Applied image recognition: guidelines for using deep learning models in practice

Matthias Griebel[1], Alexander Dürr[1], and Nikolai Stein[1]

[1] University of Würzburg, Department of Business and Economics, Würzburg, Germany
{matthias.griebel, alexander.duerr, nikolai.stein}
@uni-wuerzburg.de

Abstract. In recent years, novel deep learning techniques, greater data availability, and a significant growth in computing powers have enabled AI researchers to tackle problems that had remained unassailable for many years. Furthermore, the advent of comprehensive AI frameworks offers the unique opportunity for adopting these new tools in applied fields. Information systems research can play a vital role in bridging the gap to practice. To this end, we conceptualize guidelines for applied image recognition spanning task definition, neural net configuration and training procedures. We showcase our guidelines by means of a biomedical research project for image recognition.

Keywords: Deep learning, Image Recognition, Object Detection, Instance Segmentation, Artificial Intelligence.

1 Introduction

In recent years, novel deep learning techniques, greater data availability, and a significant growth in computing powers have enabled AI researchers to tackle problems that had remained unassailable for many years. This holds especially true for voice or image recognition tasks where deep learning has demonstrated its remarkable capability of revealing structures in unstructured high-dimensional data. Given the wide availability of such data, deep learning applications can be used in many areas of science, business and administration [1]. At this point, a *McKinsey* study estimates the potential of AI applications to create between $3.5 trillion and $5.8 trillion in value annually across nine business functions in 19 industries [2].

A case in point for image recognition applications is the health care sector where deep learning in conjunction may offer a critical complement to the gold standard of randomized controlled trials by supporting massive observational studies that were not feasible before [3]. While there are already many successful biomedical applications enabled by deep learning applications, there is still a great need for innovative solutions. *Grand Challenge* [4] lists 167 data science competitions for biomedical image analysis over the last decade. These challenges comprise a wide range of applications, from ultra sound nerve segmentation, determination of skeletal age, and multiple sclerosis segmentation to different sorts of cancer detection and classification. A recent example

14th International Conference on Wirtschaftsinformatik,
February 24-27, 2019, Siegen, Germany

is the *Kaggle Data Science Bowl 2018* [5] that aims to develop algorithms to speed up research for almost every disease, from lung cancer and heart disease to rare disorders or to the common cold. While the IS community actively engages in various healthcare-oriented fields such as health care management [6], health care services [7] or mental health therapy programs [8], there has been little activity towards supporting researchers with cutting-edge tools such as advanced image recognition. Yet, our community should assume a more active role in this field as it is "uniquely positioned to provide the appropriate mix of rigor along with humanistic and instrumental relevance" [9].

In recent years, comprehensive new AI frameworks such as *Keras* [10] have emerged. They focus on fast experimentation and prototyping through user-friendliness, modularity, and extensibility. The corresponding democratization of AI allows non-AI researches to easily access powerful deep learning applications. This shifts the focus of attention from the technology to the use case. We feel that this development offers a unique opportunity for information systems researchers in facilitating the use of these tools in practical applications. Alongside this development, the availability of unstructured data, notably image data, is increasing dramatically. Images are not only present on social media platforms (Instagram, Facebook), video platforms (YouTube), satellite images (such as Planet.com), but also a growing constituent in scientific research [11]. As the volume of image data has vastly exceeded the capacity of manual analysis, AI is henceforth a key component for automated evaluation [12].

For research purposes AI applications, as with traditional machine learning applications, are typically embedded in data mining pipelines. Existing data mining frameworks such as the guidelines put forward by Müller et al. [13] or CRISP-DM [14] only vaguely describe machine learning applications as part of the *modeling* phase, whereas they focus on tasks such as feature engineering in the data *preparation* phase and the data mining process itself. However, modeling is a critical and extremely complex task for the distinctive nature of deep learning (AI) methods.

To this end, we seek to outline the current state of advanced image recognition and contribute to the literature by providing tangible guidelines for non-AI researchers on how to incorporate state-of-the-art AI algorithms into data mining pipelines. Thereby, we follow up on the call for embracing the value of unstructured data in the design of analytical information system put forward by Müller et al. [13].

2 Building Blocks for Image Recognition Applications

Supervised learning for image recognition requires a data set of labeled images (e.g., magnetic resonance or microscopy images labelled healthy or infected) [1].[1] To facilitate the usage by researchers outside the AI world we want to establish general guidelines for setting up computer vision projects.

[1] In the following, we assume the availability of such data and do not address the also challenging collection task in the remainder of this paper.

