Artificial generated images for Lymphoma diagnosis by fine-tuning the popular Stable Diffusion model

Kuijpers Nick, Diego Perazzolo

Abstract

In this project, we address the challenge of limited training examples in medical image classification, focusing on the classification of lymphoma. We propose an algorithm that generates artificial images derived from the original dataset to augment the dataset and improve classification accuracy. The algorithm utilizes a fine-tuned stable diffusion model to create synthetic images that exhibit similar characteristics to the original lymphoma images. These artificial images, along with the original dataset, are used to train a convolutional neural network (CNN) for lymphoma classification. Additionally, we explore the effectiveness of classical data augmentation techniques such as Discrete Cosine Transform (DCT) as a benchmark for comparison. Extensive experiments are conducted to evaluate the proposed approach, comparing the performance of the CNN trained on the original dataset alone with the performance of the CNN trained on two combined datasets: one consisting of the original and artificial images, and another comprising the original and DCT-augmented images. We anticipate that the integration of artificial images will enhance classification accuracy and demonstrate the potential of this data augmentation technique in addressing the limited training examples challenge in medical image classification.

Introduction

Data augmentation techniques are widely used in image classification tasks to enhance the training dataset and improve the generalization capability of models. One such technique is Discrete Cosine Transform (DCT), commonly employed in image compression but also applicable for data augmentation purposes.

DCT-based data augmentation involves applying the Discrete Cosine Transform to an image, manipulating the transformed coefficients, and then applying the inverse DCT to generate the augmented image. This process introduces variations while preserving the visual content of the image. As part of our research, we created two benchmark datasets using two DCT methods.

To develop a Convolutional Neural Network (CNN) model for our specific classification task, we used transfer learning, a popular technique in deep learning. Transfer learning allows us to leverage pre-trained models that have learned general features from large-scale datasets, saving computational time and resources.

We utilized the well-known AlexNet architecture as the base model and trained it on three different datasets: the original dataset, the dataset augmented with classical DCT techniques, and the dataset augmented with artificially generated images using the fine-tuned stable diffusion model. The first two training scenarios served as benchmarks to assess the performance of our model trained on the artificially generated images.

In GANs, a large generator network is trained on a dataset to produce images. Another network, the discriminator, will have to distinguish between real and fake images. Training involves providing both real and fake images to the discriminator, which learns to differentiate them. However, GANs can be challenging to train due to issues like mode collapse, where the generator produces the same image repeatedly, and the lack of incentive for the network to generate diverse and interesting images.

In diffusion models, noise is gradually added to the initial image at each step, following either a linear or non-linear schedule. The training algorithm aims to train a network capable of reversing the noise addition process and recovering the original image. Rather than directly predicting the original image, the network predicts the noise added at each step, allowing the estimation of the original image by subtracting the predicted noise from the noisy image. This iterative estimation process progressively reduces the noise until an image resembling the original is achieved.

Diffusion models become more complex when guiding the generation process using additional information, such as captions. By conditioning the network on such information and leveraging techniques like transformer embeddings similar to those used in GPT models, it becomes possible to direct the generation towards specific concepts or ideas.

Execution

To adapt the pre-trained AlexNet model to our classification problem, we made modifications to the final layers. Specifically, we replaced the last fully connected layer with a new layer that aligns with the number of classes in our dataset, enabling the model to generate predictions for the target classes. Additionally, we added a log-SoftMax activation layer to normalize the predicted class probabilities.

To expedite the training process and prevent overfitting, we froze the parameters of the pre-trained layers, focusing on updating the parameters of the newly added layers. We employed an Adam optimizer, a popular choice for optimizing neural networks, and utilized the negative log-likelihood loss criterion suitable for multi-class classification tasks.

During training, we performed iterations over a specified number of epochs, conducting both training and validation steps within each epoch. We computed the loss and accuracy for each batch, updating the model's parameters accordingly. After each epoch, we calculated the average loss and accuracy for both the training and validation sets.

The trained model, along with the training history encompassing loss and accuracy metrics, was saved for further analysis and evaluation. This resulting model can be employed for inference on new images, facilitating accurate predictions for our classification problem.

A Stable Diffusion model consists of several models that contribute to its functionality:

- Text Encoder: This component projects the input prompt into a latent space.
- Variational Autoencoder (VAE): The VAE projects an input image into a latent space, which acts as a vector representation of the image.
- Diffusion Model: This model refines a latent vector and generates another latent vector, conditioned on the encoded prompt.
- Decoder: generates images given a latent vector from the diffusion model.

We found that fine-tuning the diffusion model can be performed in two ways: either by directly fine-tuning the diffusion model itself or by employing a technique called textual inversion. Textual inversion involves learning a token embedding for a new text token while keeping the remaining components of Stable Diffusion frozen.

