Simulierung möglicher Waldbrandausbreitung mittels GANs

Ersan Baran, Dennis Fischer, Sebastian Wewer

Gliederung

- Erläuterung der Grundidee (Sebastian)
- Zusammenstellung des Datensets (Sebastian)
- Satellitenbilder-Ansätze (Bot, Google Earth) (Sebastian)
- Einstieg GANs (Dennis)
- Welche GANs sind für unseren Zweck geeignet (Dennis)
- Tiefer in CycleGAN und DeepGAN (Ersan + Dennis)
- Weitere Ansatz Pix2Pix (Ersan + Dennis)

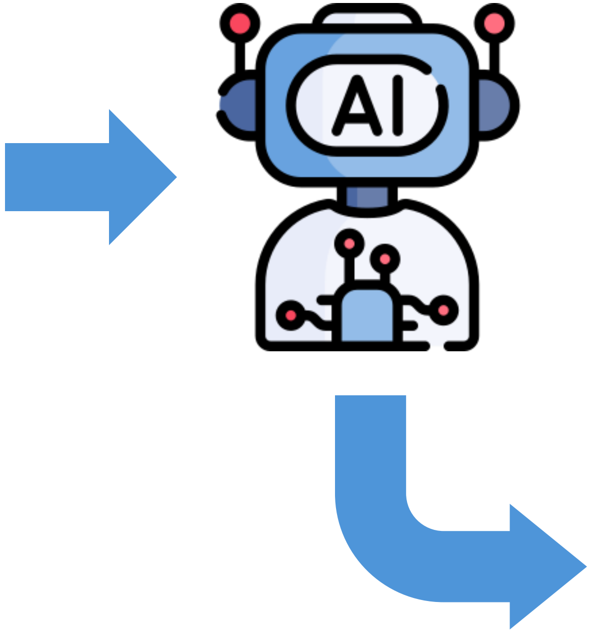
Grundidee

Können GANs dazu verwendet werden die Ausbreitung von Waldbränden realistisch darzustellen?

Paradise Pines, California, vor, während und nach einem Camp-Feuer 2018.

Grundidee

Input



Output (expected)

Output (actual)

Datenset

Wildfire Prediction Dataset (Satellite Images)[1]

- 22.710 Bilder von Waldbränden
- 20.140 Bilder ohne Waldbrände

Datenset

Problem:

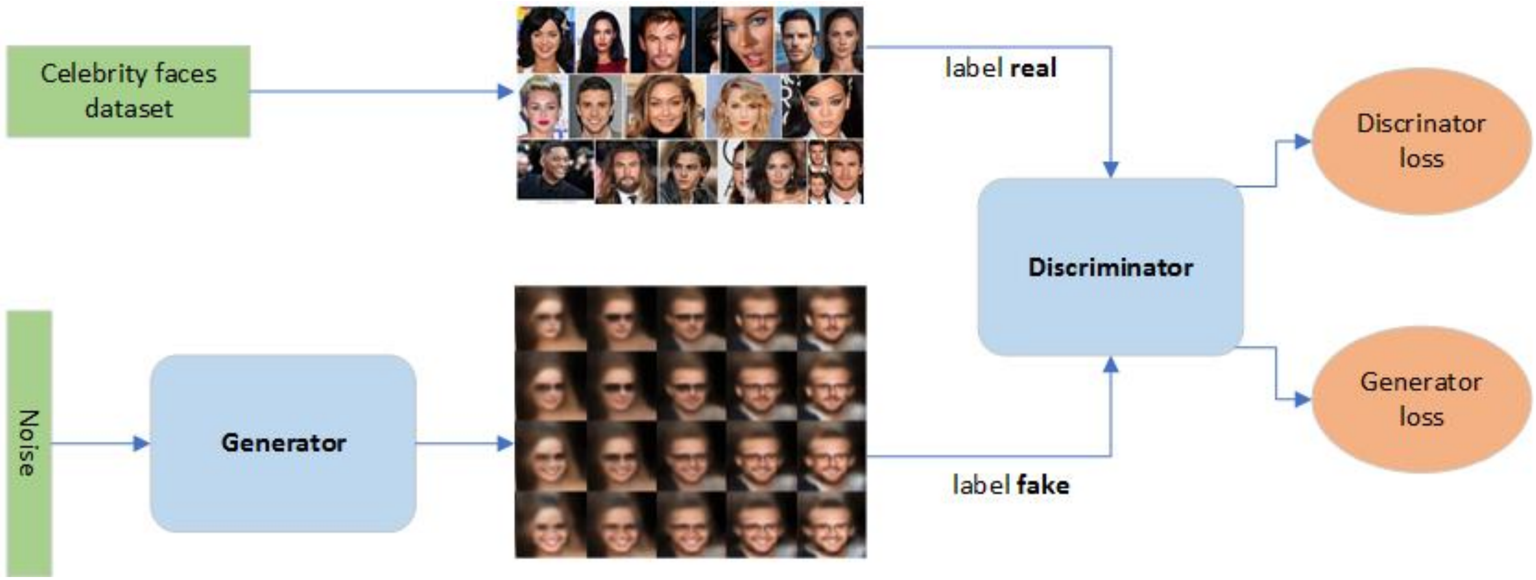
- Keine vor- und nachher Bilder der gleichen Umgebung
- Kein Vergleich des Outputs mit tatsächlichen Waldbränden möglich

Lösung

- Erweitern des Datensets um aktuelle Sattelitenbilder von Gegenden, für die wir Waldbrände haben
- Erster Ansatz: Download von Sattelitenbildern via Bot
- Zweiter Ansatz: Download via Google Maps API

Was sind GANs?

