


Figure 8: Results from the mitosis detection competition. Our method is named *DTU*. Image source: `http://amida13.isi.uu.nl`.

combine many feature extractors.

2.5 Contribution: BOVW-based identification of meat cuts

In this section, we describe a method for identifying individual meat cuts in images. The related paper is attached in Appendix B:

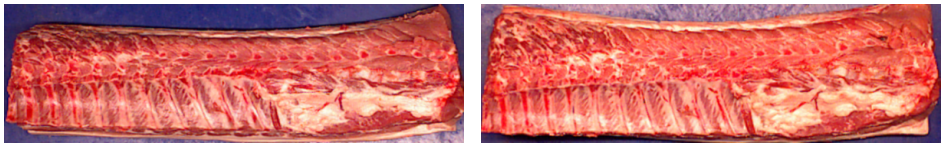


Figure 9: Two images of the same pork loin.

– Larsen, A. B. L., Hviid, M. S., Jørgensen, M. E., et al. (2014). "Vision-based method for tracking meat cuts in slaughterhouses". In: *Meat Science* 96.1, pp. 366–372.

In the food industry, traceability of products is important for public health concerns. The use of ordinary chip-based tracking is problematic for meat product because the chips cannot be attached as they may disappear into the product. We apply visual recognition to the problem of recognizing/identifying images of meat cuts, thereby establishing traceability non-intrusively. For this purpose, we have constructed a dataset of 211 pork loins for which we have taken two images a day apart, see Figure 9 for an example.

To represent the images, we propose to extract BOVW features (Eq. 2.20) sampled in a 4×2 grid along the pork loins. We use gradient orientation histograms with the DAISY scheme for sampling local gradient histograms (Tola et al. 2010). Based on the feature descriptions, we simply match up pork loins based on the χ^2 distance between their BOVW histograms.

Our results show that the BOVW representation is sufficiently robust and distinctive to correctly match up all 211 pork loins, even pork loins that a mistreated altering their visual appearance.

3 Learned image representations

This part of the thesis presents an alternative approach to traditional visual recognition pipelines where machine learning is applied on top of hand-engineered features. In the feature learning approach, the learning signal is propagated all the way through to the input pixels which allows the intermediate representations to adjust according to the given task. Conceptually, feature learning yields a more unified pipeline since the feature transformations become part of the machine learning objective.

The paradigm shift towards learning features is caused by recent advances in machine learning under the name of *deep learning* (LeCun et al. 2015; Schmidhuber 2015). The idea of deep learning is to model high-level abstractions directly from raw data through multiple layers of transformations. The assumption underlying this approach is that the layered representation of the input will correspond to levels of increasing abstraction. Because the layers build on each other, the upper layers can learn higher-level concepts expressed in terms of lower-level concepts from the previous layer. Ideally, we want the learned representation to be *disentangled* meaning that concepts are expressed in terms of lower-level concepts in a hierarchical manner (Bengio et al. 2013). E.g. we want a face feature to be expressed from features corresponding to eyes, nose and mouth, not in terms of edge structures.

In recent years, the deep paradigm for building machine learning systems have shown an impressive range of results. Thanks to the availability of large datasets and more processing power, neural networks have shown that highly expressive models can successfully be trained to disentangle meaningful structure in complex data distributions. In the following, we will go through the basics of deep learning for images and discuss a promising direction for deep unsupervised models that may allow us to learn good representations without the need for large amounts of labeled data.

3.1 Neural networks

We can think of neural networks as a function approximators. Thus, we represent a neural network as function f with input $\boldsymbol{x}$ and parameters $\boldsymbol{\theta}$:

$$f(\boldsymbol{x}; \boldsymbol{\theta}) \tag{3.1}$$

The universal approximation theorem (Hornik 1991) states that a multi-layer feedforward neural network can approximate any function $f: \mathbb{R}^d \to \mathbb{R}$. While this is not a new result, only the recent wave of deep neural networks have begun living up to the theory. Arguably, being able to learn arbitrary functions from data is one of the most important messages of the deep learning era. In feedforward neural networks, f typically consists of n hidden layers with the sequential composition

$$f(\boldsymbol{x}; \boldsymbol{\theta}) = h^{(n)}\left(\ldots h^{(2)}\left(h^{(1)}\left(\boldsymbol{x}; \boldsymbol{\theta}^{(1)}\right); \boldsymbol{\theta}^{(2)}\right)\ldots; \boldsymbol{\theta}^{(n)}\right) \quad . \tag{3.2}$$

In the following, we may omit the parameters in notation for brevity. The classic implementation of these network layers is *fully-connected* meaning that each output unit u of a layer is computed from a weighted sum of all its input units plus a bias term:

$$u = \text{act}\left(\boldsymbol{w} \cdot \boldsymbol{x} + b\right) \quad , \quad \boldsymbol{x} \in \mathbb{R}^d \, , \, \boldsymbol{w} \in \mathbb{R}^d \tag{3.3}$$

A pointwise activation function act, e.g. tanh, allows the layer to perform a nonlinear stretch which increases the expressiveness (Cybenko 1989). In fact without the nonlinearity, we could collapse the sequential linear operations to a single linear operation. When a layer has multiple outputs, the dot product becomes a matrix multiplication which we plug in to the formulation of multi-layer neural networks:

$$h^{(0)}(\boldsymbol{x}) = \boldsymbol{x} \tag{3.4}$$

$$h^{(l)}\left(\boldsymbol{x}; \boldsymbol{\theta}^{(l)}\right) = \text{act}\left(\boldsymbol{W}^{(l)} h^{(l-1)}(\boldsymbol{x}) + \boldsymbol{b}^{(l)}\right) \quad , \quad l = 1, \ldots, n \quad , \tag{3.5}$$

where $\boldsymbol{\theta}^{(l)} = \{\boldsymbol{W}^{(l)}, \boldsymbol{b}^{(l)}\}$ are the layer-wise parameters. Figure 10 depicts this architecture.

The goal of neural network is to learn the function mapping f from given example pairs $\boldsymbol{x}, \boldsymbol{y}$. Training the network consists of optimizing the network parameters $\boldsymbol{\theta}$ wrt. some loss function $\mathcal{L}$ between the predicted output and the real output. For example, we can choose the squared Euclidean distance:

$$\mathcal{L}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \|\boldsymbol{y} - \hat{\boldsymbol{y}}\|^2 \tag{3.6}$$

Optimization is usually limited to first-order methods due to the high number of network parameters. To efficiently compute parameter gradients, we use the backpropagation algorithm (Rumelhart et al. 1986) that takes a dynamic programming approach using the chain rule.

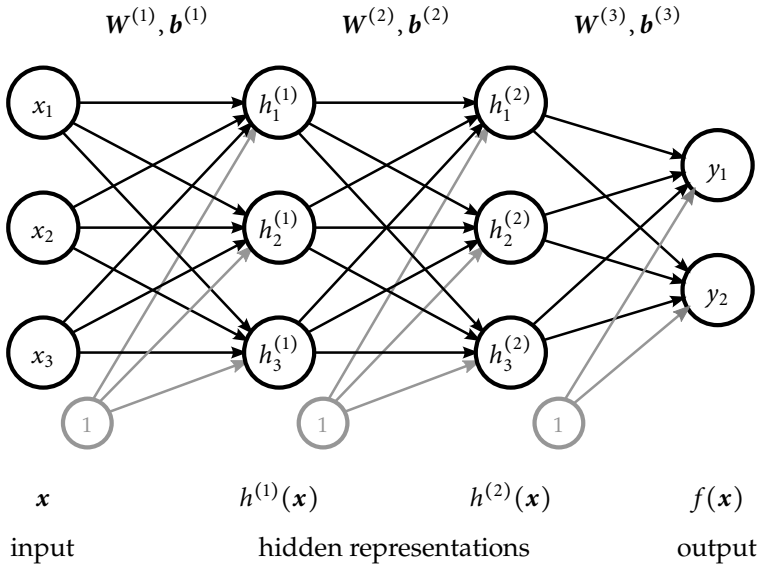


Figure 10: Vanilla feedforward neural network. The hidden representations come from the layered composition of f. Black arrows represent connectivity of the weight matrices $\boldsymbol{W}$ while grey arrows represent connectivity of the bias terms $\boldsymbol{b}$

3.2 Convolutional neural networks

When the network input is an image, we can exploit its spatial structure by replacing fully-connected with convolutional layers. Thus, the weights take the form of convolutional filters. We call this architecture a *convolutional neural network* (convnet) as pioneered by Fukushima (1980) and LeCun et al. (1998). See Figure 11 for a simple comparison between fully-connected and 1D convolutional connectivity. The 2D convolution operation computes a each pixel u in the output image $\boldsymbol{U}$ as a weighted sum of the local neighborhood around the corresponding pixel from the input image $\boldsymbol{X}$.

$$u_{i,j} = \text{act}\left(b + \sum_{k=1}^{c}\sum_{i'=1}^{n'}\sum_{j'=1}^{m'} w_{k,i',j'} \cdot x_{k,i+i',j+j'}\right) \tag{3.7}$$
$$\boldsymbol{U} \in \mathbb{R}^{n\times m}\ ,\ \boldsymbol{X} \in \mathbb{R}^{c\times n\times m}\ ,\ \boldsymbol{W} \in \mathbb{R}^{c\times n'\times m'}$$

Here, the filter response is summed over all c input image channels and $n' \times m'$ is the window size of the convolutional filter and $n \times m$ is the size of the image. To obtain multiple output channels, one has to stack the results from multiple convolutions. By construction, the operation is translation equivariant meaning that a translation in the original input followed by convolution is equivalent to a convolution followed by a translation. In

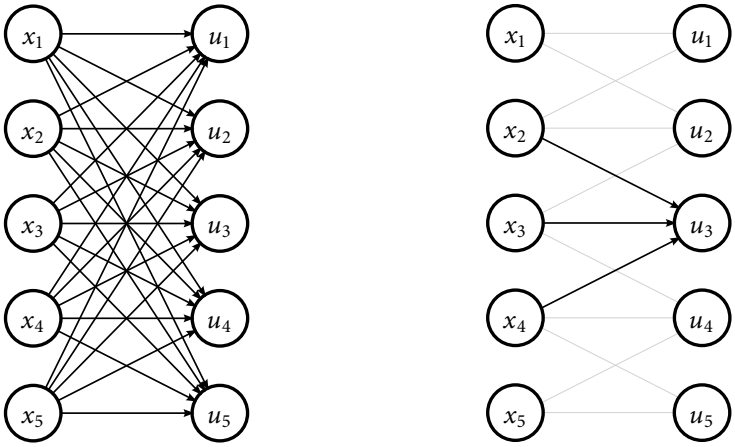


Figure 11: Full connectivity versus 1D convolutional connectivity. The convolution window spans 3 elements. By sliding the windows over the 1D input signal, we can compute an output with same spatial layout as the input and where output elements are only connected to neighboring input elements.

relation to the concept of invariance (Eq. 2.1), we define equivariance as

$$\text{transf}(\text{feat}(x)) = \text{feat}(\text{transf}(x)) \quad . \tag{3.8}$$

This property is especially beneficial in deep architectures because translation of the input will not change feature representations up in the network but simply translate them as well allowing subsequent layers to also exploit the spatial structure. Comparing convolutional to full connectivity, convolution causes a substantial reduction in the number of parameters since the weights only span a neighborhood. The fewer parameters means that the convolutional layers have lower capacity making them less prone to overfitting.

Another important component in convnets is pooling/subsampling. Images are high-dimensional and not really suitable for taking decisions at an abstract level. Therefore, we subsample by a factor s along the spatial dimensions of the image to reduce dimensionality. The subsampling is usually preceded by a pooling function to aggregate features locally, e.g. by

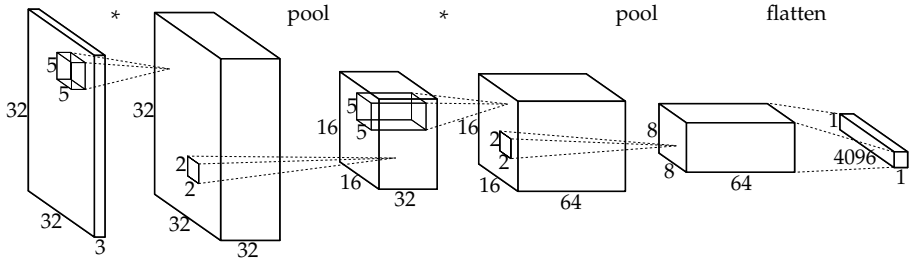


Figure 12: 2D convnet taking a 32 × 32 image with 3 color channels as input. Convnets are usually designed to such that upper layers trade off spatial resolution for features.

taking the average or the maximum value in a $n' \times m'$ pixel neighborhood:

$$\text{avg.:} \quad u_{k,i,j} = \frac{1}{n'm'} \sum_{i'=1}^{n'} \sum_{j'=1}^{m'} x_{k,i+i',j+j'} \tag{3.9}$$

$$\text{max:} \quad u_{k,i,j} = \max_{\substack{i' \in \{1,\dots,n'\}, \\ j' \in \{1,\dots,m'\}}} x_{k,is+i',js+j'} \tag{3.10}$$

$$\boldsymbol{U} \in \mathbb{R}^{c \times n \times m} \ , \ \boldsymbol{X} \in \mathbb{R}^{c \times n \times m} \ ,$$

Note that the aggregation is performed channel-wise. Pooling is an important part of convnets because we loose the exact spatial origins of the features thereby causing invariance to small image translations.

In Figure 12, we show a convnet with two consecutive blocks of convolution and pooling/subsampling. The last step of the convnet is a flattening operation in which we simply bring the image structure on a vector form allowing it to be processed further in a fully-connected manner.

In terms of image representation, convnets enforce a hierarchical feature disentangling in the network because of the limited spatial support of the weights. E.g. at a low level we may learn features corresponding to small objects such as an eye because the weights cannot span a larger area. At a higher level, the convolutional weights span a larger area and can begin to capture larger objects in terms of the learned underlying features, e.g. that a face is constructed from eyes and a mouth.

3.3 Encoder-decoder networks

By construction, neural networks lend themselves to supervised learning as they require both an input and a target output to produce the error signal required for training. However, by creatively combining neural network

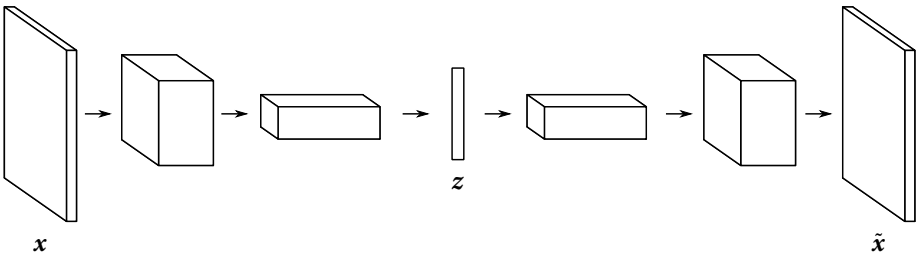


Figure 13: Overview of a 2D convolutional autoencoder.

functions and devising alternative targets it is possible to create a meaningful error signal without the use of supervised information. While the networks are not trained towards solving a particular task, this form of unsupervised training often serves the purpose of *feature learning*. That is, learning features that may be useful in other contexts.

A popular architecture for unsupervised learning is the autoencoder (Bourlard et al. 1988). The autoencoder takes an input $\boldsymbol{x}$ and encodes it using a neural network enc to a low-dimensional representation $\boldsymbol{z}$. We then employ a second neural network dec to decode $\boldsymbol{z}$ and reconstruct the original input as $\tilde{\boldsymbol{x}}$:

$$\boldsymbol{z} = \text{enc}(\boldsymbol{x}; \boldsymbol{\theta}_{\text{enc}}) \quad , \quad \tilde{\boldsymbol{x}} = \text{dec}(\boldsymbol{z}; \boldsymbol{\theta}_{\text{dec}}) \tag{3.11}$$

We train this setup by optimizing the parameters of both networks according to a chosen loss function measuring the discrepancy between the input and its reconstruction, e.g. the Eucliean distance (Eq. 3.6). Because the latent space is low-dimensional compared to the input, the $\boldsymbol{z}$ representation becomes a bottleneck forcing the encoder network to disentangle features to represent them in a compact manner.

The decoder network architecture defies the traditional view on neural networks as functions that condense high-dimensional inputs to low-dimensional outputs. For many years, decoder networks have mainly been fully-connected architectures capable of generating small images of digits. Recent convnet-based decoders, however, have convincingly shown capable of producing high-dimensional structured outputs like natural images (Dosovitskiy et al. 2015; Kulkarni et al. 2015; Larsen, Sønderby, et al. 2016; Radford et al. 2015). The architecture of a convnet decoder is constructed inversely to the encoder in order to dilute the high-level information back to the image space, see Figure 13 for a depiction.

Lately, more general forms of encoder-decoder architectures have become popular across many data domains. Instead of reconstructing an

input, they exploit the context in which the input occurs as supervisory signal during training. For words or sentences as input the decoder tries to predict preceding/succeeding text (Kiros et al. 2015; Mikolov et al. 2013). For images, the context can be spatial (Doersch et al. 2015). For video, the decoder performs sequence prediction (Lotter et al. 2016; Mathieu et al. 2015). The encoder-decoder approach has also shown applicability across different input domains. E.g. for machine translation between different written languages (Sutskever et al. 2014) or for image captioning (Mao et al. 2014).

Encoder-decoder networks have played an important part among advances towards unsupervised learning. In the beginning of the deep learning wave, autoencoders were used to pretrain network parameters to prevent networks from overfitting during subsequent supervised training. This went out of fashion as network training techniques improved through better regularization and more efficient error signal propagation (Glorot et al. 2011; Srivastava et al. 2014). Though there since have been attempts at unsupervised image feature learning for supervised tasks, the gains of this approach have not yet been substantial for natural images (Makhzani et al. 2015; Zhao et al. 2015). In comparison, encoder-decoder architectures have had better luck for semi-supervised learning driving generative models. In this setup, the decoder network acts as teacher in the unlabeled case for the discriminative model. If the discriminative model predicts incorrectly, the generative model will produce a poor reconstruction allowing the discriminative model to learn from the mistake. This approach has led to substantial performance improvements on problems with very few labeled examples (Maaløe et al. 2016; Rasmus et al. 2015).

3.4 Contribution: Learned image segmentation for automatic particle size measurements

In this section, we describe an image segmentation method that relies on a learned pixel classification. The related paper is attached in Appendix C:

– Larsen, A. B. L., Schultz, L. N., Dahl, A. B., et al. (2016). "Automatic particle size measurements from a learned image segmentation". In: *submitted*.

The goal is to segment images of calcite particles in *scanning electron microscopy* (SEM) images in order to measure the particle sizes for further analysis. Because the particles agglomerate and are difficult to separate, the measuring task is performed manually. We suggest that the user instead annotates a small training image from which we can learn a pixel-wise

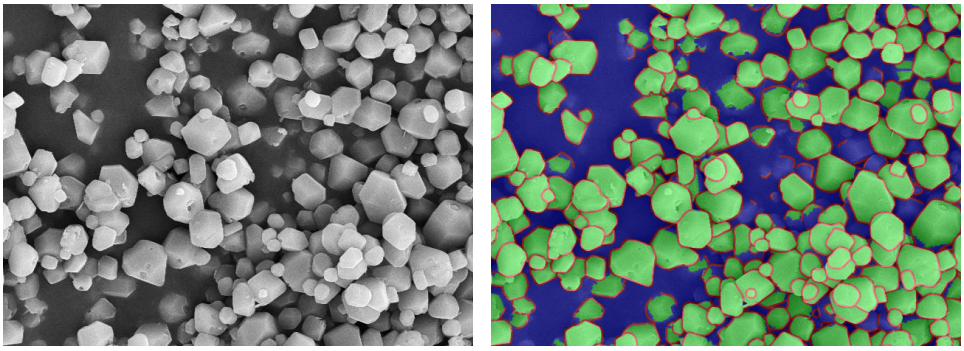


Figure 14: Image of calcite particles with annotations for learning a pixel classifier. In the right image, the three colors represent the classes: crystal (green), border (red) and background (blue). For simplicity, we show only a single class per pixel, though, we allow pixels belonging to multiple classes.

classifier. The annotation assigns the class labels *particle*, *particle border* and *background* for each pixel, see Figure 14. We then learn a feedforward convnet to predict class probabilities for all pixels. Using the probability images, the segmentation task can be solved using simple image processing techniques.

Like it has been the case for feature engineering, the devil is in the details for feature learning. Initially, we cast the prediction problem as ordinary multi-class classification where each pixel could only belong to one class. However, this is not ideal because pixels on the border between particles are also particle pixels. Instead, we let the network predict separate binary classification tasks which yields a better probability images for the segmentation task.

3.5 Contribution: Autoencoding beyond pixels using a learned similarity measure

In this section, we present an extension of the autoencoder framework addressing the issues of element-wise similarity measures for images. The related paper is attached in Appendix D:

– Larsen, A. B. L., Sønderby, S. K., Larochelle, H., et al. (2016). "Autoencoding beyond pixels using a learned similarity metric". In: *Proceedings of the 33rd International Conference on Machine Learning (ICML)*.

We argue against element-wise loss functions that are usually employed when training encoder-decoder architectures for images. The loss does not reflect the properties of visual perception, e.g. by being sensitive to translations, which causes decoder networks to generate unnatural and

Figure 15: Autoencoder reconstructions of face images. The VAE model is trained with ordinary pixel-wise measures. The VAE/GAN model is trained with feature-wise similarity measures which allows it to generate sharp and more natural-looking image structures.

overly smoothed images. To go beyond element-wise measures, we propose to use a learned feature representation of the images instead. This idea is inspired by the recent findings regarding convnet feature representations for images. Notably, Gatys et al. (2015) have demonstrated impressive results by combining two images in terms of their convnet feature representation to form a third image combining subject and style from the inputs.

To implement our idea, we combine a *variational autoencoder* (VAE) (Kingma et al. 2014; Rezende et al. 2014) with a *generative adversarial network* (GAN) (Goodfellow et al. 2014) by letting their decoder/generator networks share parameters. We train the two models together and express the log-likelihood (the reconstruction error) of the VAE using a feature representation of the GAN discriminator.

We train our method on a dataset of face images (Liu et al. 2015) and show that the generated images have higher visual fidelity and are capable of better imitating dataset samples as shown in Figure 15. Moreover, high-level facial features like glasses or bangs are truly disentangled in the latent representation $\boldsymbol{z}$ of the autoencoder as we can express these using simple vector arithmetic in $\boldsymbol{z}$ space.

4 Discussion and conclusion

This thesis places itself between two different approaches to representing images for visual recognition, namely engineering versus learning. In the following, we compare the two from a more general perspective.

Traditional features for computer vision are based on low-level image structures often extracted using hand-crafted convolutional filters. These filters usually contain edges or blobs which have also been shown to exist as visual stimuli in nature (Hubel et al. 1962; Marĉelja 1980; Young 1987). Interestingly, the same structures occur in the first layers of convnets when training on large natural image datasets like ImageNet (Russakovsky et al. 2015). This may suggest that the first levels of visual perception in nature are the result of a similar optimization problem. See Figure 16 where we show hand-engineered filters (Leung et al. 2001) in comparison with learned filters (Krizhevsky et al. 2012).

On top of low-level image structure, the two approaches start diverging. For hand-crafted feature descriptions, we typically summarize feature responses locally in histograms and sample the histograms at a lower spatial resolution. Like pooling in convnets, this introduces robustness towards translation and reduces the feature dimensionality. Moreover, hand-crafted features are built with robustness towards image rotations which can easily be achieved for e.g. gradient orientation histograms by smoothing bin contributions. We can build similar properties with convnets if we apply convolutional filters rotated at different angles followed by a pooling operation over the rotated feature responses (Cohen et al. 2016). For this approach

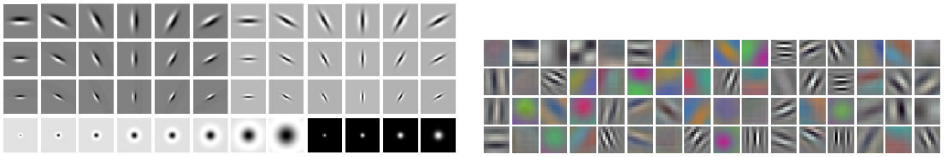


Figure 16: Hand-crafted convolutional filters (left) versus learned convolutional filters (right). Image sources: `http://www.robots.ox.ac.uk/~vgg/research/texclass/filters.html` and `https://github.com/akrizhevsky/cuda-convnet2`.

to be tractable, however, efficient parallel implementations are required which is a nontrivial task. Explicit rotation invariance/equivariance has not yet been shown to be worthwhile for natural images indicating that convnets are already capable of learning some robustness during training. In fact, robustness towards small rotations is implicitly modeled in most dataset augmentation schemes (Hauberg et al. 2016; Simard et al. 1992).

Hand-engineered features are often criticized for being hand-engineered. That is, humans have made qualified decision about feature extraction parameters (e.g. the scales α, β, σ from the locally orderless image representation). To some extent, the same criticism also applies to neural networks and especially convnets that have plenty of hyperparameters mostly due to the many architectural choices that go into designing an efficient network. In fact, the explosion of architectural tweaks and alternative layer formulations in recent deep learning literature is reminiscent of the feature description literature following the popularization of SIFT. E.g. He et al. (2015), Jarrett et al. (2009), Springenberg et al. (2014), Szegedy et al. (2015), Urban et al. (2016), and Zagoruyko et al. (2016) versus Bay et al. (2008), Dalal et al. (2005), Larsen et al. (2012), Mikolajczyk et al. (2005), and Tola et al. (2010). That said, the choice of image structure captured in hand-crafted features is fixed compared to the structure that convnets learn.