To this end, we break down the image recognition into its main building blocks - task definition, the design of the neural network and finally the training approach. The design task features several sub-tasks (choice of architecture, loss function, evaluation metric). To offer concise recommendations for these highly technical sub-tasks we link the design of the neural net to the initial task definition.

2.1 Defining the Task

In order to effectively address the abundance of image recognition applications, it is imperative to understand the underlying problem set. Consequently, any applied computer vision project must ultimately start with a proper definition of the image recognition task at hand. The majority of applications are capture by the following main task categories:

- *Image classification* [15] assigns the whole image to a particular class.
- *Semantic segmentation* (also referred to as pixel-wise classification) identifies every pixel that is part of a specific class, while neglecting distinct instances [12], [17].
- *Object classification* (also referred to as object detection) distinguishes between different objects (instances) of classes in a picture, returning their approximate location using a bounding box [18].
- *Instance segmentation* localizes objects on a pixel basis [19].

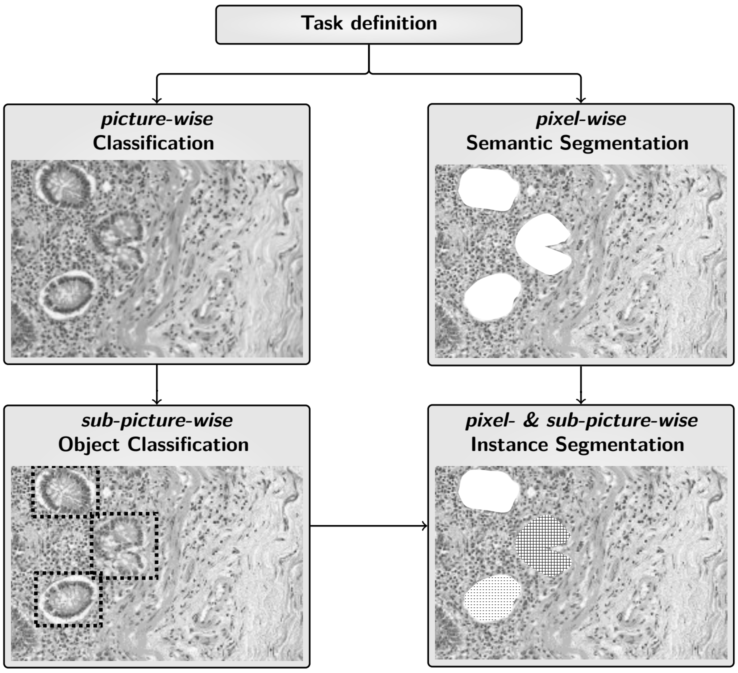

Figure 1. Taxonomy of image recognition tasks using the example of histology images that show cancer cells.

We want to illustrate these categories by means of a histology image containing cancer cells (Fig. 1). The histology images were adapted from "The GlaS Challenge Contest" data set [20]. Depending on the focus of the study the following questions can be addressed using image recognition:

- Classification: Does this image contain any cancer cells? If yes, assign this image to the class “cancer”.
- Semantic Segmentation: What pixels belong to the class “cancer”?
- Object Classification: How many cancer cells are in the image and what is their approximate location?
- Instance segmentation: How many cancer cells are in the image and what is the exact (pixel) position?

2.2 Composing the Neural Network

Having identified the image recognition task, the underlying neural network for image analysis must be set up. Unlike other classification or regression techniques, this is a highly non-trivial task and requires interacting with oftentimes cryptic concepts and an overwhelming number of design options.

While artificial neural networks, i.e., multilayer perceptrons, have been successfully applied to various tasks since the 1980s, convolutional neural networks (CNN) have emerged as the standard for image recognition in the last decade [1]. Consequently, we focus on explaining the essential building blocks of this class of neural networks and establish best practices for each task category.