- GANs = Generative Adversarial Networks
- Generator
- Diskriminator
- Anwendung



https://towardsdatascience.com/generative-adversarial-network-gan-for-dummies-a-step-by-step-tutorial-fdefff170391

[2]

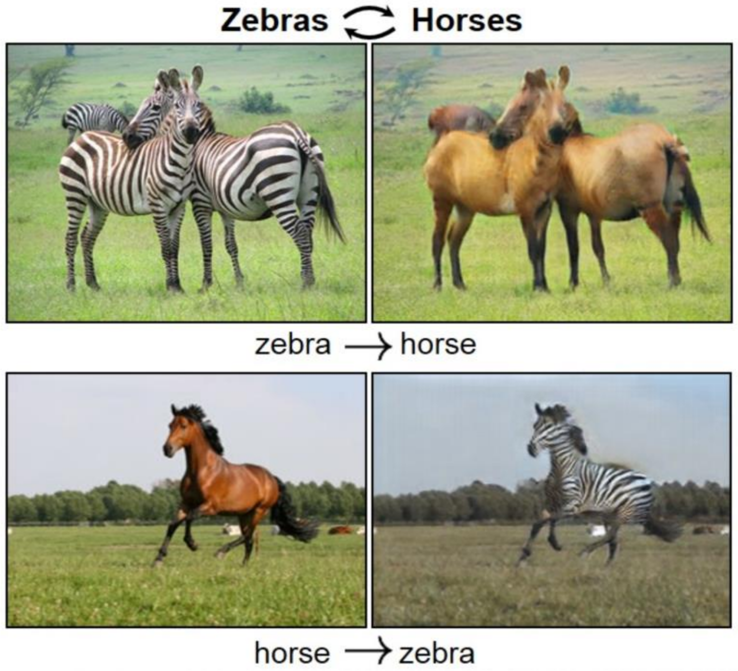
Funktionsweise von GANs

- Initialisierung mit zufälligen Werten
- Training des Generators und Diskriminators
- Iterationen

[2]

Welche GANs gibt es?

- DCGAN (Deep Convolutional GAN)
- CGAN (Conditional GAN)
- WGAN (Wasserstein GAN)
- CycleGAN
- StyleGAN
- Pix2Pix
- Weitere GANs mit Vor- und Nachteilen(https://docs.google.com/document/d/1MYEDzg9DSmH47d-IqOXHvddCKhRWP5WZy4DRDH97C98/edit)



https://aws.amazon.com/de/what-is/gan/

[3]

Welches GAN eignet sich für unsere Anwendung?

- DCGAN
- CycleGAN
- Pix2Pix

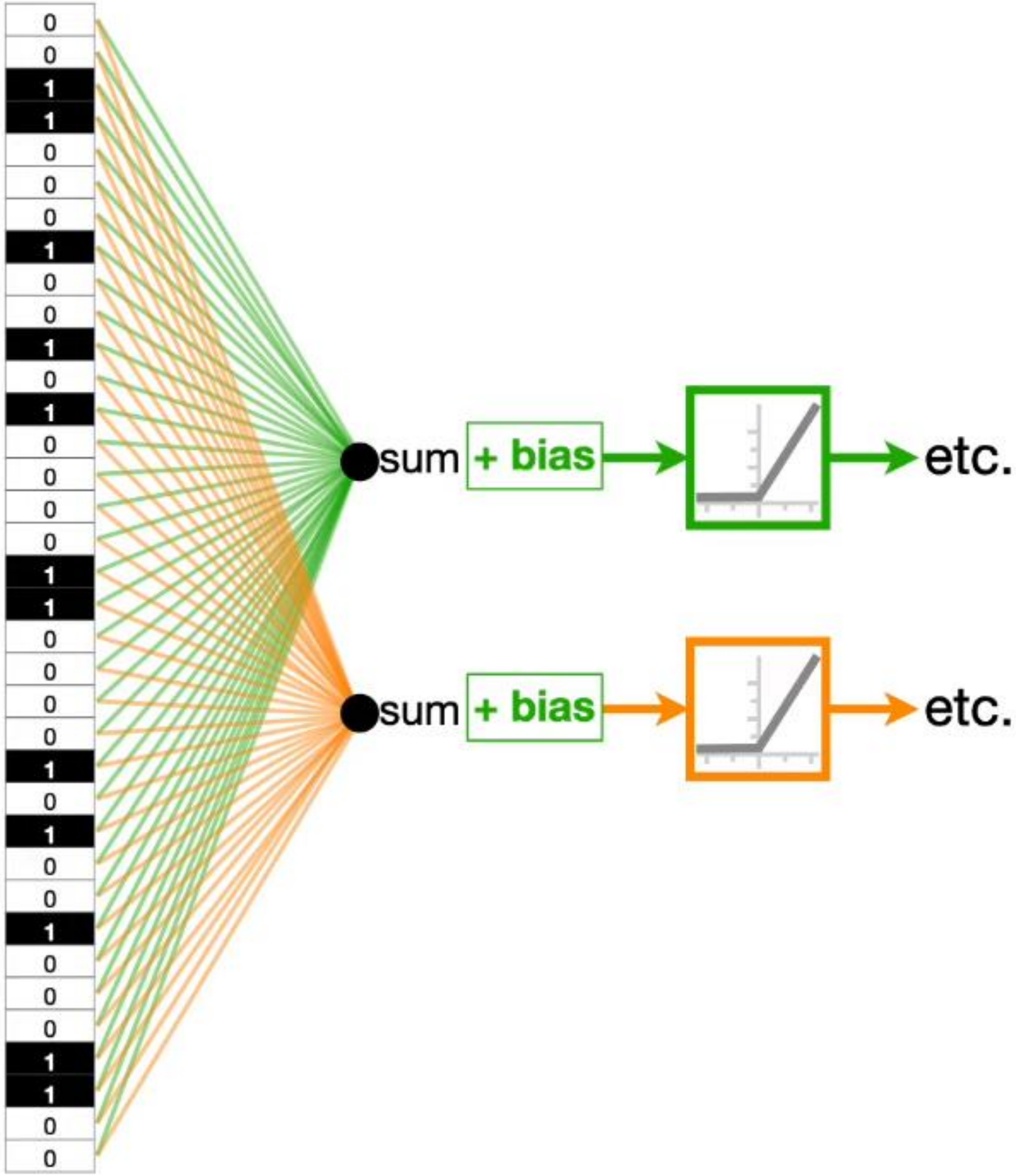
[3]

CycleGAN

- Diskriminator
- Generator
- Trainingsprozess

Warum Convolutions?

