Machine Vision and Applications (2024) 35:25
https://doi.org/10.1007/s00138-023-01501-3

RESEARCH

A review of adaptable conventional image processing pipelines and deep learning on limited datasets

Friedrich Rieken Münke[1] · Jan Schützke[1] · Felix Berens[2] · Markus Reischl[1]

Received: 3 July 2023 / Revised: 13 September 2023 / Accepted: 12 December 2023

Abstract
The objective of this paper is to study the impact of limited datasets on deep learning techniques and conventional methods in semantic image segmentation and to conduct a comparative analysis in order to determine the optimal scenario for utilizing both approaches. We introduce a synthetic data generator, which enables us to evaluate the impact of the number of training samples as well as the difficulty and diversity of the dataset. We show that deep learning methods excel when large datasets are available and conventional image processing approaches perform well when the datasets are small and diverse. Since transfer learning is a common approach to work around small datasets, we are specifically assessing its impact and found only marginal impact. Furthermore, we implement the conventional image processing pipeline to enable fast and easy application to new problems, making it easy to apply and test conventional methods alongside deep learning with minimal overhead.

Keywords Deep learning · Conventional image processing · Comparison · Synthetic data

1 Introduction

Semantic segmentation is a crucial task in computer vision, widely used in fields like autonomous driving, medical tissue evaluation, and remote sensing image analysis. Deep learning (DL) methods, including convolutional neural networks (CNN) [1–3] and visual transformers (ViT) [4], have become the preferred approach to solve this type of problem due to their outstanding performance.

DL approaches are adaptive and easily applicable to a wide range of tasks, with little effort. Consequently, they have become the go-to solution for this type of problem, while conventional image processing techniques, such as Thresholding, Watershed, Active Contour, (Super) Pixel Classification and Handcrafted Features, are often overlooked. Nevertheless, there are still automated and sophisticated conventional image processing pipelines (CIPPs) [5–8] available.

DL methods, however, have their downsides as well. The training process for DL involves representation learning and requires a significant amount of computational resources. Although researchers are currently exploring interpretability and explainability in DL [9, 10], the available methods, such as class activation maps and gradient analysis, are only applicable for image classification.

In contrast, CIPP approaches excel in areas such as computational complexity, inference speed, and explainability. The decision process of a CIPP can be easily analyzed by executing and visualizing each step separately, as the CIPP consists of many understood steps. CIPPs can be used especially if the problem at hand is easy to solve or an efficient and simple solution is needed [11]. An expert can inject implicit knowledge into a CIPP, reducing the amount of information that needs to be learned. Therefore, CIPPs can be successful when few data points or computational resources are available [12].

These properties of DL and CIPP show the potential of both approaches and their ability to complement each other when applied at the right time and scenario. The general consensus states that DL performs best on large and diverse datasets while CIPPs are applied to small and easy datasets. Studies comparing DL and conventional image processing in the field of image classification [13–18] or semantic segmentation [17, 19–27] show that DL methods consistently

✉ Friedrich Rieken Münke
friedrich.muenke@kit.edu

[1] Institute for Automation and Applied Informatics (IAI), Karlsruhe Institute of Technology (KIT), Hermann-von-Helmholtz-Platz, 76344 Karlsruhe, Baden Württemberg, Germany

[2] Institute for Artificial Intelligence, University of Applied Sciences Ravensburg-Weingarten, Doggenriedstraße, 88250 Weingarten, Baden Württemberg, Germany

Published online: 31 January 2024

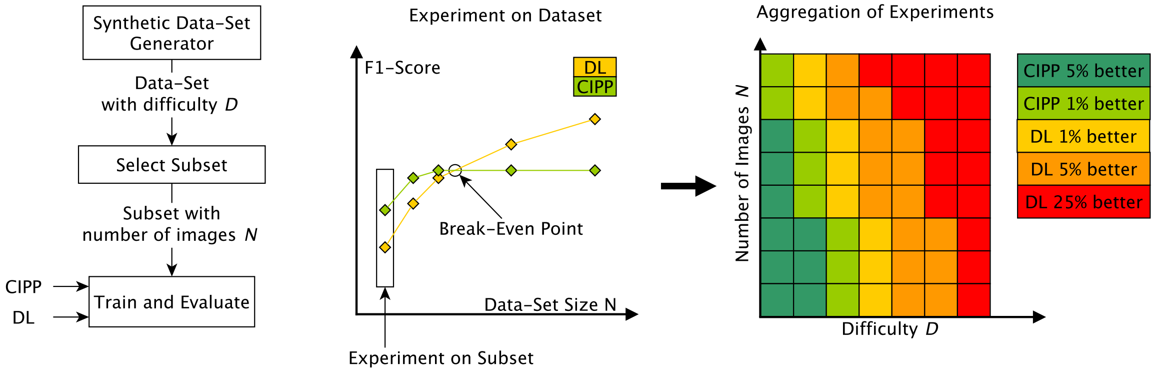


Fig. 1 Concept: A synthetic dataset generator creates a dataset with a given difficulty D. A subset with N images is sampled from this dataset. On this subset, we train a DL model and CIPP and compare the performances of both approaches. The performance is measured using the $F1$-score. This allows us to estimate the "Break-Even-Point" (BEP), up to which the CIPP is still able to outperform DL. In the end, we aggregate the experimental results of each dataset to define the specifications for the usability of CIPPs over DL

exceed or at least match the the performance of conventional techniques. All these comparisons where performed on individual datasets not evaluating the underlying dataset properties. A neutral and systematic evaluation of the applicability of CIPPs and DL in relation to the properties of semantic segmentation datasets and guidelines for application are currently missing.

In this paper, we aim to address this gap by analyzing the strengths and weaknesses of DL models compared to CIPPs in terms of dataset properties. We introduce an automatically optimized conventional image processing pipeline, which is as easy to apply to a problem as a DL method, and provide a novel synthetic dataset generator enabling us to conduct experiments and investigate the behavior of DL and CIPP for various difficulties and different numbers of images. The benchmark dataset supports different tunable noises to increase the difficulty. Additionally, we evaluate different dataset sizes with respect to the influences of stochastic errors and heterogeneous errors in training and testing. Finally, we provide guidelines for choosing the appropriate algorithm (CIPP/DL) based on the characteristics of the dataset and problem.

2 Concept

In this paper, we conduct a study on the performance of DL and CIPP approaches for semantic segmentation to discover effects that let CIPP perform better than DL. The concept of the study is shown in Fig. 1. We focus specifically on the impact of the amount of training data and the difficulty of the task.

Therefore, we introduce a synthetic dataset generator which enables us to quantify and isolate the properties of a semantic segmentation task. Synthetic data is generated with a clearly defined difficulty D. From each dataset, we randomly draw a number of images N and train with this subset a DL model and a CIPP. For each subset with difficulty D and number of images N, we can compare the performance of the DL model and the CIPP and determine the ”Break-Even-Point” (BEP) for each dataset. We expect the CIPP to perform well on easy datasets when there are few training images provided. To confirm this hypothesis, we aggregate the results over all datasets to specify the area of usability where a CIPP outperforms DL in relation to the number of training samples and the difficulty of the dataset.

3 Synthetic dataset

To generate synthetic datasets for the comparison of semantic segmentation approaches, we model an image generation pipeline as depicted in Fig. 2. Each generated dataset contains $\hat{N}$ unique images with $\hat{N}_{\text{train}} = 512$ in the train set and $\hat{N}_{\text{test}} = 512$ in the test set. An image I_i with $i \in [1, \hat{N}]$ is a square with image height and width $s_{\text{img}} = 400px$ and three RGB color channels with a corresponding binary label map L_i of the same size. In the images and their respective label maps, we place an elliptical object on top of heterogeneous structures that constitute our background. The object and background are slightly altered, e.g., texture on the object and different background colors, to ensure a *baseline* difficulty for our segmentation task. Subsequently, different types of noise are added with a defined rate D to increase the difficulty further.

In detail, the images are generated as illustrated in Fig. 2 using the following steps:

Create background: The background is drawn first and covers the entire image I_i with the purpose of giving the segmentation problem a baseline difficulty. In this study, we used a

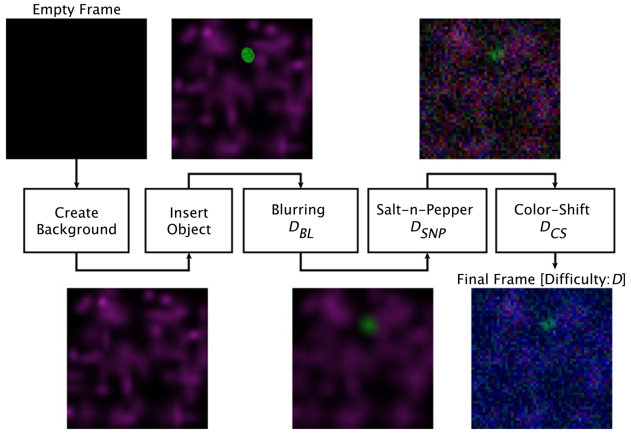


Fig. 2 The data generation process: we start with an empty frame (top left) and create a background (Gaussian Blobs) on the frame, before the object (ellipse) to be identified is inserted on top. As specified by the user, three noise types (Blurring, Salt-n-Pepper, Color-Shift) are applied

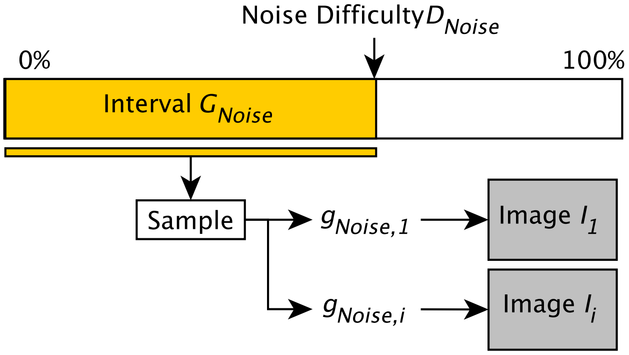


Fig. 3 The noise difficulty D_{Noise} is set by the user for the whole dataset and defines the upper limit of the interval G_{Noise}. The noise parameter $g_{\text{Noise},i}$ is then uniformly sampled from the interval G_{Noise} and applied to the image I_i. This is repeated for all images in the dataset

background consisting of 50–200 randomly generated Gaussian distributions. The color of all the blobs in a single image I_i is randomly chosen from the candidates brown, purple, and teal, which all differ from the color of the object to identify (added in the next step).

Insert object: Then an elliptical object is placed in a random position of the image I_i and the respective label map L_i. Here, we use a green ellipse that has a salt-n-pepper texture and varies slightly in shape, color, and degree of texture.