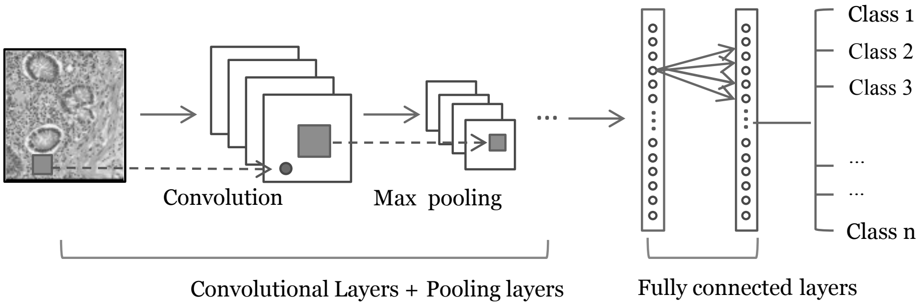

Figure 2. Example of a CNN Architecture

Architectures for Convolutional Neural Networks. The general CNN architecture is composed of three main neural layers, namely convolutional, pooling and fully connected layers as shown in Fig. 2 [1]. Convolutional layers consist of filters (“neurons”) and feature maps to discover conspicuous local pattern-like edges, lines, and other visual elements. Pooling layers are typically considered as a technique to compress or generalize feature representations and reduce the overfitting on the training data by the model [21]. Fully connected layers are used at the end of the network after feature extraction and consolidation by the convolutional and pooling layers. They integrate all feature responses and provide the final classification results [21].

The overwhelming success of AlexNet [21], a large CNN for image classification, in the ILSVRC 2012 challenge [22] has sparked significant interest in the CNN approach. Since then a vast number of architecture tweaks have emerged, each offering incremental improvements of image classification for different data sets. By using their original configuration, these networks perform the task of image classification. Due to their remarkable ability to extract features from images, they are also used as a *backbone* architecture for other image recognition tasks. Fig. 3 provides an overview of the current main architecture choices.
The basic VGG family, introduced by [23], is typically used for its simple and easily understandable architecture (see Fig. 3).

The *Inception* family of networks (introduced by [24]) relies on Inception modules (Fig. 3), where the input is processed by several parallel convolutional layers of different sizes whose outputs are then merged back. This enables the network itself to converge towards an optimal level of abstraction to represent a feature. Finally, the *ResNet* family [25] introduces residual blocks to the CNN (see Fig. 3). Their special features are shortcut connections parallel to the convolutional layers. This facilitates the efficient training of even deeper and more powerful networks [25]. Moreover, these architectures are frequently used as a foundation to tailor *customized* CNN architectures towards a specific use.

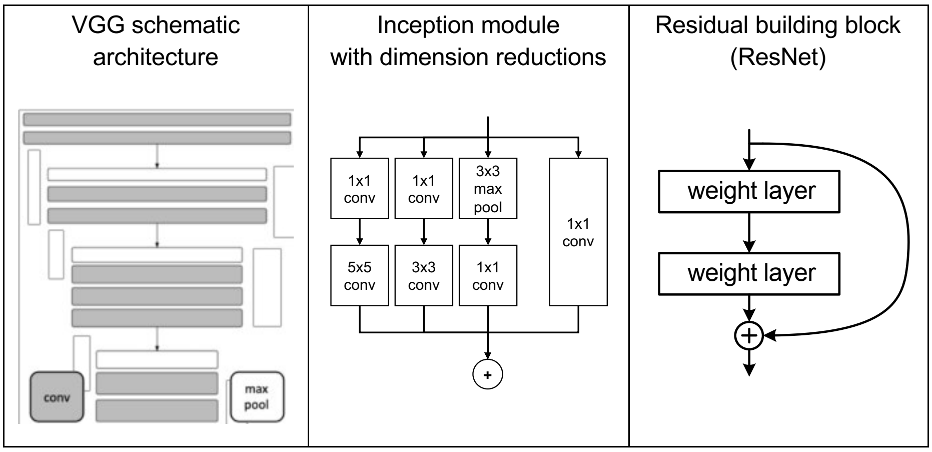
VGG schematic architecture
conv
max pool
Inception module with dimension reductions
1x1 conv
1x1 conv
3x3 max pool
1x1 conv
5x5 conv
3x3 conv
1x1 conv
+
Residual building block (ResNet)
weight layer
weight layer
+

Figure 3. Backbone architecture characteristics

Next, we want to match CNN architectures to the image recognition task categories. These suggestions should provide an informative starting point for determining a suitable architecture:

- Classification: At present, the best performing classification models are, e.g., Inception-Resnet-V2 [26] different version of ResNet (i.e., ResNet51, ResNet101) or VGG.

- Semantic segmentation: Depending on the purpose, variants of the U-Net [27] (using a VGG backbone) perform well on biomedical images such as 2D light microscopy cell segmentation. The 3D version of the U-Net is called V-Net [28]. For more general purpose applications we suggest a VGG based architecture such as a Fully Convolutional Network (FCN) [16].
- Object classification and instance segmentation: As the approach of Mask R-CNN [19] allows both object detection and instance segmentation within the same setting it is the best option for most multi-class segmentation applications. However, the U-Net variants can be extended by additional post-processing steps to enable instance segmentation. In particular, this approach showed very strong performance in the *Kaggle Data Science Bowl 2018* [5]. Depending on the problem at hand, it can be rewarding to implement and evaluate both approaches.