0	0	1	1	0	0
0	1	0	0	1	0
1	0	0	0	0	1
1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0

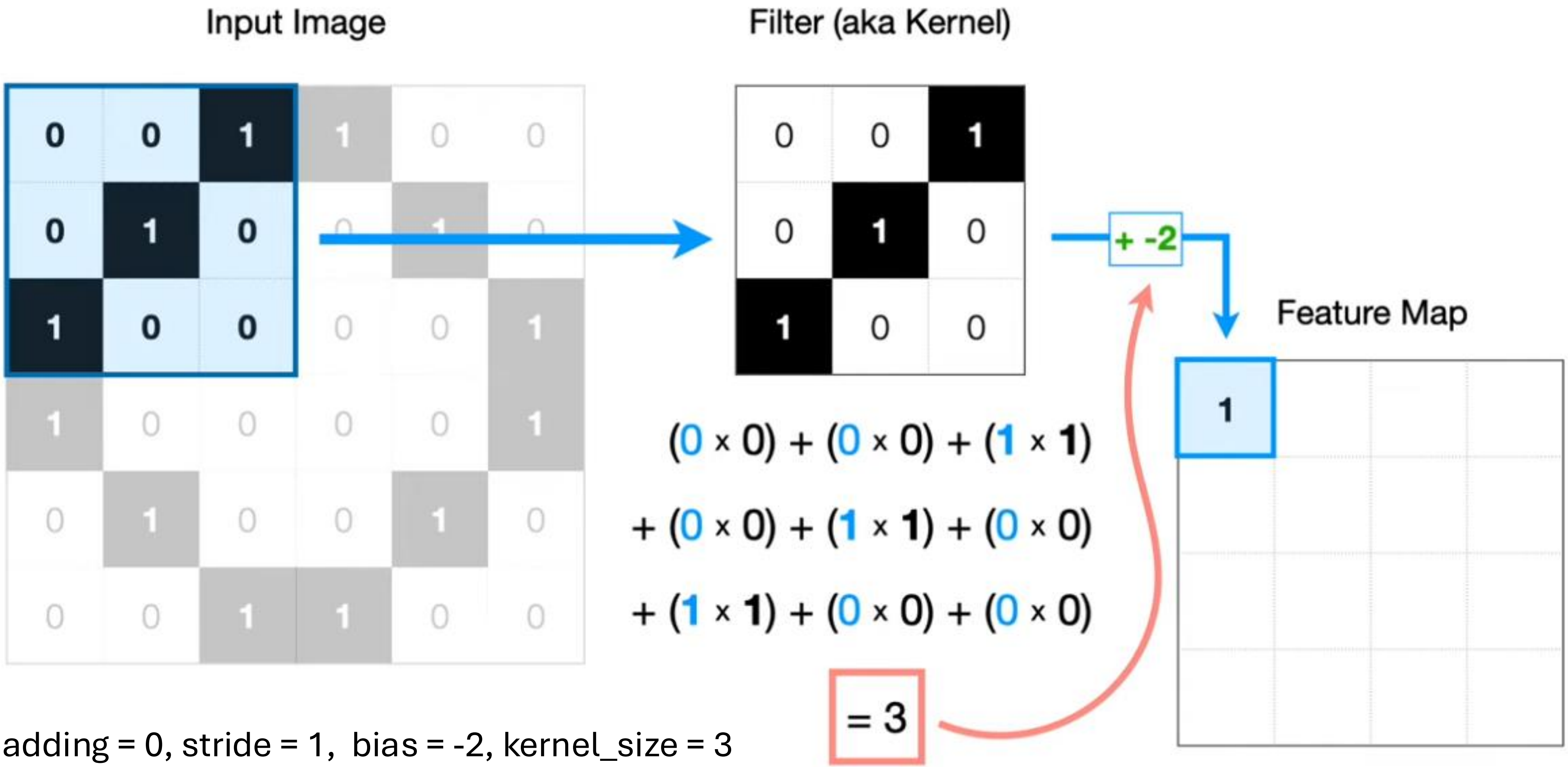


Im Beispiel: 6 x 6 = 36 Gewichte pro Knoten

Bei 100 x 100 = 10000 Gewichte pro Knoten

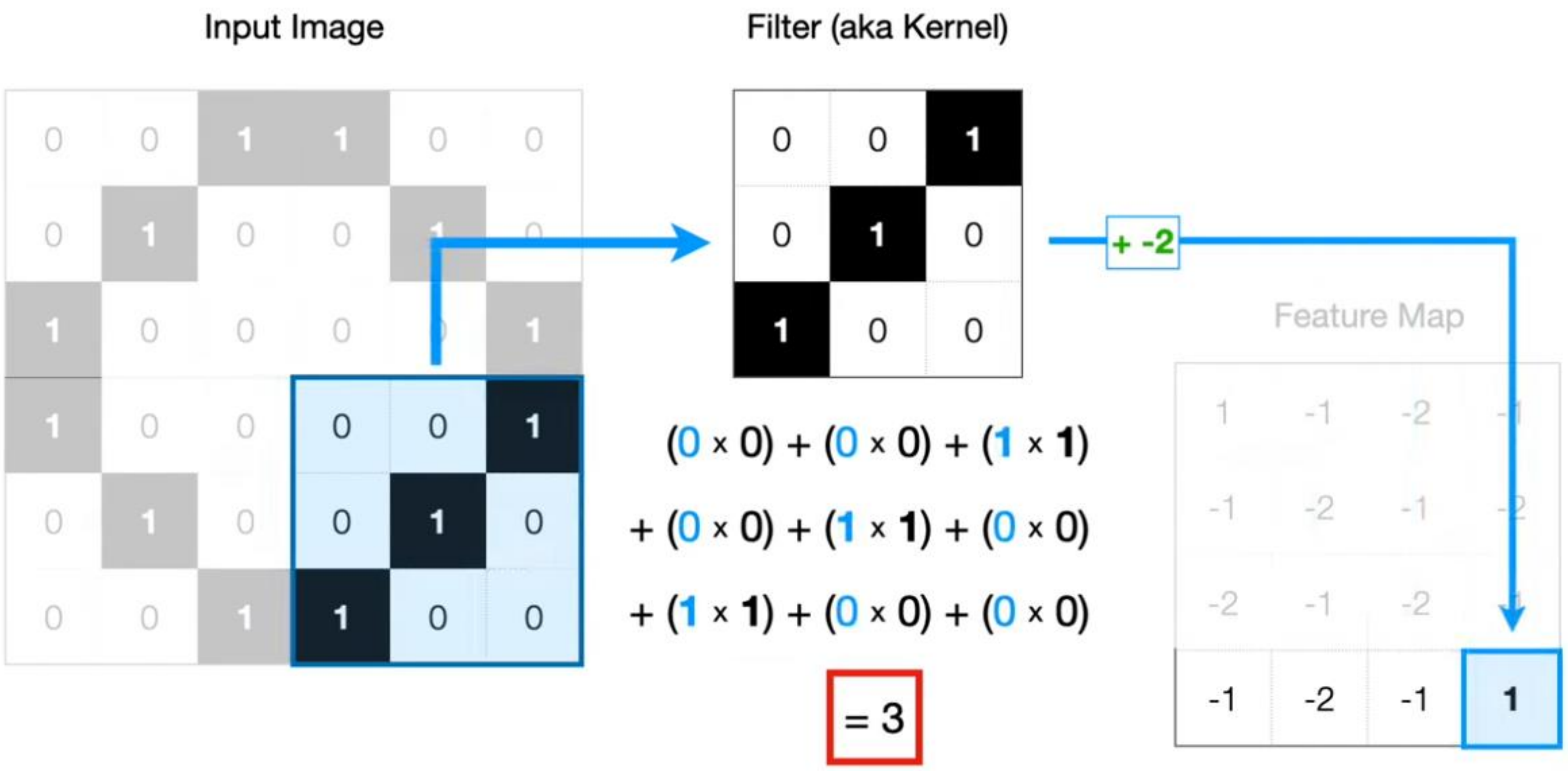
[7]