In terms of feature capacity/complexity, the continuing performance improvements using deep convnet representations (Russakovsky et al. 2015) leave little doubt that traditional computer vision features are no longer competitive for describing images structures much beyond the texture level. Hand-engineering simply does not scale to abstract concept description because we cannot describe the variability of object appearances in a meaningful way. For example describing the appearance of a person with a standing posture is straightforward. But when we have to take into account the variation caused by lighting, viewpoint, changing postures, occlusion, etc., the appearance variations grow exponentially. In comparison, convnets have been shown to learn robust high-level object detectors (e.g. faces) as part of their intermediate feature representation (Yosinski et al. 2015). This is particularly intriguing since we are not modeling any prior knowledge about the visual variations of objects which is often the case for traditional image features.

Today, one of the biggest caveats of deep learning is the need for large quantities of labeled training data in order learn general object representations and avoid overfitting. Unfortunately, labeled datasets are expensive to construct which prohibits deep learning from being applied to a large number of visual recognition problems, especially in many research domains where the amount of data is also scarce. While transfer learning has

shown to works surprisingly well for convnet features (Yosinski et al. 2014), it cannot be applied across widely different image domains. In this light, the recent progress of deep unsupervised learning is of great importance with the prospects of facilitating representations learning in many application areas where feature engineering is still king.

Bibliography

Andreopoulos, A. and Tsotsos, J. K. (2013). "50 Years of object recognition: Directions forward". In: *Computer Vision and Image Understanding* 117.8, pp. 827–891.

Bay, H., Ess, A., Tuytelaars, T., and Gool, L. V. (2008). "Speeded-Up Robust Features (SURF)". In: *Computer Vision and Image Understanding* 110.3. Similarity Matching in Computer Vision and Multimedia, pp. 346–359.

Bengio, Y., Courville, A., and Vincent, P. (2013). "Representation Learning: A Review and New Perspectives". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8, pp. 1798–1828.

Bourlard, H. and Kamp, Y. (1988). "Auto-association by multilayer perceptrons and singular value decomposition". In: *Biological Cybernetics* 59.4, pp. 291–294.

CiteSeerX (2016). *Most Cited Computer Science Articles*. URL: `http://citeseer.ist.psu.edu/stats/articles` (visited on 05/22/2016).

Cohen, T. S. and Welling, M. (2016). "Group Equivariant Convolutional Networks". In: *CoRR* abs/1602.07576.

Crosier, M. and Griffin, L. (2010). "Using Basic Image Features for Texture Classification". English. In: *International Journal of Computer Vision* 88.3, pp. 447–460.

Csurka, G., Dance, C., Fan, L., Willamowski, J., and Bray, C. (2004). "Visual categorization with bags of keypoints". In: *Workshop on statistical learning in computer vision, ECCV*. Vol. 1. 1-22. Prague, pp. 1–2.

Cybenko, G. (1989). "Approximation by superpositions of a sigmoidal function". In: *Mathematics of Control, Signals and Systems* 2.4, pp. 303–314.

Dalal, N. and Triggs, B. (2005). "Histograms of oriented gradients for human detection". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 1, 886–893 vol. 1.

Doersch, C., Gupta, A., and Efros, A. A. (2015). "Unsupervised Visual Representation Learning by Context Prediction". In: *The IEEE International Conference on Computer Vision (ICCV)*.

Dosovitskiy, A., Springenberg, J. T., and Brox, T. (2015). "Learning to Generate Chairs with Convolutional Neural Networks". In: *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ejiri, M. (2007). "Machine Vision in Early Days: Japan's Pioneering Contributions". In: *Proceedings of the 8th Asian Conference on Computer Vision - Volume Part I*. ACCV'07. Tokyo, Japan, pp. 35–53.

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., and Ramanan, D. (2010). "Object Detection with Discriminatively Trained Part-Based Models". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.9, pp. 1627–1645.

Fukushima, K. (1980). "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position". In: *Biological Cybernetics* 36.4, pp. 193–202.

Gatys, L. A., Ecker, A. S., and Bethge, M. (2015). "A Neural Algorithm of Artistic Style". In: *CoRR* abs/1508.06576.

Glorot, X., Bordes, A., and Bengio, Y. (2011). "Deep sparse rectifier neural networks". In: *International Conference on Artificial Intelligence and Statistics*, pp. 315–323.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems 27*, pp. 2672–2680.

Hauberg, S., Freifeld, O., Larsen, A. B. L., III, J. W. F., and Hansen, L. K. (2016). "Dreaming More Data: Class-dependent Distributions over Diffeomorphisms for Learned Data Augmentation". In: *Proceedings of the 19th international Conference on Artificial Intelligence and Statistics (AISTATS)*. Vol. 51, pp. 342–350.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). "Deep Residual Learning for Image Recognition". In: *CoRR* abs/1512.03385.

Hobson, P., Lovell, B. C., Percannella, G., Vento, M., and Wiliem, A. (2015). "Benchmarking human epithelial type 2 interphase cells classification methods on a very large dataset". In: *Artificial Intelligence in Medicine* 65.3, pp. 239–250.

Hornik, K. (1991). "Approximation capabilities of multilayer feedforward networks". In: *Neural Networks* 4.2, pp. 251–257.

Hubel, D. H. and Wiesel, T. N. (1962). "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex". In: *The Journal of Physiology* 160.1, pp. 106–154.

Jarrett, K., Kavukcuoglu, K., Ranzato, M., and LeCun, Y. (2009). "What is the Best Multi-Stage Architecture for Object Recognition?" In: *Proc. International Conference on Computer Vision (ICCV'09)*.

Jégou, H., Douze, M., Schmid, C., and Pérez, P. (2010). "Aggregating local descriptors into a compact image representation". In: *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 3304–3311.

Jowett, B. (1941). *Plato's The Republic*. New York: The Modern library.

Kashioka, S., Ejiri, M., and Sakamoto, Y. (1976). "A Transistor Wire-Bonding System Utilizing Multiple Local Pattern Matching Techniques". In: *IEEE Transactions on Systems, Man, and Cybernetics* SMC-6.8, pp. 562–570.

Kingma, D. P. and Welling, M. (2014). "Stochastic Gradient VB and the Variational Auto-Encoder". In: *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*.

Kiros, R., Zhu, Y., Salakhutdinov, R. R., Zemel, R., Urtasun, R., Torralba, A., and Fidler, S. (2015). "Skip-Thought Vectors". In: *Advances in Neural Information Processing Systems 28*, pp. 3294–3302.

Koenderink, J. J. and Doorn, A. J. van (1992). "Surface shape and curvature scales". In: *Image and Vision Computing* 10.8, pp. 557–564.

Koenderink, J. and Van Doorn, A. (1999). "The Structure of Locally Orderless Images". English. In: *International Journal of Computer Vision* 31.2-3, pp. 159–168.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems 25*, pp. 1097–1105.

Kulkarni, T. D., Whitney, W. F., Kohli, P., and Tenenbaum, J. (2015). "Deep Convolutional Inverse Graphics Network". In: *Advances in Neural Information Processing Systems 28*, pp. 2539–2547.

Larsen, A. B. L. (2014). *CUDArray: CUDA-based NumPy*. Tech. rep. DTU Compute 2014-21. Department of Applied Mathematics and Computer Science, Technical University of Denmark.

Larsen, A. B. L. (2016). *DeepPy: Pythonic deep learning*. Tech. rep. DTU Compute 2016-6. Department of Applied Mathematics and Computer Science, Technical University of Denmark.

Larsen, A. B. L., Dahl, A. B., and Larsen, R. (2015). "Oriented Shape Index Histograms for Cell Classification". In: *Proceedings of the 19th Scandinavian Conference Image Analysis (SCIA)*, pp. 16–25.

Larsen, A. B. L., Darkner, S., Dahl, A. L., and Pedersen, K. S. (2012). "Jet-based local image descriptors". In: *Proceedings of the 12th European Conference on Computer Vision*, pp. 638–650.

Larsen, A. B. L., Hviid, M. S., Jørgensen, M. E., Larsen, R., and Dahl, A. L. (2014). "Vision-based method for tracking meat cuts in slaughterhouses". In: *Meat Science* 96.1, pp. 366–372.

Larsen, A. B. L., Schultz, L. N., Dahl, A. B., Larsen, R., and Sørensen, H. O. (2016). "Automatic particle size measurements from a learned image segmentation". In: *submitted*.

Larsen, A. B. L., Sønderby, S. K., Larochelle, H., and Winther, O. (2016). "Autoencoding beyond pixels using a learned similarity metric". In: *Proceedings of the 33rd International Conference on Machine Learning (ICML)*.

Larsen, A. B. L., Vestergaard, J. S., and Larsen, R. (2014). "HEp-2 Cell Classification Using Shape Index Histograms With Donut-Shaped Spatial Pooling". In: *Medical Imaging, IEEE Transactions on* 33.7, pp. 1573–1580.

Laws, K. I. (1980). "Rapid texture identification". In: *24th annual technical symposium*. International Society for Optics and Photonics, pp. 376–381.

Lazebnik, S., Schmid, C., and Ponce, J. (2006). "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories". In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. Vol. 2, pp. 2169–2178.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11, pp. 2278–2324.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). "Deep learning". In: *Nature* 521.7553, pp. 436–444.

Leung, T. and Malik, J. (2001). "Representing and Recognizing the Visual Appearance of Materials using Three-dimensional Textons". In: *International Journal of Computer Vision* 43.1, pp. 29–44.

Lindeberg, T. (1998). "Feature Detection with Automatic Scale Selection". In: *International Journal of Computer Vision* 30.2, pp. 79–116.

Liu, Z., Luo, P., Wang, X., and Tang, X. (2015). "Deep Learning Face Attributes in the Wild". In: *Proceedings of International Conference on Computer Vision (ICCV)*.

Lotter, W., Kreiman, G., and Cox, D. (2016). "Deep Predictive Coding Networks for Video Prediction and Unsupervised Learning". In: *CoRR* abs/1605.08104.

Lowe, D. G. (2004). "Distinctive Image Features from Scale-Invariant Keypoints". In: *International Journal of Computer Vision* 60.2, pp. 91–110.

Maaløe, L., Sønderby, C. K., Sønderby, S. K., and Winther, O. (2016). "Auxiliary Deep Generative Models". In: *Proceedings of the 33rd International Conference on Machine Learning (ICML)*.

Makhzani, A. and Frey, B. J. (2015). "Winner-Take-All Autoencoders". In: *Advances in Neural Information Processing Systems 28*, pp. 2791–2799.

Mao, J., Xu, W., Yang, Y., Wang, J., and Yuille, A. L. (2014). "Explain Images with Multimodal Recurrent Neural Networks". In: *CoRR* abs/1410.1090.

Marĉelja, S. (1980). "Mathematical description of the responses of simple cortical cells$*$". In: *J. Opt. Soc. Am.* 70.11, pp. 1297–1300.

Mathieu, M., Couprie, C., and LeCun, Y. (2015). "Deep multi-scale video prediction beyond mean square error". In: *CoRR* abs/1511.05440.

Mikolajczyk, K. and Schmid, C. (2005). "A performance evaluation of local descriptors". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27.10, pp. 1615–1630.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). "Distributed Representations of Words and Phrases and their Compositionality". In: *Advances in Neural Information Processing Systems 26*, pp. 3111–3119.

Nobel-Jørgensen, M., Nielsen, J. B., Larsen, A. B. L., Olsen, M. D., Frisvad, J. R., and Bærentzen, J. A. (2013). "Pond of illusion: interacting through mixed reality". In: *SIGGRAPH Asia 2013 Posters*. ACM, p. 26.

Ojala, T., Pietikainen, M., and Maenpaa, T. (2002). "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 24.7, pp. 971–987.

Papert, S. (1966). *The Summer Vision Project*. `https://dspace.mit.edu/handle/1721.1/6125`. MIT AI Memo 100, Massachusetts Institute of Technology, Project Mac.

Radford, A., Metz, L., and Chintala, S. (2015). "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks". In: *CoRR* abs/1511.06434.

Rasmus, A., Berglund, M., Honkala, M., Valpola, H., and Raiko, T. (2015). "Semi-supervised Learning with Ladder Networks". In: *Advances in Neural Information Processing Systems 28*, pp. 3546–3554.

Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). "Stochastic Backpropagation and Approximate Inference in Deep Generative Models". In: *Proceedings of The 31st International Conference on Machine Learning*, pp. 1278–1286.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). "Learning representations by back-propagating errors". In: *Nature* 323, pp. 533–536.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). "ImageNet

Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision (IJCV)* 115.3, pp. 211–252.

Schmidhuber, J. (2015). "Deep learning in neural networks: An overview". In: *Neural Networks* 61, pp. 85–117.

Simard, P., Victorri, B., LeCun, Y., and Denker, J. (1992). "Tangent Prop - A formalism for specifying selected invariances in an adaptive network". In: *Advances in Neural Information Processing Systems 4*, pp. 895–903.

Springenberg, J. T., Dosovitskiy, A., Brox, T., and Riedmiller, M. A. (2014). "Striving for Simplicity: The All Convolutional Net". In: *CoRR* abs/1412.6806.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research* 15, pp. 1929–1958.

Stallkamp, J., Schlipsing, M., Salmen, J., and Igel, C. (2011). "The German Traffic Sign Recognition Benchmark: A multi-class classification competition". In: *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pp. 1453–1460.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). "Sequence to Sequence Learning with Neural Networks". In: *Advances in Neural Information Processing Systems 27*, pp. 3104–3112.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). "Going Deeper With Convolutions". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tola, E., Lepetit, V., and Fua, P. (2010). "DAISY: An Efficient Dense Descriptor Applied to Wide-Baseline Stereo". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 32.5, pp. 815–830.

Ullman, S., Assif, L., Fetaya, E., and Harari, D. (2016). "Atoms of recognition in human and computer vision". In: *Proceedings of the National Academy of Sciences* 113.10, pp. 2744–2749.

Urban, G., Geras, K. J., Ebrahimi Kahou, S., Aslan, O., Wang, S., Caruana, R., Mohamed, A., Philipose, M., and Richardson, M. (2016). "Do Deep Convolutional Nets Really Need to be Deep (Or Even Convolutional)?" In: *CoRR* abs/1603.05691.

Veta, M., Diest, P. J. van, Willems, S. M., Wang, H., Madabhushi, A., Cruz-Roa, A., Gonzalez, F., Larsen, A. B. L., Vestergaard, J. S., Dahl, A. B., Ciresan, D. C., Schmidhuber, J., Giusti, A., Gambardella, L. M., Tek, F. B., Walter, T., Wang, C.-W., Kondo, S., Matuszewski, B. J., Precioso, F., Snell, V., Kittler, J., Campos, T. E. de, Khan, A. M., Rajpoot, N. M., Arkoumani, E., Lacle, M. M., Viergever, M. A., and Pluim, J. P. (2015). "Assessment of algorithms for mitosis detection in breast cancer histopathology images". In: *Medical Image Analysis* 20.1, pp. 237–248.

Wang, X., Han, T. X., and Yan, S. (2009). "An HOG-LBP human detector with partial occlusion handling". In: *2009 IEEE 12th International Conference on Computer Vision*, pp. 32–39.

Yin, F., Wang, Q. F., Zhang, X. Y., and Liu, C. L. (2013). "ICDAR 2013 Chinese Handwriting Recognition Competition". In: *2013 12th International Conference on Document Analysis and Recognition*, pp. 1464–1470.

Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. (2014). "How transferable are features in deep neural networks?" In: *Advances in Neural Information Processing Systems 27*, pp. 3320–3328.

Yosinski, J., Clune, J., Nguyen, A., Fuchs, T., and Lipson, H. (2015). "Understanding Neural Networks Through Deep Visualization". In: *Deep Learning Workshop, International Conference on Machine Learning (ICML)*.

Young, R. A. (1987). "The Gaussian derivative model for spatial vision: I. Retinal mechanisms". In: *Spatial Vision* 2.4, pp. 273–293.

Zagoruyko, S. and Komodakis, N. (2016). "Wide Residual Networks". In: *CoRR* abs/1605.07146.

Zhao, J., Mathieu, M., Goroshin, R., and LeCun, Y. (2015). "Stacked What-Where Auto-encoders". In: *CoRR* abs/1506.02351.

A HEp-2 cell classification using shape index histograms with donut-shaped spatial pooling

HEp-2 Cell Classification Using Shape Index Histograms With Donut-Shaped Spatial Pooling

Anders Boesen Lindbo Larsen*, *Student Member, IEEE*, Jacob Schack Vestergaard, and Rasmus Larsen

***Abstract*—We present a new method for automatic classification of indirect immunoflourescence images of HEp-2 cells into different staining pattern classes. Our method is based on a new texture measure called shape index histograms that captures second-order image structure at multiple scales. Moreover, we introduce a spatial decomposition scheme which is radially symmetric and suitable for cell images. The spatial decomposition is performed using donut-shaped pooling regions of varying sizes when gathering histogram contributions. We evaluate our method using both the ICIP 2013 and the ICPR 2012 competition datasets. Our results show that shape index histograms are superior to other popular texture descriptors for HEp-2 cell classification. Moreover, when comparing to other automated systems for HEp-2 cell classification we show that shape index histograms are very competitive; especially considering the relatively low complexity of the method.**

***Index Terms*—Cell classification, feature histograms, indirect immunofluorescence, shape index, spatial pooling, texture description.**

I. Introduction

INDIRECT immunofluorescence (IFF) of Human Epithelial Cells type 2 (HEp-2 cells) is a popular technique for diagnosing autoimmune diseases where the human defense mechanism erroneously produces antibodies against its own antigens. The test is carried out by searching for the presence of antibodies in a patient's blood serum: a serum sample is incubated with HEp-2 cells such that antibodies will bind to the antigens on the cells. Moreover, a fluorescent tag that binds to the antibodies is added to make them visible under a fluorescence microscope. The cell staining patterns are indicative for the type of autoimmune disease. The recognition of these patterns is a typical medical image analysis problem where reliable automation is the key to an efficient diagnosis. For a more thorough motivation and description of the problem, we refer the reader to the overview presented in [1].

In this work, we seek to develop an automatic method for classifying staining patterns of HEp-2 cell IIF images. We evaluate our method on the dataset from the *Competition on Cells Classification by Fluorescent Image Analysis* at ICIP 2013.[1] From the dataset, we are given 13 596 categorized cell images of 83 patients. Each cell belongs to one of the six classes:

- centromere, 2741 samples;
- golgi, 724 samples;
- homogeneous, 2494 samples;
- nucleolar, 2598 samples;
- nuclear membrane (NuMem), 2208 samples;
- speckled, 2831 samples.

Examples from each class are shown in Fig. 1. We see that the classification problem is largely a texture classification problem as the cell images contain little spatial structure but rather texture patterns.

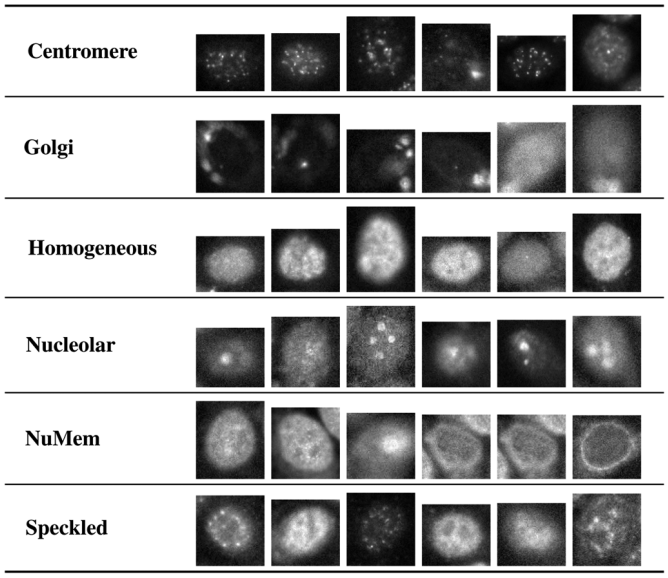


Fig. 1. Examples of the six different staining pattern classes. In general, we have good inter-class variation among the classes. Intra-class variation, however, may give rise to cases of doubt; e.g., the third speckled cell that looks similar the Centromere cells.

The novelty in our classification method consists of a texture descriptor based on second-order image structure captured by the shape index [2], [3]. We summarize the statistics of the shape index in histograms using a new pooling scheme consisting of concentric donut-shaped rings centered on the cell. This pooling scheme allows for a rotation invariant spatial decomposition of the cell image. The novelty of our method is in the feature description part of the classification pipeline, and hence, this ar-

Manuscript received February 19, 2014; revised April 13, 2014; accepted April 15, 2014. Date of publication April 18, 2014; date of current version June 27, 2014. *Asterisk indicates corresponding author.*

*A. B. L. Larsen is with the Department of Applied Mathematics and Computer Science, Technical University of Denmark, 2800 Kongens Lyngby, Denmark (e-mail: abll@dtu.dk).

J. S. Vestergaard and R. Larsen are with the Department of Applied Mathematics and Computer Science, Technical University of Denmark, 2800 Kongens Lyngby, Denmark (e-mail: jsve@dtu.dk; rlar@dtu.dk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMI.2014.2318434

[1]Competition website and dataset available at http://nerone.diiie.unisa.it/contest-icip-2013/index.shtml

ticle will not focus on the classifier built on top of the image features.

Our method has been evaluated at the ICIP 2013 competition and got the second place just 0.11% below the winner (the accuracy scores were 83.54% and 83.65%, respectively). Our method received the title *merit winner* as it was the top performer in most classes.

A. Related Work

Several methods have been proposed to perform staining pattern classification. At ICPR 2012, the first competition in the field was organized with a total of 28 teams participating [1], [4]. We categorize the different classification methods into the following two approaches. One approach is the popular bag of visual words (BoVW) pipeline where each method customize one or more of the pipeline stages. 1) Spatial decomposition (using cell boundary information [5], image intensities [6]). 2) Local feature description (gradient histograms [6], sparse coding [5], covariance of Gabor features, linear projections[7]). 3) Feature encoding (k-means, Fisher tensors [8]). 4) Classification (linear/kernel/multi-kernel support vector machines). The second approach skips the visual word encoding and train classifiers directly on a broad range of image features (*gray-level co-occurrence matrix* (GLCM) [9] or *local binary patterns* (LBP) [10], *scale-invariant feature transform* (SIFT) [11]) [12]–[15]. Among these methods, the top performer of ICPR 2012 employs an extension of LBPs where the spatial relations among adjacent LBPs are encoded in a rotation invariant manner [16]. Our method belongs to the second approach as we apply a classifier directly on top of the shape index features.

The shape index is a curvature measure that maps second-order differential structure to a real-valued index. In recent years it has gained popularity for range imaging where it is used as a robust surface descriptor, typically for visual recognition [17], [18]. In some cases, the shape index is summarized in histograms [19]–[21] following the trend of the popular local image features like SIFT and *histograms of oriented gradients* (HOG) [22]. A related approach to texture description is *basic image features* (BIF) [23] based on first- and second-order differential structure. This descriptor captures multi-scale differential structure by summarizing the distribution of quantized image structure in a histogram. Because the histogram captures the joint distribution over image structure at multiple scales, it becomes high dimensional (1296 dims) and sparse.

Spatial pooling schemes for histogram-based local feature descriptors exist in different flavors. The traditional 4×4 grid decomposition of SIFT with spatial interpolation across adjacent histograms is by far the most popular. Alternatives include GLOH [24] with a log-polar grid and DAISY [25] with Gaussian windows sampled in circles with increasing radii. None of these pooling schemes are rotation invariant since their goal is to provide a distinct image structure description. For feature description tasks that require rotation invariance, a dominant orientation of the local image structure is typically detected and the description rotated accordingly. However, this is not possible for cell classification because the cell images have no distinct orientation. Therefore, we seek a spatial decomposition that is also rotation invariant. Our donut-shaped rings are similar to the ones proposed in the RIFT descriptor [26]. Note that the donuts do not require information about the HEp-2 cell borders like the spatial decomposition schemes presented in [5].

B. Contributions

In this paper, we do the following.

- Introduce shape index histograms for texture description.
- Propose a rotation-invariant spatial pooling scheme for the shape index histograms.
- Perform a comparative evaluation of our texture descriptor against other popular texture measures using the ICIP 2013 dataset.
- Show that: 1) shape index histograms are very effective at discriminating between the HEp-2 cell staining patterns compared to other popular texture measures; 2) the spatial decomposition increases the discriminative performance.

II. Shape Index Histograms

The shape index is an image geometry measure proposed by Koenderink and van Doorn [2]. It captures second-order image structure in a continuous interval which allows us to summarize the distribution of curvatures in a histogram. In the following we shall first see how the shape index is calculated, then how we can represent shape index statistics. Note that we have made our implementation of shape index histograms available online.[2]

A. Differential Image Structure

We formulate the shape index using the Gaussian scale-space framework [27] where the linear scale-space of a 2-D image $I(\boldsymbol{x}) : \Omega \rightarrow \mathbb{R}, \Omega \subseteq \mathbb{R}^2$, defined by

$$L(\boldsymbol{x}; \sigma) = (G * I)(\boldsymbol{x}; \sigma), \quad \boldsymbol{x} = (x, y). \tag{1}$$

Convolution is denoted by $*$ and G is the Gaussian aperture function

$$G(\boldsymbol{x}; \sigma) = \frac{1}{(\sigma\sqrt{2\pi})^2} \exp\left(-\frac{\boldsymbol{x}^\top \boldsymbol{x}}{2\sigma^2}\right), \quad \sigma > 0 \tag{2}$$

where σ (also known as the *inner scale* parameter) defines the width of the Gaussian kernel. The semicolon in $L(\boldsymbol{x}; \sigma)$ indicates that the convolution is performed over the spatial coordinates $\boldsymbol{x}$ while the arguments following the semicolon are parameters of the observation.