We chose to implement the second solution, which involves updating only the parameters of the diffusion model while keeping the pre-trained text and image encoders frozen. This approach ensures that the learned representations and features captured by the encoders are preserved. The fine-tuning process follows a series of steps:

1. The input text prompt is projected into a latent space using the text encoder, while the input image is projected into a latent space using the image encoder of the VAE.
2. A small amount of noise is added to the image's latent vector at each timestep. Leveraging latent vectors from both the text and image spaces, along with a timestep embedding, the diffusion model predicts the noise that was added to the image latent vector.
3. By calculating the reconstruction loss, which measures the disparity between the predicted noise and the original noise added in the previous step. The diffusion model parameters are optimized using gradient descent.

Results

I conducted training experiments using a pre-trained CNN network on two datasets: the original dataset and an augmented dataset consisting of artificially created images. I fine-tuned the diffusion model using the DiffusionFineTuning.ipynb script, which was executed on Google Colab due to its computational requirements.

Due to limited resources, I encountered challenges when fine-tuning stable diffusion. The computations could only be performed on a V100 or A100 GPU, which I could do only once on one epoch. Because of extensive testing and resource utilization, I had to resort to training on a T4 GPU available on Google Colab for free once my calculation units were utilised.. Regrettably, this GPU lacked sufficient RAM memory to train the network effectively. Consequently, the generated images were produced by an undertrained network. Ideally, the network should have been trained for 40 to 60 epochs for optimal results.

The results of the CNN training, both with and without the fine-tuned stable diffusion-generated images, are presented below. Unfortunately, my teammate was unable to complete the implementation of the DCT (Discrete Cosine Transform) for the CNN, and therefore, his results are not included. However, they did contribute the two methods for DCT images, namely method1_DCT and method2_DCT.

Base dataset		Loss	Accuracy (%)
Epoch 1	Training	1.2144	34.49
	Validation	1.1670	48
Epoch 2	Training	1.0146	50.67
	Validation	0.9904	54.67
Epoch 3	Training	0.9267	55.18
	Validation	0.9801	48
Epoch 4	Training	0.8704	56.19
	Validation	0.9946	44
Epoch 5	Training	0.8630	60.87
	Validation	0.9118	56

Stable diffusion dataset		Loss	Accuracy (%)
Epoch 1	Training	1.0912	41.65
	Validation	1.0360	45.33
Epoch 2	Training	0.99511	51.16
	Validation	0.9583	54.67
Epoch 3	Training	0.8504	59.89
	Validation	0.9083	57.33
Epoch 4	Training	0.7985	62.72
	Validation	0.8196	66.67
Epoch 5	Training	0.8082	61.70
	Validation	0.9506	49.33

Upon analyzing the results, it is evident that the CNN's performance on the original dataset is unsatisfactory. This outcome was anticipated due to the dataset's limited number of images. However, the CNN exhibits a slightly improved performance when trained on the augmented dataset, with the accuracy reaching 66% at epoch 4, compared to a maximum accuracy of 56% with the base model. It is worth noting that this improvement is likely due to random variations, and repeated runs of the base model might yield similar results.

The results highlight the challenges faced when training on limited datasets and the potential benefits of utilizing artificially generated images. However, further investigations and experiments are required to establish the significance and reliability of these findings.
At appendice 1, nine images are portrayed which were generated from the fine-tuned stable diffusion model. As imagined, these images are too far away from our base image and cannot provide a reliable support as a data-augmentation technique.

Conclusion

In this project, we tackled the challenge of limited training examples in medical image classification, focusing specifically on lymphoma classification. We proposed an algorithm that generates artificial images derived from the original dataset to augment the dataset and improve classification accuracy. Our approach involved utilizing a fine-tuned stable diffusion model to create synthetic images that exhibit similar characteristics to the original lymphoma images. These artificial images, along with the original dataset, were used to train a convolutional neural network (CNN) for lymphoma classification.

To evaluate our proposed approach, we conducted experiments, comparing the performance of the CNN trained on the original dataset alone with the performance of the CNN trained on the datasets consisting of the original and artificial images. Due to deadline conditions (19/06) we could not put into practise our last implementation of a DCT image augmented dataset.

Despite encountering resource limitations, we were able to perform training experiments using pre-trained CNN networks. However, due to hardware constraints, we could only fine-tune the diffusion model on a limited number of epochs. This resulted in an undertrained network and suboptimal performance. Ideally, a more extensive training regime consisting of 40 to 60 epochs would have yielded better results.

