GENERATING SIMULATED INFRARED SENSOR IMAGES USING DEEP LEARNING NEURAL NETWORKS

N. Peinecke
DLR, Institute of Flight Guidance, Braunschweig, Germany

Abstract
This study presents a method for generating infrared images using deep learning techniques, specifically Generative Adversarial Networks (GANs), trained on data from real flight experiments. Implemented in Python, the approach leverages a U-Net adapted for gray-scale IR image generation. After training on a multi-core computer, the method achieved accurate infrared image generation, applicable in flight simulation and autonomous navigation.

Keywords
deep learning; sensor simulation; infrared simulation; GAN

NOMENCLATURE

Abbreviations

GAN	Generative Adversarial Networks
IR	infrared

1. INTRODUCTION

One of the most in-depth explored fields of simulation is the simulation of real-time imagery for flight simulation, and, related to this, electro-optical camera simulation. The simulation of infrared sensors, especially IR cameras is closely related and can be seen as a special case of optical camera simulation. Consequently, different implementations exist for application areas, such as airborne combat training [1], autonomous robot navigation [2], and flight testing [3]. Most of these application areas require real-time performance. Thus, these solutions are mostly based on the same graphics techniques as the visual simulations. However, since a larger effort has been spent on modelling visual appearance adequate models of real-world objects like houses, buildings and entire terrains do not exist for infrared simulation. Visual models usually lack thermal material properties that would be necessary to utilize such models for IR simulation. Consequently, a huge part of modelling efforts need to be spent for augmenting existing visual models with thermal properties.
The idea of this paper is to implicitly generate IR images from a stream of visual images by utilizing data that is available in a visual or electro-optical simulation. This study presents a method for generating infrared images by leveraging deep learning techniques, eliminating the need for additional modeling efforts. The approach utilizes Generative Adversarial Networks (GANs) and similar technologies, trained on data from real flight experiments. This method enhances the efficiency and accuracy of infrared image generation, paving the way for applications in the field of flight simulation, sensor processing, and autonomous navigation.

2. TRAINING DATA

Two sets of training data were used during the experiments:

- A set of 1600 optical camera and infrared camera image pairs generated using a sensor simulator. This set was used during the test phase of the experimentation to find a suitable combination of network architectures, loss function, optimizer and training method.
- A set of over 4000 optical camera and infrared camera image pairs taken from a flight campaign in 2012 [4]. Fig. 2 shows a pair of an electro-optical camera image and an IR image from the training data-set.

3. METHOD

The task is to generate synthetic IR images from a stream of (possibly simulated) camera images. Fig. 1 illustrates the workflow of the method: Given a pre-recorded set of not necessarily aligned corresponding camera and infrared images a Neural Network was trained using a sub-set of these. Then, the remaining images could be used to evaluate the efficiency of the approach. Several possible methods were considered:

- **Autoencoder:** Autoencoders are optimised to imitate the input image directly. Therefore, they poorly adapt to the training inputs since here the expected output does not match the input. Changing the output layer to instead produce false-colored IR images was not very successful. Results remained blurry, and the loss function did not converge after a certain point.
- **Neural network with “full mesh”:** Full mesh networks connect every input neuron to every other output neuron. This ensures that each input pixel can in principle influence each other output pixel. However, when choosing a lower number of internal nodes the complexity of the network seemed to be not sufficient for the task. The loss function did not converge at all. Higher node numbers tend to increase the training effort exponentially without a visible improvement in convergence.
- **Convolutional network with residual feed-forward (“U-Net”):** Convolutional neural networks overcome the problem of full-meshed networks by limiting the influence of single input neurons locally. Further, the residual feed-forward technique introduces higher-frequency earlier iterations of the input again at a later stage in the network to enable the network to re-introduce higher image frequencies without loosing efficiency, see Fig. 5. The U-Net architecture was originally developed for segmentation and pixel classification. Using a network originally meant for image segmentation [5], we adopted an approach with 256