Apply noise: Noise is added last to an image I_i and applied to both the background and the object. The user defines the noise difficulty $D_{\text{Noise}} \in [0\%, 100\%]$ which determines the diversity and maximum strength of the applied noise for the entire data set. The exact degree of noise applied to an individual image I_i is defined by the noise parameter $g_{\text{Noise},i}$ that is sampled from an interval G_{Noise} as shown in Fig. 3. To be precise, the noise parameter $g_{\text{Noise},i}$ for an image I_i is sampled uniformly from the interval G_{Noise}, which is defined as follows:

$$G_{\text{Noise}} = [g_{\text{Noise}}^{\min}, D_{\text{Noise}} \cdot g_{\text{Noise}}^{\max}]. \tag{1}$$

The lower limit of the interval G_{Noise} is defined by the minimum possible noise parameter $g_{\text{Noise}}^{\min}$ and the upper limit is defined by the maximum possible noise parameter $g_{\text{Noise}}^{\max}$ scaled with the defined difficulty D_{Noise}.

This sampling process ensures that the noise difficulty D_{Noise} defines the diversity of noise and the maximum amount of noise applied. The concept of applying a varying degree of noise to every generated image is inspired by real-world applications where some samples are easier to identify, while others are noisier. $D_{\text{Noise}} = 0\%$ means that no additional noise is added to a dataset, but the properties of the object, as well as the background, still differ between images, which constitutes a baseline difficulty for our synthetic dataset. By increasing the difficulty of noise D_{Noise}, a larger interval G_{Noise} of noise parameters is covered, thus raising the overall level and diversity of noise in a dataset. The specific noise options are the following:

- *Blurring*: A normalized box filter is applied to the image, thus blurring the object to identify. The noise parameter corresponds to the size of the kernel $g_{\text{BL}}^{\min} = 0$ and $g_{\text{BL}}^{\max} = 400px$ as the maximum image side s_{img}.
- *Salt-n-pepper*: For each pixel, a random value is generated, which is added or subtracted from the original pixel value. The noise parameter limits the maximum pixel value that can be generated with $g_{\text{SNP}}^{\min} = 0$ and $g_{\text{SNP}}^{\max} = 255$.
- *Color-shift*: For each channel, a random value is generated which is added or subtracted from the original channel. The noise parameter corresponds to the value added with $g_{\text{CS}}^{\min} = 0$ and $g_{\text{CS}}^{\max} = 255$.

In real-world applications, the three types of noise are influenced by various properties of the recording device, such as the employed optics or the resolution of the detector, and therefore not directly related to each other. Consequently, a general parameter D to describe the degree of noise in a dataset can be calculated as the mean of the individual difficulties:

$$D = \frac{1}{3}(D_{\text{BL}} + D_{\text{SNP}} + D_{\text{CS}}). \tag{2}$$

To simplify matters, we generate our synthetic dataset using equal noise levels for all types, e.g., $D = 5\% = D_{\text{BL}} = D_{\text{SNP}} = D_{\text{CS}}$.

In conclusion, the generation pipeline produces pairs of RGB images and binary label maps with elliptical objects for

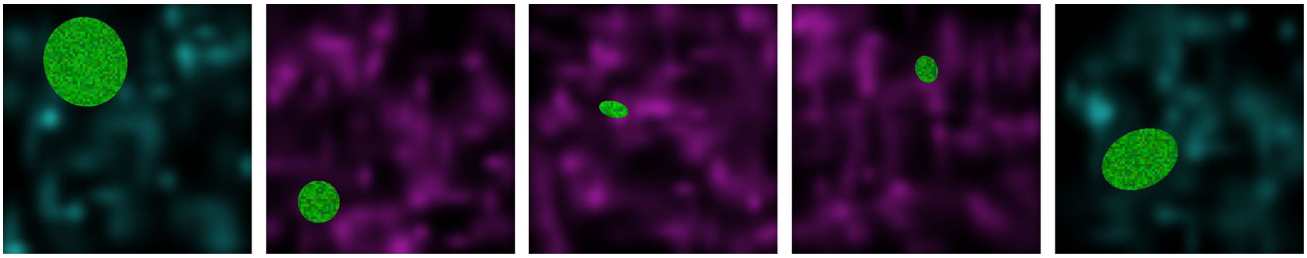


Fig. 4 Five randomly generated images of the baseline dataset with an overall difficulty $D = 0\%$ (baseline). The difficulty D is then increased by applying noise

the purpose of semantic segmentation. The elliptical objects exhibit a textured surface and vary slightly, but differ from the blurred background in their sharp edges and color. Figure 4 presents examples of a dataset with $D = 0\%$, the baseline difficulty. By increasing the level of noise, the edges of the objects are blurred, the texture is added across the entire image, and the colors of the total image are shifted, complicating the segmentation task. The code to create synthetic datasets can be found here: https://github.com/FMuenke/synthetic-dummy-dataset.

4 Semantic segmentation models

Conventional image processing relies on simple operations such as thresholding, edge-detection, or morphological operations, where each operation can be specified with individual parameters. We define a CIPP model as a static sequence of conventional image processing operations. As depicted in Fig. 5, our implementation of a CIPP model provides a framework for an expert to stack these operations without manually setting parameters. Each operation has a pre-defined set of parameters. In this paper we select the best parameters by running all available training images through all possible combinations of parameterized pipelines (grid-search) and selecting the sequence of parameters with the best performance on the training data. Our framework provides besides grid-search other optimization strategies as random search or genetic algorithm.

The CIPP model is specifically designed to use simple techniques to ensure intuitive application to a problem, explainable results, and fast inference even when few data points and computational resources are available. The strengths of CIPP are only useful when they are as easy to apply to a problem as DL. Thus, we have created an easily installable Python package to enable the simple use of CIPPs.

The CIPP is designed to solve the synthetic data set presented in Sect. 3. The segmentation target features two distinct attributes: salt-n-pepper texture and bright green color, which are detectable with edge-detection and thresholding. The CIPP used is visualized in Fig. 6. We aim to increase the processing speed by reducing the image size to 200px x 200px and only applying the CIPP to the green channel. Afterward, the CIPP has the option to apply blurring of different scales to the image to remove noise. The following inversion operation enables the CIPP to select whether the image should be inverted from maximum to minimum. Segmentation is performed by applying Thresholding, Otsu-Thresholding [28] or Edge-Detection. The segmentation mask is post-processed by applying Closing and Eroding to the segmentation. Further details on the image processing operations are found in Tab. 1. The implementation of the CIPP can be found here https://github.com/FMuenke/cipp

In the domain of image segmentation, the U-Net [1] is a prominently used neural network model [29–32] that we employ as our representative for DL. We use the implementation from [33]. The hyperparameters for training the U-Net were determined through a brief random search[1] to fit the synthetic dataset. The final parameters are the following:

- Input size: 256 × 256,
- Backbone: ResNet18 [34],
- Loss: Dice,
- Optimizer: Adam, Learning rate: 10^{-5},
- Early Stopping after 100 Epochs without improving the validation loss,
- Learning Rate Scheduling (factor 0.5 after 50 epochs),
- Augmentations: horizontal/vertical flip, rotation, cropping.

During the random search, it became evident that a batch size of 8 significantly (+30% $F1$-score) improved performance compared to a batch size of 1. When training with a few images, the batch size is set to the maximum number of images until a batch size of 8 is reached.

During training, the only augmentation techniques used are horizontal/vertical flips, rotation, and cropping, since the synthetic dataset already uses salt-n-pepper noise, blurring,

[1] We are not considering grid-search since it is too resource intensive and the general effect of the number of training images correlating with the BEP can still be observed. Especially for simpler synthetic data sets, the DL configuration does not change the result.

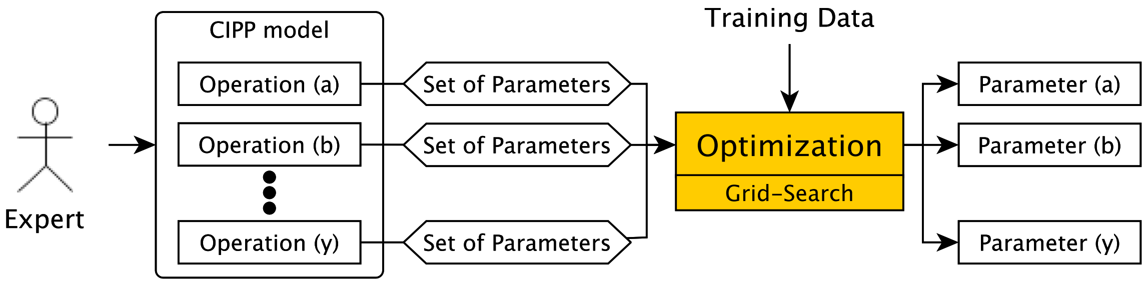


Fig. 5 Optimization process of a CIPP. The order of operations is set by the user and each operation has a predefined set of parameters associated with it. During the optimization process, the optimal parameters are determined based on the provided training data by grid-search. All provided images are processed with all possible parameter combinations and finally the set of parameters with the highest $F1$-score on the training data is picked

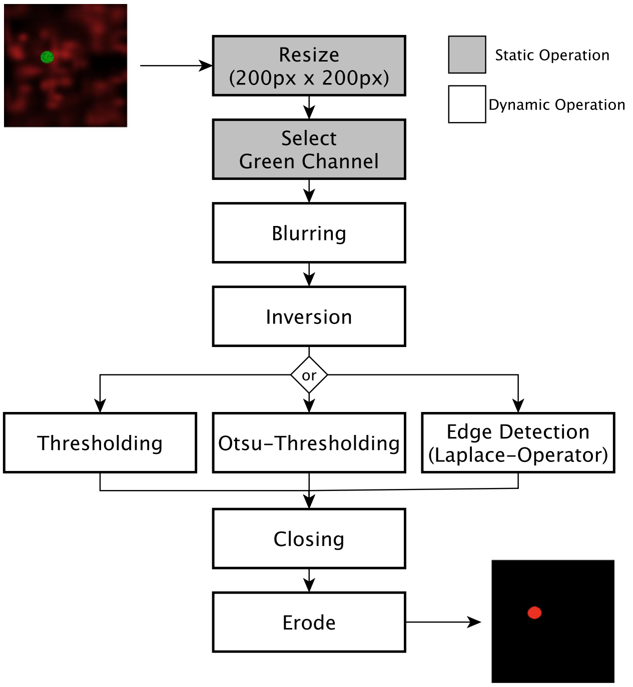


Fig. 6 Structure of the CIPP: An image I_i is resized to 200px × 200px and the green channel is selected for further processing. Afterward Blurring and Inversion are used as pre-processing steps. The target is segmented by applying Thresholding, Otsu-Thresholding or Edge-Detection to the image. Finally Closing and Eroding are used to post-process the output. (Static operations are defined by the user and do not contain variable parameters. Dynamic operations have variable parameters which are optimized during the training process)

and channel shift to increase difficulty. These augmentation techniques are not useful for a CIPP model and thus are not used during their training.

For each set of N training images, we select the same number of additional validation images. These images are used to evaluate the performance of the DL model during training. As the final DL model, we choose the best performing model on the validation dataset. Transfer-learning utilizing pretrained weights is a common strategy to improve data efficiency.