Convolution



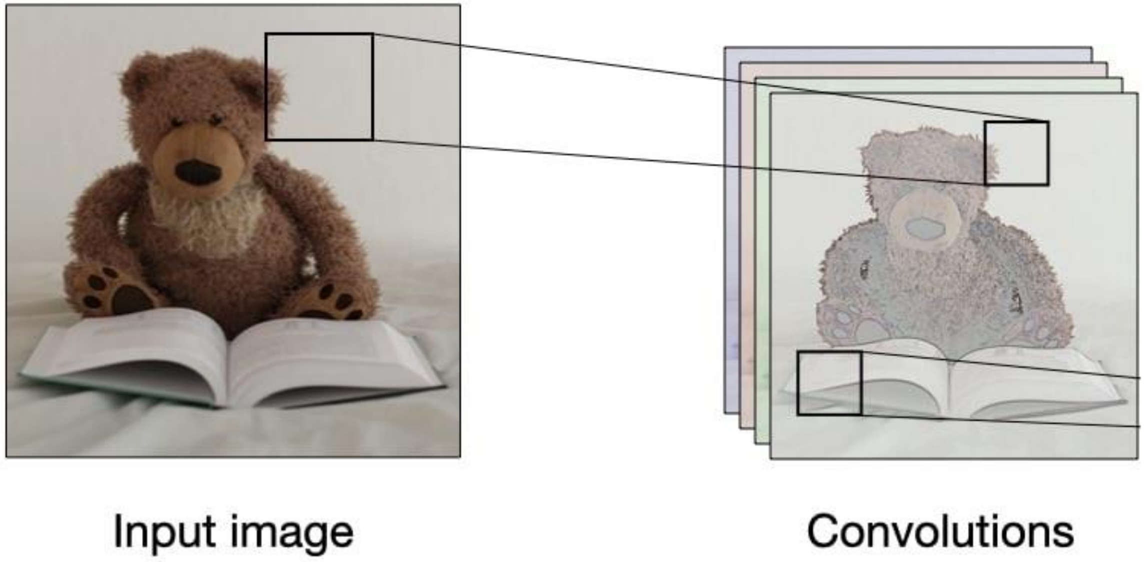
padding = 0, stride = 1, bias = -2, kernel_size = 3

[7]


Input Image
0 0 1 1 0 0
0 1 0 0 1 0
1 0 0 0 0 1
1 0 0 0 0 1
0 1 0 0 1 0
0 0 1 1 0 0
Filter (aka Kernel)
0 0 1
0 1 0
1 0 0
+ -2
Feature Map
1 -1 -2 -1
-1 -2 -1 -2
-2 -1 -2 1
-1 -2 -1 1
(0 × 0) + (0 × 0) + (1 × 1)
+ (0 × 0) + (1 × 1) + (0 × 0)
+ (1 × 1) + (0 × 0) + (0 × 0)
= 3

[7]