Loss Function and Optimizer. The loss function (objective) and optimizer are the main components to configuring the learning process of a neural network. During the learning phase the weights are adjusted so that the loss decreases. The loss function has to be chosen according to the task, the number of classes or potential class imbalances. Due to its robustness and ability to handle nonlinear effects, the binary cross entropy loss is commonly used as the standard loss for binary classification tasks (picture- or sub-picture-wise). Accordingly, the categorical cross entropy loss works well for all multi-class classification tasks.

In pixel-wise segmentation tasks there is typically an imbalance between pixel classes (i.e., many background pixels and few foreground pixels). There are two common approaches to cope with this problem. On the one hand, [27] propose the use of the weighted cross entropy loss. On the other hand, the dice coefficient loss yields promising results as it handles true negatives as uninteresting defaults [28].

The optimizer determines the update process of the CNN by calculating the gradient. To tackle the high volumes of image recognition tasks, it is of paramount importance that the optimizer computational efficient, has little memory usage and requires little tuning. We suggest to use the optimizer Adam as it outperforms other common choices (e.g., SGD, AdaDelta and RMSProp) with respect to computational overhead [29].

Evaluation Metrics. A suitable evaluation metric is needed to assess a model's performance on the image recognition task. In contrast to the loss function, metrics do neither require to be mathematically differentiable nor used to train the model. Understanding the importance of the evaluation metric is fundamental for every data science project [30], including image recognition tasks.

Accuracy and the area under the curve (AUC) are metrics to evaluate the quality of classification results. For class-imbalanced problems, the Mathew correlation coefficient (MCC) is considered a robust measure [31]. Recall, precision and F-Measure focus on the positive examples to capture information about the rates and kinds of errors made. The intersection-over-union (IoU) metric measures the similarity between the predicted region and the ground-truth region for an object present in the set of images. This is particularly suited for pixel-wise image segmentation tasks. There is clearly no gold standard for evaluation metrics, as they have to account for the specific properties of the given task and underlying data set. We suggest using a combination of different metrics in order to cover different aspects of the evaluation requirements. An exemplary combination of metrics for instance segmentation could be the IoU and recall. While the IoU measures the quality of the segmentation task, the recall accounts for the ability to detect all relevant instances.

2.3 Training Strategy

Having determined the composition of the neural network (by choosing an appropriate CNN architecture, loss function, optimizer and evaluation metric) the final task of training this network on the data needs to be tackled. To this end, we introduce different concepts and best practices for model generalization, hyperparameter optimization, and hardware requirements.

Model Generalization. The advantage of deep and complex CNN architectures is to better extract information from unstructured data. However, a large number of available parameters (weights) renders these networks prone to overfitting which prevents the model from generalizing well to unseen instances [32]. We consider *data-oriented techniques*, *transfer learning*, and *architectural tweaks* to limit the overfitting tendencies of a model.

Data-oriented techniques prevent overfitting by restricting full access of the network to the training data. To this end we apply methods such as data splitting and data augmentation. Data splitting partitions the data set into two subsets: training and validation. The model is then trained on the training data and evaluated on the validation data. Thus, it is possible to stop the training as soon as overfitting occurs. In a k-fold cross validation this procedure is repeated k times [33].

Data augmentation artificially generates additional data without incurring extra labeling costs. In the case of image recognition this is easily achieved by means of transformative methods, such as rotation, shearing, translation, flipping, elastic deformations, and random intensity jitter. This is especially useful for small data sets [21]. Depending on the data set, some transformations should not be performed, i.e., in case of an object recognition task where objects are characterized by their shape, the shape should not be distorted.

Moreover, transfer learning leverages a pre-trained model as feature extractor. To this end, the CNN is initialized with pre-trained parameters of a network that has been trained on another data set such as ImageNet [22] or MS COCO [34]. There are plenty of pre-trained models publicly available, e.g., in the repository of *Keras* [10]. The pre-trained model is fine-tuned subsequently. Thereby, the pre-trained parameters of the initialized network are gradually adjusted to the new images during additional training steps. Depending on the problem, oftentimes the parameters of the majority of the layers are fixed while only a few parameters on top layers[2] are adjusted. Optionally, some custom layers can be introduced and trained in parallel to fine-tune these layers

[2] Here, we define *top layers* as the layers close to the input layer of the CNN.

on the new data set. In general, transfer learning accelerates the learning process and improves the generalization ability of a network [32].