The results demonstrated that the CNN's performance on the original dataset was unsatisfactory, as expected. The CNN exhibited a slight improvement when trained on the augmented dataset, with the accuracy reaching 66% at epoch 4, compared to a maximum accuracy of 56% with the base model. It is important to note that this improvement is likely due to random variations, and further experiments and iterations may yield similar outcomes with the base model alone.

Although my teammate was unable to complete the DCT implementation for the CNN, we acknowledge the potential of classical data augmentation techniques like DCT as a benchmark for future comparisons and investigations.

Furthermore, we recognize that there is potential for further improving our results by exploring additional fine-tuning techniques for the stable diffusion model. While we focused on fine-tuning the diffusion model itself, another approach worth investigating is fine-tuning a model that has already

undergone fine-tuning with textual inversion. I did found a fine-tuned stable diffusion model with textual inversion for the prompt “microscopic”, which if we had more time would have been a great addition.[1]

Contributions to this project

Nick :

- Wrote this paper
- Wrote MainCNN.py
- Wrote ArtificialCNN.py
- Wrote generateArtificialImages.ipynb
- Wrote DiffusionFineTuning.ipynb

Diego :

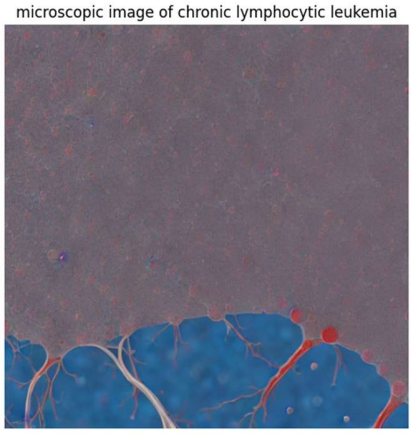
- Wrote Method_1_DCT.py
- Wrote Method_2_DCT_variant.py
- Wrote workFile_DCT_tests.ipynb

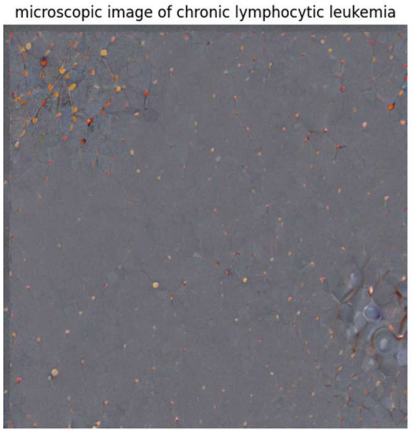
[1] https://huggingface.co/Fictiverse/Stable_Diffusion_Microscopic_model

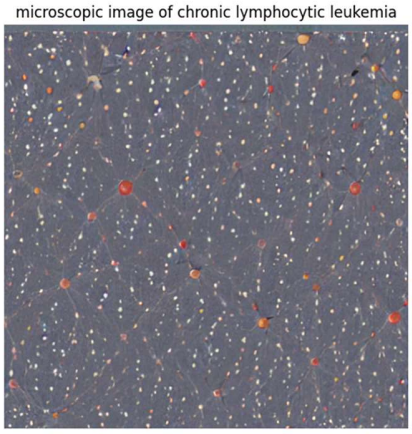
Appendice

1) Images generated with fine-tuning stable diffusion on one epoch.

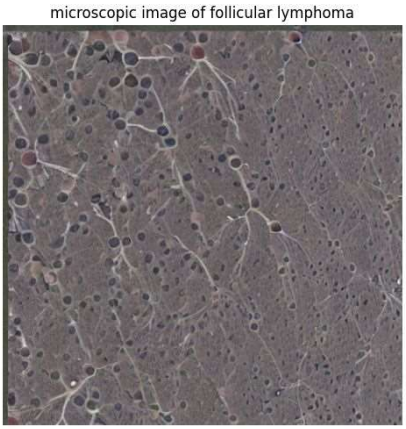
Label 1 :