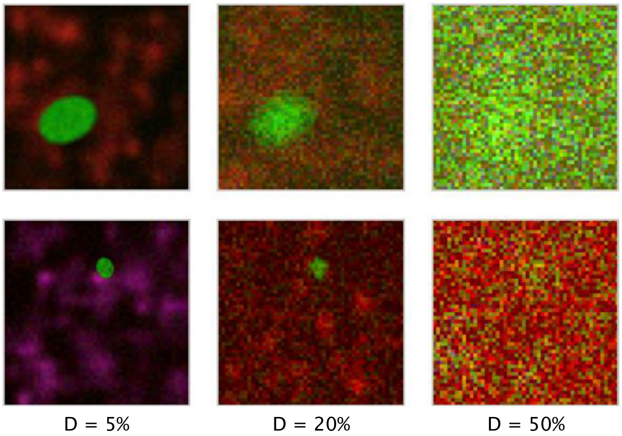


Fig. 7 Example Images from our dataset for the difficulties $D = 5\%$, 20% and 50%, as introduced in Sect. 3

Thus, we are considering the baseline U-Net as described (U-Net-R18) and the same U-Net with an encoder pretrained on Imagenet [35] (U-Net-R18-I) in our experiments.

5 Results

5.1 Overview

We train three types of models in our experiments as introduced in Sect. 4. Each model is trained on a synthetic dataset, which covers all types of noise (blurring, salt-n-pepper, and color-shift) simultaneously. This dataset increases its difficulty D by raising the separate noise difficulties D_{BL}, D_{CS}, and D_{SNP} equally, as shown in Fig. 7. The difficulties 0% to 50% in steps of 5% and additionally 100% are evaluated.

We train with different numbers of training images $N = \{4, 8, 16, 32, 64, 128\}$ for each difficulty D. N corresponds only to the number of images used to train. Since the U-Net models require validation data to determine the optimal time to stop training, we always supply the U-Nets with an equal number of validation images in parallel to the number of training images N.[2] We test CIPP and DL on each subset and compare their $F1$-score on the full test set. Since the images are selected randomly, we repeat each training 20 times to reduce the random deviation introduced by the initialization of the U-Net and the choice of training images. During the sampling of images, we ensure that the approaches are both trained on the same images by setting the random seed (e.g., the first iteration of CIPP is trained on identical images as the first iteration of the U-Net models). The difficulty, as described in Sect. 3, represents the strength of applied noise, as well as the diversity of the data set.

5.2 Baseline U-Net

The average results of the U-Net-R18 on our dataset are displayed in Fig. 8. We can see that the U-Net-R18 performs well on the difficulties $D \leq 5\%$ regardless of the amount of training images with performance over 93% $F1$-score. As expected the performance starts to decrease with an increase in difficulty and the performance increases with an increase in the amount of training images. The U-Net-R18 is still able to reach a performance of above 73% even for higher difficulties $D \leq 50\%$ provided enough training images. Only for the difficulty $D = 100\%$ the U-Net-R18 is not able to learn adequate filters for the segmentation task and cannot exceed 6% $F1$-score.

5.3 Pretrained U-Net

Figure 8 presents the average results of U-Net-R18-I, which was pretrained on ImageNet. Our findings indicate that the performance of U-Net-R18-I is closely aligned with that of U-Net-R18. Specifically, as the level of difficulty increases, the performance of both models decreases, while increasing the number of training images improves their performance. However, we observed that U-Net-R18-I performs better than U-Net-R18 by an average of 0.56% across all combinations of difficulties and training images. Notably, the performance gap between the two models is generally below 9%, and the majority of differences larger than 3% occur when the number of training images is less than 16. Our experiments also demonstrate that the effect of pretrained weights on model performance in this scenario is negligible. We assume that this could be attributed to the fact that the pretrained weights available are not specifically tailored to the domain they are being applied to.

5.4 CIPP

We assess the performance of CIPP and present the results in Fig. 8. Unlike the U-Nets, the CIPP is less sensitive to the number of training images. We observed that increasing the number of training images from $N = 16$ to $N = 128$ leads to a maximum improvement of 7% for all difficulty levels. Notably, the performance gain is more pronounced when the number of training images is increased from $N = 4$ to $N = 16$, with an average improvement of around 11%. Although the CIPP's performance decreases as difficulty increases, it still maintains a relatively high performance level of 26% even at the highest difficulty level of 100%.

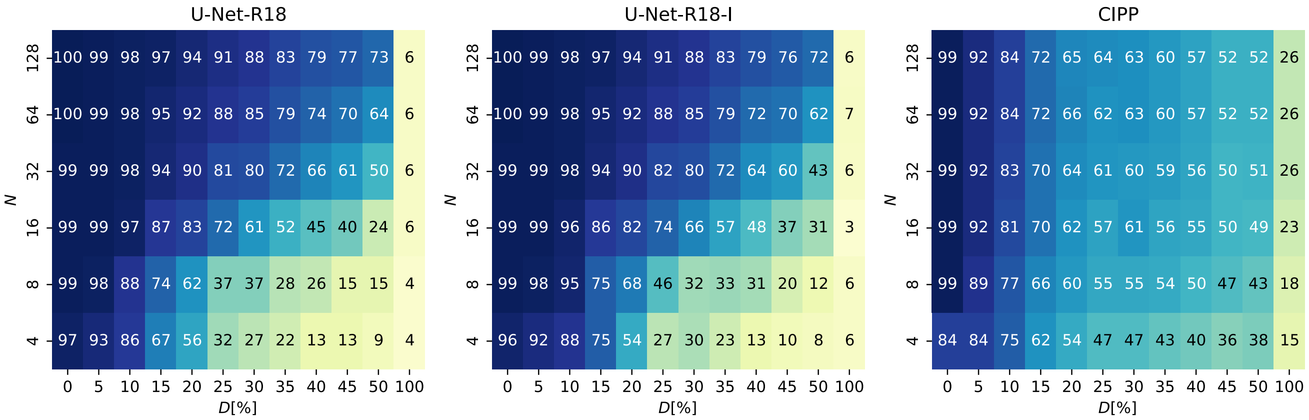
5.5 Comparison

We conducted a side-by-side comparison of the three models, evaluating their performance at three different difficulty levels, as shown in Fig. 9. Rather than presenting only the average performance, we provide the results of all 20 experimental runs, which enables us to observe the variation in performance for different numbers of training images N. The results indicate that the variation decreases as the number of training images N increases for all models. Moreover, we observed that the deviation between separate runs increases clearly as the difficulty level of the dataset increases for both U-Nets.

In Fig. 10, we compare the average performances of the three models by subtracting the performance matrix of the CIPP from those of the U-Nets. This yields a matrix that highlights the differences between the U-Nets and the CIPP. A positive value indicates superior performance by the U-Nets, while a negative value indicates superior performance by the CIPP. We observed that both matrices are similar, as the performances of the U-Nets are comparable. The CIPP outperforms the U-Nets at $N = 4$ and $D = 25\%$. With increasing difficulty, all models exhibit a drop in performance, but the CIPP maintains a more stable performance. Further, the CIPP is able to outperform the U-Nets at $D = 50\%$ even for $N = 32$ training images. At the highest difficulty level of 100%, the CIPP performs better across all numbers of training images.

Overall, the CIPP exhibits a more stable and consistent performance than the U-Nets, and is less affected by changes in dataset difficulty and the number of training images. Additionally, the spread of the results from the 20 distinct test runs is more stable for the CIPP than for the U-Nets at higher difficulty levels, as seen in Fig. 9. It is worth noting that the U-Nets exhibit outstanding performance for a small number of training images, particularly for difficulties $D \leq 15\%$. Our suspicion is that the U-Nets are capable of fitting the provided data due to the limited diversity of the dataset and

[2] We have to acknowledge that the CIPP would perform potentially even better since it does not require validation data. If the validation data used during the DL training process had been used to train the CIPP, its performance would have increased even further.



U-Net-R18

N \ D[%]	0	5	10	15	20	25	30	35	40	45	50	100
128	100	99	98	97	94	91	88	83	79	77	73	6
64	100	99	98	95	92	88	85	79	74	70	64	6
32	99	99	98	94	90	81	80	72	66	61	50	6
16	99	99	97	87	83	72	61	52	45	40	24	6
8	99	98	88	74	62	37	37	28	26	15	15	4
4	97	93	86	67	56	32	27	22	13	13	9	4

U-Net-R18-I

N \ D[%]	0	5	10	15	20	25	30	35	40	45	50	100
128	100	99	98	97	94	91	88	83	79	76	72	6
64	100	99	98	95	92	88	85	79	72	70	62	7
32	99	99	98	94	90	82	80	72	64	60	43	6
16	99	99	96	86	82	74	66	57	48	37	31	3
8	99	98	95	75	68	46	32	33	31	20	12	6
4	96	92	88	75	54	27	30	23	13	10	8	6

CIPP

N \ D[%]	0	5	10	15	20	25	30	35	40	45	50	100
128	99	92	84	72	65	64	63	60	57	52	52	26
64	99	92	84	72	66	62	63	60	57	52	52	26
32	99	92	83	70	64	61	60	59	56	50	51	26
16	99	92	81	70	62	57	61	56	55	50	49	23
8	99	89	77	66	60	55	55	54	50	47	43	18
4	84	84	75	62	54	47	47	43	40	36	38	15

Fig. 8 The performance matrix of the U-Nets and the CIPP for all difficulties D and the number of training images N. The performance is measured using the average $F1$-score