Finally, architectural considerations such as dropout layers [35] can incorporate generalization approaches within the CNN composition. Dropout layers prevent the network from overfitting by randomly deactivating a share of the neurons during the training phase. Thereby, the model is forced to learn the same patterns using different neurons. During the prediction phase the dropout is deactivated and all neurons can be utilized.

Hyperparameter Optimization. There are numerous configuration settings in a CNN that can be tuned to improve the performance. Such parameters include, e.g., the activation function, learning rate, the number of training epochs, the batch size, the, initial weight choices and many more.

- Each weight layer in a CNN is typically ensued by a non-linear *activation function*. The simplest activation function for binary classification decisions is the sigmoid function which is bounded between 0 and 1. The ReLU (Rectified Linear Unit) activation function [36] is commonly used for all layers except for the output layer in practice because of the constant slope for positive values.
- The *learning rate* controls the magnitude weights adjustment after each iteration. If the learning rate is low, the training progresses slowly. In contrast, a high learning rate can prevent from converging to a possible minimum loss.
- One *epoch* is when an entire dataset is passed through the neural network for training.
- The *batch size* defines the number of samples propagated through the network in each step of gradient descent, i.e., learning.

Given the vast number of parameters manual tuning is impossible. Consequently, we suggest to conduct an automated hyperparameter search based on either a random grid search [37] or a Bayesian optimization search [38] to identify the promising parameter choices.

Hardware Requirements. The training of CNNs requires a vast number of convolutional operations resulting in an enormous demand for computing power. Training the model on purpose-built hardware such as GPUs or TPUs are far more efficient than training on a universal CPU. The increased availability and reliability of cloud-computing services provides a strategic dynamic capability to scale up or down the IT infrastructure [39]. Therefore, we suggest using Machine Learning as a Service (MLaaS) solutions. Such services are offered by all leading cloud operators.

3 Applying the Guidelines: A Biomedical Case Study

We illustrate the execution of an image recognition project based on the guidelines put forward above. To this end, we report learnings from a research collaboration with a group of neuroscientists. In a joint project we developed an data mining pipeline to automatically detect fluorescently stained neurons in tissue images of mice brains [40].

3.1 Defining the Task

To define the task, we first need to understand the underlying problem and data set. Fig. 4 shows an excerpt of image dataset obtained using a confocal microscope. The data comprises three different sub-regions of the dorsal hippocampus: dentate gyrus (DG), Cornu ammonis 1 (CA1) and CA3. As there is no ground truth for fluorescent signal segmentation, neurons are determined by their relative brightness (signal strength) to the background. For this purpose, the resulting segmentation maps are generated either by means of a heuristic, manual identification process or by means of a (partially) automated threshold-based analysis. Due to the low signal-to-noise ratio of the data, threshold-based approaches do not work reliably as they fail to detect most of the fluorescent areas (see Fig. 4).