We compute the differential structure of the image from the normalized scale space derivatives

$$\begin{aligned} L_{x^n y^m}(\boldsymbol{x}; \sigma) &= \sigma^{n+m} \frac{\partial^{n+m}}{\partial x^n \partial x^m} (G * I)(\boldsymbol{x}; \sigma) \\ &= \sigma^{n+m} \left(\left(\frac{\partial^{n+m}}{\partial x^n \partial y^m} G \right) * I \right)(\boldsymbol{x}; \sigma) \end{aligned} \tag{3}$$

where n and m indicate the order of the differential along the x and y axis, respectively. For notational convenience we omit the arguments and substitute $L_{x^n y^m}(\boldsymbol{x}; \sigma)$ with simply $L_{x^n y^m}$. Thus, $L_{x^n y^m}$ is implicitly assumed to be computed at some scale σ and location $\boldsymbol{x}$.

[2]Python implementation available at http://compute.dtu.dk/abll/

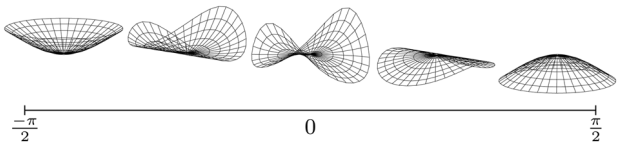


Fig. 2. Second-order curvatures along the shape index interval $]-\pi/2, \pi/2[$. We see cup, rut, saddle, ridge, and cap shapes. Shapes are aligned vertically and have the same curvedness.

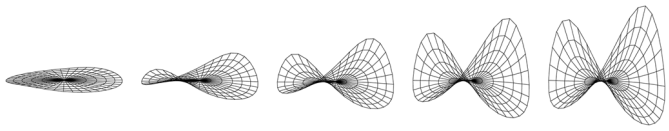


Fig. 3. Saddle shape $s = 0$ with increasing curvedness.

B. The Shape Index

The shape index is derived from the Hessian matrix $\nabla^2 L$ that captures the second-order curvature at scale σ

$$\nabla^2 L(\boldsymbol{x};\sigma) = \begin{bmatrix} L_{x^2} & L_{xy} \\ L_{xy} & L_{y^2} \end{bmatrix}. \tag{4}$$

The Hessian matrix is square and symmetric allowing us to compute the pair of real eigenvalues κ_1 and κ_2

$$\begin{aligned} \kappa_1(\boldsymbol{x};\sigma) &= \frac{1}{2}\left(L_{x^2}+L_{y^2}-\sqrt{(L_{x^2}-L_{y^2})^2+4L_{xy}^2}\right) \\ \kappa_2(\boldsymbol{x};\sigma) &= \frac{1}{2}\left(L_{x^2}+L_{y^2}+\sqrt{(L_{x^2}-L_{y^2})^2+4L_{xy}^2}\right) \end{aligned} \tag{5}$$

κ_1 and κ_2 are the *principal curvatures*. They describe the strength of the curvature along the *extremal directions* where the curvatures are minimal and maximal, respectively. Note that κ_1 and κ_2 are invariant to the orientation of the curvature. The eigenvectors of $\nabla^2 L$ capture the extremal directions. However, we discard these as we want our texture measure to be rotation invariant. The shape index $s \in](-\pi)/2), \pi/2[$ is defined as

$$s(\boldsymbol{x};\sigma) = \arctan\left(\frac{\kappa_1+\kappa_2}{\kappa_1-\kappa_2}\right). \tag{6}$$

The shape index has the attractive property that it maps all second-order shapes onto a continuous interval providing a smooth and intuitive transition between the shapes, see Fig. 2. In addition to the shape index a measure of *curvedness* c is defined

$$c(\boldsymbol{x};\sigma) = \sqrt{\kappa_1^2+\kappa_2^2}. \tag{7}$$

The curvedness indicates the strength of the shape described by the shape index. A small c means that the shape is almost flat and not very distinct while a large c means that the shape is very prominent, see Fig. 3 for an example.

An alternative formulation of s and c that skips the explicit calculation of the Hessian eigenvalues is given by

$$s(\boldsymbol{x};\sigma) = \arctan\left(\frac{-L_{x^2}-L_{y^2}}{\sqrt{(L_{x^2}-L_{y^2})^2+4L_{xy}^2}}\right) \tag{8}$$

$$c(\boldsymbol{x};\sigma) = \sqrt{L_{x^2}^2+2L_{xy}^2+L_{y^2}^2}. \tag{9}$$

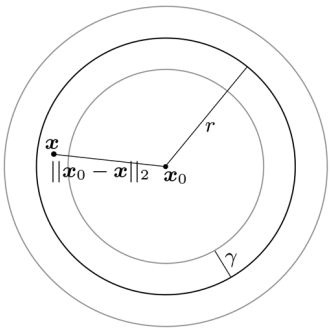


Fig. 4. Diagram of the donut weight calculations. Black circle represents the mode of the distribution. Gray circles indicate the standard deviation γ.

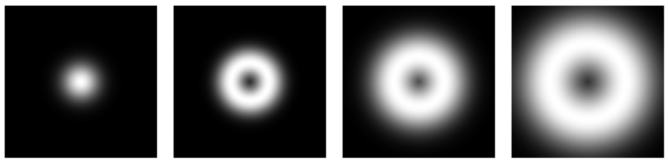


Fig. 5. Different donuts with increasing r and γ values. Note that we have set $r = 0$ in the first donut which results in a Gaussian window.

As we shall see shortly, the measures s and c allow for a simple yet powerful texture description. For readers familiar with histograms of gradient orientations (e.g., as used in SIFT) we note that s can be thought of as the gradient orientation and c as the gradient magnitude which is used for weighting the histogram contribution of the gradient orientation.

To construct multi-scale shape index responses, we propose to select n_σ different scales such that

$$\sigma_i = \sigma_{\text{base}} \cdot \sigma_{\text{ratio}}^i, \quad i = 1, \ldots, n_\sigma. \tag{10}$$

σ_{base} is the smallest shape index scale and σ_{ratio} is the ratio between σ_i and σ_{i+1}. Typically, $\sigma_{\text{ratio}} = 2$ as this is the natural step size in order to utilize efficient factor 2 subsampling.

C. Spatial Decomposition

Inspired by the success of feature descriptors like SIFT, we propose to represent an image as a concatenation of histograms gathered from a spatial decomposition of the image. However, we cannot split a cell image in e.g., a 4×4 grid like SIFT, because the cells have no fixed orientation. Instead, we propose an alternative spatial pooling scheme using overlapping concentric donut-shaped rings. A donut D centered around the cell center $\boldsymbol{x}_0$ is calculated from

$$D(\boldsymbol{x};\boldsymbol{x}_0,\gamma,r) = \exp\left(-\frac{(\|\boldsymbol{x}_0-\boldsymbol{x}\|_2-r)^2}{2\gamma^2}\right). \tag{11}$$

The parameter γ determines the width of the Gaussian fall-off from the ring radius given by r. The idea is to use a donut for spatial weighting when gathering bin contributions for a histogram. This allows us to give large weights to shapes at distance r to the center. To illustrate the parameters we show a diagram of the donut calculation in Fig. 4. Moreover, in Fig. 5 we show examples of different donuts. We argue that the proposed spatial pooling scheme is well-suited for the cell staining

patterns (shown in Fig. 1), because the different texture patterns occur approximately at the same radius in all cell images. Thus, the donut shapes allow for spatial decomposition without committing to a specific cell orientation. Note that this spatial decomposition is a result of using different spatial weights when gathering histogram contributions. In comparison, SIFT and HOG subdivide the image and generate a histogram per sub-image.

D. Histogram Construction

We generate shape index histograms by choosing a set of n_b bin centers $b_1, \ldots, b_{n_b}$ uniformly distributed along the shape index interval $]-\pi/2, \pi/2[$. For a bin B centered at b, we calculate its total contributions from a weighted sum using both the spatial donut weighting D and the curvedness measure c,

$$B(\sigma, \boldsymbol{x}_0, \gamma, r, b, \beta) = \frac{\sum_{\boldsymbol{x}} Dc \exp\left(-\frac{(b-s)^2}{2\beta^2}\right)}{\sum_{\boldsymbol{x}} D}. \quad (12)$$

We apply a Gaussian window in the shape index range (aka. the tonal range) such that a shape index may contribute to neighboring bins. The tonal scale parameter β adjusts this smoothing. We also perform a normalization of the bin contributions relative to the total spatial weight of the donut. This makes histograms with spatial pooling from small donuts to have the same influence as histograms with spatial pooling from large donuts.

The reader familiar with Koenderink's *locally orderless image* formulation [28] should notice that the donuts serve as spatial apertures from which feature response are gathered.

At this point, it should be clear that (12) allows us to construct histograms capturing image structure at different scales (σ) and with different spatial pooling (γ and r). Thus, we can construct a feature vector by concatenating histograms with different parameters such that image structure at multiple scales is captured with different spatial decompositions.

As the final step we normalize the feature vector. That is, we normalize across all histograms in order to make the description robust to image contrast variations among the cell images. We experimented with different normalization schemes that try to diminish the dominance of large bin values in order to emphasize the influence of smaller bin values get higher influence. For example, the normalization proposed by Lowe for SIFT (L_2 normalization followed by clipping followed by L_2 normalization) or the square root normalization described in [29] (L_1 normalization followed by element-wise square rooting followed by L_2 normalization). However, we have not found significant performance gains from any of these techniques, and we therefore use plain L_1 normalization.

III. Experiments

In this section, we investigate the potential of shape index histograms for HEp-2 cell classification. For the first part we use the ICIP 2013 competition dataset for our experiments (only the training set since the test set has not been published). For the last part we evaluate shape index histograms on the ICPR 2012 dataset to compare our method with other HEp-2 cell classification methods.

A. Comparison With Other Texture Measures

First, we want to compare shape index histograms with other popular texture measures. For simplicity, we perform this experiment without any spatial decomposition. That is, we regard the cell classification task a pure texture classification problem with the cell image being the texture. For the comparison we have included the following textures descriptors.

- The GLCM properties *contrast*, *dissimilarity*, *homogeneity*, *energy*, *correlation*, and *angular second moment*. We calculate features from multiple GLCMs with different offsets.
- Local binary patterns (LBP). We use the rotation invariant uniform LBP and extract multiple histograms using LBPs with different radii.
- Basic image features (BIF) extracted over different scales. Because all cell images in the dataset are at the same scale, we disregard the scale-invariant similarity measure proposed in the original paper. Instead we use the RBF kernel of the SVM classifier.
- Shape index histograms (SIH) extracted at multiple scales.

We perform the classification using an RBF kernel SVM with a fixed penalty parameter $C = 1$ and a kernel coefficient $\gamma_{\mathrm{RBF}} = 1/N$ where N is the dimensionality of the feature vector. Multi-class support is achieved with a one versus one comparison scheme. We argue that this fixed classifier configuration is suitable for our comparison since the focus of this paper is on feature description and because our (nonshown) experiments indicate that only insignificant performance improvements can be achieved from adjusting these parameters. We assess classification performance from a leave-one-out cross-validation study across all 83 patients in the dataset. Because the patients have different numbers of cells, we measure performance as the weighted average over classification accuracies for the 83 patients where the weights are the number of cells per patient. In an attempt to make the comparison as unbiased as possible, we optimize the parameters for each texture measure using Bayesian optimization [30] with the framework provided in [31]. For each texture measure, we let the framework perform around 150 function evaluations before selecting the optimal configuration. To avoid selecting an accidentally good parameter setting among the 150 parameter configurations that overfits to the dataset, we perform the Bayesian optimization using cross-validation on 40 randomly selected patients.

For the GLCM features we have determined the number of grey-scale levels (135), the different offsets used (1, 3,. . ., 17), and the number of equally spaced angles (3). This gives a 180-dimensional feature vector. For the LBP we have determined the radii (2, 4, . . ., 20) and the number of angular samples (24). This gives a 275-dimensional feature vector. For the BIF we have determined the base scale (1.2), the number of scales (4), the scale ratio (2.3), and the noise threshold (0). Note that these parameters are fairly close to the ones proposed in the original paper. The BIF feature vector has 1296 elements. For the SIH we have determined the number of scale ($n_\sigma = 5$), the base scale ($\sigma_{\mathrm{base}} = 1$), scale ratio ($\sigma_{\mathrm{ratio}} = 2.25$), the number of bins per histogram ($n_b = 18$), and the tonal scale ($\beta = 0.22$). The SIH feature vector has dimensionality $n_\sigma \cdot n_b = 90$.

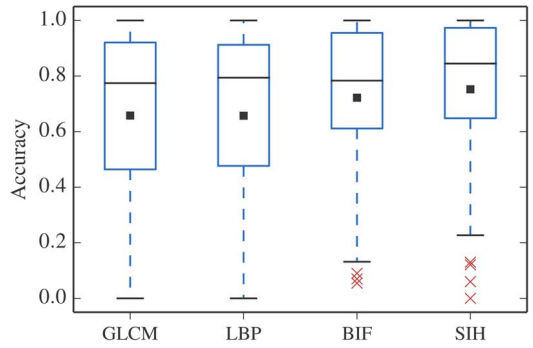


Fig. 6. Comparison of texture descriptors. Boxplot statistics are gathered from the cross-validation across 83 patients. Boxplot is generated using weighted percentiles with the weights being equal to the number of cells per patient. Black square indicates the weighted average accuracy. For the four texture measures the weighted average accuracies are 0.66, 0.66, 0.72, 0.75.

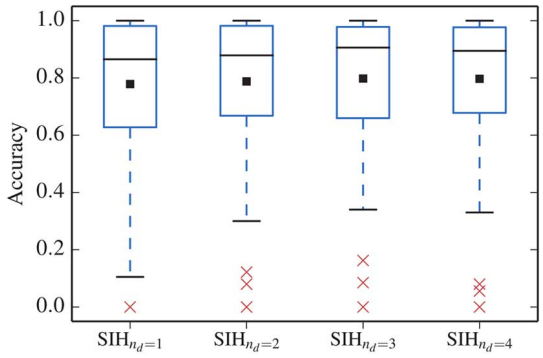


Fig. 7. Performance comparison of different donut configurations. Weighted average accuracies are 0.75, 0.78, 0.79, 0.80, 0.80.

The results are shown in Fig. 6. We see that the GLCM properties and the LBP yield similar accuracies around 0.66 while BIF scores around 0.72. Compared to these texture measures, SIH offers an improvement with an accuracy of 0.75. Moreover, the median accuracy of SIH is clearly higher than the other texture measures.

B. Spatial Decomposition

In the next experiment, we investigate how to augment the shape index histograms using our donut-shaped spatial pooling scheme. To minimize the number of free parameters, we use the SIH configuration from earlier and adjust only the donut parameters. We compare different spatial decompositions by fixing the number of donuts to $n_d = 1$,2,3,4 and optimizing the radii r and widths γ for each choice of n_d. We perform Bayesian optimization using cross-validation similar to the previous experiment. Thus, the dimensionality of the feature vectors are $90 \cdot n_d$.

The performance of the feature descriptors is shown in Fig. 7. Moreover, Fig. 8 shows the optimal donut configuration for the different n_d. Interestingly, we see both blobs and donuts among the donut configurations. There is a clear improvement by performing the spatial decomposition and it seems that the improvement converges at $n_d = 3$. We therefore choose $n_d = 3$, $r = \{0.0, 6.7, 7.8\}$, $\gamma = \{19.0, 8.0, 4.1\}$ as the optimal configuration yielding a weighted average accuracy of 0.80. We will refer to this configuration as SIH$_{n_d=3}$. The performance

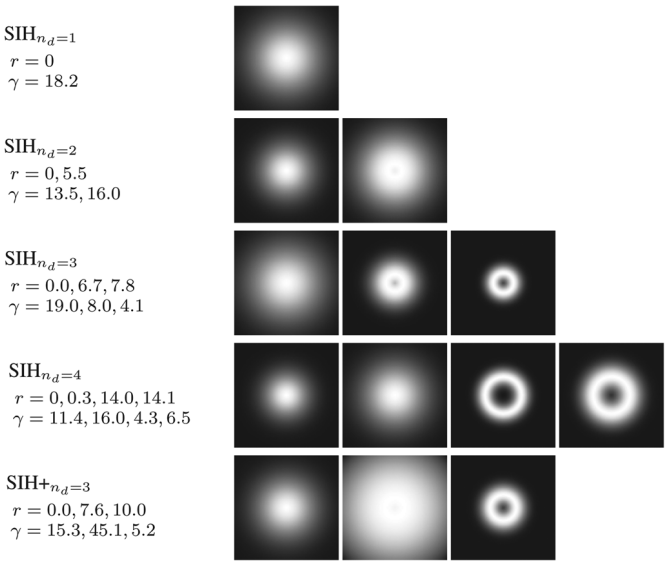


Fig. 8. Donut layouts for different choices of n_d. SIH+$_{n_d=3}$ denotes the final feature descriptor.

improvement from SIH to SIH$_{n_d=3}$ is significantly different at a 0.0038 significance level.[3]

C. Final Feature Descriptor

In our last experiment, we wish to find the optimal configuration of our feature description. We optimize over the joint parameter space of both shape index histograms and donuts. We choose $n_d = 3$ since this has shown a good trade-off between descriptor dimensionality and performance. Moreover, as the ICIP 2013 dataset provide more information per cell than just the image, we include the following in the feature vector.

- A single integer encoding the fluorescent intensity level of the cell image (*positive* or *intermediate*).
- Image features extracted from the binary cell mask (including area, eccentricity, major and minor axis length, perimeter, and different image moments). This adds 44 elements to the feature vector.

We have arrived at the following shape index parameters: $n_\sigma = 5$, $\sigma_{\text{base}} = 1.6$, $\sigma_{\text{ratio}} = 1.5$, $n_b = 18$, $\beta = 0.27$. Fig. 8 shows the optimized donut layout. This gives a total feature dimensionality of $1 + 44 + n_\sigma \cdot n_b \cdot n_d = 315$ We call this descriptor variant SIH+$_{n_d=3}$.

On the ICIP 2013 dataset we achieve a weighted average accuracy of 0.82 which is a small improvement over SIH$_{n_d=3}$ considering the extra information we are utilizing. This performance improvement is significantly different at a 0.0757 significance level. In Fig. 9 we show the classification performance for the different classes in the dataset. The Golgi and Speckled classes stand out as difficult classes with average accuracies lower than 0.82. We note in this connection that part

[3]Our hypothesis test for determining if descriptor A has greater performance than descriptor B is based on bootstrapping [32]. We generate 10 000 bootstrap samples. Each sample is obtained by drawing 83 patients with replacement and computing $\mu_{\text{diff}} = \mu_{\text{A}} - \mu_{\text{B}}$ where μ_{A} is descriptor A's weighted mean performance on the drawn patients. That is, our bootstrap estimator is the difference between the two weighted mean performances. We wish to test $H_0 : \mu_{\text{A}} \leq \mu_{\text{B}}$ versus $H_1 : \mu_{\text{A}} > \mu_{\text{B}}$. We report the significance level as the bootstrap estimated p-value $P(H_0 : \mu_A \leq \mu_B)$.

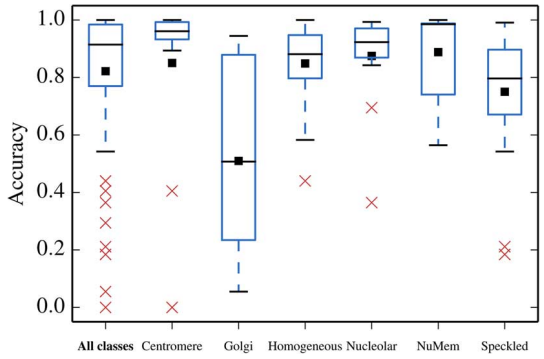


Fig. 9. Performance of SIH+$_{n_d=3}$ for the entire dataset and for the different staining patterns. Weighted average accuracy for the dataset is 0.82.

	Centromere	Golgi	Homogeneous	Nucleolar	NuMem	Speckled
Centromere	2331	3	0	121	4	282
Golgi	2	369	24	103	216	10
Homogeneous	5	4	2117	6	46	316
Nucleolar	185	27	19	2273	15	79
NuMem	2	71	131	26	1961	17
Speckled	187	10	388	108	14	2124

Fig. 10. Confusion matrix for SIH+$_{n_d=3}$. Rows are actual (true) labels. Columns are predicted labels.

of the explanation for the Golgi class performance may be that it contains cells from only four patients. Thus, we cannot expect the class variance to be represented by three patients when doing cross-validation. In Fig. 10 we show the confusion matrix among the different classes. In Fig. 11 we show examples of cell classifications. The misclassifications clearly illustrate the inter-class overlap, e.g., the confusion between nucleolar and homogeneous. Interestingly, the classification examples reveal which samples from class X that try to imitate class Y. For example, the Speckled cell patterns do a good job imitating the Centromere class. We also see a large intra-class variation by comparing the cell images in the same row, i.e., from the same class.

D. Classification Performance on the ICPR 2012 Dataset

To compare with other HEp-2 cell classification methods in the literature we have evaluated our method on the ICPR 2012 dataset [1]. Compared to the ICIP 2013 dataset, this dataset is smaller containing 1457 cells from 28 image slides. Moreover, the image acquisition setup differs and the staining classes are different (centromere, coarse speckled, cytoplasmic, fine speckled, homogeneous and nucleolar).

We optimize our method parameters using Bayesian optimization (as in Section III-A) with leave-one-out cross-validation over the 14 image slides in the training set. In Fig. 12 we list our classification performance on the test set alongside other methods.

IV. Discussion

As shown in the first part of our experiments, shape index histograms are superior to other popular texture descriptors for indirect immunoflourescence image analysis. We speculate that the second-order structure captured by the shape index is ideal for the blob-like structures in the staining patterns. Another attractive property of shape index histograms is that its parameters σ (shape scale) and γ (pooling scale) are very intuitive compared to LBP or GLCM where one has to specify pixel offsets and angular spacing. However, like the other histogram-based texture measures SIH also requires setting the number of bins which typically is determined in an *ad hoc*-manner. Note that we could have chosen to formulate the shape index histograms using fewer parameters. For example, we could have disregarded the smoothing along the tonal range and just let each shape index contribute to its nearest bin. According to our (non-shown) experiments, the change in performance is statistically insignificant on this dataset. However, this simplification is not justified because the absence of the β parameter is just an implicit parameter choice. Therefore, we have prefered making our formulation as generic as possible.

Our donut pooling scheme adds the radius r and the Gaussian width γ parameters to the model but offers a significant performance improvement. We believe the donut shapes are able to perform a meaningful spatial decomposition of the cell because cells are round and have the different staining patterns occurring at certain radii. Note that the feature description becomes rotation invariant by construction because the donuts are radially symmetric and because the shape index is based on the eigenvalues of the Hessian matrix. This contrasts the approach taken in [15], [16] where LBPs are made rotation invariant by estimating a canonical orientation and modifying the LBP accordingly. Moreover, we remark that our pooling scheme is not available for only shape index histograms since it is possible to e.g., perform a similar weighting of the contributions to the GLCM, BIF or LBP histograms. We suspect, however, that it will be less suitable for high-dimensional histograms (e.g., GLCM and BIF), as the donuts decrease the number of bin contributions meaning the histograms may fail to capture the image feature statistics.

Shape index histograms are comparable to BIF since they are both based on differential image geometry and therefore capture similar structures. The difference in the features extracted is that BIF uses first-, second-, and third-order information while SIH only relies on second-order information. Our results in favor of SIH could therefore indicate that the discriminative image structure between cell staining patterns is mainly second-order structure. Furthermore, our results also suggest that the joint multi-scale histogram of BIF can be replaced effectively by the marginal histograms across feature scales of SIH. This gives shape index histograms the advantage of having much lower feature dimensionality (90 versus 1296).

Regarding our performance comparison on the ICPR 2012 dataset we have seen that shape index histograms are competitive with other HEp-2 cell classification methods from the literature. By performing well on two datasets with different image acquisition setups, our method has demonstrated robustness towards changes in laboratory setups which are likely to occur in practice. It should be noted that the two methods that outperform shape index histograms are more complex as they rely on learning a codebook for BOVW [6], [15]; combine multiple features [15]; and learn a dissimilarity measure of features [15].