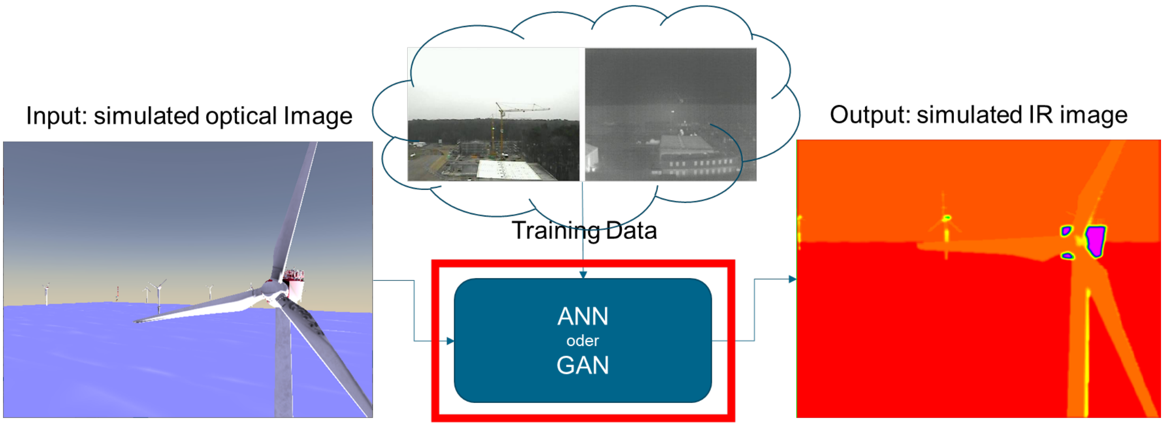


FIG 1. Workflow of generating synthetic IR images from real or simulated camera data

