Machine Learning Engineer Task

CRT-A Team
Corporate Research & Technology | Algorithms
Carl Zeiss AG

2025–01–14

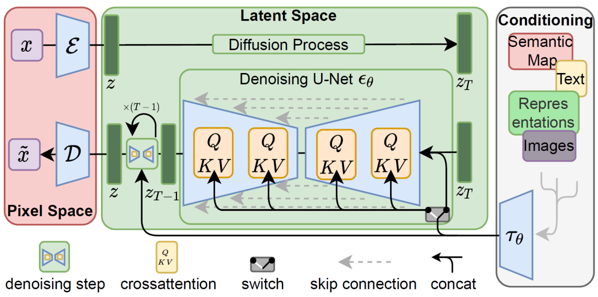
Introduction

The Corporate Reasearch & Technology (CRT) team, based in Oberkochen, Jena, and Munich, drives innovative, customer-focused research for ZEISS business units. Our team of scientists and engineers works with cutting-edge technologies in an inspiring environment. Read more about our work on ZEISS website:Artificial Intelligence in ZEISS, AI in Industry 4.0, Docker in ZEISS Microscopy or read some of our papers: N2V2 [1].
We seek passionate machine learning engineers to develop next-generation products and algorithms. As part of a collaborative team, you will advance the entire machine learning lifecycle, turning concepts into real-world applications. You'll work across the ZEISS product portfolio, applying advancements in machine learning, computer vision, and optical metrology. Together, we'll enhance codebases and infrastructure, ensuring stability and scalability.
In this role, you'll research, develop, and share best practices, contributing to a thriving ZEISS machine learning community. You'll also build a strong network in academia and industry, leveraging technology to address future challenges.
This task evaluates key expertise areas essential for a CRT engineering role.

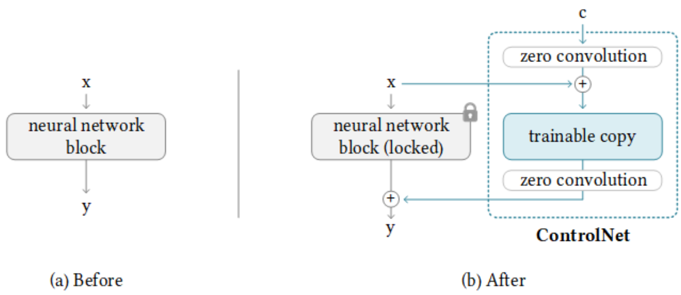
Your Task

In what context is this task?

Generative AI is undoubtedly one of the most significant and transformative technology trends today, fundamentally altering our approach to solving a wide range of problems. Beyond large language models (LLMs), another branch of generative AI focuses on creating images from text or other prompts. Over the past few years, image-domain generative AI has showcased remarkable capabilities, largely thanks to the development of diffusion models.
Diffusion models [2] are specialized neural networks trained to reverse the so-called "forward diffusion process". This process involves gradually adding random noise to an image until it loses all recognizable structure and becomes pure noise. By reversing this process, diffusion models can generate high-quality, realistic images starting from an initial image of random noise. This is achieved through a series of iterative denoising steps using the trained neural network. Initially, diffusion models simply generated samples from the data distribution. However, the technology has since evolved to condition the image generation process on text prompts (as seen in Stable Diffusion, Fig. 1a) or more specific image prompts (illustrated in ControlNet [3], Fig. 1b).



(a) Stable Diffusion conditioning of LDMs either via concatenation or by a more general cross-attention mechanism. [2]



(b) ControlNet adds a "trainable copy" connecting it to original network with zero convolution layers

Figure 1: Example images of Stable Diffusion and ControlNet

In the upcoming task, you will harness GenAI to enhance ZEISS imaging technology, specifically by generating on-demand synthetic data to support machine learning applications in manufacturing and medical domains.

What is the usecase?