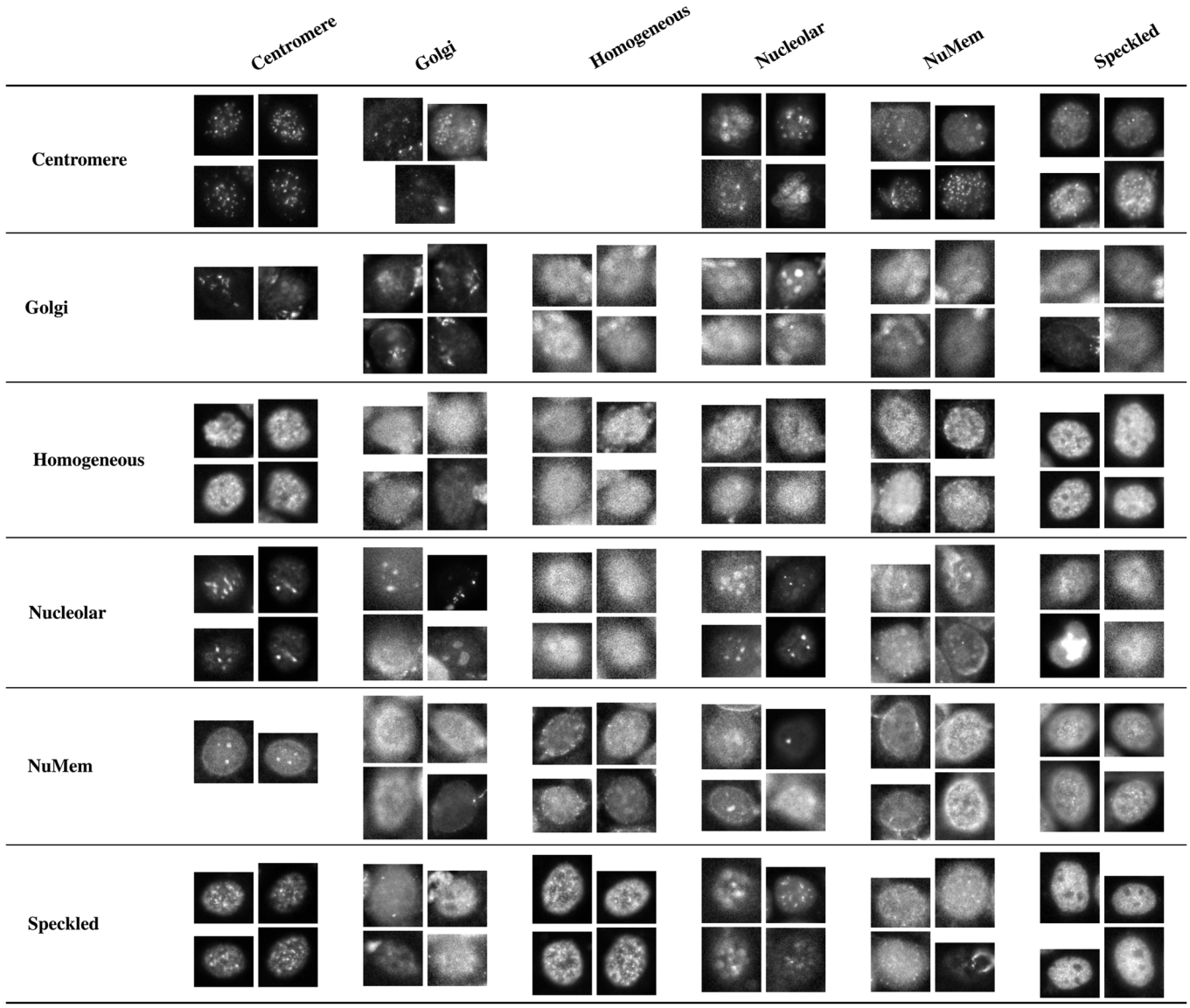


Fig. 11. Classification examples for SIH+$_{n_d=3}$ organized in a confusion table. Rows are actual (true) labels. Columns are predicted labels. Cell images are selected based on their SVM probability estimates [33], such that the cell images that look most like the predicted class are chosen. The off-diagonal elements are misclassifications, i.e., the least likely misclassifications according to the SVM. Diagonal contains correct classifications. When there are less than four examples in a table entry it means that we have misclassified less than four images of the row class as the column class.

Ref.	Method summary	Accuracy
[8]	Fisher tensor-based BOVW with region covariance descriptors.	70.2%
[16]	Rotation invariant co-occurrence of LBPs.	68.5%
[6]	BOVW w. gradient orientation histograms and intensity-based histogram pooling.	74.4%
[15]	Dissimilarity representation of SIFT and rotation invariant co-occurrence of LBPs	75.1%
[5]	BOVW w. discrete cosine transformation features and spatial decomposition from cell boundary.	67.4%
Ours	Shape index histograms.	71.5%

Fig. 12. Performances comparison on the ICPR 2012 dataset of the top performing methods in the literature.

In comparison, our classification is performed directly in the shape index histogram feature space (similar to the winner of the ICPR 2012 competition [1], [16]). That said, we speculate that the shape index could be well-suited for capturing second-order image structure as visual words in a BOVW pipeline.

We round of our discussion by remarking that shape index histograms are very fast to extract. The image derivatives are computed by convolutions while the shape index and the histogram contributions are computed from element-wise operations. For example, the SIH$_{n_d=3}$ descriptor takes around 40 ms to extract from a cell image. In terms of computational complexity, the feature extraction from an $M \times N$ image runs in $O(MN(\log N + \log M))$ when performing convolutions in the Fourier domain.

V. Conclusion

In this work, we have developed a new method for automatic classification of HEp-2 cell IIF images. We have used the scale-space framework to derive shape index histograms for texture description. The shape index captures second-order image

structure and is well suited for the blob-like structures prominent in the cell images. We have demonstrated this on the ICIP 2013 competition dataset, where shape index histograms are superior to the other texture measures included in our study. Furthermore, we have proposed a spatial decomposition of the cell images based on donut shapes. This extension improves the discriminative abilitiy of shape index histograms significantly and resulted in the title *merit winner* at the ICIP 2013 competition. On the ICPR 2012 dataset, shape index histograms show very competitive performance compared to other HEp-2 cell classification methods, especially when considering the low complexity of the method.

REFERENCES

[1] P. Foggia, G. Percannella, P. Soda, and M. Vento, "Benchmarking HEp-2 cells classification methods," *IEEE Trans. Med. Imag.*, vol. 32, no. 10, pp. 1878–1889, Oct. 2013.

[2] J. J. Koenderink and A. J. van Doorn, "Surface shape and curvature scales," *Image Vis. Comput.*, vol. 10, no. 8, pp. 557–564, 1992.

[3] M. Lillholm and L. D. Griffin, "Statistics and category systems for the shape index descriptor of local 2nd order natural image structure," *Image Vis. Comput.*, vol. 27, no. 6, pp. 771–781, 2009.

[4] P. Foggia, G. Percannella, A. Saggese, and M. Vento, "Pattern recognition in stained hep-2 cells: Where are we now?," *Pattern Recognit.* vol. 47, no. 7, pp. 2305–2314, 2014.

[5] A. Wiliem, C. Sanderson, Y. Wong, P. Hobson, R. F. Minchin, and B. C. Lovell, "Automatic classification of human epithelial type 2 cell indirect immunofluorescence images using cell pyramid matching," *Pattern Recognit.*, vol. 47, no. 7, pp. 2315–2324, 2013.

[6] L. Shen, J. Lin, S. Wu, and S. Yu, "HEp-2 image classification using intensity order pooling based features and bag of words," *Pattern Recognit.*, vol. 47, no. 7, pp. 2419–2427, 2014.

[7] L. Liu and L. Wang, "HEp-2 cell image classification with multiple linear descriptors," *Pattern Recognit.*, vol. 47, no. 7, pp. 2400–2408, 2014.

[8] M. Faraki, M. T. Harandi, A. Wiliem, and B. C. Lovell, "Fisher tensors for classifying human epithelial cells," *Pattern Recognit.*, vol. 47, no. 7, pp. 2348–2359, 2014.

[9] R. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Trans. Syst., Man Cybernet.*, vol. 3, no. 6, pp. 610–621, Nov. 1973.

[10] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.

[11] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, pp. 91–110, 2004.

[12] R. Stoklasa, T. Majtner, and D. Svoboda, "Efficient k-nn based HEp-2 cells classifier," *Pattern Recognit.*, vol. 47, no. 7, pp. 2409–2418, 2014.

[13] Y. Yang, A. Wiliem, A. Alavi, B. C. Lovell, and P. Hobson, "Visual learning and classification of human epithelial type 2 cell images through spontaneous activity patterns," *Pattern Recognit.*, vol. 47, no. 7, pp. 2325–2337, 2014.

[14] I. Ersoy, F. Bunyak, J. Peng, and K. Palaniappan, "HEp-2 cell classification in IIF images using Shareboost," in *Proc. 21st Int. Conf. Pattern Recognit.*, 2012, pp. 3362–3365.

[15] I. Theodorakopoulos, D. Kastaniotis, G. Economou, and S. Fotopoulos, "Hep-2 cells classification via sparse representation of textural features fused into dissimilarity space," *Pattern Recognit.*, vol. 47, no. 7, pp. 2367–2378, 2014.

[16] R. Nosaka and K. Fukui, "HEp-2 cell classification using rotation invariant co-occurrence among local binary patterns," *Pattern Recognit.*, vol. 47, no. 7, pp. 2428–2436, 2014.

[17] U. Bonde, V. Badrinarayanan, and R. Cipolla, "Multi scale shape index for 3d object recognition," in *Scale Space and Variational Methods in Computer Vision*, ser. Lecture Notes in Computer Science, A. Kuijper, K. Bredies, T. Pock, and H. Bischof, Eds. Berlin, Germany: Springer, 2013, vol. 7893, pp. 306–318.

[18] N. Bayramoglu and A. Alatan, "Shape index SIFT: Range image recognition using local features," in *Proc. 20th Int. Conf. Pattern Recognit.*, 2010, pp. 352–355.

[19] L. Broadbent, K. Emrith, A. Farooq, M. Smith, and L. Smith, "2.5d facial expression recognition using photometric stereo and the area weighted histogram of shape index," in *RO-MAN, 2012 IEEE*, 2012, pp. 490–495.

[20] G. Hetzel, B. Leibe, P. Levi, and B. Schiele, "3D object recognition from range images using local feature histograms," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2001, vol. 2, pp. II-394–II-399.

[21] T.-W. R. Lo and J. P. Siebert, "Local feature extraction and matching on range images: 2.5d SIFT," *Comput. Vis. Image Understand.*, vol. 113, no. 12, pp. 1235–1250, 2009.

[22] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2005, vol. 1, pp. 886–893.

[23] M. Crosier and L. Griffin, "Using basic image features for texture classification," *Int. J. Comput. Vis.*, vol. 88, no. 3, pp. 447–460, 2010.

[24] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1615–1630, Oct. 2005.

[25] E. Tola, V. Lepetit, and P. Fua, "Daisy: An efficient dense descriptor applied to wide-baseline stereo," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 5, pp. 815–830, May 2010.

[26] S. Lazebnik, C. Schmid, and J. Ponce, "A sparse texture representation using local affine regions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1265–1278, Aug. 2005.

[27] T. Lindeberg, *Scale-Space Theory in Computer Vision*. New York: Springer, 1993.

[28] J. Koenderink and A. Van Doorn, "The structure of locally orderless images," *Int. J. Comput. Vis.*, vol. 31, no. 2–3, pp. 159–168, 1999.

[29] R. Arandjelovic and A. Zisserman, "Three things everyone should know to improve object retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 2911–2918.

[30] N. Srinivas, A. Krause, S. Kakade, and M. Seeger, "Gaussian process optimization in the bandit setting: No regret and experimental design," in *Proc. 27th Int. Conf. Mach. Learn.*, J. Fürnkranz and T. Joachims, Eds., Jun. 2010, pp. 1015–1022.

[31] J. Snoek, H. Larochelle, and R. Adams, "Practical Bayesian optimization of machine learning algorithms," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds. Red Hook, NY: Curran.

[32] A. C. Davison and D. V. Hinkley, *Bootstrap Methods and Their Application*. Cambridge, U.K.: Cambridge Univ. Press, 1997.

[33] J. C. Platt, *Probabilities for SV Machines*. Cambridge, MA: MIT Press, 2000, pp. 61–74.

B Vision-based method for tracking meat cuts in slaughterhouses

Meat Science 96 (2014) 366–372

Contents lists available at ScienceDirect

Meat Science

journal homepage: www.elsevier.com/locate/meatsci

Vision-based method for tracking meat cuts in slaughterhouses

Anders Boesen Lindbo Larsen [a,*], Marchen Sonja Hviid [b], Mikkel Engbo Jørgensen [b], Rasmus Larsen [a], Anders Lindbjerg Dahl [a]

[a] Technical University of Denmark, Lyngby, Denmark
[b] Danish Meat Research Institute, Roskilde, Denmark

ARTICLE INFO

Article history:
Received 16 April 2013
Received in revised form 10 July 2013
Accepted 16 July 2013

Keywords:
Tracking
Traceability
Computer vision
Image processing
Object recognition

ABSTRACT

Meat traceability is important for linking process and quality parameters from the individual meat cuts back to the production data from the farmer that produced the animal. Current tracking systems rely on physical tagging, which is too intrusive for individual meat cuts in a slaughterhouse environment. In this article, we demonstrate a computer vision system for recognizing meat cuts at different points along a slaughterhouse production line. More specifically, we show that 211 pig loins can be identified correctly between two photo sessions. The pig loins undergo various perturbation scenarios (hanging, rough treatment and incorrect trimming) and our method is able to handle these perturbations gracefully. This study shows that the suggested vision-based approach to tracking is a promising alternative to the more intrusive methods currently available.

1. Introduction

In recent years, traceability has become an increasingly important aspect of the meat industry. For consumers, meat safety and quality is a persistent concern strengthened by reoccurring food recalls and scandals as well as increased animal welfare awareness (Vanhonacker, Verbeke, Poucke, & Tuyttens, 2008). In Western markets, this public concern has lead to legislations and regulations regarding food traceability to ensure quality and safety standards (Trienekens & Zuurbier, 2008). For producers, traceability adds extra value to their end products (Wang & Li, 2006). Demand for traceability information is on the rise yielding a competitive advantage to the producers who can deliver better guarantees of origin and handling (Buhr, 2003; Carriquiry & Babcock, 2007; Pouliot & Sumner, 2008).

In industrial abattoirs individual meat cuts become hard to trace after having cut up the carcass. Today most tracking systems are based on secondary systems like boxes or Christmas trees with RFID technology or conveyor belts. These systems offer only batch-level tracking of meat cuts because the secondary devices cannot be attached to the products individually.

In this work we propose a new technology for enabling meat traceability of individual meat cuts in slaughterhouse environments. Our approach is based on modern methods from the field of computer vision and image processing. Instead of attaching identification information to an object in order to track it we capture an image of the object and can identify the same object at a later point by capturing a new image. That is, we extract a *description* of an object from its appearance and use it as identifier for that object. We believe that this approach offers attractive advantages compared to current technology. While our experiments are limited to tracking pork loins, the method is sufficiently generic to be applied in other domains where the objects exhibit adequate diversity in appearance like the meat cuts considered in this work.

1.1. Related work

Food traceability has been approached from many angles with different applications in mind. This has led to a diverse literature with a limited agreement on how to implement food traceability. For an overview of food traceability literature, we refer to Karlsen, Dreyer, Olsen, and Elvevoll (2013).

In this article, we focus on a single aspect of traceability in the meat industry; the technology that enables object tracking along a production line. In recent literature, the use of RFID tags as underlying food tracking technology is dominating (Cimino & Marcelloni, 2012; Lefebvre, Castro, & Lefebvre, 2011; Regattieri, Gamberi, & Manzini, 2007). However, RFID tagging of meat in a slaughterhouse environment has drawbacks for mainly one reason: Tags may disappear into the meat product and turn up on the consumer's plate. This is a very critical point with the consequence that slaughterhouses avoid tagging meat cuts directly; instead they attach a tag to the device carrying the meat.

Regarding tracking technology in the meat industry, the following approaches have been suggested. Mousavi, Sarhadi, Fawcett, Bowles, and York (2005) present a conveyor belt system capable of tracking

* Corresponding author at: Richard Petersens Plads, bldg. 324, 2800 Lyngby, Denmnark. Tel.: +45 45255228.
E-mail addresses: abll@dtu.dk (A.B.L. Larsen), mahd@teknologisk.dk (M.S. Hviid), mej@teknologisk.dk (M.E. Jørgensen), rlar@dtu.dk (R. Larsen), abda@dtu.dk (A.L. Dahl).

0309-1740/$ – see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.meatsci.2013.07.023

meat cuts in a boning hall. To facilitate the tracking, RFID chips are embedded in carrier hooks for the meat cuts. Fröschle, Gonzales-Barron, McDonnell, and Ward (2009) examine the usability of barcodes printed on the beak and legs of chickens. This approach does not generalize well to other meat tracking scenarios because it requires the meat product to have non-edible parts suitable for barcode printing. Arana, Soret, Lasa, and Alfonso (2002); Suekawa et al. (2010) perform breed identification of beefs based on DNA analysis, and Tate (2001) investigates the possibility of using DNA analysis for tracing individual meat cuts back to the original carcass. Our vision-based approach is reminiscent of DNA identification in the way identification is derived from the object rather than from a tag attached to the object. However, DNA identification is still a cumbersome process for a slaughterhouse environment.

Of the three tracking technologies mentioned above, the conveyor belt system is most representative of current slaughterhouse practice. Typically, meat cuts are tracked individually or in a batch by attaching a tag to the container or carrier device. A drawback of this method is that it is prone to accidents where pieces are lost or exchanged between carrier devices. Such accidents may happen since the meat cuts cannot be directly connected to the carrier device at all times. With a vision-based approach, this scenario will not be a problem since the meat cut carries identification in its appearance.

For both the food and the non-food industry we have not been able to find examples of visual recognition methods similar to ours applied in a tracking/identification setup. Weichert et al. (2010) propose combining RFID tracking with a vision system that can recognize and decode 2D barcodes. Using cheap cameras they can offer a more continuous identification and localization of the products and thereby improve fault detection. Again, this approach is not viable for meat cuts as the goal is to avoid foreign objects (both barcodes and RFID tags) that can end up in the product. Therefore, to the best of our knowledge, tracking from visual recognition of the products directly has not been attempted before.

1.2. Contributions

In this work we investigate a new technology for enabling traceability of individual meat cuts in a slaughterhouse environment. The investigation extends the work presented by Hviid, Jørgensen, and Dahl (2011) by scaling up the experiment to 211 pork loins and introducing *nuisance factors* to simulate a slaughterhouse environment. We show that the pork loins can be recognized and identified correctly between the two photo sessions. These results indicate that current computer vision methods for object recognition are mature for integration in production lines.

2. Experiment setup

The dataset for our experiment is constructed using 211 pig loins. The pig loins are photographed in two sessions separated by 1 day. Overnight, the loins are hanging on *Christmas trees* stored in a chill room, see Fig. 1.

The photographing setup (see Fig. 2) is the same for both photo sessions. We use the popular and inexpensive Microsoft Kinect camera that captures a depth map along with a standard RGB image of the loin. Examples of both images are shown in Fig. 3. Next to the camera a fluorescent tube is mounted spreading light at a wide angle. The loins are photographed separately by placing them one by one on a table and capturing a photo.

A selection of the loins undergoes different perturbation scenarios in an attempt to simulate slaughterhouse treatment. All perturbations occur after the first and before the second photo session. The perturbations are:

Rough treatment 19 loins are knocked hard onto a table before the second photo session.

Fig. 1. Pork loins are stored overnight on *Christmas trees* between the two photo sessions.

Incorrect trimming Pieces of meat and bones are cut off from 18 loins before the second photo session.

Incorrect hanging 19 loins are stored overnight by hanging them sideways on Christmas trees which causes bends.

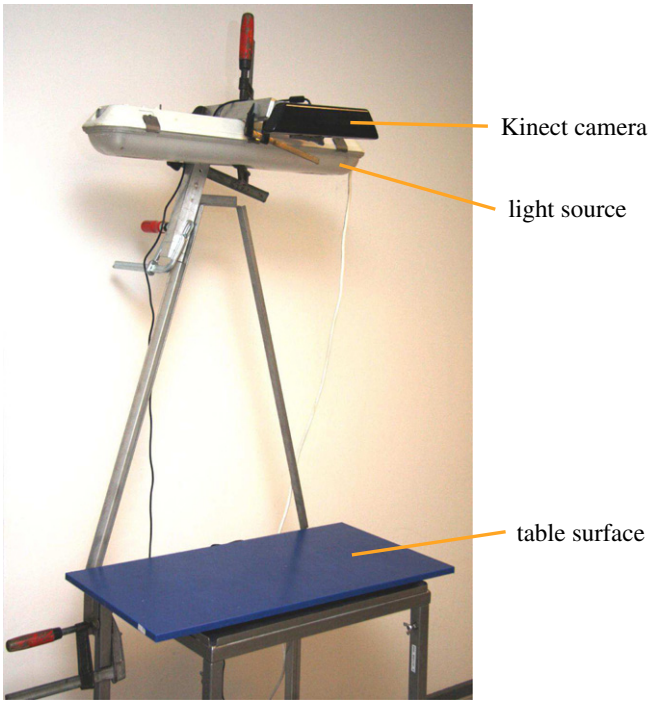


Fig. 2. Camera setup. Pork loins are placed on the table and are photographed from above.

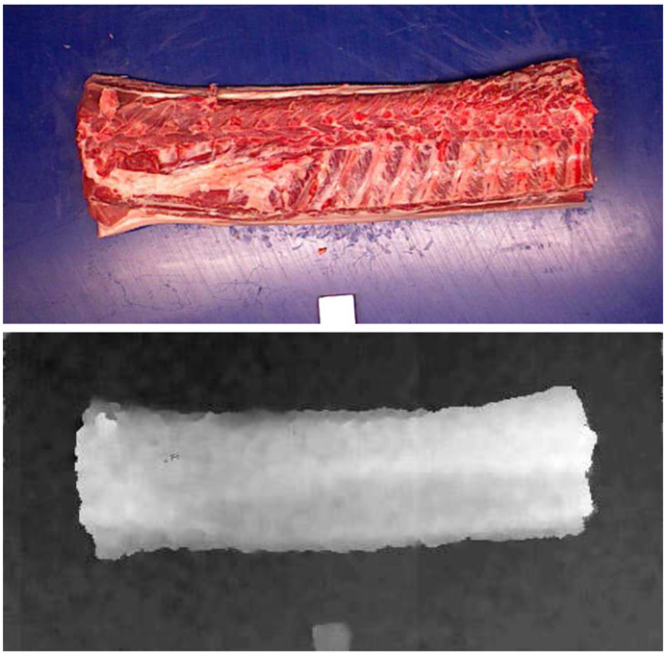


Fig. 3. RGB and depth images of a pork loin as captured by the Kinect camera.

Fig. 4. Pork loin segmentation. Top image shows the segmentation mask derived from the depth image (see Fig. 3). Bottom image shows the pork loin cut out using the segmentation mask.

Illumination and orientation changes 37 loins are rotated between 45° and 180° around the optical axis before being photographed. This creates variations in lighting because the light falls differently on a rotated object. Moreover, the rotated loins serve as a check to see if our algorithm is invariant to different object orientations.

All loins except those subject to incorrect hanging are photographed normally on day 2 before any perturbations occur. Because some loins are reused in multiple perturbation scenarios (e.g. a loin is photographed at a different orientation and again after a trimming), we cannot perform a matching on all 211 loins by combining all perturbations. Instead, we combine each perturbation scenario with the remaining unperturbed images from day 2 in 4 separate experiments. This means that for the incorrect hanging scenario we want to match all 211 loins whereas for the other scenarios we want to match 192 loins.

3. Visual recognition method

The purpose of the visual recognition method is to match the pork loin images correctly between the two photo sessions. Our recognition method is divided into 4 steps listed here as an overview.

1. *Segmentation*. First, we perform a segmentation of the pork loin. That is, we cut the pork loin out from the background image pixels.
2. *Canonization*. The segmented pork loin images are then brought to a canonized form that minimizes variability from external sources, e.g. illumination.
3. *Description*. From the canonized images we generate a description of the image structure.
4. *Matching*. Finally, we perform the pork loin matching by comparing the descriptors from the previous step.

In total, the recognition method takes under 2 s per image in processing time on a 2.67 GHz CPU. It should be possible to speed this up significantly since our method has not been implemented with speed efficiency as a priority. Note that we have made all implementation details available online.[1]

3.1. Segmentation

To separate the loin from the background we use the depth image provided by the Kinect camera. We know the depth of the table surface which makes it easy to differentiate between the surface and the meat. To account for noisy depth data, we employ the *max-flow/min-cut* graph cut algorithm to perform segmentation of the depth image (Boykov & Kolmogorov, 2004). This yields a binary *mask* specifying which pixels belong to the loin and which pixels belong to the background. The result of the segmentation algorithm is shown in Fig. 4.

3.2. Canonization

The goal of the canonization step is to bring the pork loin images to a common form making them invariant to changes in *illumination*, *rotation* and *size*.

Since the pork loin primarily consists of red color nuances we can discard the colors by converting the image to greyscale without losing significant information. Moreover, we perform a *histogram equalization* to increase the contrast and compensate for differences in lighting.

To ensure the same orientation for all pork loins, we use the segmentation mask and calculate the *image moments* of the mask region. The second-order moments can be used to derive the covariance matrix of the image region. The dominant orientation of the region is then calculated from the angles of the covariance matrix eigenvectors. We rotate the loin such that the dominant orientation is parallel to the x axis, that is, the loin is orientated horizontally along its broad side. Notice that this rotation does not consider if the loin is placed upside-down. We handle this situation by performing a pixel-wise comparison of a loin image with the average of 20 upright loin images. If a loin is pixel-wise closer to the upside-down version rather than upright version, it should be rotated 180°.

[1] Source code available at http://compute.dtu.dk/~abll/meat_recognition.

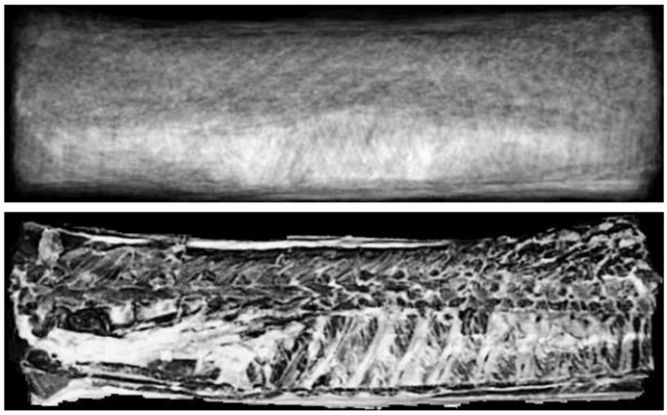


Fig. 5. The result of the canonization step. Top image shows an average of 20 pork loin images with the same orientation. The average image is used to check if loin images are placed upside down. Bottom image shows a canonized loin image.