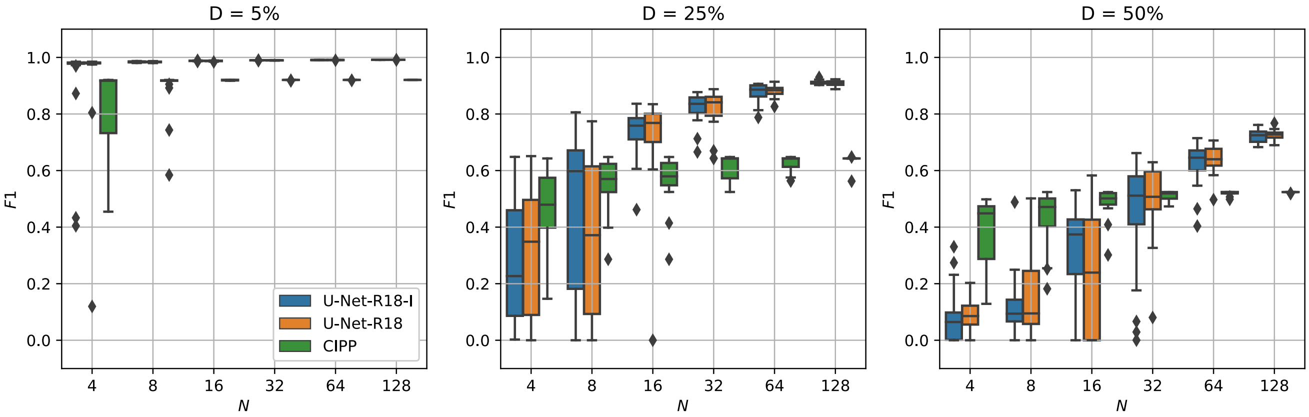
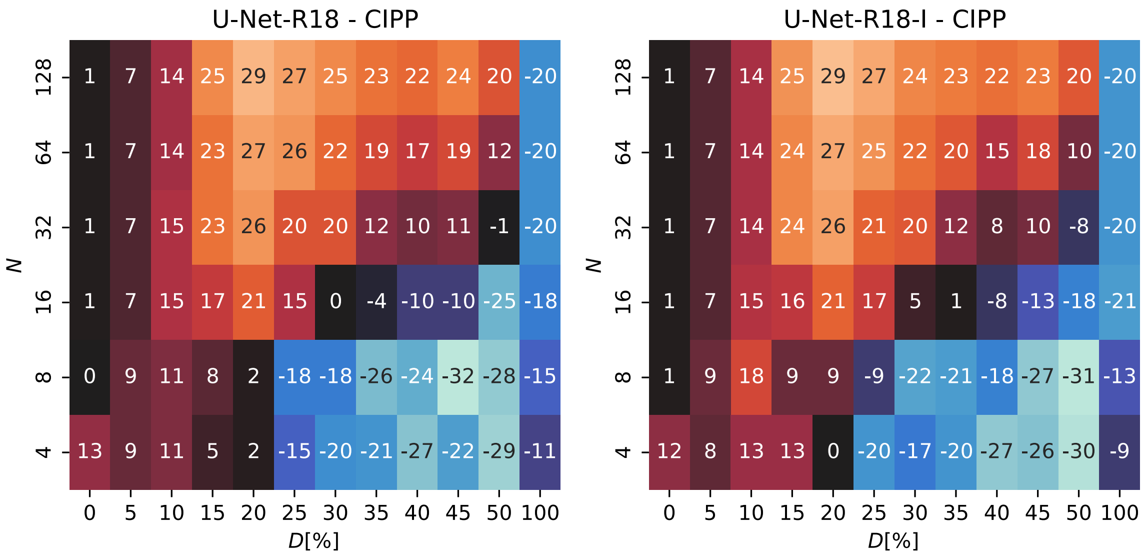


Fig. 9 Exemplary results for different difficulties over the number of training images N



U-Net-R18 - CIPP

N \ D[%]	0	5	10	15	20	25	30	35	40	45	50	100
128	1	7	14	25	29	27	25	23	22	24	20	-20
64	1	7	14	23	27	26	22	19	17	19	12	-20
32	1	7	15	23	26	20	20	12	10	11	-1	-20
16	1	7	15	17	21	15	0	-4	-10	-10	-25	-18
8	0	9	11	8	2	-18	-18	-26	-24	-32	-28	-15
4	13	9	11	5	2	-15	-20	-21	-27	-22	-29	-11

U-Net-R18-I - CIPP

N \ D[%]	0	5	10	15	20	25	30	35	40	45	50	100
128	1	7	14	25	29	27	24	23	22	23	20	-20
64	1	7	14	24	27	25	22	20	15	18	10	-20
32	1	7	14	24	26	21	20	12	8	10	-8	-20
16	1	7	15	16	21	17	5	1	-8	-13	-18	-21
8	1	9	18	9	9	-9	-22	-21	-18	-27	-31	-13
4	12	8	13	13	0	-20	-17	-20	-27	-26	-30	-9

Fig. 10 Comparison of the U-Net-R18 and the U-Net-R18-I with the CIPP. Where the differences of the performance matrices between CIPP and the U-Nets are visualized. (Positive Values: U-Nets outperforms, Negative Values: CIPP outperforms)

the fact that the validation images closely resemble the general dataset.

When comparing the inference speed of DL and CIPP on a Mac Book with a 2,3 GHz Quad-Core Intel Core i7 processor, the DL approach is able to process 2.36 images per second compared to 62.1 images per second for the CIPP. This makes CIPP especially relevant for devices with low computational capacities, such as microcontrollers.

5.6 Transferability

The results presented in this paper are derived from synthetic data, raising the question of whether these findings can be extrapolated to real-world datasets. In the domain of biomedical image processing, datasets often show high diversity within and between datasets, and are typically limited in size. Our research suggests that datasets sharing similar inherent features yields comparable results to those obtained from our synthetic dataset. We have evaluated the effectiveness of the CIPP on four real-world dataset in Appendix 1. The LIVECell dataset [36] and the DOORS dataset [37] exhibit significant diversity between their training and testing subsets. This diversity leads to the anticipated superiority of the CIPP over the U-Net-R18-I. In the case of the Derma ISIC dataset [38], both models demonstrate comparable performance, owing to the dataset's relatively limited diversity. Conversely, on the CryoNuSeg dataset [39], the CIPP exhibits a comparatively inferior performance due to the limited diversity among segmentation targets.

6 Conclusions

So far, there is no comprehensive study, comparing conventional image processing to modern deep learning algorithms considering dataset specific properties. Thus, we introduced a synthetic dataset with tunable degrees of difficulty and conducted a exhaustive study on DL approaches and our own easy-to-apply implementation of a CIPP. The dataset serves as a versatile benchmark dataset and will be used for future studies as well. Furthermore, it can be used to educate students and researchers in understanding and comparing the performance of semantic segmentation approaches.

Our findings show that DL performs best on tasks with low difficulty/diversity and large amounts of training data. Deep learning is able to consider context and shapes which makes it effective in recognizing the target even with few training images. However, if only a few training images are provided, the diversity of the dataset is not properly represented, leading to decreased DL performance. In such cases, the CIPP is able to generalize better due to human expert input and limited parameter space to optimize.

Overall, we recommend the use of our implementation of a CIPP in all scenarios due to its ease of application and low resource requirements. Our proposed CIPP implementation can work with the same data format as most DL frameworks, reducing the additional effort required for adoption. Additionally, CIPPs allow for easy understanding and adaptation of the processing pipeline to new data, making them useful in laboratory settings with few experimental modalities that require quick adaptation with minimal computational costs. Finally, the CIPP can also be used to post-process outputs of DL approaches by removing artifacts or supporting the labeling process by providing quickly label-maps, which can be corrected by a human operator.

Our study highlights the importance of understanding the strengths and weaknesses of both deep learning methods and conventional image processing pipelines. Researchers and practitioners can use this knowledge to choose the most appropriate approach for their specific task and dataset, based on the available resources and desired performance metrics.

In our future research, we plan to expand the capabilities of our CIPP implementation and assess its ability to assist human annotators in fast and efficient pre-labeling. Specifically, we aim to enhance our CIPP with additional image processing techniques and optimize its performance on various types of image datasets. Additionally, we will investigate the potential of our CIPP to be used in combination with DL methods to further improve semantic image segmentation accuracy. We will also explore the possibility of integrating our CIPP into existing annotation tools to facilitate the labeling process for human annotators.

Funding Open Access funding enabled and organized by Projekt DEAL.

Data availibility All synthetic data can be generated by executing the provided code. The data including the presented results can as well be acquired by contacting the corresponding author.

Declarations

Conflict of interest The authors declare that they have no financial, non-financial or other competing interests.

Code availability: The code is made fully available.

Appendix A: Conventional image processing operations

In this section, we give further insights into the image processing operations used by the CIPP. In Tab. 1 all operations utilized by the CIPP in this paper are listed and explained. The single operations can be picked by a user and assembled to a CIPP which is then able to optimize itself. The optimization considers the parameters which are listed in the Parameter column.

Appendix B: Supplementary test series

In this section, we delve into additional experiments that we conducted. First, we evaluate further DL models and compare them to the U-Net. Second, we discuss the performances of the U-Net-R18-I and the CIPP on four distinct benchmark datasets. The subsequent experiments focused on a specific type of noise using synthetic datasets. These synthetic datasets were easier compared to the main test series, and therefore, we evaluated the full range of difficulties (D_{Noise}) from 0 to 100% in 10% steps. For all following experiments, we used the same implementation of the CIPP as introduced in Sect. 4.

B.1: Deep learning models

Besides the U-Net there are multiple other DL models capable of semantic segmentation. We repeat the test series from Sect. 5 with the LinkNet [40], FPN [41] and PSPNet [2] as implemented in [33]. The DL models were trained with the same parameters as the U-Net in Sect. 5 with a ResNet18 backbone pretrained on ImageNet. Each training was performed ten times to reduce random deviation.

The results are displayed in Fig. 11. The DL models show a similar performance characteristics as the U-Net. The DL models excel when many training images are available or when the complexity and diversity is low ($D < 20\%$). The performance generally drops ($F_1 \leq 40\%$) for $D \geq 35\%$ and $N \leq 8$. The LinkNet-R18-I, FPN-R18-I and PSPNet-R18-I are all outperformed by the CIPP at $D > 25\%$ for $N = 4$, as the U-Net-R18-I. Thus, we conclude that the U-Net-R18-I is a good representative for DL models.

B.2: Benchmark datasets

We have assessed the U-Net-R18-I and the CIPP on a selection of four benchmark datasets. We conducted 10 training runs for $N = 4, 8, 16$ training images. Example images for each datasets are shown in Fig. and the results are summarized in Fig.
LIVECell [36] is a dataset of phase-contrast images for cell segmentation. We select a subset of this dataset which has a unique feature where training crops have a resolution of 256px × 256px, and test images have different resolutions of 704px × 520px. The CIPP displays no deviation and can reliably select its optimal parameters, even with just $N = 4$ training images. Conversely, the U-Net's deviation is generally higher and decreases with an increasing number of training images. Although the dataset lacks diversity in appearance, the resolution shift greatly affects the U-Net's performance, whereas the CIPP, as demonstrated in previous tests, is more robust to data diversity and is not affected. While we expect the U-Net to perform well on images of the same resolution this example underlines the robustness of the CIPP model.

DOORS [37] is a synthetic dataset to detect boulders on the surface of small bodies. We train on images showing one boulder and test on images with multiple boulders to evaluate the impact of diversity between train and test set. We observe that the CIPP as expected outperforms the U-Net-R18-I as it is more resilient against diversity in the dataset when few training images are available.

Derma ISIC [38] focuses on skin lesion analysis and melanoma detection. The performance of the U-Net-R18-I and the CIPP are similar and overlap. The median performance of the CIPP is slightly larger.

Table 1 Details for the image processing operations used by the CIPP

Operation	Parameter set	Description
Blurring	Kernel size: {3, 5, 9, 17}, inactive	This pre-processing step to removes potential noise by blurring the input. The parameter corresponds to the kernel size
Inversion	Active/inactive	Depending on the following operations it may be necessary to invert the image from minimum to maximum
Threshold	Threshold: {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}	The input is min-max-scaled to values between 0 and 1 before the threshold is used to binarize the image
Otsu-Threshold	Active/inactive	The Otsu-Threshold [28] is applied to the input
Edge-detection	Kernel size: {3, 5, 9, 17}, inactive	The Laplace-operator is applied to the image. The output is then min-max-scaled between 0 and 1
Closing	Size: {3, 5, 9, 17}, inactive	A morphological closing operation is applied to the input. The structuring element is an ellipse
Erosion	Size: {3, 5, 9, 17}, inactive	A morphological erosion operation is applied to the input. The structuring element is an ellipse

Where the column parameter contains all values the CIPP can choose from during training

CryoNuSeg [39] is a dataset of Hematoxylin and Eosin (H&E)-stained images for nuclei segmentation from 10 different organs. The performances of both models are similar but in this case the U-Net-R18-I slightly outperforms the CIPP.

