9/18

Read paper on weakly supervised learning model using WeCLIP. Also downloaded dataset to practice training a small model locally to make sure code works.

Paper looks like they take prompt and image and by pipelining two models they perform image segmentation based on the text. Idk what “freeze” the input means but I will likely use their code on github to train a model in october.

Train deeplab v3 on new cancer set

Iou and f1 score evaluation for both sets

Generate diffused images for 2 ultrasound datasets

10/8

Tried running Huang’s code, first had to change cv2 functions to PIL functions. Then code kept on crashing in the middle of training first epoch, I think because of gpu issues?

Retrained original breast cancer model with k-fold implemented on training with 5 folds. Also increased to 25 epochs. This reduced Iou to 0.4917, but had an f1 score to 0.6289. Just worse performance so I'm gonna work on this.

10/9

Tried a different method of training improvement. Loaded data but on each image also loaded a randomly flipped/rotated version to increase dataset size. This allowed me to increase batch size to 32 as well. Training on this for 50 epochs got an average iou of .5369 and an F1 score of .6758.

Changing the batch size to 8 and 25 epochs got an iou of 0.6129 and f1 of 0.7536

10/10

Yesterday apparently I was training on cpu, got gpu working again today.

Trained a second breast cancer model on the new set with same transforms (doubling dataset size) and batch size of 4 with 50 epochs, got average iou of \_\_\_\_\_\_ and f1 of \_\_\_\_\_\_\_\_\_ (smaller batch size needs more epochs for convergence).

10/11

Get professor’s code to work. Performance of the second dataset is worse but should not be a big gap.

Get diffusion to work (not easy):

<https://github.com/mueller-franzes/medfusion/tree/main>

Get questions for code.

10/13

Got professor’s code to work, printing a new line for each batch is annoying but whatever we’re training. Gonna run it on both datasets. Kept all hyperparameters unchanged, just replaced cv2 operations in “dataset.py” with PIL. GPU also being successfully utilized locally so that’s good news!

Results for usg set:

Iou = 0.6125

F1 = 0.753

Actually similar result to my code.

Results for busi set:

Iou = 0.6974

F1 = 0.7806

Questions:

What does this code do to make better performance, random flips and rotations to vary?

10/18

Optimizer changed with learning rate of 10^-4 and scheduler shrinks the optimizer because at end of training we don't want to move very much

Dam optimizer use smaller learning rate

Diffusion model

Step 1:

Train embedder and save the VAE and visualize the output

Step 2:

Train the diffusion to generate image from noise

For this week get the embedder to work and visualize the output!

10/23

Got vae training code almost wokring

10/25

For november report:

Iou and recall table for the two datasets

Train latent embedder and generate some images

10/26

Reinstalled cudnn libraries, gpu now detected in virtual environment

Refactored some functions

Disabled multiple optimizers causing problems with pytorch lightning

Started training, each epoch taking about 30s

Now I need to figure out how to use the model and save some resulting images

10/27/2024

Sorted out embedder evaluation script. Had to modify some code:

Put evaluation calculation portion in \_\_main function (prevents whole script from rerunning during multiprocessing)

Num\_workers 8 to 0 (to resolve infinite loop)

Fixed transform lambda (they’re not picklable??? So switched it to regular function)

11/1/2024

Generate 1000 synthetic images

Use mean teacher method to train deeplab using part images with labels and part non