Finally, the pork loin images are trimmed to remove the background border followed by a scaling to 600 × 180 pixels giving all loins the same dimensions. An example of the canonization is shown in Fig. 5.

3.3. Description

In the image description step we seek to achieve an image representation that captures the image structure in a manner suitable for comparative purposes. E.g., the standard pixel representation is not suitable because it is sensitive to object translations.

We employ the popular *bag-of-words* approach (Prince, 2012) and perform *K-means clustering* on image patches extracted from 30 out of the 211 loin images. The cluster centers yield a finite set of different image patches (the *visual vocabulary*). An image can now be described by extracting numerous image patches and mapping each patch to its nearest entry (aka. *visual word*) in the vocabulary and counting the number of occurrences of each visual word. The bag-of-words image characterization thus constitutes a histogram over visual words. An overview of bag-of-words description is shown in Fig. 6.

3.3.1. Feature description

Instead of using raw image patches as mentioned above, we perform *feature description* of these patches. Feature description yields a low-dimensional representation of an image patch that attempts to capture the image structure while being invariant to various image perturbation factors. A wide selection of feature descriptors exists in the literature but their performance varies only little for general applications (Dahl, Aanæs, & Pedersen, 2011; Kaneva, Torralba, & Freeman, 2011). We use the DAISY descriptor (Tola, Lepetit, & Fua, 2010) as it is formulated for dense extraction. However, we use our own variation of DAISY since the original is based on unsigned gradient orientations spanning 180° (whereas signed orientations span 360°). Unsigned orientations offer invariance towards more complicated illumination situations where light areas become dark and vice versa due to surface reflectance properties. Our scenario is sufficiently constrained allowing us to benefit from signed orientations.

DAISY follows a popular approach to feature description based on image gradient orientations summarized in histograms. From an image I we can extract the x and y directional derivatives,

$$\mathbf{L}_x = \frac{\partial \mathbf{G}_\sigma}{\partial x} * \mathbf{I} \quad , \quad \mathbf{L}_y = \frac{\partial \mathbf{G}_\sigma}{\partial y} * \mathbf{I} \quad , \tag{1}$$

by convolving the image with a Gaussian window $\mathbf{G}$ differentiated along the x and y axis respectively. $*$ denotes convolution and σ adjust the width of the Gaussian kernel, i.e., the scale at which we extract the derivatives. We can then calculate the image gradient orientations θ and their magnitude $\mathbf{m}$ from

$$\theta = \arctan 2\left(\mathbf{L}_x, \mathbf{L}_y\right) \quad , \quad \mathbf{m} = \sqrt{\mathbf{L}_x^2 + \mathbf{L}_y^2}. \tag{2}$$

To describe the gradient orientation statistics we select a number of bins N in the angular range. For each angle $a_i, i = \{1,\ldots,N\}$ we calculate the bin contribution $\mathbf{b}_i$ using the circular normal distribution to smooth out contributions among neighbor bins to be invariant towards small rotations. Moreover, we weigh the bin contributions by the gradient magnitude such that small gradients have less influence than large gradients.

$$\mathbf{b}_i = exp\ \ (\kappa * cos\ \ (\theta - a_i)) \circ \mathbf{m} \tag{3}$$

$\circ$ denotes the element-wise product. κ adjusts the scale of the bin smoothing in the angular range. To gather bin contributions spatially, we convolve with a Gaussian window of scale γ.

$$\mathbf{b}_{\gamma,i} = \mathbf{G}_\gamma * \mathbf{b}_i \tag{4}$$

Finally, we assemble a histogram at the spatial location (u,v) from

$$\mathbf{h}_\gamma(u,v) = \left[\mathbf{b}_{\gamma,1}(u,v), \ldots, \mathbf{b}_{\gamma,N}(u,v)\right]. \tag{5}$$

To perform DAISY description of an image patch, we sample $\mathbf{h}_\gamma$ in a log-polar grid similar to the original formulation. The histogram vectors are then concatenated and the entire descriptor vector is L1-normalized. This normalization makes the descriptor invariant to affine illumination variations.

Feature descriptors like DAISY have become very popular for visual recognition. They are effective at capturing both texture information and local image structure while being robust towards various image perturbations.

3.3.2. Pork loin image representation

The bag-of-words representation disregards all spatial layout of the extracted image patches. This is good for achieving invariance to object translations, but not so good for providing a distinctive object description. We reestablish some of the spatial layout information by sampling multiple bag-of-words histograms at different positions in the image. More specifically, we generate 8 histograms from the pork loin image by weighing the different histogram contributions using Gaussian windows placed in a 2 × 4 grid to reflect the oblong shape of a pork loin. See Fig. 7. The reason for using Gaussian windows to gather bin contributions is because the smoothed weighting handles object translations more gracefully leading to a more robust description.

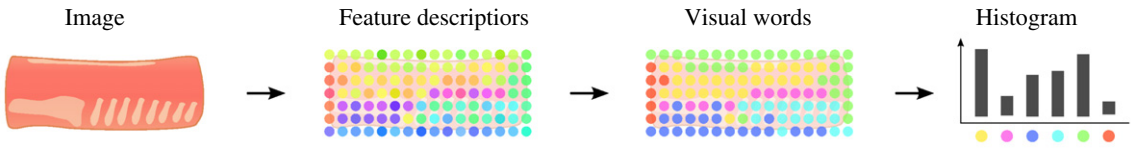


Fig. 6. Overview of the description pipeline. From the raw image we perform a dense extraction of local feature descriptors. The feature descriptors are then quantized into visual words. Finally the occurrences of visual words are summarized in a histogram that becomes the final image description.

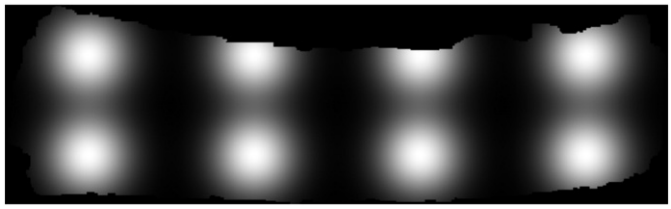


Fig. 7. The visual word contribution to each bag-of-words histogram is weighted using a Gaussian window. Visual words that lie outside the segmentation have 0 weight.

The final description of a pork loin image is the concatenation of the 8 bag-of-words histograms. Our bag-of-words vocabulary consists of 1500 visual words meaning that each histogram can be represented by a 1500-dimensional vector. Thus, the concatenation of the 8 bag-of-words histograms yields a 12,000-dimensional image description vector.

3.4. Matching

We assess the similarity of two pork loin images by calculating the histogram distance between their two description vectors generated in the previous step. For every pork loin from day 1 a match is established to the pork loin from day 2 with the smallest χ^2 distance defined as

$$\chi^2(\mathbf{x},\mathbf{y}) = \sum_{n=1}^{D} \frac{(\mathbf{x}(n)-\mathbf{y}(n))^2}{\mathbf{x}(n)+\mathbf{y}(n)} \quad , \tag{6}$$

where D is the dimensionality of the vectors $\mathbf{x}$ and $\mathbf{y}$ and $\mathbf{x}(n)$ is the nth element of $\mathbf{x}$.

4. Results

We run our recognition method on the 4 different experiments listed in Section 2. In all experiments we are able to match all pork loins between the two photo sessions correctly.

To demonstrate the visual impact of the perturbation scenarios, we show examples of pork loins from both days in Fig. 8. We show the canonized versions rather than the original camera images as the canonization makes visual comparison easier.

Fig. 8 shows a loin without perturbations. i.e. proper hanging overnight. We observe both local pixel translations due to minor object deformations and global pixel translations due to improper alignment in the canonization step. In Fig. 8b and c, we observe local deformations caused by rough handling of the meat and incorrect trimming. Fig. 8d shows perturbations due to incorrect hanging overnight. The twist causes translation, local deformation in the right end of the loin, and minor local rotation. Finally in Fig. 8e and f, the illumination changes caused by object rotation are shown. These perturbations are significantly diminished by the canonization step, however, we still see that specularities and shadows change indicating that the experiment setup could be improved with a more diffuse illumination.

To investigate the robustness of the recognition method we inspect loins that have been poorly matched in our experiments. We measure the quality of a match by its distinctiveness d computed by subtracting the descriptor distance of the nearest incorrect match from the descriptor distance of the correct match.

A large difference means that the matching pork loin image pair from day 1 and 2 stands out from the rest of the loins. A small difference means that there exists a mismatching loin from day 2 with an image description similar to the pork loin from day 1. In Fig. 9, we show 3 examples of poorly matched pork loin image pairs along with the second-closest match from day 2. In Fig. 9a and b we see two examples where the appearances of the second-closest matches are similar to the loins from day 1. If a human were to tell the loins apart, he/she would most likely rely on smaller details in their appearances. In Fig. 9c a significant bend affects the aspect ratio of the loin image leading to a poor canonization caused by improper alignment. Thus, it is the canonization rather than the image description that fails.

Finally in Fig. 10, we illustrate the distinctiveness statistics for each experiment. We see that our recognition method is very close to yielding a few mismatches as the distinctiveness of the lowest outliers come close to 0 (a negative value means an incorrect match). However, the main part of the remaining loins (around 200) are matched with a comfortable margin to the nearest incorrect match.

5. Discussion

In Fig. 8, we have seen examples of different perturbation scenarios in a slaughterhouse environment. Our image description algorithm is

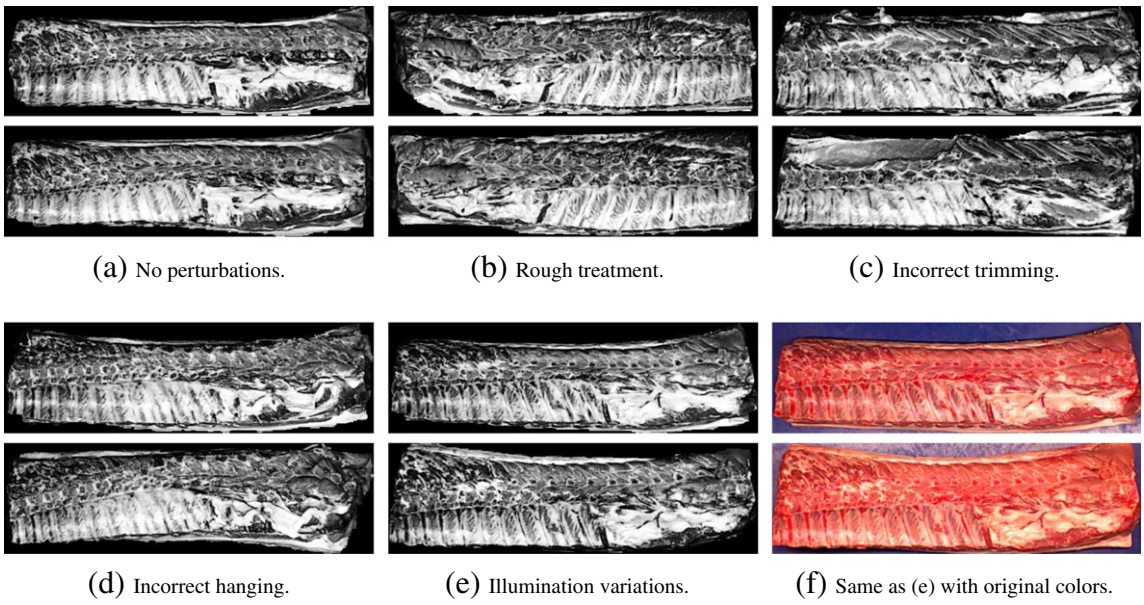
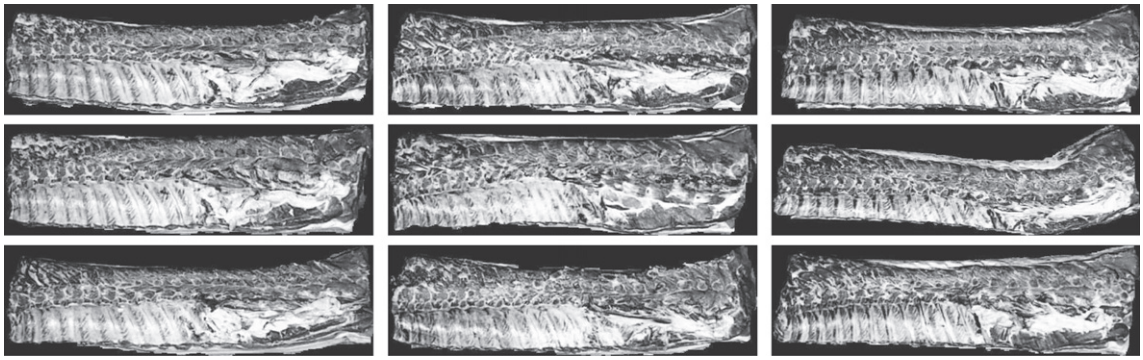


Fig. 8. Examples of perturbation scenarios between day 1 (upper image) and day 2 (lower image). Canonized images are shown for better visual comparison (except for (f)).



(a) Rough treatment. (b) Incorrect trimming. (c) Incorrect hanging.

Fig. 9. Examples of pork loins for which our recognition method yield image descriptions with little distinctiveness compared to the other image descriptions. Top row shows the pork loin on day 1. Middle row shows the same loin on day 2. Bottom row shows the closest candidate among the other loins.

constructed to be robust towards such perturbations and our experiments have confirmed this. While our results seem promising, we should note that we do not have sufficient image data to create proper *training*, *validation* and *test* sets. Therefore, our method parameters are likely to be overfitted because of improper training on the test set. However, as we consider this work a proof of concept, we still believe that the results show that our approach is feasible. In this connection we add that pork loins in slaughterhouses are typically processed in batches of significantly fewer pieces than in our experiment.

From the results in Fig. 9c, we have identified an important shortcoming of our canonization method. The image alignment is not suitable for non-rigid deformations because it leads to improper scaling and placement of the object. We consider this an important bottleneck of our current recognition method because bad alignment directly influences the image description.

It is possible to improve the recognition task even further by disallowing a loin from day 1 to be matched to multiple loins from day 2 and vice versa (by considering the problem an instance of *bipartite matching*). However, since the experiment images do not challenge our recognition method sufficiently, it will be difficult to draw conclusions from improvements to the method. It should be noted that by introducing bipartite matching, we lose the ability to perform any matching before all loins have been photographed for the second time. We have not investigated whether this will be a critical point in practice along a production line.

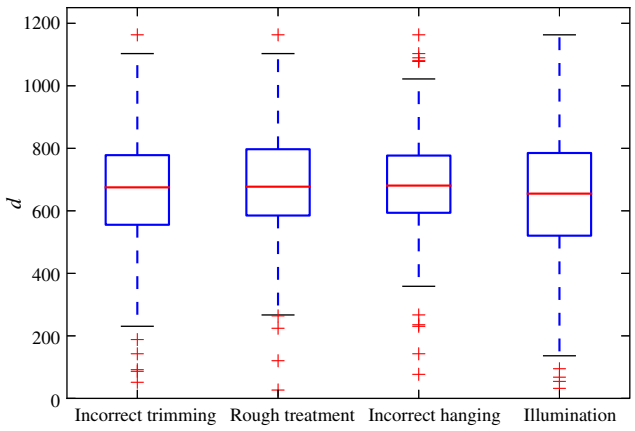


Fig. 10. A box plot describing the statistics of the match distinctiveness d for each experiment. Rectangles represent the *interquartile range* IQR = Q3 − Q1. The whiskers are placed at Q1 − 1.5 IQR and Q3 + 1.5 IQR. The plusses denote outliers.

5.1. Perspectives in a slaughterhouse environment

Based on our results, we believe that the proposed method is a competitive alternative to current technology relying on RFID tags of carrier devices. Vision-based tracking is less intrusive as it does not require physical contact with the tracked objects. Moreover, our relatively simple camera setup should be easy to integrate in a production line. As our experiments shows, our method does not enforce strict requirements to the camera stations wrt. lighting or light shielding. Though, one should still strive for a good diffuse illumination of the objects as it improves the matching distinctiveness.

Regarding the IT infrastructure needed to implement this system, we believe that the requirements of vision-based tracking are similar to what is currently required by RFID tracking. For both tracking methods we need IT systems for bookkeeping to keep track of which products have been seen where and when. A consequence of image-based identification is that the amount of identification data is many orders of magnitude bigger than with physical tags (the entire image description versus a single number per tag). With current computer networking speeds, however, we do not believe that this will impose any problems.

We imagine that the visual recognition should supplement the RFID tracking of carrier devices and ameliorate the tracking granularity from batches to individual meat cuts. Thus, from the RFID tag we can identify which batch is currently being processed and perform visual recognition within this batch. This is a subject for further investigation when our approach is to be tested on a real production line.

So far, we have only experimented with pork loins that exhibit a very characteristic image structure. It is likely that other meat cuts are more difficult to represent distinctively using our method. More experiments are needed to assess the robustness of our recognition method in more challenging situations.

Finally, as a more speculative perspective, we imagine that the image data gathered can be used for further analysis as a part of a quality assurance and process control stage. E.g. the fat percentage or the quality of the cutting process could be quantified by an image-analysis program using images from camera stations along the production line.

6. Conclusion

Tracking of individual meat cuts is an important part of facilitating meat traceability from farmer to consumer. In this work we have demonstrated a vision-based system that enables meat traceability in a slaughterhouse environment. By combining off-the-shelf vision and image processing technology we are able to track around 200 pig loins between two points without errors. This approach is meant as an alternative to current more intrusive tracking methods and our investigation shows that it is feasible. Further experiments are needed to determine the limitations of our method.

References

Arana, A., Soret, B., Lasa, I., & Alfonso, L. (2002). Meat traceability using DNA markers: Application to the beef industry. *Meat Science*, *61*, 367–373. http://dx.doi.org/10.1016/S0309-1740(01)00206-6.

Boykov, Y., & Kolmogorov, V. (2004). An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *26*, 1124–1137. http://dx.doi.org/10.1109/TPAMI.2004.60.

Buhr, B. (2003). Traceability and information technology in the meat supply chain: Implications for firm organization and market structure. *Journal of Food Distribution Research*, *34*, 13–26.

Carriquiry, M., & Babcock, B.A. (2007). Reputations, market structure, and the choice of quality assurance systems in the food industry. *American Journal of Agricultural Economics*, *89*, 12–23. http://dx.doi.org/10.1111/j.1467-8276.2007.00959.x (arXiv:http://ajae.oxfordjournals.org/content/89/1/12.full.pdf+html).

Cimino, M., & Marcelloni, F. (2012). Enabling traceability in the wine supply chain. In G. Anastasi, E. Bellini, E. Nitto, C. Ghezzi, L. Tanca, & E. Zimeo (Eds.), *Methodologies and Technologies for Networked Enterprises. Lecture Notes in Computer Science*, *Vol. 7200*. (pp. 397–412). Springer Berlin Heidelberg. http://dx.doi.org/10.1007/978-3-642-31739-2_20.

Dahl, A., Aanæs, H., & Pedersen, K. (2011). Finding the best feature detector–descriptor combination. *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2011 International Conference on* (pp. 318–325). http://dx.doi.org/10.1109/3DIMPVT.2011.47.

Fröschle, H. K., Gonzales-Barron, U., McDonnell, K., & Ward, S. (2009). Investigation of the potential use of e-tracking and tracing of poultry using linear and 2d barcodes. *Computers and Electronics in Agriculture*, *66*, 126–132. http://dx.doi.org/10.1016/j.compag.2009.01.002.

Hviid, M., Jørgensen, M., & Dahl, A. (2011). Vision based meat tracking. *58th International Congress of Meat Science and Technology*.

Kaneva, B., Torralba, A., & Freeman, W. (2011). Evaluation of image features using a photorealistic virtual world. *Computer Vision (ICCV), 2011 IEEE International Conference on* (pp. 2282–2289). http://dx.doi.org/10.1109/ICCV.2011.6126508.

Karlsen, K. M., Dreyer, B., Olsen, P., & Elvevoll, E. O. (2013). Literature review: Does a common theoretical framework to implement food traceability exist? *Food Control*, *32*, 409–417. http://dx.doi.org/10.1016/j.foodcont.2012.12.011.

Lefebvre, L. A., Castro, L., & Lefebvre, É. (2011). *RFID: An emerging paradigm for the agrifood supply chain*. Wiley-Blackwell, 109–129 (chapter 7).

Mousavi, A., Sarhadi, M., Fawcett, S., Bowles, S., & York, M. (2005). Tracking and traceability solution using a novel material handling system. *Innovative Food Science & Emerging Technologies*, *6*, 91–105. http://dx.doi.org/10.1016/j.ifset.2004.10.006.

Pouliot, S., & Sumner, D. A. (2008). Traceability, liability, and incentives for food safety and quality. *American Journal of Agricultural Economics*, *90*, 15–27. http://dx.doi.org/10.1111/j.1467-8276.2007.01061.x (arXiv:http://ajae.oxfordjournals.org/content/90/1/15.full.pdf+html).

Prince, S. (2012). *Computer vision: Models learning and inference*. Cambridge University Press.

Regattieri, A., Gamberi, M., & Manzini, R. (2007). Traceability of food products: General framework and experimental evidence. *Journal of Food Engineering*, *81*, 347–356. http://dx.doi.org/10.1016/j.jfoodeng.2006.10.032.

Suekawa, Y., Aihara, H., Araki, M., Hosokawa, D., Mannen, H., & Sasazaki, S. (2010). Development of breed identification markers based on a bovine 50 k snp array. *Meat Science*, *85*, 285–288. http://dx.doi.org/10.1016/j.meatsci.2010.01.015.

Tate, M. (2001). Traceability of meat products—application of DNA technology. *Proceedings of the New Zealand Grassland Association* (pp. 255–258).

Tola, E., Lepetit, V., & Fua, P. (2010). Daisy: An efficient dense descriptor applied to wide-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*, 815–830.

Trienekens, J., & Zuurbier, P. (2008). Quality and safety standards in the food industry, developments and challenges. *International Journal of Production Economics*, *113*, 107–122. http://dx.doi.org/10.1016/j.ijpe.2007.02.050 (research and Applications in E-Commerce and Third-Party Logistics Management — Special Section on Meta-standards in Operations Management: Cross-disciplinary perspectives.).

Vanhonacker, F., Verbeke, W., Poucke, E. V., & Tuyttens, F. A. (2008). Do citizens and farmers interpret the concept of farm animal welfare differently? *Livestock Science*, *116*, 126–136. http://dx.doi.org/10.1016/j.livsci.2007.09.017.

Wang, X., & Li, D. (2006). Value added on food traceability: A supply chain management approach. *Service Operations and Logistics, and Informatics, 2006. SOLI'06. IEEE International Conference on* (pp. 493–498). http://dx.doi.org/10.1109/SOLI.2006.329074.

Weichert, F., Fiedler, D., Hegenberg, J., Mller, H., Prasse, C., Roidl, M., & Hompel, M. (2010). Marker-based tracking in support of rfid controlled material flow systems. *Logistics Research*, *2*, 13–21. http://dx.doi.org/10.1007/s12159-010-0025-6 (URL: http://dx.doi.org/10.1007/s12159-010-0025-6).

C Automatic particle size measurements from a learned image segmentation

Automatic particle size measurements from a learned image segmentation

Anders Boesen Lindbo Larsen[a,*], Logan Nicholas Schultz[b,c,*], Anders Bjorholm Dahl[a], Rasmus Larsen[a], Henning Osholm Sørensen[b],

[a] *Department of Applied Mathematics and Computer Science, Technical University of Denmark, Kgs. Lyngby, Denmark*
[b] *Nano-Science Center, Department of Chemistry, University of Copenhagen, Copenhagen, Denmark*
[c] *Department of Chemical and Biological Engineering, Center for Biofilm Engineering, Montana State University, Bozeman, MT*

Abstract

We present a system to automatize the cumbersome task of estimating particle size distributions from *scanning electron microscopy* (SEM) images. We propose a user interface that requires the user to annotate a small training image from which the program learns image structures. The learned structures allow us to efficiently segment an image and extract crystal size measurements using off-the-shelf image processing techniques. We apply the method to inspect grain coarsening of calcite over time and compare automatic versus manual estimates of particle size distributions. The results show that for 11 out of 15 test images, the measurements of our method are insignificantly different to manual measurements with $p > 0.05$. This suggests that our method can facilitate large-scale experiments for particle measurements where the amount of manual work has previously been a bottleneck. A Python-based implementation of the method is available online.

Keywords: crystals/particles, image segmentation, machine learning, deep learning, convolutional neural network

Preprint submitted to ...

1. Introduction

In many fields of science it is important to determine the sizes of objects, e.g. particles or crystals from microscopy images. Since these particles often partly overlap or are found in large agglomerates it is difficult to automatically determine the sizes of the individual particles. Often available algorithms for such image analysis only work well on images of well separated particles. Recently Schultz et al. (2014) conducted an investigation of calcite, $CaCO_3$, crystal growth under different conditions. The calcite crystals formed in the growth experiments were imaged by SEM and they tended to agglomerate. Hence the generally available methods for determining the size distributions could not be used and manual measurement of the crystals was performed. The task of manually determining the size of crystals in SEM images is tedious and time-consuming. Therefore, replacing manual measurements by our method seeks to save man-hours while allowing for much larger quantities of image data to be processed, thus improving statistical analyses to support quantification of the results. Central to our method is recent advances in machine learning, which allows us to perform a good pixel classification of the SEM images. Instead of hand-crafting image features to solve the task, we learn the morphology of the features from an example image *annotated* by the user. The annotations describe simple image contents (e.g. is this pixel part of a crystal or the background) and allows the segmentation method to extract the crystals easily. Thus, we require that the user annotates a small image region into three components: background, crystals and crystal borders. Generating the training data manually might take a bit of time, however, as the amount of image data to be processed typically is magnitudes higher than the image region to be annotated, the time spent annotating quickly becomes profitable.