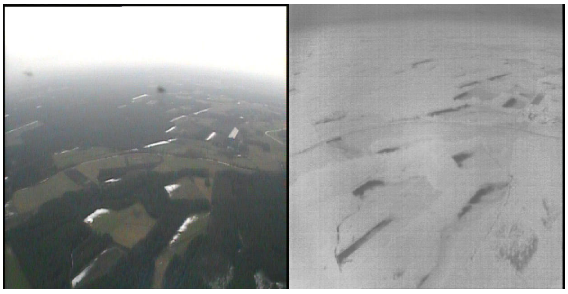


FIG 2. Example pair of an electro-optical camera image and an IR image

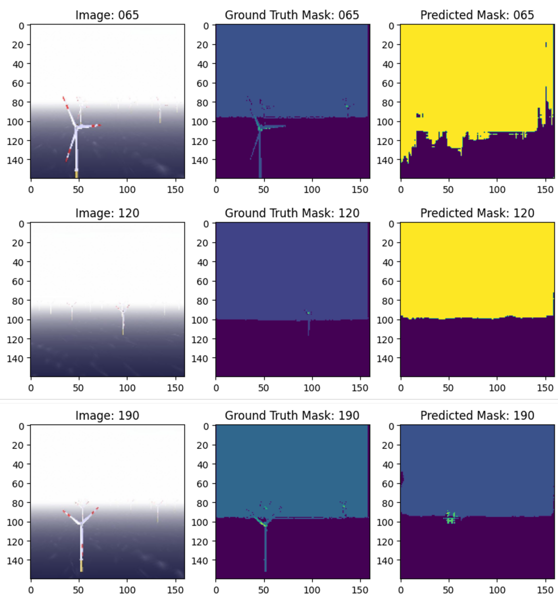


FIG 3. Training results for a discrete U-Net with ADAM optimizer

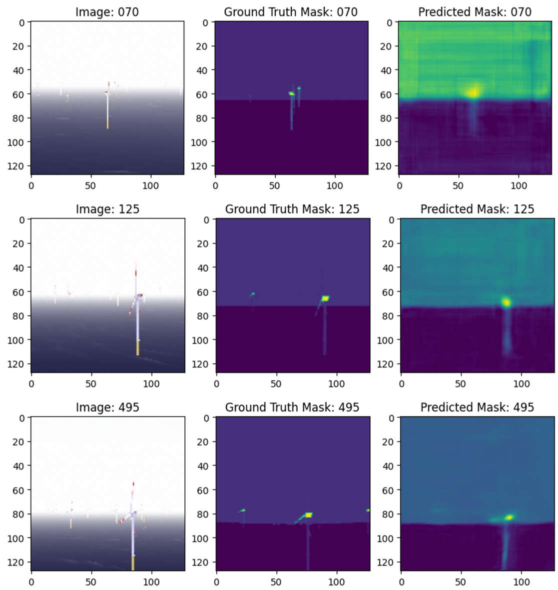


FIG 4. Training results for a floating-point U-Net with NADAM optimizer

"classes" corresponding to the grey levels of the IR image. As the loss-function binary cross-entropy was chosen, which is a standard for classificator networks, as the optimizer ADAM [6] was used. The result is depicted in Fig. 3. While the network quickly adopts the overall segmentation of sky and sea in the training set, bright heat spots are only coarsly approximated even after over 150 epochs. Therefore, a modified approach utilizing floating-point grey scales was implemented. The NADAM [7] approach was chosen for the optimizer, using Mean Square Error as the loss function. The result can be seen in Fig. 4. While the network quickly picks up the overall positions of brighter spots, finer details are not reproduced even after over 400 epochs of training.

- **GAN:** The term Generative Adversarial Network (GAN) describes a method for mutual training of two networks: The generator and the discriminator (see Fig. 6). In principle, since this is a training method GANs can be used with any network architecture. We combined it with the U-Net architecture from the previous iteration which was up to that point the most promising method. This yielded the best method in this trial. The combination U-Net with GAN was chosen as the final implementation.

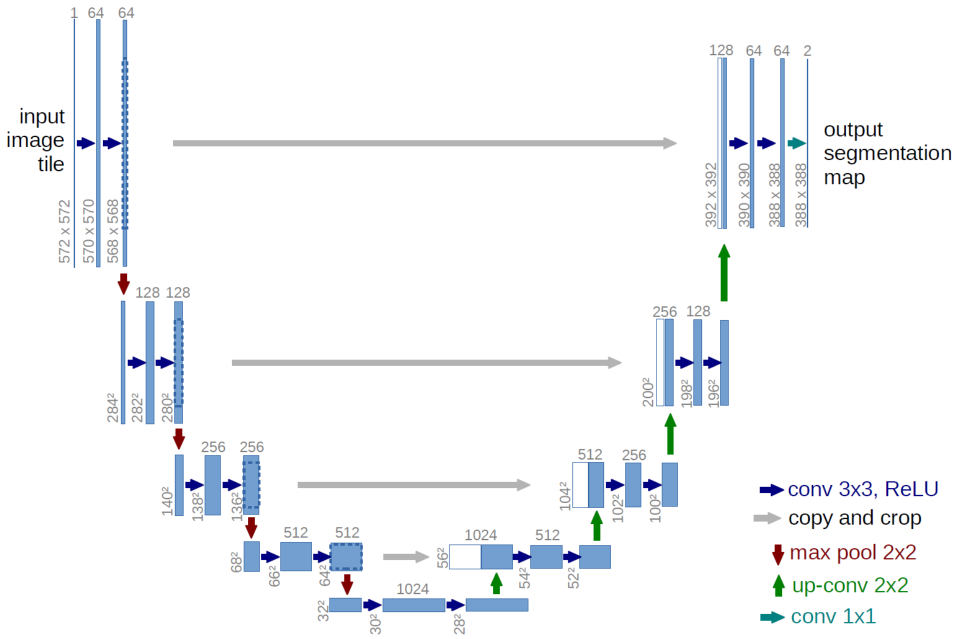


FIG 5. Architecture of a U-Net, adapted from [8]