B.3: Blurring

In this test series, example images are visualized in Fig. 12, we have applied only blurring to gradually increase the difficulty D_{BL}. The results for the difficulties $D_{\mathrm{BL}} = 10\%$, 50% and 90% are shown in Fig. 13. The U-Net used in this synthetic experiment did not utilize any pretraining or specific backbone. As expected, the deviation of independent training runs decreases with the number of training images N used for training. The CIPP performs on average (65.64%) significantly lower than the U-Net (80.20%), which indicates that the U-Net generally is better equipped to deal with blurring noise. With an increase in difficulty D_{BL} the performance of both methods drops significantly. The U-Net is more sensitive to higher difficulties than the CIPP. This allows the CIPP to outperform the U-Net for very few images and high difficulties as visible in Fig. 14. In direct comparison, it is apparent that the U-Net is able to handle blurring noise better compared to the CIPP in nearly every test case. We suppose that the ability of the U-Net to assess the shape and context of the image provides in this specific case a crucial advantage.

B.4: Color-shift

In this test series, we have only applied the noise color-shift to the images, as visualized in Fig. 15. Three exemplary difficulties are visualized in Fig. 16 to showcase the impact of random initialization and image choice on both approaches. The U-Net used in this synthetic experiment did not utilize any pretraining or specific backbone. With increasing difficulty the deviation of performance increases for both approaches. It is visible that the CIPP reaches peak perfor-

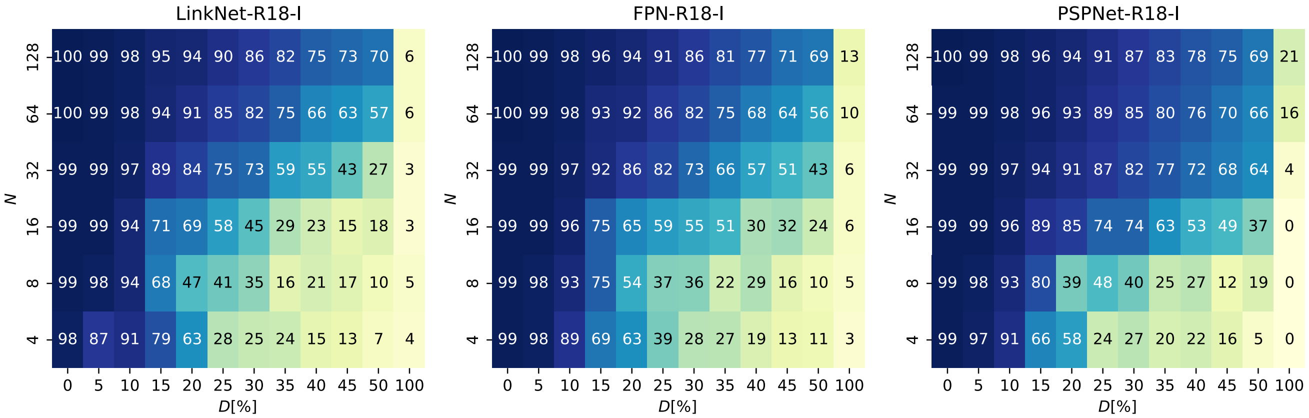


Fig. 11 The performance matrix of the LinkNet, FPN and PSPNet for all difficulties D and the number of training images N. The performance is measured using the average $F1$-score on the synthetic dataset as in Sect. 5

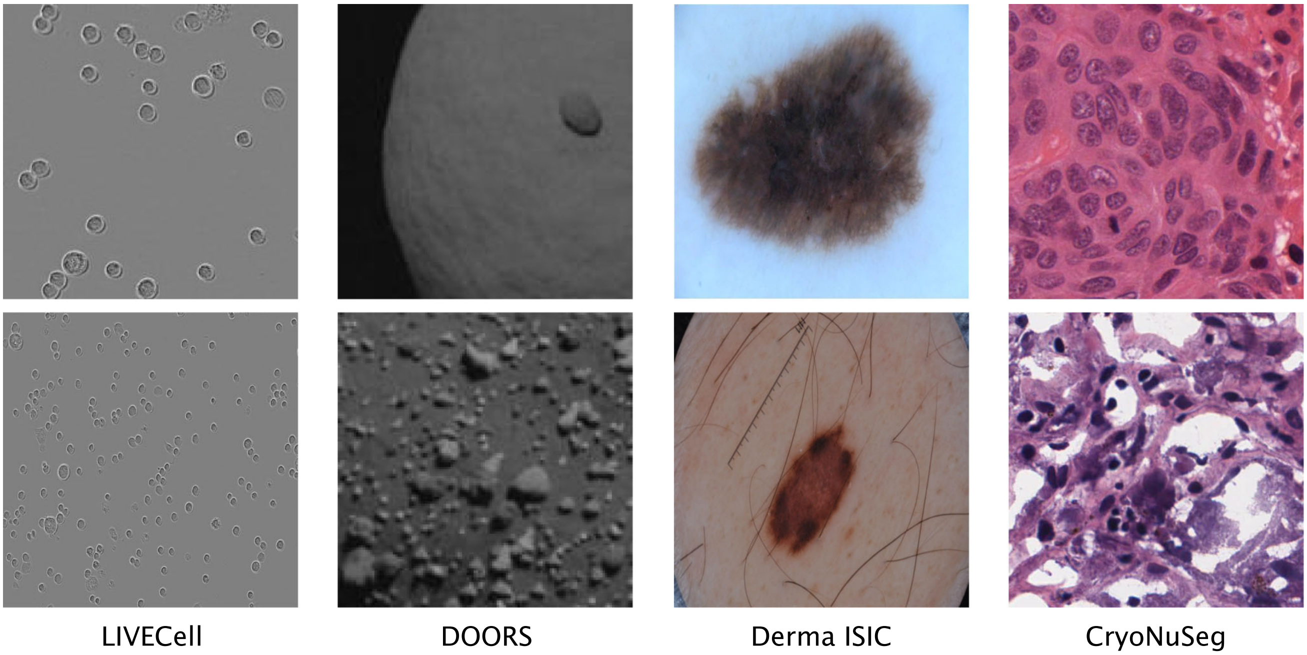


Fig. 12 Example images from the selected benchmark datasets. The first row shows an example from the training set and the second row shows an example from the test set

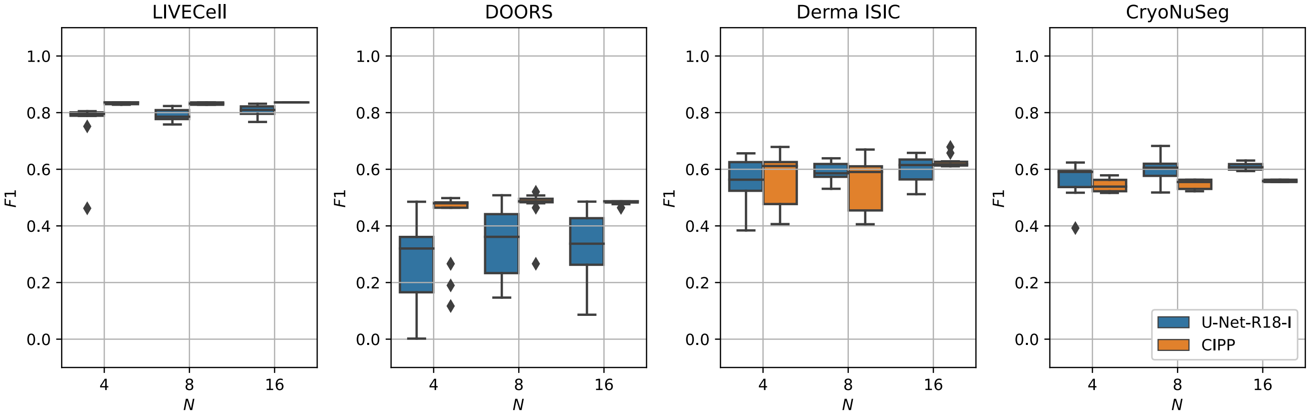
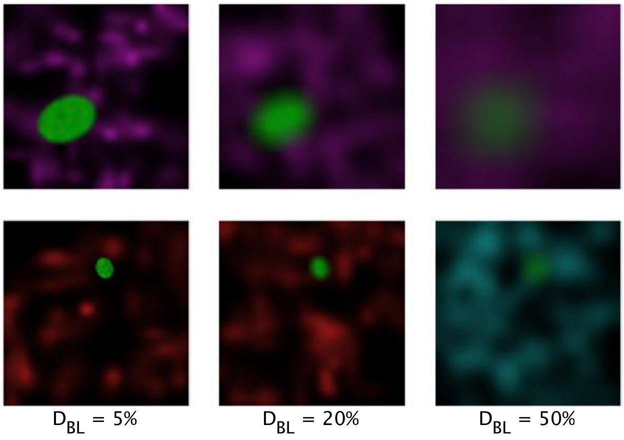


Fig. 13 The performances of the U-Net-R18-I and the CIPP compared side by side over different amounts of training images N for each benchmark dataset