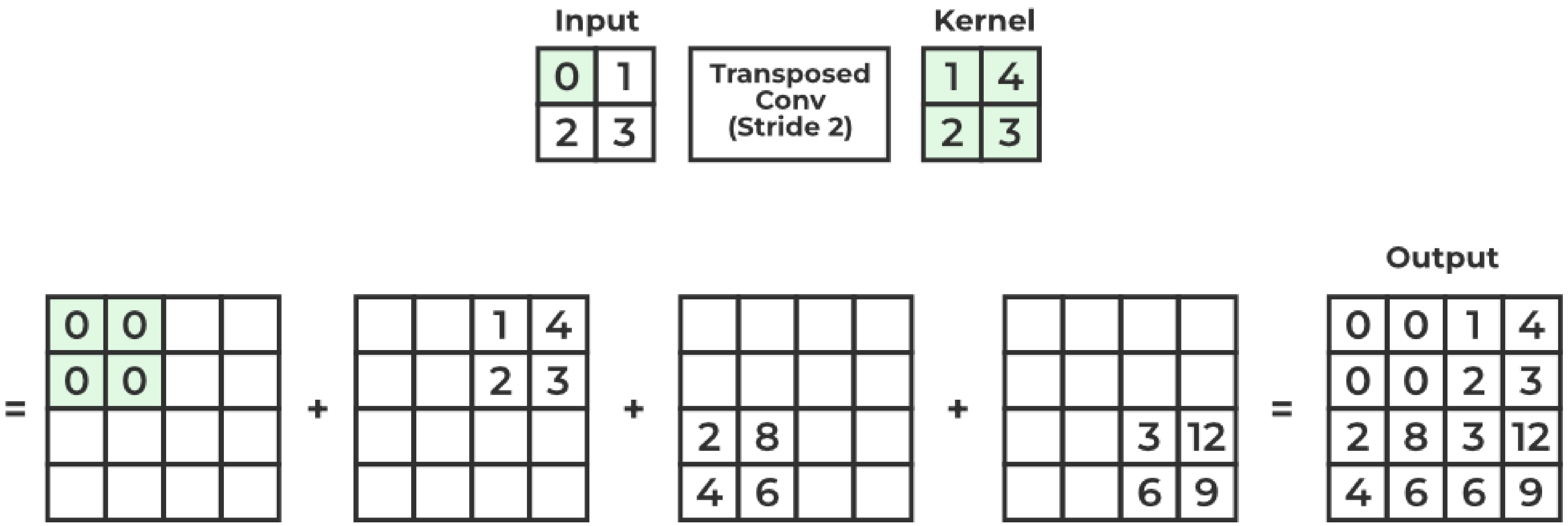
Beispiel einer Feature Map



Jeder Filter erkennt verschiedene Features im Bild

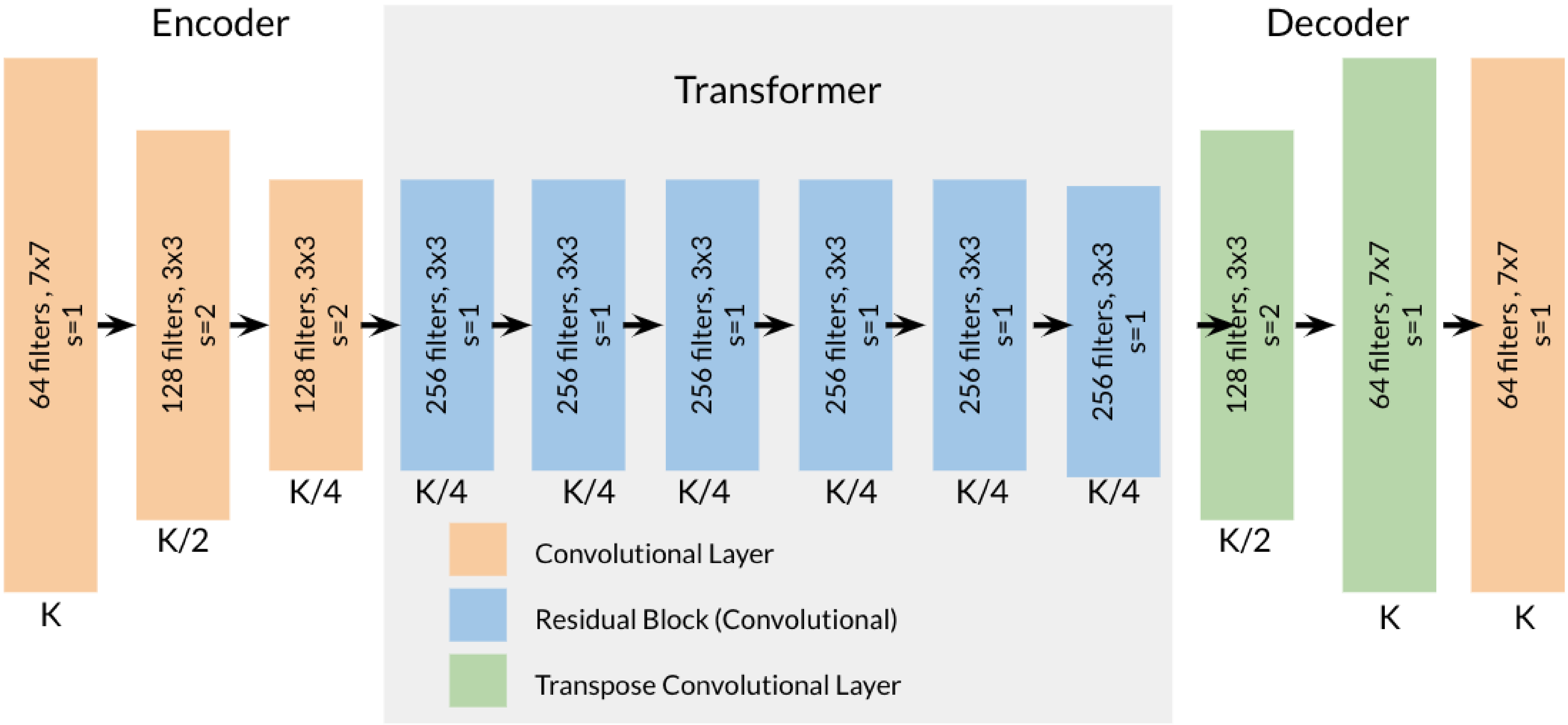
[8]

Transposed Convolution



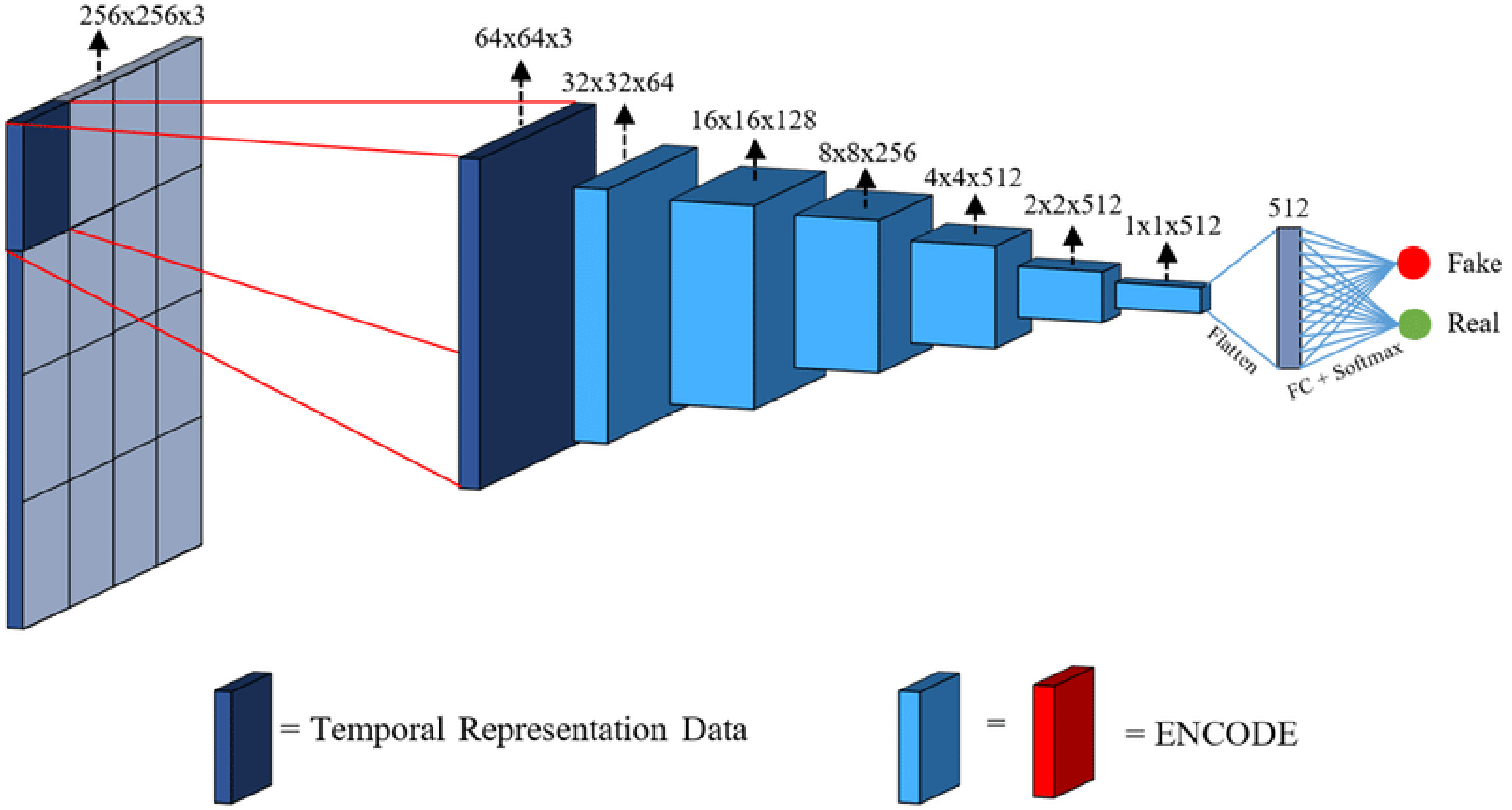
[7]