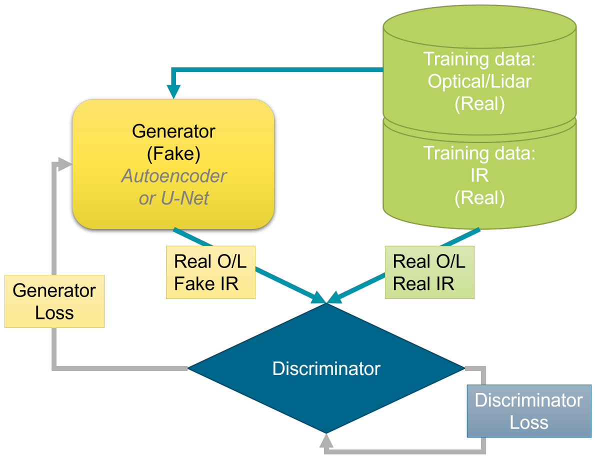


FIG 6. Principle of a GAN

The method implemented was based on the satellite-to-map technique developed by Nhu [9] which in turn was based on the well-known *pix2pix* technique by Isola et al. [10].

The method was implemented using a development environment in Python and/or Jupyter-Notebook. A combination of the following methods was used:

- Convolutional Network with Residuum-Feed-Forward ("U-Net"): Initially designed for segmentation and pixel classification, the U-Net was adapted to generate gray-scale IR images.
- GAN: The best method identified for training the network, utilizing a Generative Adversarial Network with mutual training of two networks (Generator and Discriminator).

4. RESULTS

Successful implementation of a functioning concept was achieved, which was then ported to a single Python module and executed on a multi-core computer with 96 threads. This allowed the training to be carried out for more than 90000 epochs in a reasonable time. Consequently, a higher fidelity of the generated images was achieved. Fig. 8 shows selected intermediate results from the training process. Note the red arrows highlighting visible deviations of the synthetic image from the ground truth image. These deviations disappear in higher epochs.

After training for 98000 epochs a sufficiently accurate imitation of infrared images from input optical images was achieved (Fig. 7). The resulting network could then be used to generate arbitrary infrared images from input camera images during a simulated flight. A result can be seen in Fig. 9: The original image on the left was not contained in the training set. Instead, it was taken from a different flight of the same campaign. One observes that several details like the snow line next to the street on the left and the snow covered roof on the right are reproduced (as dark structures) in the synthetic IR image. Further, the different camera pose from the training set is imitated properly.

5. CONCLUSION

It was demonstrated that a U-Net combined with a GAN is able to imitate an IR camera stream based on data available in a common flight simulation. The authors have yet to implement the demonstrated solution in a real setup to bench-mark the performance in daily use. Further, it seems obvious that the network would adapt to special cases of IR simulation only when trained with data from a single campaign. Thus, larger more diverse sets of training data would be needed to construct a solution viable for different simulation uses, e.g., on- and offshore flight missions or indoor use of IR cameras.

Although in this study some experiments made use of Lidar data as an additional input to the generator network it is not clear if this is really beneficial since in large parts of the training data Lidar data were either not available or objects were beyond the range of the sensor. Consequently, follow-up studies should investigate if depth data provide improvements when using synthetic input data.