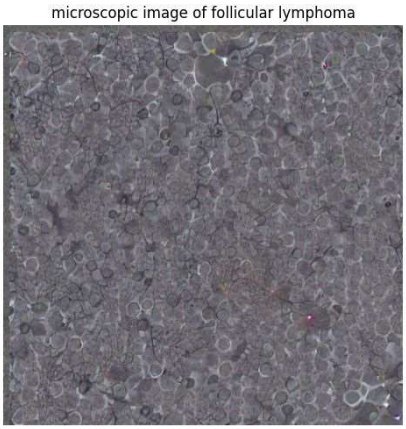


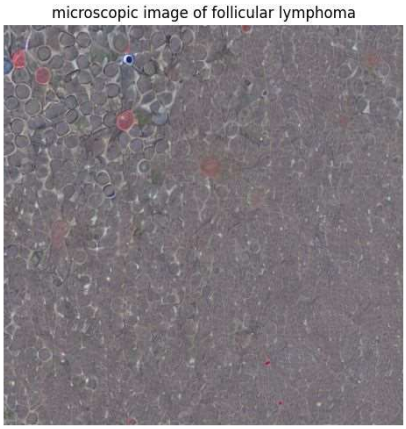




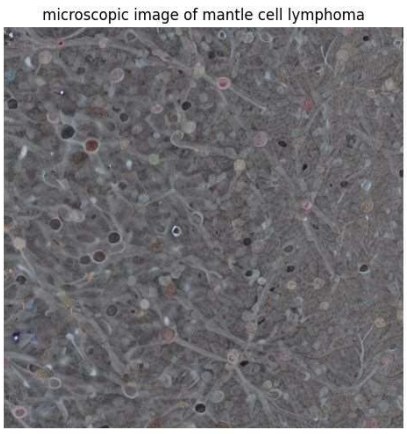
Label 2 :

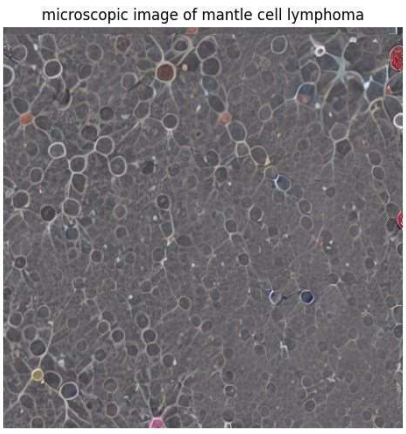


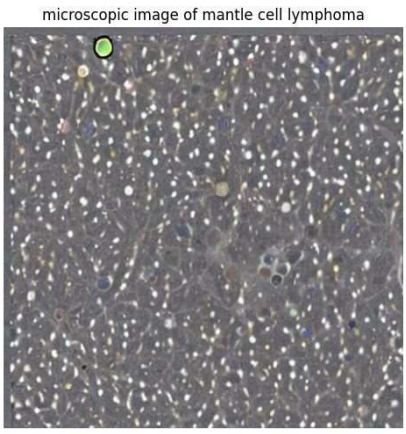




Label 3 :







References

All the work that was included in the original project handout and :

- LI, Dongguang, BLEDSOE, Jacob R., ZENG, Yu, et al. A deep learning diagnostic platform for diffuse large B-cell lymphoma with high accuracy across multiple hospitals. Nature communications, 2020, vol. 11, no 1, p. 6004.
- MA, Boyuan, WEI, Xiaoyan, LIU, Chuni, et al. Data augmentation in microscopic images for material data mining. npj Computational Materials, 2020, vol. 6, no 1, p. 125.
- MA, Jingchao, HU, Chenfei, ZHOU, Peng, et al. Review of Image Augmentation Used in Deep Learning-Based Material Microscopic Image Segmentation. Applied Sciences, 2023, vol. 13, no 11, p. 6478.
- NAGHIZADEH, Alireza, XU, Hongye, MOHAMED, Mohab, et al. Semantic aware data augmentation for cell nuclei microscopical images with artificial neural networks. In : Proceedings of the IEEE/CVF international conference on computer vision. 2021. p. 3952-3961.
- YANG, Ling, ZHANG, Zhilong, SONG, Yang, et al. Diffusion models: A comprehensive survey of methods and applications. arXiv preprint arXiv:2209.00796, 2022.
- FRID-ADAR, Maayan, DIAMANT, Idit, KLANG, Eyal, et al. GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification. Neurocomputing, 2018, vol. 321, p. 321-331.
- DHARIWAL, Prafulla et NICHOL, Alexander. Diffusion models beat gans on image synthesis. Advances in Neural Information Processing Systems, 2021, vol. 34, p. 8780-8794.
- Data synthesis and adversarial networks: A review and meta-analysis in cancer imaging, Medical Image
Richard Osuala, Kaisar Kushibar, Lidia Garrucho, Akis Linardos, Zuzanna Szafranowska, Stefan Klein, Ben Glocker, Oliver Diaz, Karim Lekadir, Analysis,Volume 84,2023,102704,ISSN 1361-8415,https://doi.org/10.1016/j.media.2022.102704 (https://www.sciencedirect.com/science/article/pii/S1361841522003322)
- SAHARIA, Chitwan, CHAN, William, CHANG, Huiwen, et al. Palette: Image-to-image diffusion models. In : ACM SIGGRAPH 2022 Conference Proceedings. 2022. p. 1-10.