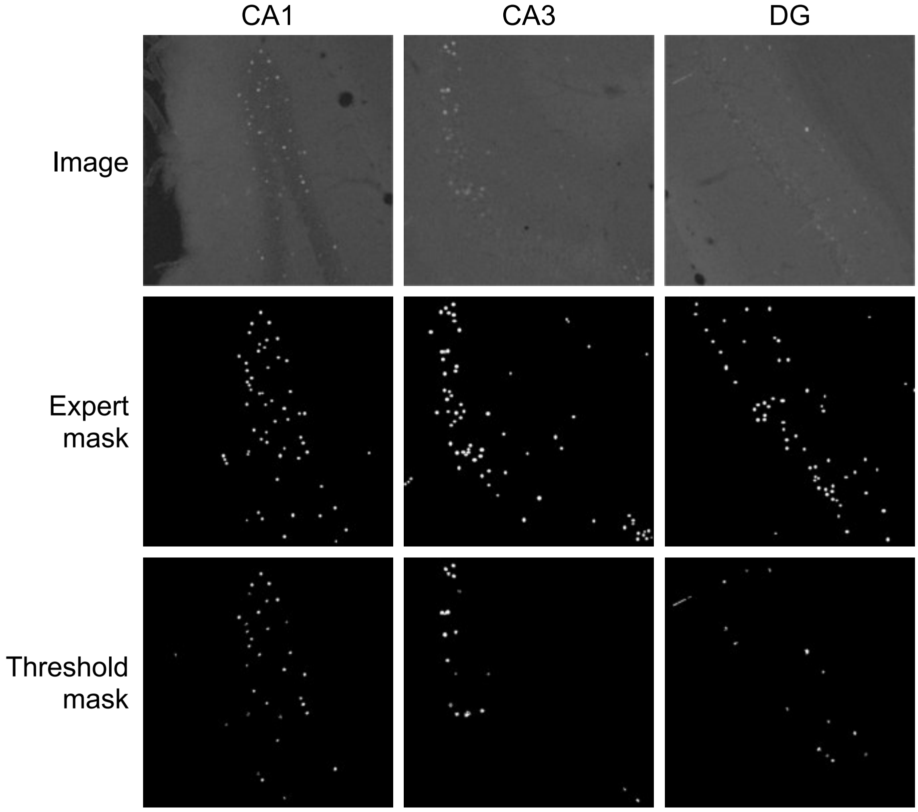

Figure 4. Different sub-regions of the dorsal hippocampus and the corresponding segmentation masks (here, the threshold only considers the 5% brightest pixels per image).

The goal of our image recognition is to automatically detect fluorescent neurons within a microscopy image. For biomedical evaluation, researches require the position, size and signal intensity of fluorescent neurons. Thus, our model needs to identify (i) object instances as well as (ii) the exact area (segmentation mask) rendering instance segmentation suitable for our task.

3.2 Composing the Neural Network

CNN Architecture. According to the task definition we first used the Mask R-CNN approach based on a ResNet backbone architecture for instance segmentation. This already yielded reasonably good results but also required a huge amount of computational resources. We also tried a U-Net based approach similar to the winning solution of the *Kaggle Data Science Bowl 2018* [5]. In this particular case, instance segmentation is achieved by (i) performing pixel-wise binary classification with the U-Net and (ii) post-processing the resulting binary segmentation map. The post-processing pipeline includes a Watershed algorithm [41] and the removal of biological implausible regions (i.e., too small or misshapen). As the U-Net approach yields the better results we continue with it for the remainder of the study.

Loss and Optimizer. As shown in Fig. 4 the total number of fluorescent neurons (positive pixel class) is far less than the background (negative pixel class) resulting in high class imbalances among the whole dataset. Thus, we optimize (Adam algorithm) our model by minimizing a weighted combination of the cross-entropy loss and the dice loss to take advantage of their respective benefits. Here, the dice coefficient loss is particularly valuable as it handles true negatives as uninteresting defaults. We found that the outcome of the whole pipeline depends on a well-suited loss function.

Metrics. To evaluate the quality of our model we compare the expert segmentation masks to the post-processed output masks of our network. This comparison can either be performed pixel-wise or on an aggregated neuron level. For the pixel-wise comparison we need to take the class imbalance into account. Hence, we leverage the IoU as we are mainly interested in identifying instances of the positive class (fluorescent neurons).

Considering the biomedical use case, researches are particularly interested in position and size of each neuron. However, in high resolution images the exact boundaries of the neurons are difficult to define for human experts on a pixel level. As a result, there are often minimal deviations on pixel level even though the same neuron is detected. To address this issue, we introduce another comparison process that (i) matches the corresponding neurons of two segmentation masks and (ii) calculates the accuracy as the proportion of matches divided by the total number of unique neurons found on both segmentation masks.

3.3 Training Strategy

Due to the high cost for both manual labeling and mice experiments, only a limited amount of training samples are available. Thus, we apply data augmentation as a combination of randomly rotating, flipping and shifting the original image-mask pairs. As the shape of the neurons is important in the identification process, we do not use techniques that distort the shape (e.g., shearing). Fig. 5 exemplifies this process with random parameters. Here, the original image-mask pair is horizontally flipped, rotated by 90 degrees clockwise and 20 percent shifted to the top and right. In light of the small dataset, data augmentation prevents from overfitting and generalizes the model, e.g. by learning to detect neurons independent of their position.

To further remedy the issue of limited training data we pre-trained our model on the Kaggle Data Science Bowl 2018 [5] data set, which contains similar microscopy tissue images. To tune the parameters of the network we use a Bayesian optimization search. The model is trained and evaluated on multiple Nvidia Tesla V-100 GPUs.