Fig. 14 Exemplary images from the test series focusing on blurring



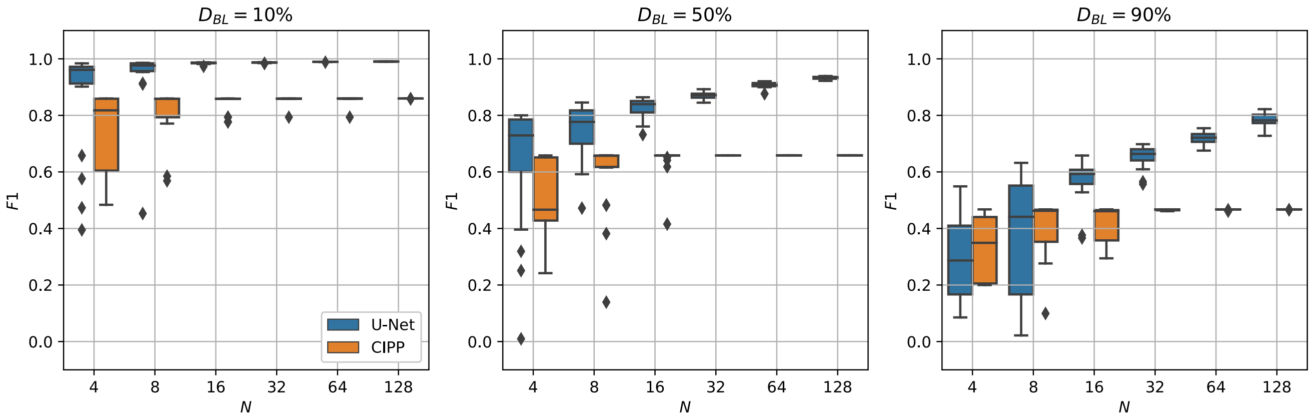
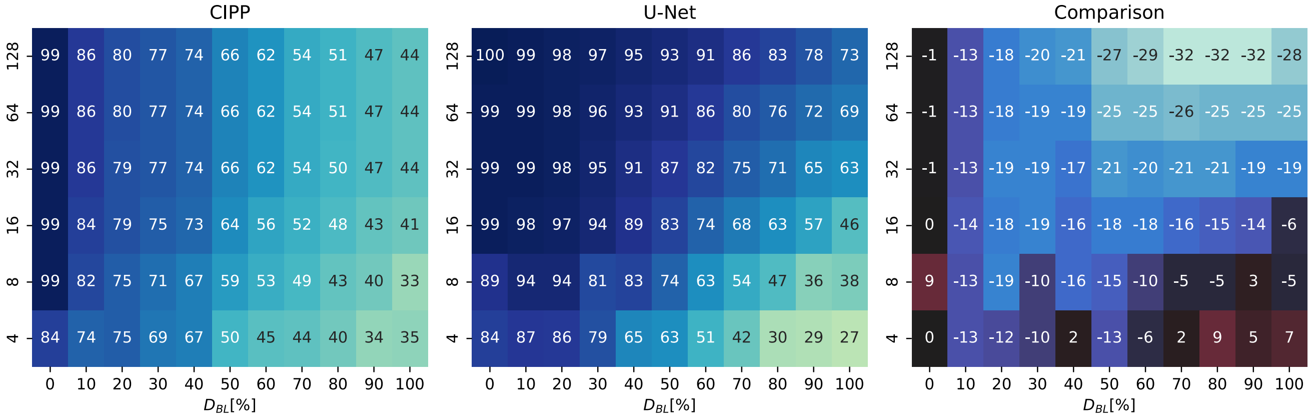


Fig. 15 Experimental results for test series only applying blurring for different difficulties $D\text{BL}$ over the number of training images N



CIPP

	0	10	20	30	40	50	60	70	80	90	100
128	99	86	80	77	74	66	62	54	51	47	44
64	99	86	80	77	74	66	62	54	51	47	44
32	99	86	79	77	74	66	62	54	50	47	44
16	99	84	79	75	73	64	56	52	48	43	41
8	99	82	75	71	67	59	53	49	43	40	33
4	84	74	75	69	67	50	45	44	40	34	35

U-Net

	0	10	20	30	40	50	60	70	80	90	100
128	100	99	98	97	95	93	91	86	83	78	73
64	99	99	98	96	93	91	86	80	76	72	69
32	99	99	98	95	91	87	82	75	71	65	63
16	99	98	97	94	89	83	74	68	63	57	46
8	89	94	94	81	83	74	63	54	47	36	38
4	84	87	86	79	65	63	51	42	30	29	27

Comparison

	0	10	20	30	40	50	60	70	80	90	100
128	-1	-13	-18	-20	-21	-27	-29	-32	-32	-32	-28
64	-1	-13	-18	-19	-19	-25	-25	-26	-25	-25	-25
32	-1	-13	-19	-19	-17	-21	-20	-21	-21	-19	-19
16	0	-14	-18	-19	-16	-18	-18	-16	-15	-14	-6
8	9	-13	-19	-10	-16	-15	-10	-5	-5	3	-5
4	0	-13	-12	-10	2	-13	-6	2	9	5	7

Fig. 16 The performance matrices of test series (BL) for CIPP and U-Net comparing dataset difficulties $D\text{BL}$ and number of training images N. The performance is measured using the average $F1$-score. The comparison shows the performance gap between CIPP and U-Net

mance until a difficulty $D_{\text{CS}} \leq 50\%$ even with $N = 4$ images besides few outliers, while the U-Net is not producing less stable results. It is as well notable that the peak performance of the CIPP drops for difficulty $D_{\text{CS}} = 90\%$. We can assume that at this point the CIPP is lacking the necessary tools to compensate for the applied noise.

The CIPP does not improve from $N = 16$ to $N = 128$ and has already reached its full potential at $N = 16$. In comparison, the U-Net improves more from $N = 16$ to $N = 128$ than from $N = 4$ to $N = 16$ with an average improvement of 23.91%. The CIPP is able to solve this task nearly perfectly. It detects the sharp texture of the object in the task. This sharp texture is not affected by the applied color-shift. This way the CIPP can solve the task by focusing on the texture while being able to ignore color changes. Only for very high color deviations, the texture can vanish when the color is changing so much that it is limited by the allowed values within an image [0, 255]. The U-Net in contrast can be confused by the differences in colors especially when few images are presented and the change in color is substantial.

As visible in Fig. 17 the U-Net is only able to outperform the CIPP at a difficulty of $D \geq 70\%$ and $N \geq 64$.

B.5: Salt-N-pepper

This test series focuses on salt-n-pepper noise. We have visualized example image in Fig. 18. The results of three different difficulties D_{SNP} are visualized in Fig. 19. The U-Net used

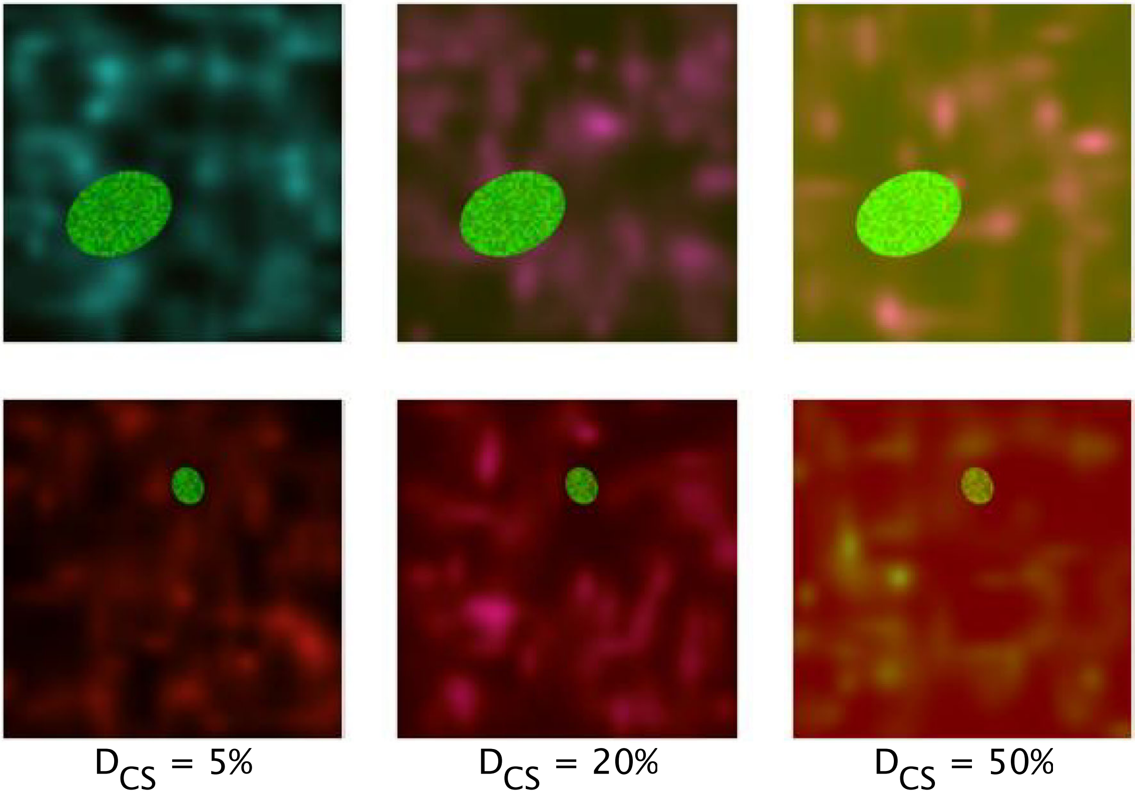


Fig. 17 Exemplary images from the test series focusing on color-shift

in this synthetic experiment did not utilize any pretraining or specific backbone. The deviation increases with higher difficulties D_{SNP} but it is only relevant for $N = 4$ training images. The top performance of the CIPP is steadily dropping with increasing difficulty. The effect on the U-Net is minimal. The CIPP in this test series is more sensitive to changes in difficulty. The U-Net is less affected on average, as visible in Fig. 20. It appears that the U-Net is able to adapt very well to the salt-n-pepper noise.

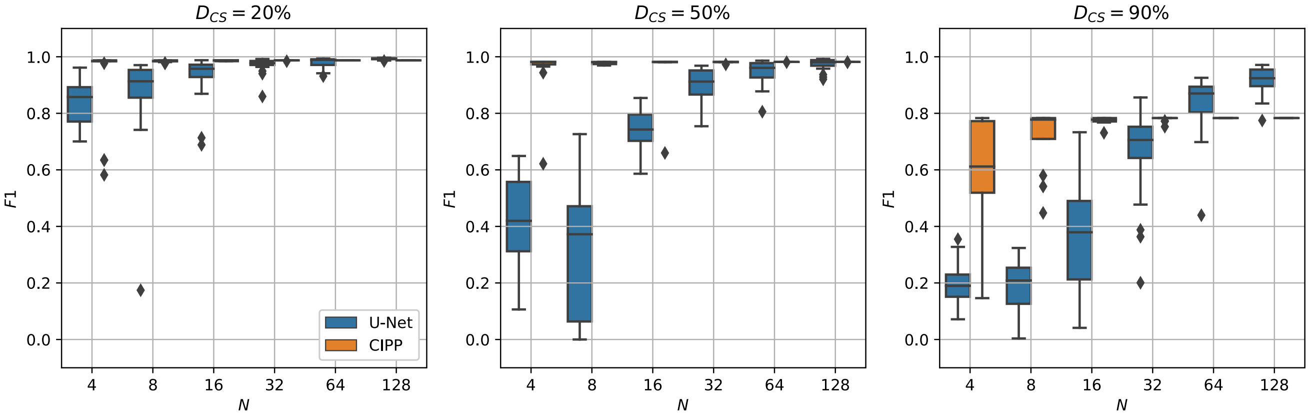


Fig. 18 Experimental results for test series only applying color-shift for different difficulties DCS over the amount of training image N