The task of quantifying particle morphology from images can provide an effective measure for further analysis. Røgen et al. (2001) extract calcite particles from *backscatter electron micrographs* which allows for a simple binary segmentation by thresholding the pixel intensity. They then extract

*Corresponding authors.

Email addresses: abll@dtu.dk (Anders Boesen Lindbo Larsen), lschultz@nano.ku.dk (Logan Nicholas Schultz), abda@dtu.dk (Anders Bjorholm Dahl), rlar@dtu.dk (Rasmus Larsen), osholm@nano.ku.dk (Henning Osholm Sørensen)

URL: http://compute.dtu.dk/~abll (Anders Boesen Lindbo Larsen)

particles by deagglomerating calcite regions by region growing. While they do not capture exact particle sizes, their measure is useful for quantifying calcite texture types. More recently, Lu et al. (2009); Mingireanov Filho et al. (2013); Pascal Asmussen and Olaf Conrad and Andreas Günther and Moritz Kirsch and Ulrich Riller (2015); Jungmann et al. (2014) have presented systems for characterizing thin section images from grain segmentation. Common to these approaches is that that they rely mostly on pixel colors and apply region growing, level set, or graphs to perform the segmentation. Moreover, they do not require any training data provided by the user as they rely on interactive corrections to assist segmentation. Ideally, we want the user to teach the program only once as this will allow the method to run on large quantities of data without requiring additional manual labor.

Image segmentation is a heavily researched field in computer vision as it often occurs as a sub-problem for a wide range of higher-level tasks. Recently, advances in *deep learning* have shown superior performance across a wide range of visual recognition problems (LeCun et al., 2015). Whereas traditional approaches in computer vision focuses on the process of devising clever image patterns to describe the image contents (feature engineering), deep learning tries to learn these patterns from data during training. As first demonstrated by Ciresan et al. (2012), *convolutional neural networks* (convnets) are very competitive for segmentation tasks compared to more classical approaches, even when these incorporate more prior knowledge about spatial structure of the problem. Laptev et al. (2012) build a binary segmentation method for *serial section transmission electron microscopy* images by first estimating correspondences between image slices using gradient orientation histograms (Lowe, 2004; Liu et al., 2011). The correspondences are used to align the image slices which allows them to extract local image features across slices. They apply a random forest classifier to assign pixel-wise class probabilities given the array of local image features. Finally, the hard pixel classification is performed using graph-cut segmentation (Boykov and Kolmogorov, 2004) on top of the class probabilities. In contrast, Ciresan et al. (2012) achieve better results on the same problem by simply perform pixel classification using a convnet on the 2D image slices separately. Thus, convnets allow us to build simple segmentation pipelines capable of learning good discriminative image features that can compete with substantially more complicated segmentation pipelines. A notable downside of neural networks is the computational burden of learning the image features which usually requires parallel architectures like GPUs to accelerate training. Though, neural networks are cheap at test

time allowing them to be deployed on ordinary computers.

An alternative data-driven method for segmentation is *sparse dictionary learning* (Dahl and Larsen, 2011), but this approach is *shallow* as it cannot learn a hierarchical representation of the image structure like deep neural networks (Bengio et al., 2013).

In this article, we will present a new approach to segmenting and determining the sizes of individual crystals or particles in images obtained by scanning electron microscopy. The segmentation task is made easy using a trainable pixel classifier and this the pixel classifier will be trained on a small annotated training image. We apply the method to a series of SEM images of calcite crystals and show that our method yields crystal size distributions that are statistically insignificant from manual measurements.

Our Python implementation with installation instructions [will be] available at `http://github.con/andersbll/particle_segmentation`. For exact details about our method and choice of parameters, we encourage the reader to inspect the code.

2. Dense pixel classification

Our particle recognition system builds on the idea that a good pixel classification into meaningful components should can greatly simplify the segmentation task and allow us to use use simple image processing techniques. In this section, we describe the method that learns to assign labels densely to all image pixels.

2.1. Problem formulation

In order to segment crystals in SEM images, we must be able to differentiate between 1) crystals and the background, and 2) adjacent crystals. This can be achieved by classifying the image pixels into $n_c = 3$ classes: crystals, crystal borders and background as shown in Figure 1. Note that a pixel can belong to several classes, e.g. a pixel on the border between two particles is both a border pixel and a particle pixel. Therefore, we model the problem as multi-label binary classification,

To simplify the pixel classification task, we process entire SEM images patch-wise. That is, we extract and process an image patch of size $Q \times Q$ pixels together with class labels from the center of the image patch. When generating labels for an entire image, we process overlapping image patches and combine the labels afterwards. More formally, we wish to learn a mapping

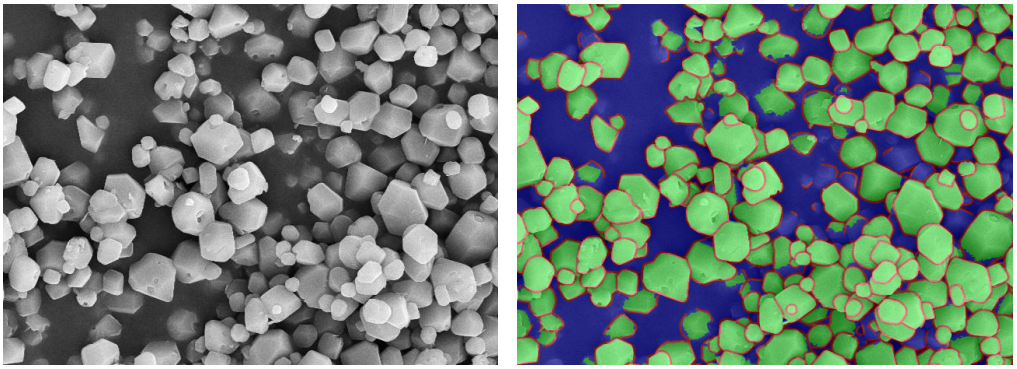


Figure 1: Training image with annotations. In the right image, the three colors represent the classes: crystal (green), border (red) and background (blue). For simplicity, we show only a single class per pixel, though, we allow pixels belonging to multiple classes. Note that we delineate crystal borders even if the crystals are recessed and part of the background.

from an image patch $\boldsymbol{x} \in \mathbb{R}^{Q \times Q}$ to the class probabilities $\boldsymbol{y} \in \mathbb{R}^{n_c}$. That is, we are modeling $p(\boldsymbol{y} \mid \boldsymbol{x})$ in probabilistic terms.

2.2. Image preprocessing

Because of high variability in crystals depths, the exposure level can vary substantially between SEM images. To compensate for this, we perform *local contrast normalization*. That is, we calculate the local pixel mean and standard deviation use these to normalize the image pixels. See Figure 3b for an example.

2.3. Network architecture and training

We employ a convolutional neural network to solve the task of mapping image patches to class labels. The network architecture follows a classical recipe consisting of consecutive layers of convolution and pooling/subsampling followed by fully-connected layers. See Figure 2 for an overview of our network architecture. We use *rectified linear units* as activation functions in our network because they are not prone to *vanishing gradients* compared to classical sigmoidal functions (Nair and Hinton, 2010). We train the network using mini-batch stochastic gradient descent with momentum (Sutskever et al., 2013).

As objective function for optimizing the network parameters, we choose the binary cross-entropy between a target t and a prediction p:

$$\mathcal{L}_{\text{BinCE}}(p, t) = t \log p + (1 - t) \log(1 - p) \tag{1}$$

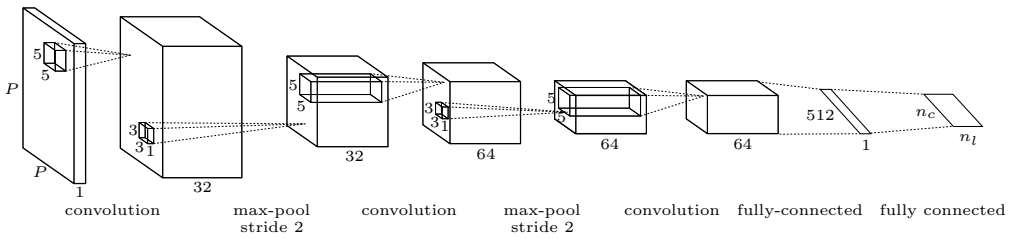


Figure 2: Convolutional network architecture. Window sizes of convolutional filters and pooling operations are provided for each operation.

Let the mapping $f(\boldsymbol{x}) : \mathbb{R}^{Q\times Q} \rightarrow \mathbb{R}^{n_c}$ represent our network output predictions with a sigmoid operation appended to ensure that the output is a probability measure in the range $[0, 1]$. Moreover, let $\boldsymbol{t} \in \mathbb{R}^{n_c}$ be the user-annotated pixel targets, that is a vector of zeros and ones denoting which classes a pixel belongs to. We then formulate the objective function as the sum over all class predictions:

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{t}) = \sum_{i=1}^{n_c} \mathcal{L}_{\text{BinCE}} \left(f(\boldsymbol{x})_i, t_i \right) \tag{2}$$

For our problem, we have found it beneficial to extend the vanilla objective function for two reasons. First, the dataset might be *unbalanced* meaning that we have uneven number of examples per class. During training, this will cause the gradient updates to favor the most occurring classes because these are seen more often. To alleviate this problem we associate a weight vector $\boldsymbol{w} \in \mathbb{R}^{n_c}$ with values inversely proportional to the number of occurrences in the training set for each class. Second, we wish to smooth our pixel classifications by averaging over several predictions. Typically, this is achieved using separately trained networks, aka. *ensembling* (Hansen and Salamon, 1990). We exploit the spatial structure of our data by letting the network predict classes for a small pixel neighborhood instead of a single pixel. Ensemble prediction is then performed by averaging overlapping predictions when classifying an entire image. To achieve this, we change our network architecture to output predictions for multiple pixels corresponding to a patch of size $L \times L$ pixels centered on the input image patch. Thus, the network becomes $f(\boldsymbol{x}) : \mathbb{R}^{Q\times Q} \rightarrow \mathbb{R}^{n_l \times n_c}, n_l = L^2$, and the target labels become $\boldsymbol{T} \in \mathbb{R}^{n_l \times n_c}$. For our problem, we find that multiple outputs in the last layer gives a visually

similar result to ordinary ensembling with the benefit of being computationally inexpensive as it does not require multiple networks. Combining the two extensions, we arrive at the following multi-label weighted cross-entropy as objective function for our network.

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{T}) = \sum_{j=1}^{n_l} \sum_{i=1}^{n_c} w_i \mathcal{L}_{\text{BinCE}} \left(f(\boldsymbol{x})_{ij}, t_{ij} \right) \tag{3}$$

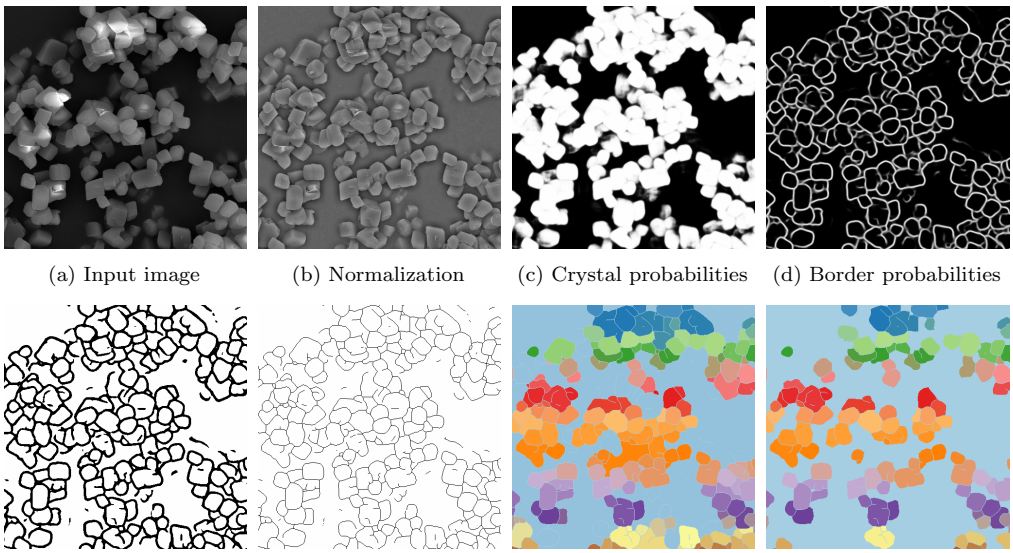
Neural networks are prone to overfitting because of the large representational capacity of their many parameters. For our application with very limited amounts of training data, this can be especially challenging. We alleviate the problem by performing *dropout* (Srivastava et al., 2014) on the first fully-connected layer. Moreover, we perform data augmentation to expand the available training data, that is, we replicate the training image with small perturbations in rotation, scale, brightness and contrast. With these precautions, we observe no overfitting on a separately annotated validation image.

We use the annotated image shown in Figure 1 to train the network that forms the basis for the rest of our results. Using a GPU to accelerate computations, the network training converges in less than 10 minutes.

3. Segmenting and measuring crystals

Given a good crystal/border/background classification of an SEM image, the task of segmenting crystals becomes fairly straightforward using off-the-shelf image processing tools (thresholding, morphology and watershed transformations).

To segment crystals individually. As example, we show a given input image (Figure 3a) and its local contrast normalized version (Figure 3b). From this we predict the class probabilities using our learned pixel classifier. In Figure 3c and 3d we show the probabilities for the crystal and the border classes respectively. We threshold the border probabilities using local threshold estimates which gives us a binary mask (Figure 3e) which we skeletonize using morphological operations (Figure 3f). We then perform a marker-based watershed transformation based on the border skeleton yielding separated image components as shown in Figure 3g. To discard background components from crystal components, we measure their average pixel probabilities and discard components with a crystal probability of less that 0.95. This remaining crystal components are shown in Figure 3h.



(a) Input image (b) Normalization (c) Crystal probabilities (d) Border probabilities

(e) Border mask (f) Border skeleton (g) Watershed labeling (h) Accepted crystals

Figure 3: Overview of the crystal recognition pipeline. The color scheme in (g) and (h) represents separated image components with unique colors.

In order to measure the size of each crystal component, we find the convex hull of the component. This is done as a simple attempt to compensate for crystals that are not entirely visible due to occlusion. Similar idiosyncratic conventions applies when measuring the crystals manually. We measure the crystal size as the diameter of the area assuming it is a circle.

4. Results

We apply our method to SEM images of calcite ($CaCO_3$) crystals aged in saturated solutions at 100 and 200 °C for up to 261 days. Mineralogists describe this type of particle growth as *Ostwald ripening*, where molecules dissolve from smaller particles and reprecipitate on larger particles. While the solution chemistry does not change, the average particle size grows to become more thermodynamically stable (Lifshitz and Slyozov, 1961). In Figure 4 we show SEM images captured from the 200 °C experiment. The images reveal a great level of variability in crystal appearance compared to the single image we use for training (Figure 1).

Each image is processed as described above yielding a histogram of crystal sizes, which we compare with distributions obtained by manually measuring

the crystal sizes as shown in Figure 5. To quantify the difference between the two distributions, we perform a two-sample Kolmogorov-Smirnov test under the null hypothesis the two samples come from the same distribution. The p values from this test for each image are shown in Figure 5. Out of 15 test images, 11 have a p value above 0.05 and 9 have a p value over 0.10. For distributions with lower p values, we observe the pattern that automatic measurements tend to count more small particles. To compensate for the skewness of the distributions, we show the log-transformed data in Figure 6. In this plot, we report the p values calculated from Welch's t-test (Welch, 1947) assuming normality of the log-transformed crystal sizes. These p values are in accordance with those of the Kolmogorov-Smirnov test.

Though we generally observe a good fit between the distributions measured automatically versus by hand, the discrepancy between distributions is relatively substantial for day 1 and day 3 from the 100 °C experiment. To investigate this, we show an image crop from day 3 in Figure 7. It is clear that the problems are caused by poor pixel classification. The image contains many overlapping crystals and a low-contrast region in the upper middle part of the image where crystals are hard to separate correctly (even for humans). Compared to our single training image from Figure 1, the crystal appearance in is simply too different making pixel classification challenging.

Finally, we report the relative number of particles found by our method versus the number of particles found by hand to 0.67. Our method finds only 2/3 of the particles found manually because we discard particles with a strict threshold based on the crystal probability measure. We have chosen to err on the side of caution as we can easily process more image data if needed.

5. Discussion and conclusion

Motivated by the study in Schultz et al. (2014), we have tried to replicate their manually measured results with automatic measurements using the same data. As there are only slight discrepancies between our and their results, we believe that our method is a promising approach. Especially since automated measurements allow for experiments of a much larger scale and thereby can provide data for richer statistical analyses.

From the results it is clear that our method has limitations. In regions where crystal borders are missing due to low contrast, the pixel classification will fail to recognize the borders. This situation is understandable since the image data is substantially different from the training data and we cannot

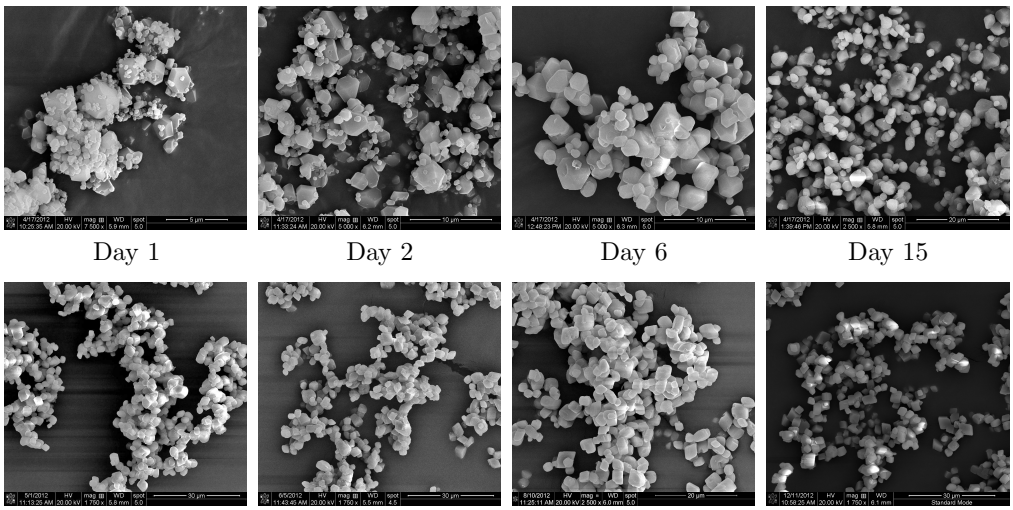


Figure 4: SEM images of calcite ($CaCO_3$) crystals coarsening over time at 200 °C.

expect our classifier to generalize well to such extreme cases. A human would simply discard the entire low-contrast region because of the many ambiguities, and ideally, we would want an automated method to detect such anomalies in order to handle them gracefully. However, it is beyond the scope of this work to perform anomaly detection since this is completely different from segmentation.

Our method sets itself apart from previous methods in the literature. Usually, these rely on simple image features, i.e. color intensities, and apply advanced segmentation techniques on top. In contrast, we argue that segmentation is difficult on top of raw pixels since these are noisy and do not incorporate knowledge about the local image structure. We propose a learning-based approach to segmentation that relies on classifying pixels into *meaningful* classes. That is, we want to know if a pixel is part of a crystal or not. Also, we want to know if a pixel lies on the border of a crystal. We leverage convnets to learn such a pixel classifier which allows us to learn directly from the image input without engineering suitable image features. Using the pixel classifier, we can perform a simple segmentation on top of the crystal/border probabilities. Moreover, the crystal probabilities allow us to discard recessed background crystal that cannot be measured properly.

Our approach is a step towards automating the segmentation pipeline

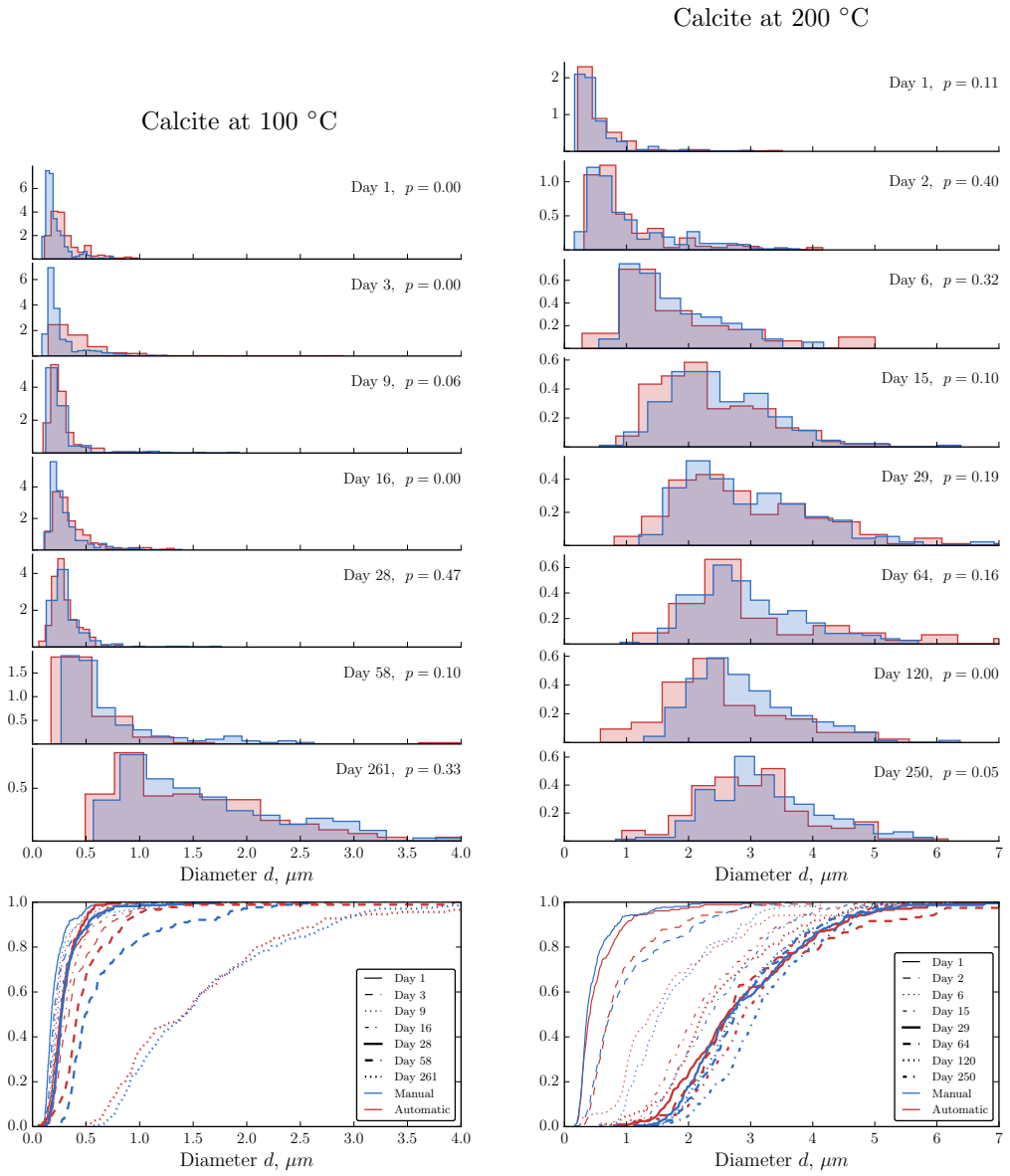


Figure 5: Evolution of crystal size distribution over time. Manually measured distributions are colored blue and automatic measurements (our method) are colored red. The two columns show crystal development in two different experiments. The bottom plots collect distributions in a cumulative histogram. The x*axis is truncated slightly for clarity. For each plot we report the significance level p for the null hypothesis that the distributions are the same.