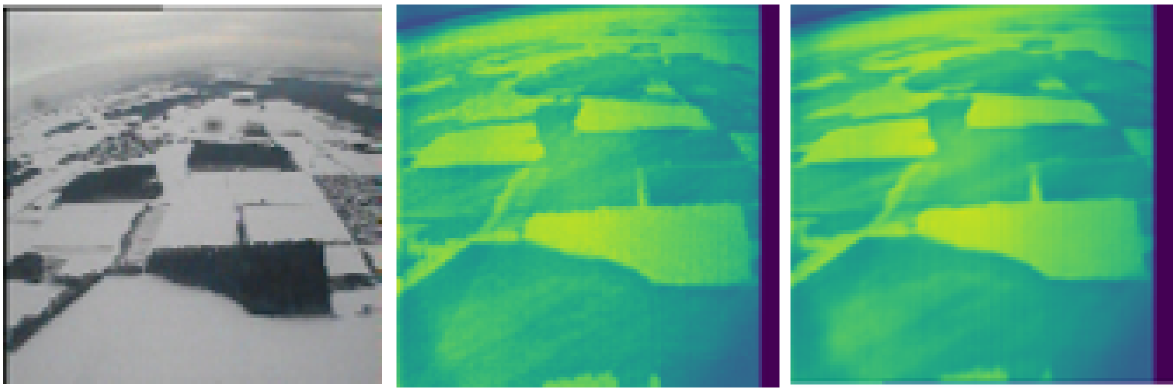
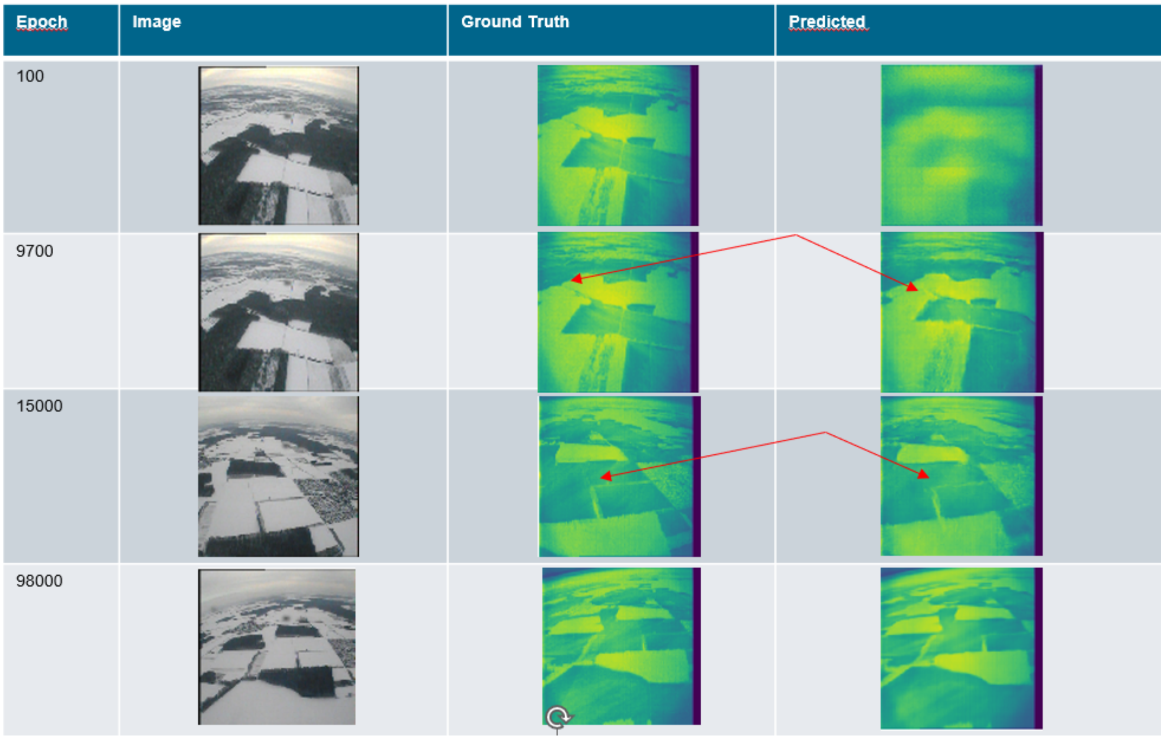


FIG 7. Input camera image (left), real IR image (middle), and synthetic IR image (right)