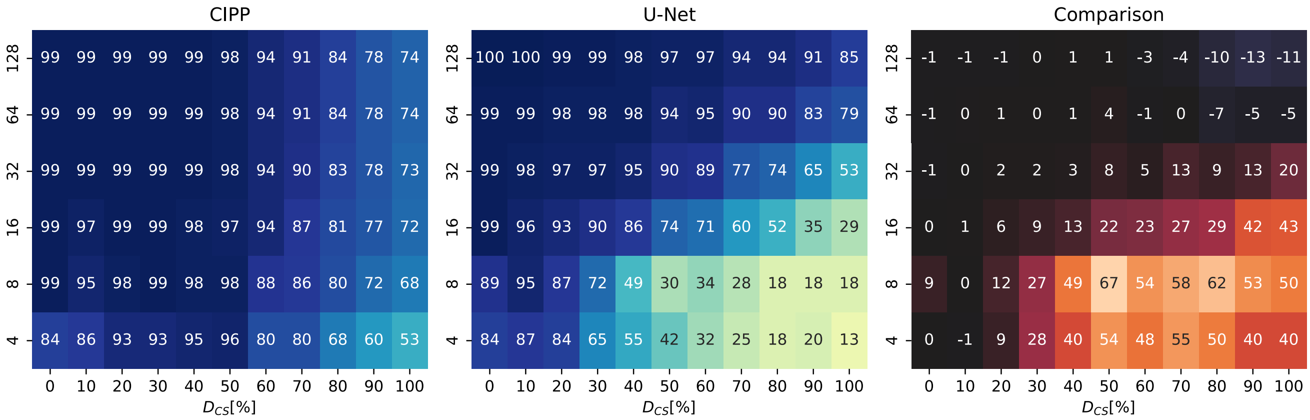


Fig. 19 The performance matrices of test series color-shift for CIPP and U-Net comparing dataset difficulties DCS and number of training images N. The performance is measured using the average $F1$-score. The comparison shows the performance gap between CIPP and U-Net

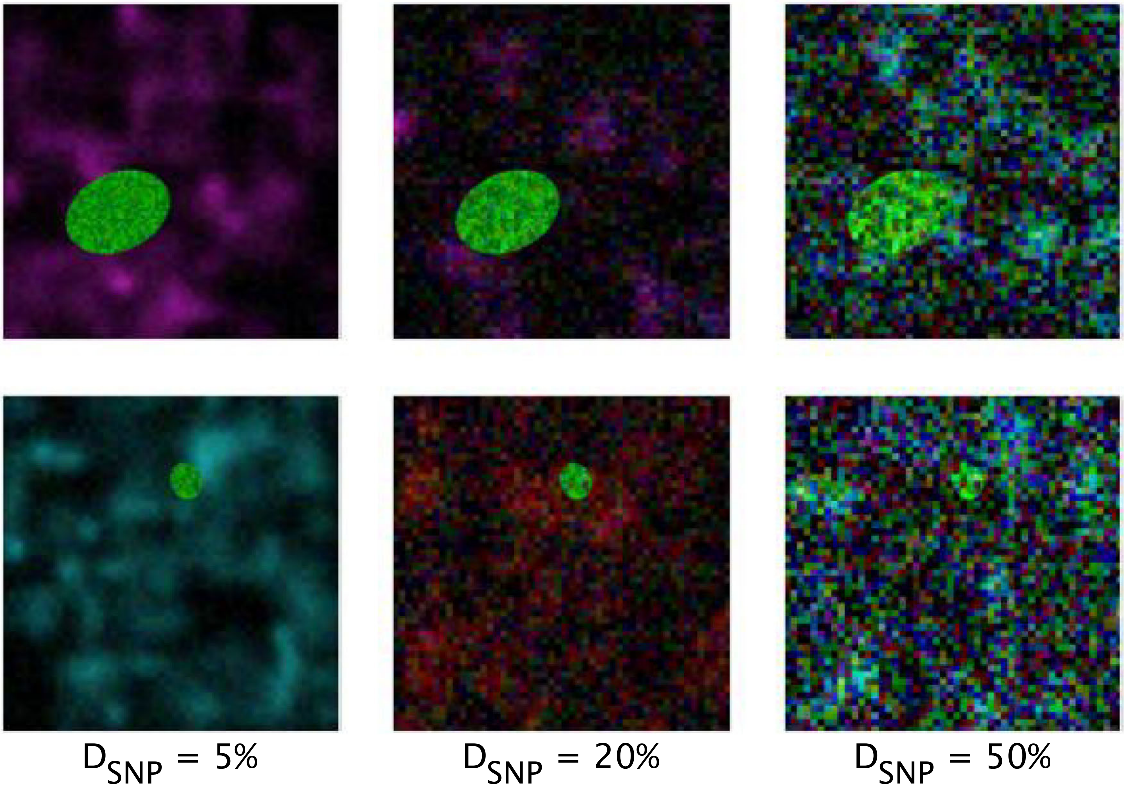


Fig. 20 Exemplary images from the test series focusing on salt-n-pepper

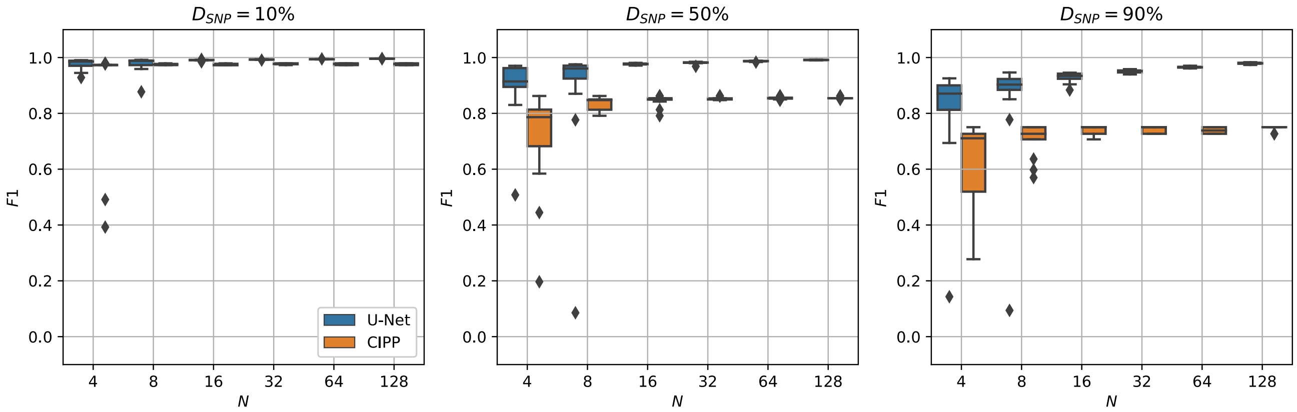
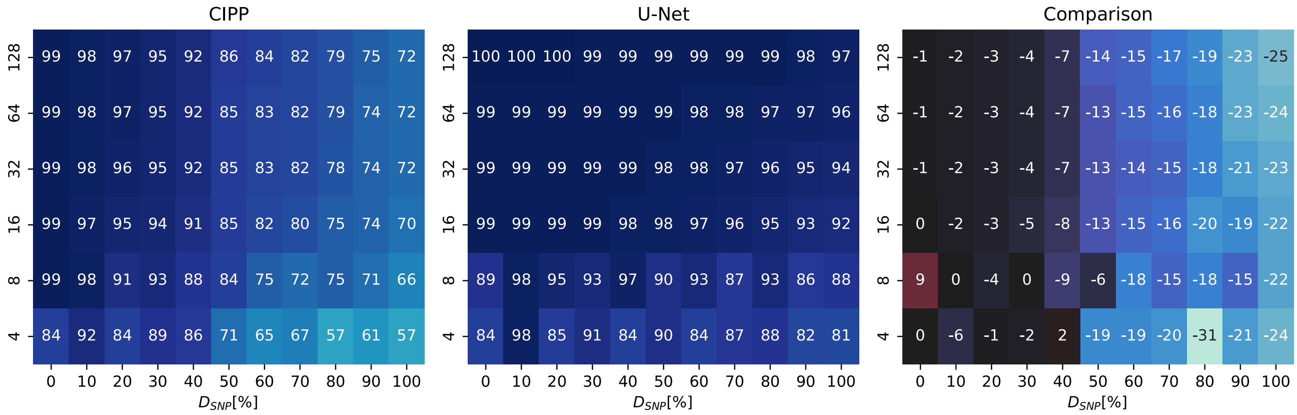


Fig. 21 Experimental results for test series only applying salt-n-pepper noise for different difficulties $DSNP$ over the amount of training image N



CIPP

N	0	10	20	30	40	50	60	70	80	90	100
128	99	98	97	95	92	86	84	82	79	75	72
64	99	98	97	95	92	85	83	82	79	74	72
32	99	98	96	95	92	85	83	82	78	74	72
16	99	97	95	94	91	85	82	80	75	74	70
8	99	98	91	93	88	84	75	72	75	71	66
4	84	92	84	89	86	71	65	67	57	61	57

U-Net

N	0	10	20	30	40	50	60	70	80	90	100
128	100	100	100	99	99	99	99	99	99	98	97
64	99	99	99	99	99	99	98	98	97	97	96
32	99	99	99	99	99	98	98	97	96	95	94
16	99	99	99	99	98	98	97	96	95	93	92
8	89	98	95	93	97	90	93	87	93	86	88
4	84	98	85	91	84	90	84	87	88	82	81

Comparison

N	0	10	20	30	40	50	60	70	80	90	100
128	-1	-2	-3	-4	-7	-14	-15	-17	-19	-23	-25
64	-1	-2	-3	-4	-7	-13	-15	-16	-18	-23	-24
32	-1	-2	-3	-4	-7	-13	-14	-15	-18	-21	-23
16	0	-2	-3	-5	-8	-13	-15	-16	-20	-19	-22
8	9	0	-4	0	-9	-6	-18	-15	-18	-15	-22
4	0	-6	-1	-2	2	-19	-19	-20	-31	-21	-24

Fig. 22 The performance matrices of test series salt-n-pepper for CIPP and U-Net comparing dataset difficulties $DSNP$ and number of training images N. The performance is measured using the average $F1$-score. The comparison shows the performance gap between CIPP and U-Net