Architektur des Generators



[10]

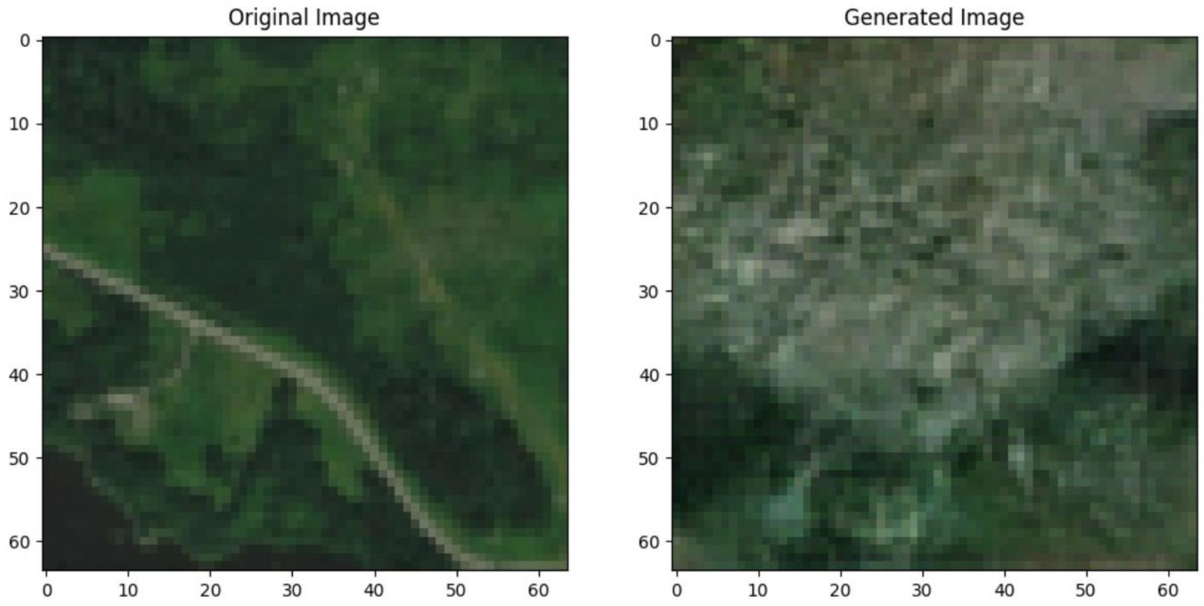
Architektur des Diskriminators



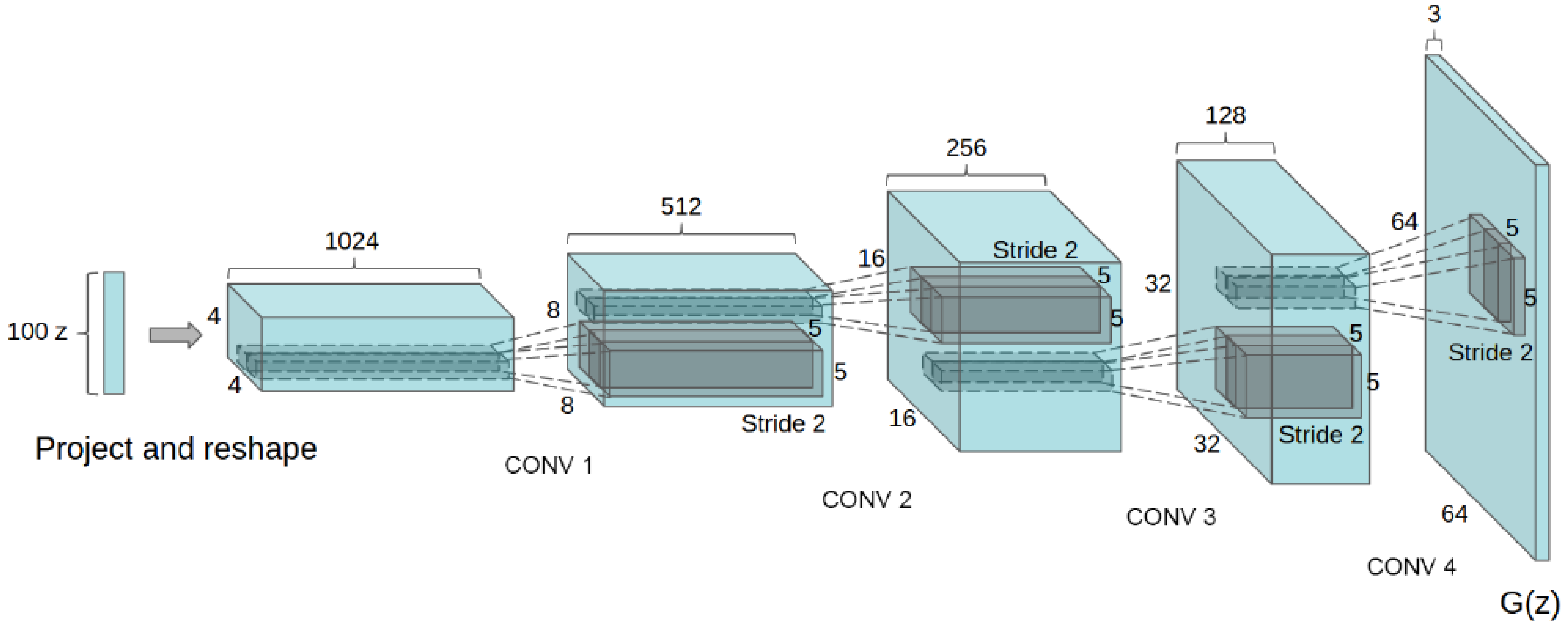
[11]