Epoch	Image	Ground Truth	Predicted
100			
9700			
15000			
98000			

FIG 8. Selected results from the training process in different epochs

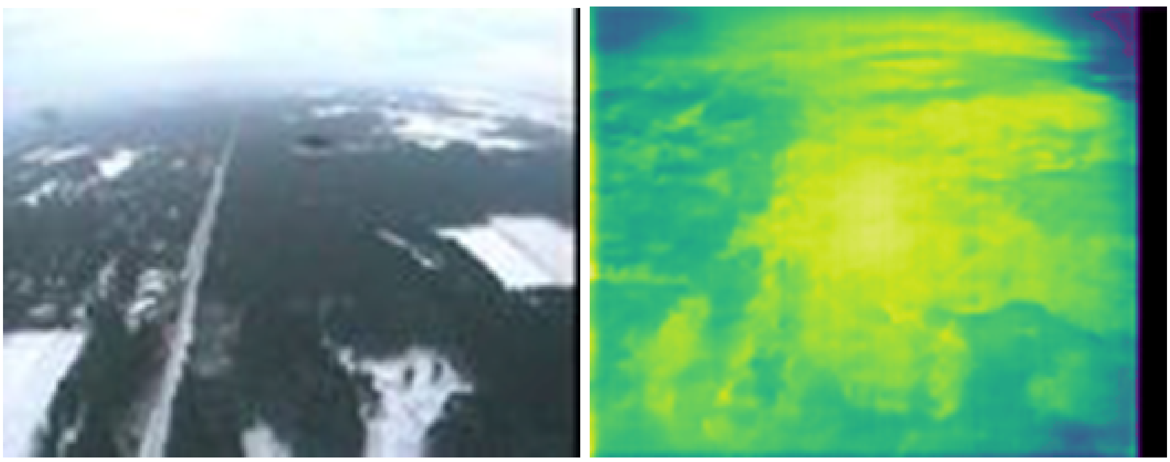


FIG 9. Input camera image (left) and synthetic IR image (right)

Contact address:

niklas.peinecke@dlr.de

References

[1] E.W. Drew, G.R. George, and S.N. Knight. Visionics simulation in the ah-64 combat mission simulator. In *Aerospace and Electronics Conference, 1988. NAECON 1988., Proceedings of the IEEE 1988 National*, pages 947–953 vol.3, 1988. DOI: 10.1109/NAECON.1988.195123.

[2] R. Codd-Downey, M. Jenkin, M. Ansell, H.-K. Ng, and P. Jasiobedzki. Simulating the c2sm 'fast' robot. In Noriaki Ando, Stephen Balakirsky, Thomas Hemker, Monica Reggiani, and Oskar Stryk, editors, *Simulation, Modeling, and Programming for Autonomous Robots*, volume 6472 of *Lecture Notes in Computer Science*, pages 26–37. Springer Berlin Heidelberg, 2010. DOI: 10.1007/978-3-642-17319-6_6.

[3] Niklas Peinecke. Simulating airborne sensors using graphics hardware. In *ODAS 2009*. DLR, October 2009.

[4] Robin Lantzsch, Steffen Greiser, Jens Wolfram, Johannes Wartmann, Mario Müllhäuser, Thomas Lüken, Hans-Ullrich Döhler, and Niklas Peinecke. ALLFlight: a full scale pilot assistance test environment. In *American Helicopter Society 68th Annual Forum*. Curran Associates, Inc., 05 2012.

[5] Francois Chollet. Image segmentation with a u-net-like architecture. *URL https://keras.io/examples/vision/oxford_pets_image_segmentation*, 2021.

[6] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[7] Timothy Dozat. Incorporating Nesterov momentum into ADAM. https://openreview.net/pdf?id=OM0jvwB8jIp57ZJjtNEZ, 2016.

[8] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.

[9] Anh N. Nhu. Satellite imagery to map translation using pix2pix-gan framework. https://github.com/anh-nn01/Satellite-Imagery-to-Map-Translation-using-Pix2Pix-GAN-framework.

[10] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *CoRR*, abs/1611.07004, 2016.