References

1. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. CoRR (2015). arXiv:1505.04597
2. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. CoRR (2016). arXiv:1612.01105
3. Chen, L., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. CoRR (2017). arXiv:1706.05587
4. Yan, H., Zhang, C., Wu, M.: Lawin transformer: improving semantic segmentation transformer with multi-scale representations via large window attention. CoRR (2022). arXiv:2201.01615
5. Martin, V., Thonnat, M.: A cognitive vision approach to image segmentation. Tools Artif. Intell. **8**, 265–294 (2008). https://doi.org/10.5772/6080
6. Taveira, L.F.R., Kurc, T., Melo, A.C.M.A., Kong, J., Bremer, E., Saltz, J.H., Teodoro, G.: Multi-objective parameter auto-tuning for tissue image segmentation workflows. J. Digit. Imaging **2019**(32), 521–533 (2018)
7. Teodoro, G., Kurç, T.M., Taveira, L.F.R., Melo, A.C.M.A., Gao, Y., Kong, J., Saltz, J.H.: Algorithm sensitivity analysis and parameter tuning for tissue image segmentation pipelines. Bioinformatics **33**(7), 1064–1072 (2017). https://doi.org/10.1093/bioinformatics/btw749
8. Carpenter, A.E., Jones, T.R., Lamprecht, M.R., Clarke, C., Kang, I.H., Friman, O., Guertin, D.A., Chang, J.H., Lindquist, R.A., Moffat, J., Golland, P., Sabatini, D.M.: Cell profiler: image analysis software for identifying and quantifying cell phenotypes. Genome Biol. **7**(10), 100 (2006). https://doi.org/10.1186/gb-2006-7-10-r100
9. Linardatos, P., Papastefanopoulos, V., Kotsiantis, S.: Explainable AI: a review of machine learning interpretability methods. Entropy **23**, 18 (2021). https://doi.org/10.3390/e23010018
10. Lin, D., Li, Y., Prasad, S., Nwe, T.L., Dong, S., Oo, Z.M.: CAM-UNET: class activation MAP guided UNET with feedback refinement for defect segmentation. In: 2020 IEEE International Conference on Image Processing (ICIP), pp. 2131–2135 (2020). https://doi.org/10.1109/ICIP40778.2020.9190900
11. Mahony, N.O., Campbell, S., Carvalho, A., Harapanahalli, S., Velasco-Hernández, G.A., Krpalkova, L., Riordan, D., Walsh, J.: Deep learning versus traditional computer vision. CoRR (2019). arXiv:1910.13796
12. Alzubaidi, L., Zhang, J., Humaidi, A.J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M.A., Al-Amidie, M., Farhan, L.: Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. J. Big Data **8**(1), 53 (2021). https://doi.org/10.1186/s40537-021-00444-8
13. Anubha Pearline, S., Sathiesh Kumar, V., Harini, S.: A study on plant recognition using conventional image processing and deep learning approaches. J. Intell. Fuzzy Syst. **36**(3), 1997–2004 (2019). https://doi.org/10.3233/JIFS-169911
14. Hegde, R.B., Prasad, K., Hebbar, H., Singh, B.M.K.: Comparison of traditional image processing and deep learning approaches for

classification of white blood cells in peripheral blood smear images. Biocybern. Biomed. Eng. **39**(2), 382–392 (2019). https://doi.org/10.1016/j.bbe.2019.01.005

15. Sharma, S., Mehra, R.: Conventional machine learning and deep learning approach for multi-classification of breast cancer histopathology images-a comparative insight. J. Digit. Imaging **33**(3), 632–654 (2020). https://doi.org/10.1007/s10278-019-00307-y
16. Okayasu, K., Yoshida, K., Fuchida, M., Nakamura, A.: Vision-based classification of mosquito species: comparison of conventional and deep learning methods. Appl Sci **9**, 18 (2019). https://doi.org/10.3390/app9183935
17. Boumaraf, S., Liu, X., Wan, Y., Zheng, Z., Ferkous, C., Ma, X., Li, Z., Bardou, D.: Conventional machine learning versus deep learning for magnification dependent histopathological breast cancer image classification: a comparative study with visual explanation. Diagnostics **11**, 3 (2021). https://doi.org/10.3390/diagnostics11030528
18. Wang, P., Fan, E., Wang, P.: Comparative analysis of image classification algorithms based on traditional machine learning and deep learning. Pattern Recogn. Lett. **141**, 61–67 (2021)
19. Van Valen, D.A., Kudo, T., Lane, K.M., Macklin, D.N., Quach, N.T., DeFelice, M.M., Maayan, I., Tanouchi, Y., Ashley, E.A., Covert, M.W.: Deep learning automates the quantitative analysis of individual cells in live-cell imaging experiments. PLoS Comput. Biol. **12**(11), 1–24 (2016). https://doi.org/10.1371/journal.pcbi.1005177
20. Brehar, R., Mitrea, D.-A., Vancea, F., Marita, T., Nedevschi, S., Lupsor-Platon, M., Rotaru, M., Badea, R.I.: Comparison of deep-learning and conventional machine-learning methods for the automatic recognition of the hepatocellular carcinoma areas from ultrasound images. Sensors **20**, 11 (2020). https://doi.org/10.3390/s20113085
21. Harangi, B., Toth, J., Bogacsovics, G., Kupas, D., Kovacs, L., Hajdu, A.: Cell detection on digitized Pap smear images using ensemble of conventional image processing and deep learning techniques. In: 2019 11th International Symposium on Image and Signal Processing and Analysis (ISPA), pp. 38–42 (2019). https://doi.org/10.1109/ISPA.2019.8868683
22. Fotin, S.V., Yin, Y., Haldankar, H., Hoffmeister M.D., J.W., Periaswamy, S.: Detection of soft tissue densities from digital breast tomosynthesis: comparison of conventional and deep learning approaches. In: Tourassi, G.D. (eds.) Medical Imaging 2016: Computer-Aided Diagnosis, vol. 9785, pp. 228–233. International Society for Optics and Photonics (2016). https://doi.org/10.1117/12.2217045
23. Bianconi, F., Fravolini, M.L., Pizzoli, S., Palumbo, I., Minestrini, M., Rondini, M., Nuvoli, S., Spanu, A., Palumbo, B.: Comparative evaluation of conventional and deep learning methods for semi-automated segmentation of pulmonary nodules on CT. Quant. Imaging Med. Surg. **11**(34249654), 3286–3305 (2021). https://doi.org/10.21037/qims-20-1356
24. Ren, Y., Huang, J., Hong, Z., Lu, W., Yin, J., Zou, L., Shen, X.: Image-based concrete crack detection in tunnels using deep fully convolutional networks. Construct. Build. Mater. **234**, 117367 (2020). https://doi.org/10.1016/j.conbuildmat.2019.117367
25. Karabağ, C., Jones, M.L., Peddie, C.J., Weston, A.E., Collinson, L.M., Reyes-Aldasoro, C.C.: Semantic segmentation of HeLa cells: an objective comparison between one traditional algorithm and four deep-learning architectures. PLoS ONE **15**(10), 1–21 (2020). https://doi.org/10.1371/journal.pone.0230605
26. King, A., Bhandarkar, S.M., Hopkinson, B.M.: A comparison of deep learning methods for semantic segmentation of coral reef survey images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (2018)
27. Ofir, N., Nebel, J.: Classic versus deep approaches to address computer vision challenges. CoRR (2021). arXiv:2101.09744
28. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Trans. Syst. Man Cybern. **9**(1), 62–66 (1979). https://doi.org/10.1109/TSMC.1979.4310076
29. Caicedo, J.C., Roth, J., Goodman, A., Becker, T., Karhohs, K.W., Broisin, M., Molnar, C., McQuin, C., Singh, S., Theis, F.J., Carpenter, A.E.: Evaluation of deep learning strategies for nucleus segmentation in fluorescence images. Cytometry A **95**(9), 952–965 (2019). https://doi.org/10.1002/cyto.a.23863
30. Scherr, T., Löffler, K., Böhland, M., Mikut, R.: Cell segmentation and tracking using CNN-based distance predictions and a graph-based matching strategy. PLoS ONE **15**(12), 1–22 (2020). https://doi.org/10.1371/journal.pone.0243219
31. Le'Clerc Arrastia, J., Heilenkötter, N., Otero Baguer, D., Hauberg-Lotte, L., Boskamp, T., Hetzer, S., Duschner, N., Schaller, J., Maass, P.: Deeply supervised UNet for semantic segmentation to assist dermatopathological assessment of basal cell carcinoma. J. Imaging **7**, 4 (2021). https://doi.org/10.3390/jimaging7040071
32. Schilling, M., Scherr, T., Münke, F.R., Neumann, O., Schutera, M., Mikut, R., Reischl, M.: Automated annotator variability inspection for biomedical image segmentation. IEEE Access **10**, 2753–2765 (2022). https://doi.org/10.1109/ACCESS.2022.3140378
33. Iakubovskii, P.: Segmentation Models. GitHub (2019)
34. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition (2015)
35. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A., Fei-Fei, L.: ImageNet large scale visual recognition challenge. In: IJCV (2015)
36. Edlund, C., Jackson, T.R., Khalid, N., Bevan, N., Dale, T., Dengel, A., Ahmed, S., Trygg, J., Sjögren, R.: LIVECell—a large-scale dataset for label-free live cell segmentation. Nat. Methods **18**, 9 (2021). https://doi.org/10.1038/s41592-021-01249-6
37. Pugliatti, M., Topputo, F.: DOORS: Dataset for Boulders Segmentation. Statistical Properties and Blender Setup (2022)
38. Codella, N.C.F., Gutman, D., Celebi, M.E., Helba, B., Marchetti, M.A., Dusza, S.W., Kalloo, A., Liopyris, K., Mishra, N., Kittler, H., Halpern, A.: Skin Lesion Analysis Toward Melanoma Detection: A Challenge at the 2017 International Symposium on Biomedical Imaging (ISBI), Hosted by the International Skin Imaging Collaboration (ISIC) (2018)
39. Mahbod, A., Schaefer, G., Bancher, B., Löw, C., Dorffner, G., Ecker, R., Ellinger, I.: CryoNuSeg: a dataset for nuclei instance segmentation of cryosectioned H&E-stained histological images. Comput. Biol. Med. **132**(104349), x (2021)
40. Chaurasia, A., Culurciello, E.: LinkNet: exploiting encoder representations for efficient semantic segmentation. In: 2017 IEEE Visual Communications and Image Processing (VCIP). IEEE (2017). https://doi.org/10.1109/vcip.2017.8305148
41. Lin, T.-Y., Dollar, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)

Friedrich Rieken Münke received the bachelor's and master's degrees in mathematics from the Karlsruhe Institute of Technology, Germany, in 2017 and 2019, respectively. Since 2019, he is doctoral candidate in the field of computer vision at the Institute for Automation and

Applied Computer Science Karlsruhe Institute of Technology, Karlsruhe, Germany. His research interests include conventional image processing methods, change detection, semantic segmentation, deep learning, and object detection.

Jan Schützke holds a master of science degree in mechanical engineering with a major in robotics and information technology, which was obtained at Karlsruhe Institute of Technology (KIT) in 2019. He is currently pursuing a Ph.D. in mechanical engineering, focusing on developing neural networks for the analysis of diffraction and spectroscopy data. Additionally, Mr. Schützke has a keen interest in the field of computer vision and actively collaborates with companies, external scientific groups, and colleagues to apply deep learning techniques across diverse fields.

Felix Berens received the bachelor's and master's degrees in mathematics from the University of Greifswald, Germany, in 2017 and 2019, respectively. Since 2019, he is doctoral candidate in the field of sensor fusion for autonomous driving at the Institute for Automation and Applied Computer Science Karlsruhe Institute of Technology, Karlsruhe, Germany, and the Institute for Artificial Intelligence, University of Applied Sciences Ravensburg-Weingarten, Weingarten, Germany. His research interests include sensor fusion, sensor configuration, machine learning, and object detection.

Markus Reischl received the Dipl.-Ing. and the Ph.D. degree in mechanical engineering from the University of Karlsruhe, Germany, in 2001 and 2006, respectively. Since 2020, he has been an Adjunct Professor at the Faculty of Mechanical Engineering. He is the head of research field "Automation for Laboratories" and heading the research group "Machine Learning for High-Throughput and Mechatronics" at the Institute for Automation and Applied Informatics at the Karlsruhe Institute of Technology. Research Interests: Man–machine interfaces, image processing, machine learning, and data analytics.