11/12/2024

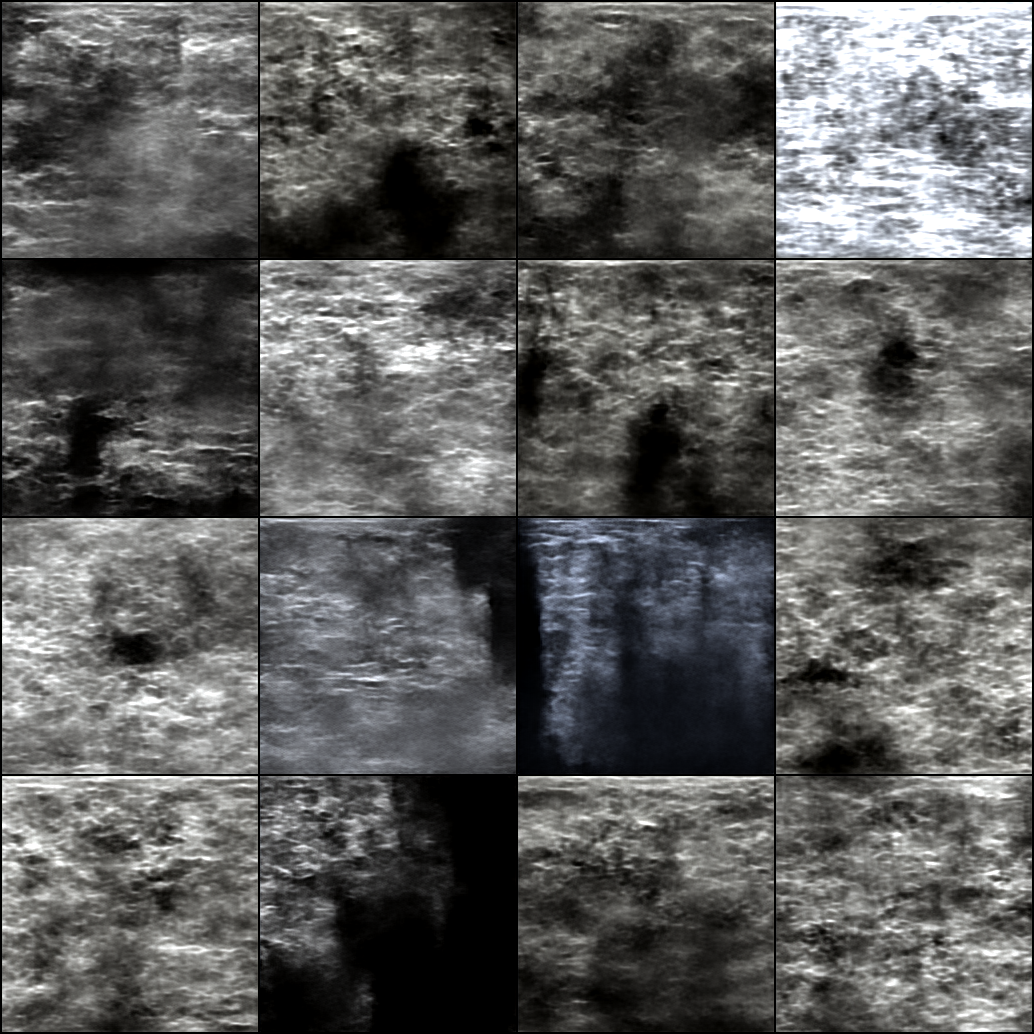
Training diffusion troubleshooting:

Had to comment out auto\_lt\_find = false because of some error

Removed optimizer\_idx from step function in diffusion pipeline

Both of those were just issues with pytorch lightning version but after that it started training!

Preliminary results are okay, need to ask professor in scripts/sample.py what diff, 0, 1, and none outputs are



11/15

Refined image set to only include images and labels with visually obvious tumors (hand picked)

Retrained embedder on this set for 50 more epochs totalling 150

New embedder metrics are as follows:

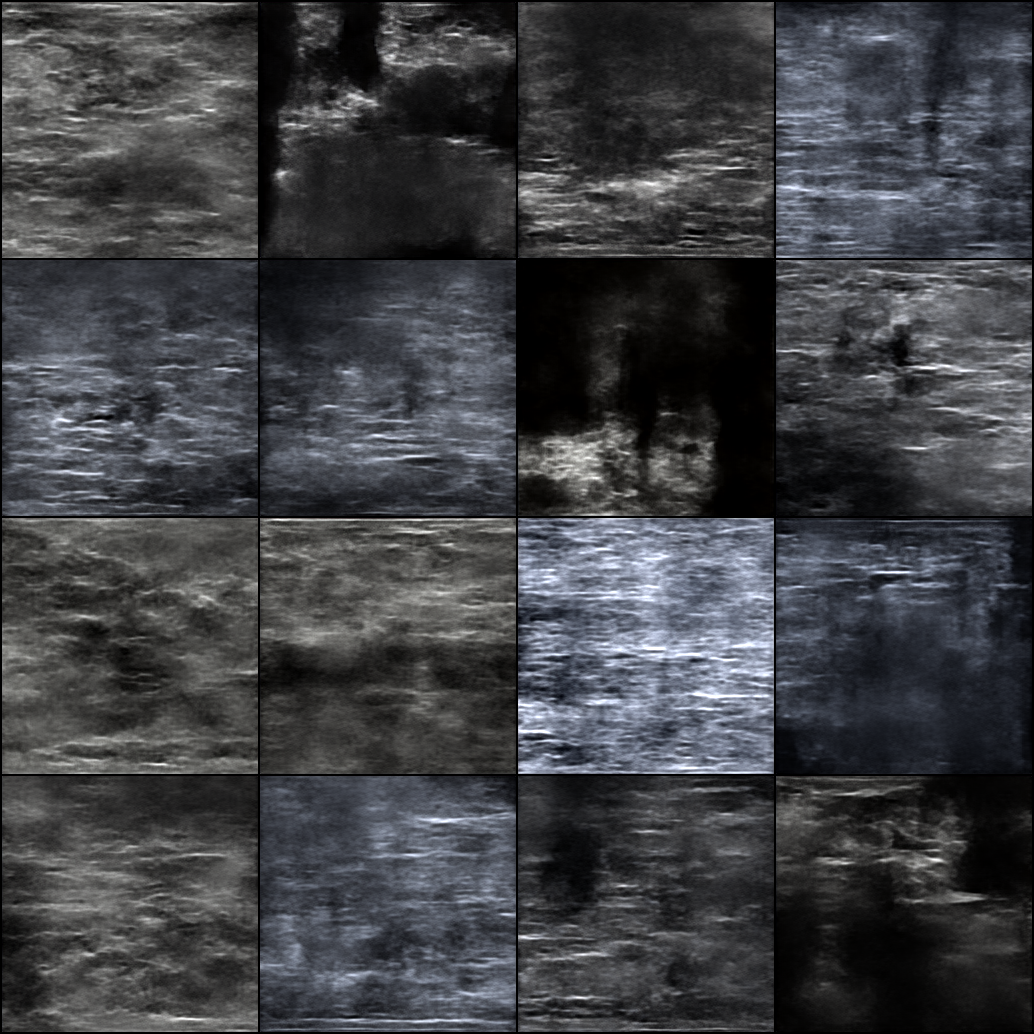
LPIPS Score = 0.8364

MS-SSIM = 0.7643 +- 0.2214

MSE = 0.0034 +- 0.0043

Trained diffusion using refined embedder for 150 epochs

Resulting test images:



Metrics on the first embedder were better

Meeting:

Look up how to refine diffusion

Generate 1000 synthetic images

Train original model using mean teacher method (script given)

Can set condition in the diffusion model

11/17

Trying to set up environment on kean gpu cluster server

Iniutal models have been based on USG dataset, have to refactor stuff to use the BUSI dataset

Had to create train and test excels with random mix of labels for embedder dataset to use, too complex to change to use image crawler

11/20

Doing a bunch of stuff to get mean teacher script working without generated images, have to wrap non function code in a main function because windows multiprocessing error, similar issue with previous hunag’s scripts, also had to modify the dataset classes because they were doing some weird stuff with the paths and then the label

Got mean teacher to “work” (the code runs smooth) and got individual diffusion images generated

Troublwe generating images do to gpu memory, going to batches of 100. This worked!

11/21

Went on a wild goose chase because dataloader for unlabeled images didn’t include the batch size and was throwing everything off. Solution ended up being to just return an empty list in the dataset as a “label” for the unlabeled images.

Training is cooking on nohup process, will test later.

True and predicted masks loaded weird asf to test script will need to look at tf goin on there, hopefully it’s not to do with training mean teacher script.

11/22

Training embedder on the BUSI set now with 1000 epochs before meeting, later today will evaluate and train diffusion 1000 epochs. Also need to fix test script foreman teacher model.

From meeting need to:

Generate better images

Train mean teacher on BUSI

Train mean teach on USG

Compare to old model performance

Embedder’s performance is better:

LPIPS: 0.8617

MS-SSIM: 0.8772 sd 0.2184

MSE: 0.0026 sd 0.0056

Training diffusion error now so had to add line to trainer instantiation arg for “strategy”?

Running smooth now sort of, many warning messages but is training.

11/23

Fixing warnings and errors:

Installed lightning[extras]

Switch logging sample every 2 because 100 is larger than num batches

Needed to sync logs because strategy change so added param to pipeline

Okay so I think I was just confused on the logging, the best checkpoint is not the “last” checkpoint but the one that is stated in the best checkpoint json log, uh duh doi, plus there is a variable to save specific number of checkpoints in a run and just set that to 4, plus modified batches so it’s not 2 batches per epoch we’re now working with 16, probably better for training, we’ll see later today if it generates better images.

11/24

Got better images generated, now running mean teacher on BUSI, will get test script working later. Batch size 16 was garbage in epochs so upped to 32 and restarted. Still plateauing quite quickly.