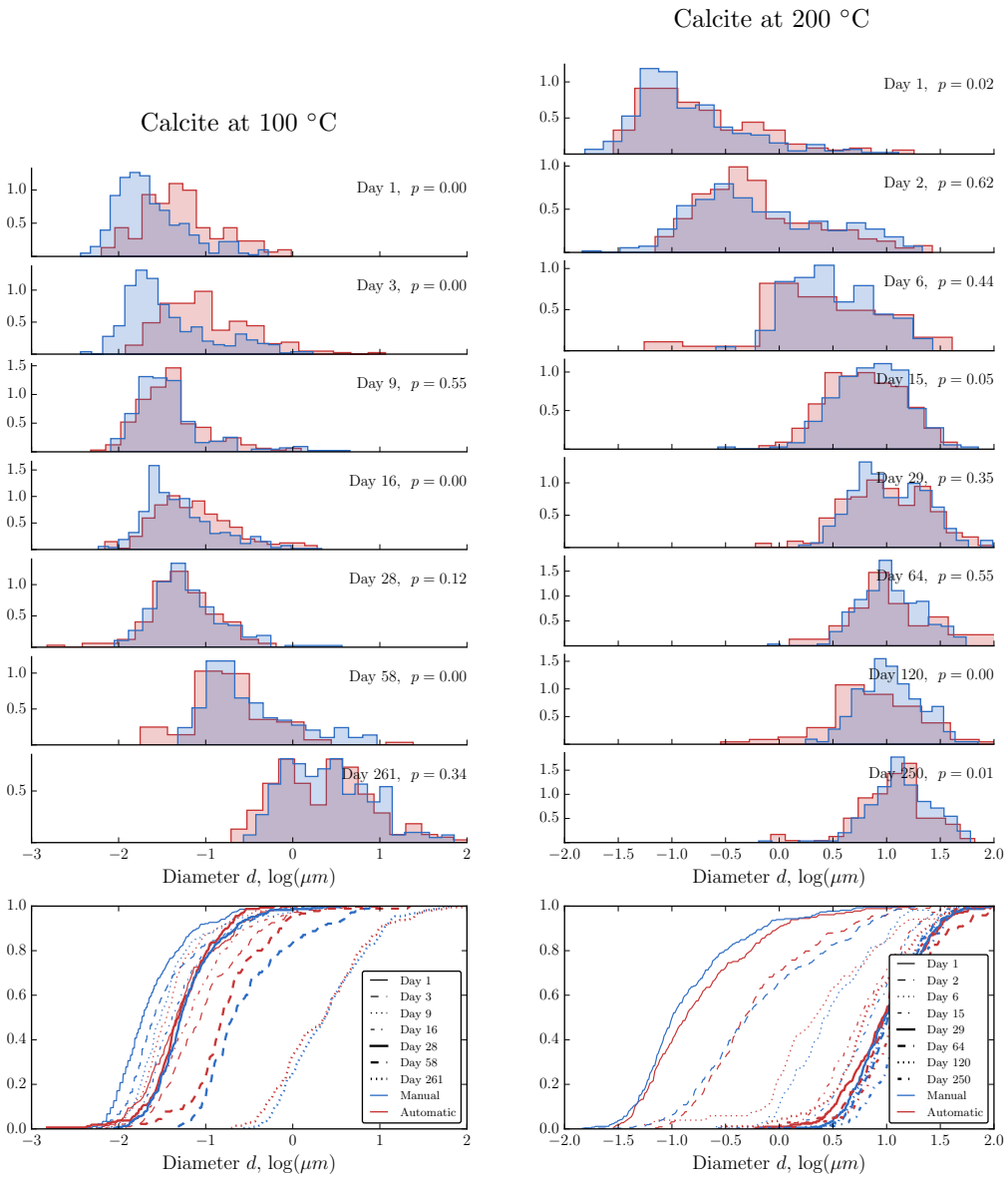
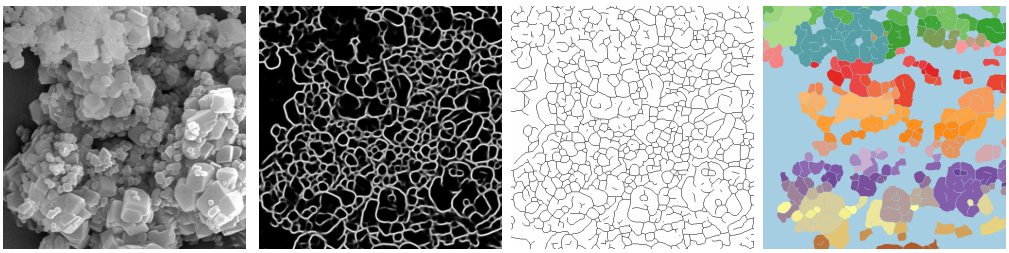


Figure 6: log-transformed version of Figure 5.



(a) Input image (b) Border probabilities (c) Border skeleton (d) Accepted crystals

Figure 7: Difficult image crop from day 3 of calcite aging at 100 °C. The image contains low contrast regions and many overlapping crystals. This causes the border segmentation to fail such that small particles are recognized as fewer larger particles.

as we try to learn the challenging part of differentiating between image structures. Ideally, we would learn the last image processing steps as well by incorporating the spatial segmentation in the learning method . However, this would require substantially more training data in order to learn about the spatial structure of entire crystals. Therefore, in this work we have sought a compromise that requires very little training data but greatly simplifies the segmentation pipeline.

Acknowledgments

We would like to thank Knut Conradsen for insightful discussions. Our method is built in Python using the libraries *SciPy* (Jones et al., 2001–), *scikit-image* (van der Walt et al., 2014), *CUDArray* (Larsen, 2014) and *DeepPy* (Larsen, 2016). A.B.L.L. is funded by the *Danish Council for Strategic Research*.

References

Bengio, Y., Courville, A., Vincent, P., Aug 2013. Representation learning: A review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (8), 1798–1828.

Boykov, Y., Kolmogorov, V., Sept 2004. An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision. IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (9), 1124–1137.

Ciresan, D., Giusti, A., Gambardella, L. M., Schmidhuber, J., 2012. Deep neural networks segment neuronal membranes in electron microscopy images. In: Pereira, F., Burges, C., Bottou, L., Weinberger, K. (Eds.), Advances in Neural Information Processing Systems 25. Curran Associates, Inc., pp. 2843–2851.

Dahl, A., Larsen, R., 2011. Learning dictionaries of discriminative image patches. In: Proceedings of the British Machine Vision Conference. BMVA Press, pp. 77.1–77.11.

Hansen, L., Salamon, P., 1990. Neural network ensembles. IEEE Transactions on Pattern Analysis and Machine Intelligence 12 (10), 993–1001.

Jones, E., Oliphant, T., Peterson, P., et al., 2001–. SciPy: Open source scientific tools for Python. [Online; accessed 2016-06-08].
URL `http://www.scipy.org/`

Jungmann, M., Pape, H., Wikirchen, P., Clauser, C., Berlage, T., 2014. Segmentation of thin section images for grain size analysis using region competition and edge-weighted region merging. Computers & Geosciences 72, 33 – 48.

Laptev, D., Vezhnevets, A., Dwivedi, S., Buhmann, J. M., 2012. Medical Image Computing and Computer-Assisted Intervention – MICCAI 2012: 15th International Conference, Nice, France, October 1-5, 2012, Proceedings, Part I. Springer Berlin Heidelberg, Berlin, Heidelberg, Ch. Anisotropic ssTEM Image Segmentation Using Dense Correspondence across Sections, pp. 323–330.

Larsen, A. B. L., 2014. CUDArray: CUDA-based NumPy. Tech. Rep. DTU Compute 2014-21, Department of Applied Mathematics and Computer Science, Technical University of Denmark.

Larsen, A. B. L., 2016. DeepPy: Pythonic deep learning. Tech. Rep. DTU Compute 2016-6, Department of Applied Mathematics and Computer Science, Technical University of Denmark.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521 (7553), 436–444.

Lifshitz, I., Slyozov, V., 1961. The kinetics of precipitation from supersaturated solid solutions. Journal of Physics and Chemistry of Solids 19 (1), 35 – 50.

Liu, C., Yuen, J., Torralba, A., May 2011. Sift flow: Dense correspondence across scenes and its applications. IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (5), 978–994.

Lowe, D. G., 2004. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60 (2), 91–110.

Lu, B., Cui, M., liu, Q., Wang, Y., 2009. Automated grain boundary detection using the level set method. Computers & Geosciences 35 (2), 267 – 275.

Mingireanov Filho, I., Spina, T. V., Falcao, A. X., Vidal, A. C., 2013. Segmentation of sandstone thin section images with separation of touching grains using optimum path forest operators. Computers and Geosciences 57, 146–157.

Nair, V., Hinton, G. E., 2010. Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th International Conference on Machine Learning (ICML-10). pp. 807–814.

Pascal Asmussen and Olaf Conrad and Andreas Günther and Moritz Kirsch and Ulrich Riller, 2015. Semi-automatic segmentation of petrographic thin section images using a "seeded-region growing algorithm" with an application to characterize wheathered subarkose sandstone. Computers and Geosciences 83, 89 – 99.

Røgen, B., Gommesen, L., Fabricius, I. L., 2001. Grain size distributions of chalk from image analysis of electron micrographs. Computers and Geosciences 27 (9), 1071 – 1080, geological Applications of Digital Imaging.

Schultz, L. N., Dideriksen, K., Lakshtanov, L., Hakim, S. S., Mter, D., Hauer, F., Bechgaard, K., Stipp, S. L. S., 2014. From nanometer aggregates to micrometer crystals: Insight into the coarsening mechanism of calcite. Crystal Growth & Design 14 (2), 552–558.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., Jan. 2014. Dropout: A simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15 (1), 1929–1958.

Sutskever, I., Martens, J., Dahl, G. E., Hinton, G. E., May 2013. On the importance of initialization and momentum in deep learning. In: Dasgupta, S., Mcallester, D. (Eds.), Proceedings of the 30th International Conference on Machine Learning (ICML-13). Vol. 28 (3). JMLR Workshop and Conference Proceedings, pp. 1139–1147.

van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., Yu, T., the scikit-image contributors, 6 2014. scikit-image: image processing in Python. PeerJ 2, e453.

Welch, B. L., 1947. The generalization of 'student's' problem when several different population variances are involved. Biometrika 34 (1/2), 28–35. URL `http://www.jstor.org/stable/2332510`

D Autoencoding beyond pixels using a learned similarity metric

Autoencoding beyond pixels using a learned similarity metric

Anders Boesen Lindbo Larsen[1] ABLL@DTU.DK
Søren Kaae Sønderby[2] SKAAESONDERBY@GMAIL.COM
Hugo Larochelle[3] HLAROCHELLE@TWITTER.COM
Ole Winther[1,2] OLWI@DTU.DK

[1] Department for Applied Mathematics and Computer Science, Technical University of Denmark
[2] Bioinformatics Centre, Department of Biology, University of Copenhagen, Denmark
[3] Twitter, Cambridge, MA, USA

Abstract

We present an autoencoder that leverages learned representations to better measure similarities in data space. By combining a variational autoencoder (VAE) with a generative adversarial network (GAN) we can use learned feature representations in the GAN discriminator as basis for the VAE reconstruction objective. Thereby, we replace element-wise errors with feature-wise errors to better capture the data distribution while offering invariance towards e.g. translation. We apply our method to images of faces and show that it outperforms VAEs with element-wise similarity measures in terms of visual fidelity. Moreover, we show that the method learns an embedding in which high-level abstract visual features (e.g. wearing glasses) can be modified using simple arithmetic.

1. Introduction

Deep architectures have allowed a wide range of discriminative models to scale to large and diverse datasets. However, generative models still have problems with complex data distributions such as images and sound. In this work, we show that currently used similarity metrics impose a hurdle for learning good generative models and that we can improve a generative model by employing a learned similarity measure.

When learning models such as the variational autoencoder (Kingma & Welling, 2014; Rezende et al., 2014), the choice of similarity metric is central as it provides the main part of the training signal via the reconstruction error objective. For this task, element-wise measures like the squared error are the default. Element-wise metrics are simple but not very suitable for image data, as they do not model the properties of human visual perception. E.g. a small image translation might result in a large pixel-wise error whereas a human would barely notice the change. Therefore, we argue in favor of measuring image similarity using a higher-level and *sufficiently invariant* representation of the images. Rather than hand-engineering a suitable measure to accommodate the problems of element-wise metrics, we want to learn a function for the task. The question is how to learn such a similarity measure? We find that by jointly training a VAE and a generative adversarial network (Goodfellow et al., 2014) we can use the GAN discriminator to measure sample similarity. We achieve this by combining a VAE with a GAN as shown in Fig. 1. We collapse the VAE decoder and the GAN generator into one by letting them share parameters and training them jointly. For the VAE training objective, we replace the typical element-wise reconstruction metric with a feature-wise metric expressed in the discriminator.

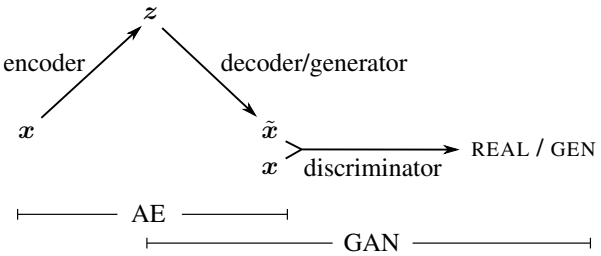


Figure 1. Overview of our network. We combine a VAE with a GAN by collapsing the decoder and the generator into one.

1.1. Contributions

Our contributions are:

Proceedings of the 33^{rd} International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48.

- We combine VAEs and GANs into an unsupervised generative model that simultaneously learns to *encode*, *generate* and *compare* dataset samples.
- We show that generative models trained with learned similarity measures produce better image samples than models trained with element-wise error measures.
- We demonstrate that unsupervised training results in a latent image representation with disentangled factors of variation (Bengio et al., 2013). This is illustrated in experiments on a dataset of face images labelled with visual attribute vectors, where it is shown that simple arithmetic applied in the learned latent space produces images that reflect changes in these attributes.

2. Autoencoding with learned similarity

In this section we provide background on VAEs and GANs. Then, we introduce our method for combining both approaches, which we refer to as VAE/GAN. As we'll describe, our proposed hybrid is motivated as a way to improve VAE, so that it relies on a more meaningful, feature-wise metric for measuring reconstruction quality during training.

2.1. Variational autoencoder

A VAE consists of two networks that *encode* a data sample $\boldsymbol{x}$ to a latent representation $\boldsymbol{z}$ and *decode* the latent representation back to data space, respectively:

$$\boldsymbol{z} \sim \mathrm{Enc}(\boldsymbol{x}) = q(\boldsymbol{z}|\boldsymbol{x}) \;, \quad \tilde{\boldsymbol{x}} \sim \mathrm{Dec}(\boldsymbol{z}) = p(\boldsymbol{x}|\boldsymbol{z}) \;. \quad (1)$$

The VAE regularizes the encoder by imposing a prior over the latent distribution $p(\boldsymbol{z})$. Typically $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ is chosen. The VAE loss is minus the sum of the expected log likelihood (the reconstruction error) and a prior regularization term:

$$\mathcal{L}_{\mathrm{VAE}} = -\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})}{q(\boldsymbol{z}|\boldsymbol{x})}\right] = \mathcal{L}^{\mathrm{pixel}}_{\mathrm{llike}} + \mathcal{L}_{\mathrm{prior}} \quad (2)$$

with

$$\mathcal{L}^{\mathrm{pixel}}_{\mathrm{llike}} = -\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}\left[\log p(\boldsymbol{x}|\boldsymbol{z})\right] \quad (3)$$

$$\mathcal{L}_{\mathrm{prior}} = D_{\mathrm{KL}}(q(\boldsymbol{z}|\boldsymbol{x})\|p(\boldsymbol{z})) \;, \quad (4)$$

where D_{KL} is the Kullback-Leibler divergence.

2.2. Generative adversarial network

A GAN consists of two networks: the *generator* network $\mathrm{Gen}(\boldsymbol{z})$ maps latents $\boldsymbol{z}$ to data space while the *discriminator* network assigns probability $y = \mathrm{Dis}(\boldsymbol{x}) \in [0, 1]$ that $\boldsymbol{x}$ is an actual training sample and probability $1 - y$ that $\boldsymbol{x}$ is generated by our model through $\boldsymbol{x} = \mathrm{Gen}(\boldsymbol{z})$ with $\boldsymbol{z} \sim p(\boldsymbol{z})$. The GAN objective is to find the binary classifier that gives the best possible discrimination between true and generated data and simultaneously encouraging Gen to fit the true data distribution. We thus aim to maximize/minimize the binary cross entropy:

$$\mathcal{L}_{\mathrm{GAN}} = \log(\mathrm{Dis}(\boldsymbol{x})) + \log(1 - \mathrm{Dis}(\mathrm{Gen}(\boldsymbol{z}))) \;, \quad (5)$$

with respect to Dis / Gen with $\boldsymbol{x}$ being a training sample and $\boldsymbol{z} \sim p(\boldsymbol{z})$.

2.3. Beyond element-wise reconstruction error with VAE/GAN

An appealing property of GAN is that its discriminator network implicitly has to learn a rich similarity metric for images, so as to discriminate them from "non-images". We thus propose to exploit this observation so as to transfer the properties of images learned by the discriminator into a more abstract reconstruction error for the VAE. The end result will be a method that combines the advantage of GAN as a high quality generative model and VAE as a method that produces an encoder of data into the latent space $\boldsymbol{z}$.

Specifically, since element-wise reconstruction errors are not adequate for images and other signals with invariances, we propose replacing the VAE reconstruction (expected log likelihood) error term from Eq. 3 with a reconstruction error expressed in the GAN discriminator. To achieve this, let $\mathrm{Dis}_l(\boldsymbol{x})$ denote the hidden representation of the lth layer of the discriminator. We introduce a Gaussian observation model for $\mathrm{Dis}_l(\boldsymbol{x})$ with mean $\mathrm{Dis}_l(\tilde{\boldsymbol{x}})$ and identity covariance:

$$p(\mathrm{Dis}_l(\boldsymbol{x})|\boldsymbol{z}) = \mathcal{N}(\mathrm{Dis}_l(\boldsymbol{x})|\,\mathrm{Dis}_l(\tilde{\boldsymbol{x}}), \mathbf{I}) \;, \quad (6)$$

where $\tilde{\boldsymbol{x}} \sim \mathrm{Dec}(\boldsymbol{z})$ is the sample from the decoder of $\boldsymbol{x}$. We can now replace the VAE error of Eq. 3 with

$$\mathcal{L}^{\mathrm{Dis}_l}_{\mathrm{llike}} = -\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}\left[\log p(\mathrm{Dis}_l(\boldsymbol{x})|\boldsymbol{z})\right] \quad (7)$$

We train our combined model with the triple criterion

$$\mathcal{L} = \mathcal{L}_{\mathrm{prior}} + \mathcal{L}^{\mathrm{Dis}_l}_{\mathrm{llike}} + \mathcal{L}_{\mathrm{GAN}} \;. \quad (8)$$

Notably, we optimize the VAE wrt. $\mathcal{L}_{\mathrm{GAN}}$ which we regard as a *style* error in addition to the reconstruction error which can be interpreted as a *content* error using the terminology from Gatys et al. (2015). Moreover, since both Dec and Gen map from $\boldsymbol{z}$ to $\boldsymbol{x}$, we share the parameters between the two (or in other words, we use Dec instead of Gen in Eq. 5).

In practice, we have observed the devil in the details during development and training of this model. We therefore provide a list of practical considerations in this section. We refer to Fig. 2 and Alg. 1 for overviews of the training procedure.

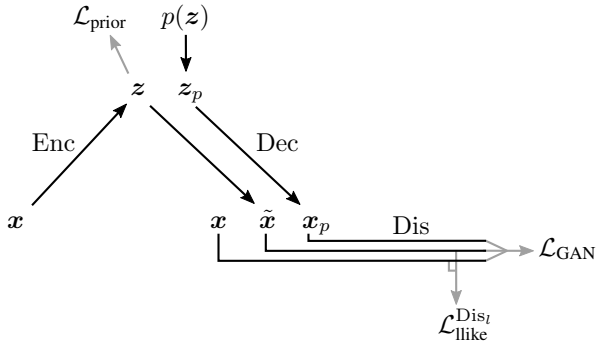


Figure 2. Flow through the combined VAE/GAN model during training. Gray lines represent terms in the training objective.

Limiting error signals to relevant networks Using the loss function in Eq. 8, we train both a VAE and a GAN simultaneously. This is possible because we do not update all network parameters wrt. the combined loss. In particular, Dis should not try to minimize $\mathcal{L}_{\text{llike}}^{\text{Dis}_l}$ as this would collapse the discriminator to 0. We also observe better results by not backpropagating the error signal from $\mathcal{L}_{\text{GAN}}$ to Enc.

Weighting VAE vs. GAN As Dec receives an error signal from both $\mathcal{L}_{\text{llike}}^{\text{Dis}_l}$ and $\mathcal{L}_{\text{GAN}}$, we use a parameter γ to weight the ability to reconstruct vs. fooling the discriminator. This can also be interpreted as weighting *style* and *content*. Rather than applying γ to the entire model (Eq. 8), we perform the weighting only when updating the parameters of Dec:

$$\boldsymbol{\theta}_{\text{Dec}} \overset{+}{\leftarrow} -\nabla_{\boldsymbol{\theta}_{\text{Dec}}}(\gamma \mathcal{L}_{\text{llike}}^{\text{Dis}_l} - \mathcal{L}_{\text{GAN}}) \tag{9}$$

Discriminating based on samples from $p(\boldsymbol{z})$ and $q(\boldsymbol{z}|\boldsymbol{x})$ We observe better results when using samples from $q(\boldsymbol{z}|\boldsymbol{x})$ (i.e. the encoder Enc) in addition to our prior $p(\boldsymbol{z})$ in the GAN objective:

$$\begin{aligned}\mathcal{L}_{\text{GAN}} = \; & \log(\text{Dis}(\boldsymbol{x})) + \log(1 - \text{Dis}(\text{Dec}(\boldsymbol{z}))) \\ & + \log(1 - \text{Dis}(\text{Dec}(\text{Enc}(\boldsymbol{x}))))\end{aligned} \tag{10}$$

Note that the regularization of the latent space $\mathcal{L}_{\text{prior}}$ should make the set of samples from either $p(\boldsymbol{z})$ or $q(\boldsymbol{z}|\boldsymbol{x})$ similar. However, for any given example $\boldsymbol{x}$, the negative sample $\text{Dec}(\text{Enc}(\boldsymbol{x}))$ is much more likely to be similar to $\boldsymbol{x}$ than $\text{Dec}(\boldsymbol{z})$. When updating according to $\mathcal{L}_{\text{GAN}}$, we suspect that having similar positive and negative samples makes for a more useful learning signal.

Algorithm 1 Training the VAE/GAN model

$\boldsymbol{\theta}_{\text{Enc}}, \boldsymbol{\theta}_{\text{Dec}}, \boldsymbol{\theta}_{\text{Dis}} \leftarrow$ initialize network parameters
repeat
  $\boldsymbol{X} \leftarrow$ random mini-batch from dataset
  $\boldsymbol{Z} \leftarrow \text{Enc}(\boldsymbol{X})$
  $\mathcal{L}_{\text{prior}} \leftarrow D_{\text{KL}}(q(\boldsymbol{Z}|\boldsymbol{X}) \| p(\boldsymbol{Z}))$
  $\tilde{\boldsymbol{X}} \leftarrow \text{Dec}(\boldsymbol{Z})$
  $\mathcal{L}_{\text{llike}}^{\text{Dis}_l} \leftarrow -\mathbb{E}_{q(\boldsymbol{Z}|\boldsymbol{X})}[p(\text{Dis}_l(\boldsymbol{X})|\boldsymbol{Z})]$
  $\boldsymbol{Z}_p \leftarrow$ samples from prior $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
  $\boldsymbol{X}_p \leftarrow \text{Dec}(\boldsymbol{Z}_p)$
  $\mathcal{L}_{\text{GAN}} \leftarrow \log(\text{Dis}(\boldsymbol{X})) + \log(1 - \text{Dis}(\tilde{\boldsymbol{X}})) + \log(1 - \text{Dis}(\boldsymbol{X}_p))$
  // Update parameters according to gradients
  $\boldsymbol{\theta}_{\text{Enc}} \overset{+}{\leftarrow} -\nabla_{\boldsymbol{\theta}_{\text{Enc}}}(\mathcal{L}_{\text{prior}} + \mathcal{L}_{\text{llike}}^{\text{Dis}_l})$
  $\boldsymbol{\theta}_{\text{Dec}} \overset{+}{\leftarrow} -\nabla_{\boldsymbol{\theta}_{\text{Dec}}}(\gamma \mathcal{L}_{\text{llike}}^{\text{Dis}_l} - \mathcal{L}_{\text{GAN}})$
  $\boldsymbol{\theta}_{\text{Dis}} \overset{+}{\leftarrow} -\nabla_{\boldsymbol{\theta}_{\text{Dis}}}\mathcal{L}_{\text{GAN}}$
until deadline

3. Related work

Element-wise distance measures are notoriously inadequate for complex data distributions like images. In the computer vision community, preprocessing images is a prevalent solution to improve robustness to certain perturbations. Examples of preprocessing are contrast normalization, working with gradient images or pixel statistics gathered in histograms. We view these operations as a form of metric engineering to account for the shortcomings of simple element-wise distance measures. A more detailed discussion on the subject is provided by Wang & Bovik (2009).

Neural networks have been applied to metric learning in form of the Siamese architecture (Bromley et al., 1993; Chopra et al., 2005). The learned distance metric is minimized for similar samples and maximized for dissimilar samples using a max margin cost. However, since Siamese networks are trained in a supervised setup, we cannot apply them directly to our problem.

Several attempts at improving on element-wise distances for generative models have been proposed within the last year. Ridgeway et al. (2015) apply the structural similarity index as an autoencoder (AE) reconstruction metric for grey-scale images. Yan et al. (2015) let a VAE output two additional images to learn shape and edge structures more explicitly. Mansimov et al. (2015) append a GAN-based sharpening step to their generative model. Mathieu et al. (2015) supplement a squared error measure with both a GAN and an image gradient-based similarity measure to improve image sharpness of video prediction. While all these extensions yield visibly sharper images, they do not have the same potential for capturing high-level structure compared to a deep learning approach.

In contrast to AEs that model the relationship between a dataset sample and a latent representation directly, GANs learn to generate samples indirectly. By optimizing the

GAN generator to produce samples that imitate the dataset according to the GAN discriminator, GANs avoid element-wise similarity measures by construction. This is a likely explanation for their ability to produce high-quality images as demonstrated by Denton et al. (2015); Radford et al. (2015).