DCGAN

- Diskriminator
- Generator
- Trainingsprozess

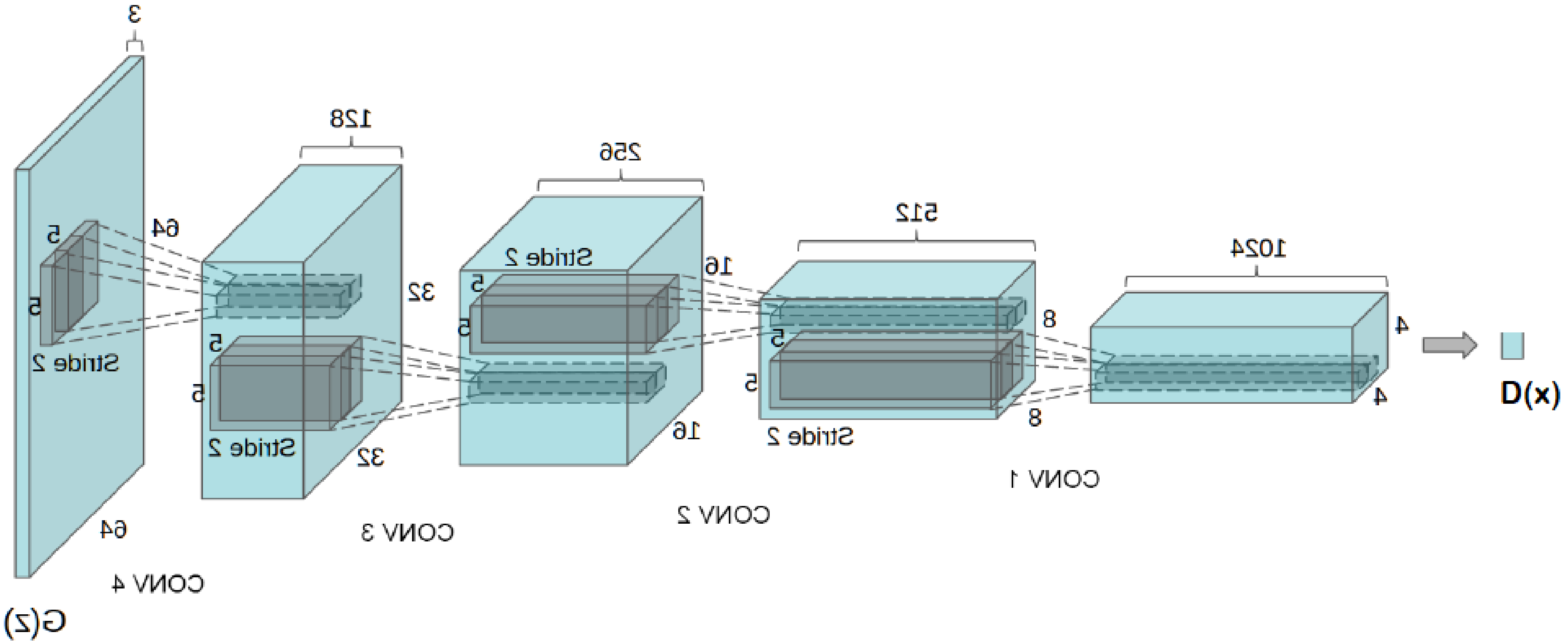


Architektur des Generators



[4]

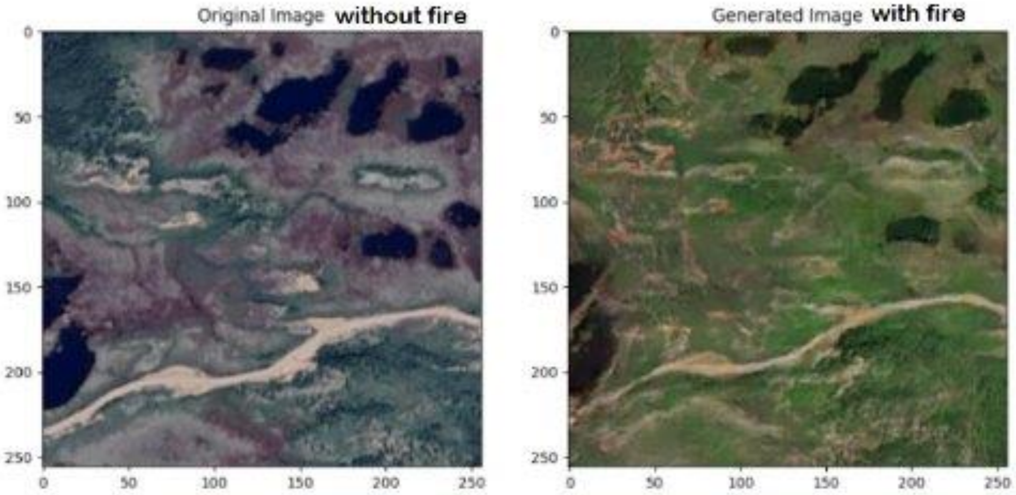
Architektur des Diskriminators



[4]