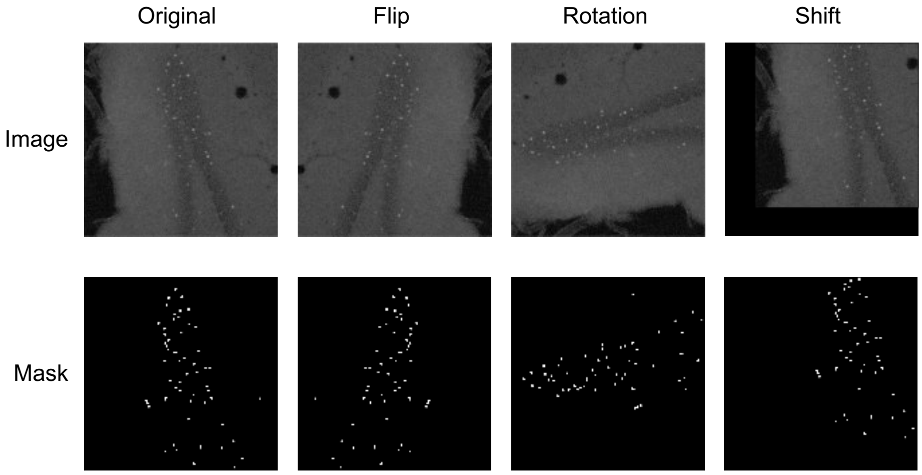

Figure 5. Data augmentation methods used in our project.

The detailed findings of our research project are described by Segebarth et al. [40] and the code is publicly available on *GitHub*[3]. In order to communicate our research, we provide a Jupyter-Notebook with free access to high computing power on *Google Colab*[4]. The execution of the notebook requires no machine learning and almost no programming expertise.

[3] https://github.com/matjesg/DeepFlaSH

[4] https://colab.research.google.com/

4 Conclusion and Outlook

In this study, we outline the current state of advanced image recognition and provide guidelines for non-AI researchers on how to incorporate state-of-the-art AI algorithms into data mining pipelines. We showcase the application of the proposed guidelines on a case study in the field of biomedical image recognition.

In particular, our research aims to make several contributions to the literature. First, we structure the variety of existing image recognition approaches and put forward a taxonomy for image recognition tasks based on the relevant literature. Second, our proposed guidelines are expected to extend data mining frameworks such as CRISP-DM. The *modeling* phase for machine learning applications is only vaguely described here, although it is a complex challenge for deep learning models. Finally, our presented case study demonstrates the potential of AI in biomedical applications. The automation of the conventional (manual) image analysis process reduces workload and enables highly qualified researchers to focus on important activities instead of tedious image labeling work.

To tap into these benefits, some limitations of our guidelines need to be considered. In the rapidly materializing field of AI and deep learning, the proposed model architectures and recommendations only represent a snapshot of the current developments and need to be updated continuously. However, the taxonomy for image recognition tasks will also apply to new technologies. In addition, modeling deep learning applications may require additional specific domain knowledge as depicted in our case study. Another difficulty is to identify the appropriate degree of abstraction for our guidelines when addressing scholars with different levels of prior knowledge.

In future research, we plan to generalize and refine our proposed guidelines by means of an evaluation on use cases from different domains. Possible topics may comprise but are not limited to fashion trend detection, satellite image analysis to predict future resource requirements and the diagnosis and management of diseases.

References

1. Lecun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature. 521, 436–444 (2015)
2. Parker, K., Stone, J.A., Arena, R., Lundberg, D., Aggarwal, S., Goodhart, D., Traboulsi, M.: Notes from the AI frontier: Insights from use cases. McKinsey Global Institute (2018)
3. Agarwal, R., Dhar, V.: Big Data, Data Science, and Analytics: The Opportunity and Challenge for IS Research. Inf. Syst. Res. 25, 443–448 (2014)
4. Grand Challenges, https://grand-challenge.org/All_Challenges/ (Accessed: 10.10.2018)
5. Kaggle Data Science Bowl, https://www.kaggle.com/c/data-science-bowl-2018/ (Accessed: 10.10.2018)
6. Wager, K.A., Lee, F.W., Glaser, J.P.: Health care information systems: a practical approach for health care management. John Wiley & Sons (2017)
7. Yaraghi, N., Du, A.Y., Sharman, R., Gopal, R.D., Ramesh, R.: Health Information Exchange as a Multisided Platform: Adoption, Usage, and Practice Involvement in Service Co-Production. Inf. Syst. Res. 26, 1–18 (2015)