Lately, convolutional networks with upsampling have shown useful for generating images from a latent representation. This has sparked interest in learning image embeddings where semantic relationships can be expressed using simple arithmetic – similar to the suprising results of the *word2vec* model by Mikolov et al. (2013). First, Dosovitskiy et al. (2015) used supervised training to train convolutional network to generate chairs given high-level information about the desired chair. Later, Kulkarni et al. (2015); Yan et al. (2015); Reed et al. (2015) have demonstrated encoder-decoder architectures with disentangled feature representations, but their training schemes rely on supervised information. Radford et al. (2015) inspect the latent space of a GAN after training and find directions corresponding to eyeglasses and smiles. As they rely on pure GANs, however, they cannot encode images making it challenging to explore the latent space.

Our idea of a learned similarity metric is partly motivated by the neural artistic style network of Gatys et al. (2015) who demonstrate the representational power of deep convolutional features. They obtain impressive results by optimizing an image to have similar features as a subject image and similar feature correlations as a style image in a pre-trained convolutional network. In our VAE/GAN model, one could view $\mathcal{L}_{\text{llike}}^{\text{Dis}_l}$ as content and $\mathcal{L}_{\text{GAN}}$ as style. Our style term, though, is not computed from feature correlations but is the error signal from trying to fool the GAN discriminator.

4. Experiments

Measuring the quality of generative models is challenging as current evaluation methods are problematic for larger natural images (Theis et al., 2015). In this work, we use images of size 64x64 and focus on more qualitative assessments since traditional log likelihood measures do not capture visual fidelity. Indeed, we have tried discarding the GAN discriminator after training of the VAE/GAN model and computing a pixel-based log likelihood using the remaining VAE. The results are far from competitive with plain VAE models (on the CIFAR-10 dataset). In an attempt to verify the idea of feature-based similarity metrics, we have trained a GAN on CIFAR-10. After training, we compute a feature representation of CIFAR-10 by propagating the images up in the GAN discriminator. We then measure the $k = 5$ nearest neighbor classification performance. Using a feature-based metric reduces the error to 33.73% from the pixel-based error of 66.02%.

In this section we investigate the performance of different generative models:

- Plain VAE with an element-wise Gaussian observation model.
- VAE with a learned distance ($\text{VAE}_{\text{Dis}_l}$). We first train a GAN and use the discriminator network as a learned similarity measure. We select a single layer l at which we measure the similariy according to Dis_l. l is chosen such that the comparison is performed after 3 convolutional layers with stride 2 downsampling.
- The combined VAE/GAN model. This model is similar to $\text{VAE}_{\text{Dis}_l}$ but we also optimize Dec wrt. $\mathcal{L}_{\text{GAN}}$. One might suspect that simultaneous training of the VAE and the GAN from noise initialization is problematic because the Dis_l representation starts out as a random projection of the data. However, we observe no instabilities in this regard.
- An alternative $\text{VAE/GAN}_{\text{Dis}_0}$ model where the VAE reconstruction error is measured in pixel space, $\mathcal{L}_{\text{llike}}^{\text{Dis}_0} = \mathcal{L}_{\text{llike}}^{\text{pixel}}$. This models serves to confirm that there is a benefit in using feature-based similarities and that the GAN is not single-handedly responsible for the more natural-looking image generation.
- A GAN. This modes has recently been shown capable of generating high-quality images (Radford et al., 2015).

All models share the same architectures for Enc, Dec and Dis respectively. For all our experiments, we use convolutional architectures and use *backward convolution* (aka. *fractional striding*) with stride 2 to upscale images in Dec. Backward convolution is achieved by flipping the convolution direction such that striding causes upsampling. Our models are trained with RMSProp using a learning rate of 0.0003 and a batch size of 64. In table 1 we list the network architectures. We refer to our implementation available online[1].

4.1. CelebA face images

We apply our methods to face images from the *CelebA* dataset[2] (Liu et al., 2015). This dataset consists of 202,599 images annotated with 40 binary attributes such as *eyeglasses*, *bangs*, *pale skin* etc. We scale and crop the images to 64×64 pixels and use only the images (not the attributes) for unsupervised training.

After training, we draw samples from $p(\boldsymbol{z})$ and propagate

[1] http://github.com/andersbll/autoencoding_beyond_pixels

[2] We use the aligned and cropped version of the dataset.

Enc	Dec	Dis
5×5 64 conv. ↓, BNorm, ReLU	8·8·256 fully-connected, BNorm, ReLU	5×5 32 conv., ReLU
5×5 128 conv. ↓, BNorm, ReLU	5×5 256 conv. ↑, BNorm, ReLU	5×5 128 conv. ↓, BNorm, ReLU
5×5 256 conv. ↓, BNorm, ReLU	5×5 128 conv. ↑, BNorm, ReLU	5×5 256 conv. ↓, BNorm, ReLU
2048 fully-connected, BNorm, ReLU	5×5 32 conv. ↑, BNorm, ReLU	5×5 256 conv. ↓, BNorm, ReLU
	5×5 3 conv., tanh	512 fully-connected, BNorm, ReLU
		1 fully-connected, sigmoid

Table 1. Architectures for the three networks that comprise VAE/GAN. ↓ and ↑ represent down- and upsampling respectively. BNorm denotes batch normalization (Ioffe & Szegedy, 2015). When batch normalization is applied to convolutional layers, per-channel normalization is used.

Figure 3. Samples from different generative models.

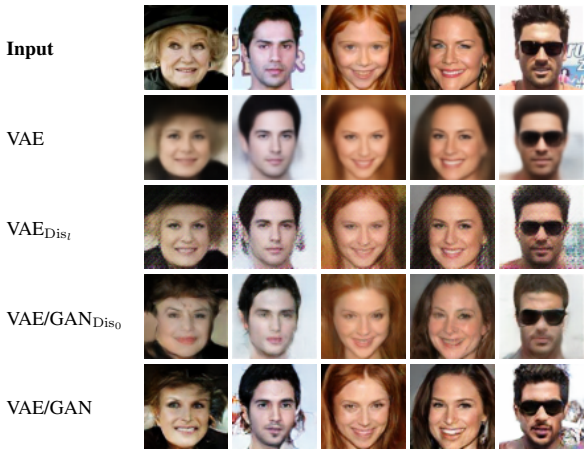


Figure 4. Reconstructions from different autoencoders.

these through Dec to generate new images which are shown in Fig. 3. The plain VAE is able draw the frontal part of the face sharply, but off-center the images get blurry. This is because the dataset aligns faces using frontal landmarks. When we move too far away from the aligned parts, the recognition model breaks down because pixel correspondence cannot be assumed. VAE$_{\text{Dis}_l}$ produces sharper images even off-center because the reconstruction error is lifted beyond pixels. However, we see severe noisy artifacts which we believe are caused by the harsh downsampling scheme of Dis. In comparison, VAE/GAN$_{\text{Dis}_0}$, VAE/GAN and pure GAN produce sharper images with more natural textures and face parts.

Next, we make the VAEs reconstruct images taken from a separate test set. Reconstruction is not possible with the GAN model as it lacks an encoder network. The results are shown in Fig. 4 and our conclusions are similar to what we observed for the random samples. Note however, that VAE/GAN$_{\text{Dis}_0}$ fails to capture the same level of detail as VAE/GAN with feature-based similarities.

Additionally, Fig. 5 shows the influence of the γ hyperparameter that balances gradient contributions to $\boldsymbol{\theta}_{\text{Dec}}$ from $\mathcal{L}_{\text{llike}}^{\text{Dis}_l}$ versus $\mathcal{L}_{\text{GAN}}$. We seek a trade-off between the two. If $\mathcal{L}_{\text{llike}}^{\text{Dis}_l}$ is too prominent we see artifacts from the feature-based reconstruction. If $\mathcal{L}_{\text{GAN}}$ is too prominent we loose details in the reconstruction, e.g. mouth shape.

4.1.1. Visual attribute vectors

Inspired by attempts at learning embeddings in which semantic concepts can be expressed using simple arithmetic (Mikolov et al., 2013), we inspect the latent space of a trained VAE/GAN model. The idea is to find directions in the latent space corresponding to specific visual features in image space.

We use the binary attributes of the dataset to extract *visual attribute vectors*. For all images we use the encoder to calculate latent vector representations. For each attribute, we compute the mean vector for images with the attribute and the mean vector for images without the attribute. We then compute the visual attribute vector as the difference between the two mean vectors. This is a very simple method for calculating visual attribute vectors that will have problems with highly correlated visual attributes such as *heavy makeup* and *wearing lipstick*. In Fig. 6, we show face images as well as the reconstructions after adding different visual attribute vectors to the latent representations. Though

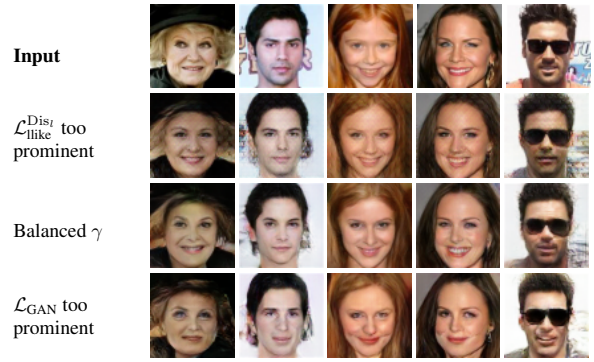


Figure 5. Adjusting the γ hyperparameter to balance gradient contributions to $\boldsymbol{\theta}_{\text{Dec}}$ from $\mathcal{L}_{\text{llike}}^{\text{Dis}_l}$ versus $\mathcal{L}_{\text{GAN}}$.

not perfect, we clearly see that the attribute vectors capture semantic concepts like *eyeglasses*, *bangs*, etc. E.g. when bangs are added to the faces, both the hair color and the hair texture matches the original face. We also see that being a man is highly correlated with having a mustache, which is caused by attribute correlations in the dataset. In comparison, the visual concepts learned by a plain VAE in the same manner are much less prominent, see Fig. 7.

4.2. Attribute similarity, Labeled faces in the wild

Inspired by the *attribute similarity* experiment of Yan et al. (2015), we seek a more quantative evaluation of our generated images. The idea is to learn a generative model for face images conditioned on facial attributes. At test time, we generate face images by retrieval from chosen attribute configurations and let a separately trained regressor network predict the attributes from the generated images. A good generative model should be able to produce visual attributes that are correctly recognized by the regression model. To imitate the original experiment, we use Labeled faces in the wild (LFW) images (Huang et al., 2007) with attributes (Kumar et al., 2009). We align the face images according to the landmarks in (Zhu et al., 2014). Additionally, we crop and resize the images to 64×64 pixels and augment the dataset with common operations. Again, we refer to our implementation online for more details.

We construct conditional VAE, GAN and VAE/GAN models by concatenating the attribute vector to the vector representation of the input in Enc, Dec and Dis similar to (Mirza & Osindero, 2014). For Enc and Dis, the attribute vector is concatenated to the input of the top fully connected layer. Our regression network has almost the same architecture as Enc. We train using the LFW training set, and during testing, we condition on the test set attributes and sample faces to be propagated through the regression network. Figure 8 shows faces generated by conditioning on attribute vectors from the test set. We report regressor performance

Model	Cosine similarity	Mean squared error
LFW test set	0.9193	14.1987
VAE	0.9030	27.59 ± 1.42
GAN	0.8892	27.89 ± 3.07
VAE/GAN	**0.9114**	**22.39 ± 1.16**

Table 2. Attribute similarity scores. To replicate (Yan et al., 2015), the cosine similarity is measured as the best out of 10 samples per attribute vector from the test set. The mean squared error is computed over the test set and statistics are measured over 25 runs.

numbers in Table 2. Compared to an ordinary VAE, the VAE/GAN model yields significantly better attributes visually that leads to smaller recognition error. The GAN network performs suprisingly poorly and we suspect that this is caused by instabilities during training (GAN models are very difficult to train reliably due to the minimax objective function). Note that our results are not directly comparable with those of Yan et al. (2015) since we do not have access to their preprocessing scheme nor regression model.

4.3. Unsupervised pretraining for supervised tasks

For completeness, we report that we have tried evaluating VAE/GAN in a semi-supervised setup by unsupervised pretraining followed by finetuning using a small number of labeled examples (for both CIFAR-10 and STL-10 datasets). Unfortunately, we have not been able to reach results competitive with the state-of-the-art (Rasmus et al., 2015; Zhao et al., 2015). We speculate that the intra-class variation may be too high for the VAE-GAN model to learn good generalizations of the different object classes.

5. Discussion

The problems with element-wise distance metrics are well known in the literature and many attempts have been made at going beyond pixels – typically using hand-engineered measures. Much in the spirit of deep learning, we argue that the similarity measure is yet another component which can be replaced by a learned model capable of capturing high-level structure relevant to the data distribution. In this work, our main contribution is an unsupervised scheme for learning and applying such a distance measure. With the learned distance measure we are able to train an image encoder-decoder network generating images of unprecedented visual fidelity as shown by our experiments. Moreover, we show that our network is able to disentangle factors of variation in the input data distribution and discover visual attributes in the high-level representation of the latent space. In principle, this lets us employ a large set of

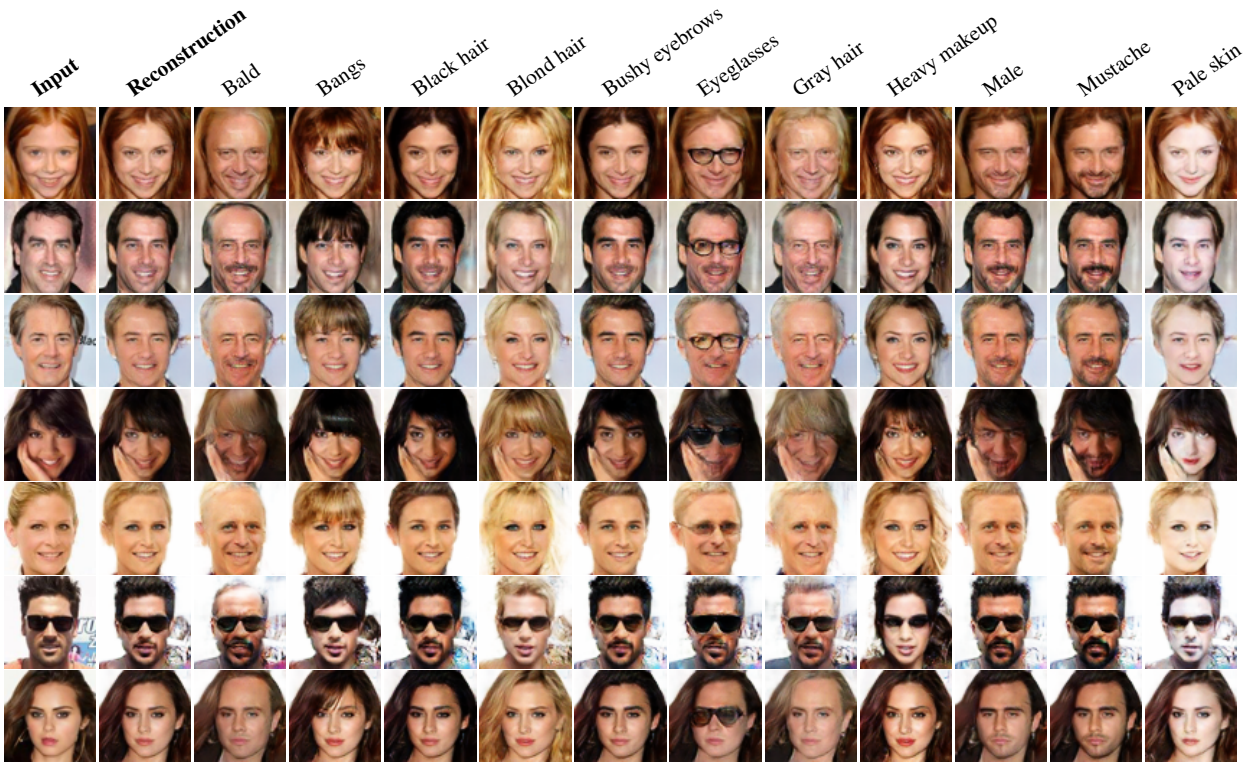


Figure 6. Using the VAE/GAN model to reconstruct dataset samples with visual attribute vectors added to their latent representations.

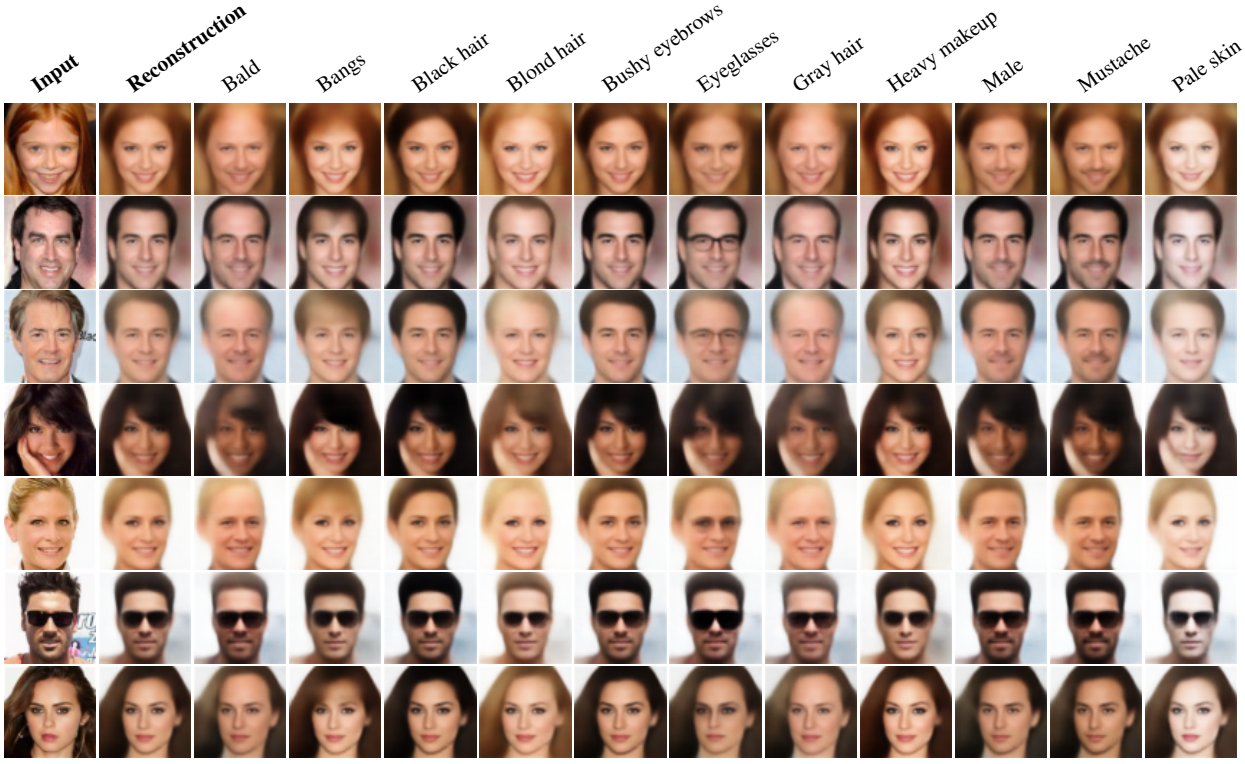


Figure 7. Using the VAE model to reconstruct dataset samples with visual attribute vectors added to their latent representations.

Figure 8. Generating samples conditioned on the LFW attributes listed alongside their corresponding image.

unlabeled images for training and use a small set of labeled images to discover features in latent space.

We regard our method as an extension of the VAE framework. Though, it must be noted that the high quality of our generated images is due to the combined training of Dec as a both a VAE decoder and a GAN generator. This makes our method more of a hybrid between VAE and GAN, and alternatively, one could view our method as an extension of GAN.

It is not obvious that the discriminator network of a GAN provides a useful similarity measure as it is trained for a different task, namely being able to tell generated samples from real samples. However, convolutional features are often surprisingly good for transfer learning, and as we show, good enough in our case to improve on element-wise distances for images. It would be interesting to see if better features in the distance measure would improve the model, e.g. by employing a similarity measure provided by a Siamese network trained on faces, though in practice Siamese networks are not a good fit with our method as they require labeled data. Alternatively one could investigate the effect of using a pretrained feedforward network for measuring similarity.

In summary, we have demonstrated a first attempt at unsupervised learning of encoder-decoder models as well as a similarity measure. Our results show that the visual fidelity of our method is competitive with GAN, which in that regard is considered state-of-the art. We therefore consider learned similarity measures a promising step towards scaling up generative models to more complex data distributions.

Acknowledgements

We would like to thank our reviewers for useful feedback, Søren Hauberg, Casper Kaae Sønderby and Lars Maaløe for insightful discussions, Nvidia for donating GPUs used in experiments, and the authors of DeepPy[3] and CUDArray (Larsen, 2014) for the software frameworks used to implement our model.

References

Bengio, Yoshua, Courville, Aaron, and Vincent, Pierre. Representation learning: A review and new perspectives. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(8):1798–1828, 2013.

Bromley, Jane, Bentz, James W., Bottou, Léon, Guyon, Isabelle, LeCun, Yann, Moore, Cliff, Säckinger, Eduard, and Shah, Roopak. Signature verification using a siamese time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence*, 07 (04):669–688, 1993.

Chopra, S., Hadsell, R., and LeCun, Y. Learning a similarity metric discriminatively, with application to face verification. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pp. 539–546 vol. 1, June 2005.

Denton, Emily L, Chintala, Soumith, Szlam, Arthur, and Fergus, Rob. Deep generative image models using a laplacian pyramid of adversarial networks. In Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 28*, pp. 1486–1494. Curran Associates, Inc., 2015.

Dosovitskiy, Alexey, Springenberg, Jost Tobias, and Brox, Thomas. Learning to generate chairs with convolutional neural networks. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1538–1546, 2015.

[3] http://github.com/andersbll/deeppy

Gatys, Leon A., Ecker, Alexander S., and Bethge, Matthias. A neural algorithm of artistic style. *CoRR*, abs/1508.06576, 2015.

Goodfellow, Ian, Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David, Ozair, Sherjil, Courville, Aaron, and Bengio, Yoshua. Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., and Weinberger, K.Q. (eds.), *Advances in Neural Information Processing Systems 27*, pp. 2672–2680. Curran Associates, Inc., 2014.

Huang, Gary B., Ramesh, Manu, Berg, Tamara, and Learned-Miller, Erik. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

Ioffe, Sergey and Szegedy, Christian. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Blei, David and Bach, Francis (eds.), *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pp. 448–456. JMLR Workshop and Conference Proceedings, 2015.

Kingma, Diederik P. and Welling, Max. Auto-encoding variational Bayes. In *Proceedings of the International Conference on Learning Representations*, 2014.

Kulkarni, Tejas D., Whitney, Will, Kohli, Pushmeet, and Tenenbaum, Joshua B. Deep convolutional inverse graphics network. *CoRR*, abs/1503.03167, 2015.

Kumar, Neeraj, Berg, Alexander C., Belhumeur, Peter N., and Nayar, Shree K. Attribute and simile classifiers for face verification. In *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 365–372, Sept 2009.

Larsen, Anders Boesen Lindbo. CUDArray: CUDA-based NumPy. Technical Report DTU Compute 2014-21, Department of Applied Mathematics and Computer Science, Technical University of Denmark, 2014.

Liu, Ziwei, Luo, Ping, Wang, Xiaogang, and Tang, Xiaoou. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

Mansimov, Elman, Parisotto, Emilio, Ba, Lei Jimmy, and Salakhutdinov, Ruslan. Generating images from captions with attention. *CoRR*, abs/1511.02793, 2015.

Mathieu, Michaël, Couprie, Camille, and LeCun, Yann. Deep multi-scale video prediction beyond mean square error. *CoRR*, abs/1511.05440, 2015.

Mikolov, Tomas, Sutskever, Ilya, Chen, Kai, Corrado, Greg S, and Dean, Jeff. Distributed representations of words and phrases and their compositionality. In Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K.Q. (eds.), *Advances in Neural Information Processing Systems 26*, pp. 3111–3119. Curran Associates, Inc., 2013.

Mirza, Mehdi and Osindero, Simon. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014.

Radford, Alec, Metz, Luke, and Chintala, Soumith. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2015.

Rasmus, Antti, Berglund, Mathias, Honkala, Mikko, Valpola, Harri, and Raiko, Tapani. Semi-supervised learning with ladder networks. In Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 28*, pp. 3532–3540. Curran Associates, Inc., 2015.

Reed, Scott E, Zhang, Yi, Zhang, Yuting, and Lee, Honglak. Deep visual analogy-making. In Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 28*, pp. 1252–1260. Curran Associates, Inc., 2015.

Rezende, Danilo Jimenez, Mohamed, Shakir, and Wierstra, Daan. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of The 31st International Conference on Machine Learning*, pp. 1278–1286, 2014.

Ridgeway, Karl, Snell, Jake, Roads, Brett, Zemel, Richard S., and Mozer, Michael C. Learning to generate images with perceptual similarity metrics. *CoRR*, abs/1511.06409, 2015.

Theis, Lucas, van den Oord, Aäron, and Bethge, Matthias. A note on the evaluation of generative models. *CoRR*, abs/1511.01844, 2015.

Wang, Zhou and Bovik, A.C. Mean squared error: Love it or leave it? a new look at signal fidelity measures. *Signal Processing Magazine, IEEE*, 26(1):98–117, Jan 2009.

Yan, X., Yang, J., Sohn, K., and Lee, H. Attribute2Image: Conditional Image Generation from Visual Attributes. *CoRR*, abs/1512.00570, 2015.

Zhao, Junbo, Mathieu, Michael, Goroshin, Ross, and LeCun, Yann. Stacked what-where auto-encoders. *CoRR*, abs/1506.02351, 2015.

Zhu, Shizhan, Li, Cheng, Loy, Chen Change, and Tang, Xiaoou. Transferring landmark annotations for cross-dataset face alignment. *CoRR*, abs/1409.0602, 2014.