Pix2Pix

- Diskriminator

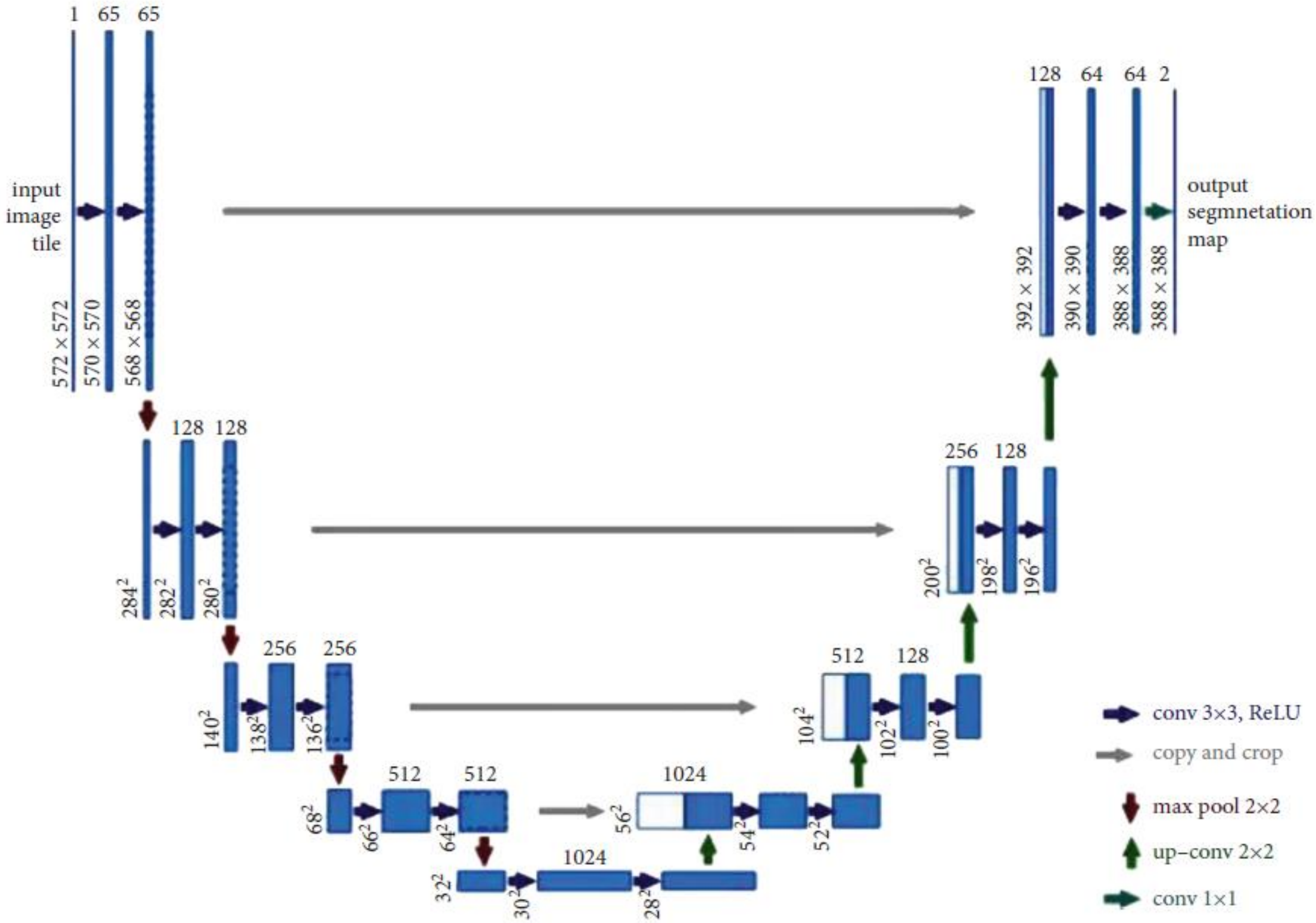


- Generator

- Trainingsprozess

Architektur des Generators

U-Net



[12]

Architektur des Diskriminators

- Wie beim CycleGAN

Vergleich von Pix2Pix und CycleGAN

Labels -> Fotos

Loss	Per-pixel acc.	Per-class acc.	Class IOU
CoGAN [32]	0.40	0.10	0.06
BiGAN/ALI [9, 7]	0.19	0.06	0.02
SimGAN [46]	0.20	0.10	0.04
Feature loss + GAN	0.06	0.04	0.01
CycleGAN (ours)	**0.52**	**0.17**	**0.11**
pix2pix [22]	0.71	0.25	0.18

FCN-Score

$$\mathrm{IoU} = \frac{\mathrm{Intersection}}{\mathrm{Union}}$$

Fotos -> Labels

Loss	Per-pixel acc.	Per-class acc.	Class IOU
CoGAN [32]	0.45	0.11	0.08
BiGAN/ALI [9, 7]	0.41	0.13	0.07
SimGAN [46]	0.47	0.11	0.07
Feature loss + GAN	0.50	0.10	0.06
CycleGAN (ours)	**0.58**	**0.22**	**0.16**
pix2pix [22]	0.85	0.40	0.32

Vergleich zu
Ground-Truth-Labels

[6]

Input

Möglicher Feuergebiete bei einem Ausbruch

Quellen

[1] Wildfire Prediction Dataset (Satellite Images): https://www.kaggle.com/datasets/abdelghaniaaba/wildfire-prediction-dataset

[2] Saxena, D., & Cao, J. (2021). Generative Adversarial Networks (GANs). *ACM Computing Surveys (CSUR)*, 54, 1 - 42. https://doi.org/10.1145/3446374.

[3] Zhengwei Wang, Qi She, & Tomas E. Ward. (2020). Generative Adversarial Networks in Computer Vision: A Survey and Taxonomy.

[4] Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." (2015). https://arxiv.org/abs/1511.06434

[5]Isola, Phillip, et al. "Image-to-image translation with conditional adversarial networks." Proceedings of the IEEE conference on computer vision and pattern recognition. (2017). https://arxiv.org/abs/1611.07004

[6]Zhu, Jun-Yan, et al. "Unpaired image-to-image translation using cycle-consistent adversarial networks." Proceedings of the IEEE international conference on computer vision. (2017). https://arxiv.org/abs/1703.10593

[7] Josh Starmer. "Neural Networks Part 8: Image Classification with Convolutional Neural Networks (CNNs)" (2021) https://www.youtube.com/watch?v=HGwBXDKFk9I

[8] Shervine Amidi . "Convolutional Neural Networks cheatsheet" (2024) https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks

[9] "What is Transposed Convolutional Layer?" (2024) https://www.geeksforgeeks.org/what-is-transposed-convolutional-layer/

[10] "GANsPainter" (2024) https://github.com/manhph2211/GANsPainter?tab=readme-ov-file

[11] Ganokratanaa, Thittaporn & Aramvith, Supavadee & Sebe, Nicu. (2020). Unsupervised Anomaly Detection and Localization Based on Deep Spatiotemporal Translation Network. IEEE Access. PP. 1-1. 10.1109/ACCESS.2020.2979869.

[12] Yin, Xiao-Xia, et al. "U-Net-Based Medical Image Segmentation." Journal of healthcare engineering 2022.1 (2022): 4189781. https://onlinelibrary.wiley.com/doi/pdf/10.1155/2022/4189781