Project Report

Machine Learning Engineer Task

Iyer, Sneha

July 11, 2025

Contents

1 Analysis of ControlNet tool 2

2 Model training and feature development 2

References 3

1 Analysis of ControlNet tool

Diffusion models in the context of machine learning can be used to generate training datasets by adding noise to images, combining text prompts and images as inputs. Further improvements to the application of the model have been made using tools like ControlNet. In the Zeiss imaging sector, diffusion models can be employed in denoising and image reconstruction to enhance imaging quality. They can also be utilized for inpainting, which recognizes missing pixels in images. Additionally, diffusion models can be applied for audio enhancement.

In the context of pharmaceutical and computational biology applications, diffusion models can be used to identify protein sequences [3]. Although ControlNets have advantages in generating images without prompts, they rely on a robust pre-trained model, particularly at low dataset sizes. Therefore, the scope of outputs is dependent on the pre-trained model, which can pose challenges in multimodal data integration [1]. The interpretation of text prompts can remain subjective and may alter image expectations, potentially not meeting the target.

Compared to computer vision techniques that introduce image variations through simple effects like shifts or blurs to generate training datasets, the computational time and resources required for using ControlNets are substantial. Moreover, generating images using text-to-image prompts can raise ethical issues within organizations due to the discriminatory nature and bias of the pre-trained model.

2 Model training and feature development

The current codebase needs to be modified according to OOP principles to improve code sustainability. To enhance the training of the model and ensure code reproducibility and retraining, I have deployed an automation pipeline that uses the Hydra package (hierarchical configuration setup) to manage various input prompts, model parameters, and result comparison modes between lab, YUV, and luminance. Users can now set parameters in a config file to run the pretrained model in serial or parallel, facilitating experimentation with different parameters. Outputs for each run are recorded according to a unique process ID.

The code can be improved to add different input images through initial parameters that define the type of input image to be generated. As future work, project managers like Hatch can be used to create a reproducible build ecosystem. Regarding the Git development workflow, pre-commit hooks can ensure good static analysis checks, such as linting, type hinting, docstrings, and code formatting that adheres to PEP8 standards before updating the project. Unit tests and integration tests can be conducted using pytest to improve code sustainability.

Furthermore, cloud AutoML tools like Microsoft Azure AutoML can be used for model tracking and analysis, which can be integrated into existing Azure DevOps pipelines. Concept drifts can be monitored using AutoML tools by tracking the Frechet Inception Distance (FID), CLIP text-to-image scores, and aesthetic scores [2]. In this task, I have assumed that the tests are conducted purely through user studies, for which output tracking is necessary and has been implemented during code formatting.

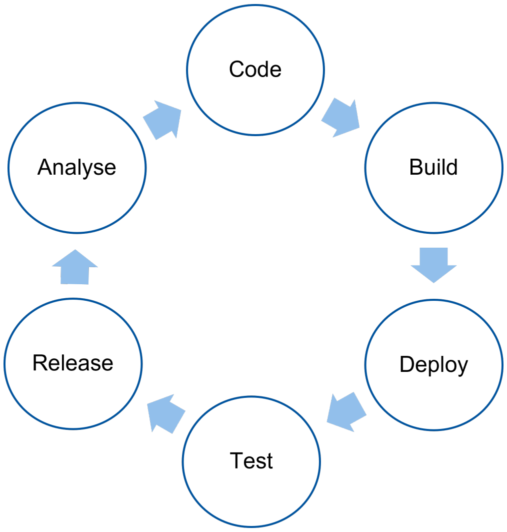

Figure 1: Software development life cycle for a scientific process

Bibliography

References

[1] Z. Guo, J. Liu, Y. Wang, M. Chen, D. Wang, D. Xu, and J. Cheng, “Diffusion models in bioinformatics: A new wave of deep learning revolution in action,” 2023. [Online]. Available: https://arxiv.org/abs/2302.10907

[2] L. Zhang, A. Rao, and M. Agrawala, “Adding conditional control to text-to-image diffusion models,” 2023. [Online]. Available: https://arxiv.org/abs/2302.05543

[3] M. Chen, S. Mei, J. Fan, and M. Wang, “An overview of diffusion models: Applications, guided generation, statistical rates and optimization,” 2024. [Online]. Available: https://arxiv.org/abs/2404.07771

ChatGPT prompts

1. Brief summary of diffusion models in machine learning
2. Hydra tool compare to automl tools
3. Improve the code to suit .yaml config
4. config_method = cls(config) config_method.get_config_params(config) config_vars = dict(vars(config_method)) return set_dataclass(config_vars, ConfigData) how to return a class object or how to retrieve the data object
5. add docstrings
6. correct grammar in writing