8. Lederman, R., Wadley, G., Gleeson, J., Bendall, S., Álvarez-Jiménez, M.: Moderated online social therapy: Designing and evaluating technology for mental health. ACM Trans. Comput.-Hum. Interact. 21, 1–26 (2014)
9. Abbasi, A., Sarker, S., Chiang, R.H.L.: Big data research in information systems: Toward an inclusive research agenda. J. Assoc. Inf. Syst. 17, 1–32 (2016)
10. Chollet, F., others: Keras, https://keras.io (Accessed: 10.10.2018)
11. Chen, H., Chiang, R.H., Storey, V.C.: Business intelligence and analytics: from big data to big impact. MIS Q. 36, 1165–1188 (2012)
12. Provost, F., Fawcett, T.: Data Science and its Relationship to Big Data and Data-Driven Decision Making. Big Data. 1, 51–59 (2013)
13. Müller, O., Junglas, I., Brocke, J.V., Debortoli, S.: Utilizing big data analytics for information systems research: Challenges, promises and guidelines. Eur. J. Inf. Syst. 25, 289–302 (2016)
14. Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., Wirth, R.: CRISP-DM 1.0 Step-by-step data mining guide. (2000)
15. Haralick, R.M., Shanmugam, K., Dinstein, I.: Textural Features for Image Classification. IEEE Trans. Syst. Man Cybern. SMC-3, 610–621 (1973)
16. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 3431–3440 (2015)
17. Noh, H., Hong, S., Han, B.: Learning deconvolution network for semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1520–1528 (2015)
18. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 580–587 (2014)
19. He, K., Gkioxari, G., Dollar, P., Girshick, R.: Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2980–2988 (2017)
20. Sirinukunwattana, K., Pluim, J.P.W., Chen, H., Qi, X., Heng, P.A., Guo, Y.B., Wang, L.Y., Matuszewski, B.J., Bruni, E., Sanchez, U., Böhm, A., Ronneberger, O., Cheikh, B.B., Racoceanu, D., Kainz, P., Pfeiffer, M., Urschler, M., Snead, D.R.J., Rajpoot, N.M.: Gland segmentation in colon histology images: The glas challenge contest. Med. Image Anal. 35, 489–502 (2017)
21. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp. 1097–1105 (2012)
22. Russakovsky, O., Deng, J., Krause, J., Berg, A., Fei-Fei, L.: The ImageNet Large Scale Visual Recognition Challenge. Int. J. Comput. Vis. IJCV. 115, 211–252 (2012)
23. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. ArXiv (2014)
24. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1–9 (2015)
25. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778. IEEE, Las Vegas, NV, USA (2016)
26. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.: Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. Pattern Recognit. Lett. 42, 11–24 (2016)

27. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention, pp. 234–241 (2015)
28. Milletari, F., Navab, N., Ahmadi, S.A.: V-Net: Fully convolutional neural networks for volumetric medical image segmentation. In: Proceedings of the 4th International Conference on 3D Vision, pp. 565–571 (2016)
29. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. ArXiv 1–15 (2014)
30. Davis, J.V., Kulis, B., Jain, P., Sra, S., Dhillon, I.S.: Information-theoretic metric learning. In: ICML 2007, pp. 209–216 (2007)
31. Powers, D.M.: Evaluation: From Precision, Recall and F-Measure To Roc, Informedness, Markedness & Correlation. J. Mach. Learn. Technol. 2, 37–63 (2011)
32. Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., Lew, M.S.: Deep learning for visual understanding: A review. Neurocomputing. 187, 27–48 (2016)
33. Kohavi, R.: A study of Cross validation and bootstrap for accuracy estimation and model selection. In: Proceedings of the International Joint Conference on Neural Networks, pp. 1137–1145 (1995)
34. Andriluka, M., Pishchulin, L., Gehler, P., Schiele, B.: 2D human pose estimation: New benchmark and state of the art analysis. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 3686–3693 (2014)
35. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A Simple Way to Prevent Neural Networks from Overfitting. J. Mach. Learn. Res. 15, 1929–1958 (2014)
36. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th international conference on machine learning (ICML-10), pp. 807–814 (2010)
37. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. J. Mach. Learn. Res. 13, 281–305 (2012)
38. Golovin, D., Solnik, B., Moitra, S., Kochanski, G., Karro, J., Sculley, D.: Google Vizier. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1487–1495 (2017)
39. Bharadwaj, A., El Sawy, O.A., Pavlou, P.A., Venkatraman, N.: Digital business strategy: toward a next generation of insights. MIS Q. 37, (2013)
40. Segebarth, D., Griebel, M., Duerr, A., R. von Collenberg, C., Martin, C., Fiedler, D., Comeras, L.B., Sah, A., Stein, N., Gupta, R., Sasi, M., Lange, M.D., Tasan, R.O., Singewald, N., Pape, H.-C., Sendtner, M., Flath, C.M., Blum, R.: DeepFLaSh, a deep learning pipeline for segmentation of fluorescent labels in microscopy images. bioRxiv (2018)
41. Beucher, S.: Use of watersheds in contour detection. In: Proceedings of the International Workshop on Image Processing. CCETT (1979)