Your teammate developed a demo based on the open-source project: ControlNet. Specifically, this can generate synthetic data and includes a file called **awesomedemo.py**, which showcases the feasibility of the technology. One of ZEISS's Business Units is excited about this and would like you to set up a small demo for them to explore it's potential in their production systems.
In order to start a demo on your local or any remote machine follow these steps:

1. Clone ControlNet repository and create a virtual environment.

```
$ git clone https://github.com/lllyasviel/ControlNet.git
$ cd ControlNet
$ conda env create -f environment.yaml
$ conda activate control
```

2. Download a model from Huggingface. In particular put control_sd15_canny.pth (5.71GB!) in ControlNet **.models** folder.

3. put the **awesomedemo.py** in the root of the repository, and copy input image into *test_imgs*

```
$ mv engineering/code/awesomedemo.py ControlNet/
$ mv engineering/img/mri_brain.jpg ControlNet/test_imgs/
```

4. Run the demo:

```
$ cd ControlNet/
$ python awesomedemo.py
```

The code will use ControlNet to generate realistic MRI Brain scans, like the ones shown on Fig. 2

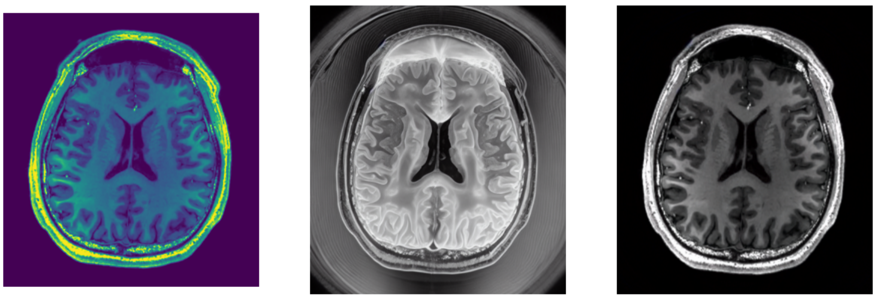


Figure 2: MRI brain scan images generated by **awesomedemo.py**

Note, you will need up to *7GB* of disk space on your machine for the models and code.

What do we want you to do?

You will be leading this project coordinating the work of multiple scientists and engineers that will contribute to and enhance this codebase in the future. Your task is as follows:

First, analyze the use case. As part of this, identify challenges that may arise in adopting the technology and explore any additional applications this technology might have for ZEISS. We are interested in your ability to lead technical projects. The deliverable for this analysis should be **one slide or one A4 page** written explanation of your ideas.

Second, plan the process for model training and feature development. Assess available codebase and consider what you would change or add to it. This also includes addressing aspects such as the overall code architecture, the git development workflow, the infrastructure needed for training models, ensuring reproducibility of experiments, handling retraining requests, and managing issues like input data or concept drift. We would like to take into account state of the art best practices for development in python programming language. The deliverable for this step should also be a **written explanation** of your ideas.

Lastly, create a small deployment demo to enable our business unit to generate synthetic images through REST API calls. This involves serializing the model, containerizing the application, and developing a REST API demo that allows image generation via POST requests. For the coding part feel free to use any available open source project with suitable license (e.g. python based full-stack-fastapi-template), and just point out which code was written by you. Additionally, feel free to compromise on image quality in this part and focus more on model deployment aspects. The deliverable for this part should include both **written explanation** of your ideas and **code with documentation**.

We recognize that the range of solutions is vast, so feel free to make any assumptions in your responses. We encourage you to aim for concise and clear answers while ensuring that key aspects are thoroughly addressed. Please note that we expect you to dedicate approximately **5-6 hours** to this assignment. You can submit your answers via E-Mail or share a repository with us.

Good luck! We look forward to seeing your results!

References

[1] Eva Höck, Tim-Oliver Buchholz, Anselm Brachmann, Florian Jug, and Alexander Freytag. N2v2 - fixing noise2void checkerboard artifacts with modified sampling strategies and a tweaked network architecture, 2022.

[2] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *CoRR*, abs/2112.10752, 2021.

[3] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023.