1. Who are you (mini-bio) and what do you do professionally? If you are on a team, please complete this block for each member of the team.

   I'm a freelance data scientist. I regularly participate in data competitions, particularly those focussed on solving problems of significant social impact.

2. What motivated you to compete in this challenge?

   In 2020, I participated in the TissueNet competition hosted on DrivenData and the PANDA challenge on Kaggle. I saw this challenge as an opportunity to expand on my previous work and make a meaningful contribution to cancer research.

3. High level summary of your approach: what did you do and why?

   Baseline Model:
   I initially trained a standard CNN classifier on the entire slide to directly predict the chance of relapse. The model's learning quickly plateaued after training for just a few epochs, and performed poorly on the test set. This suggested that the model was overfitting and potentially failing to capture meaningful features that are crucial to determine chance of relapse.

   Combining MLP and CNN:
   A Multilayer Perceptron (MLP) model trained only on the patient's metadata performed poorly. However, combining it with the baseline model improved performance on the test set. Notably, incorporating ulceration and breslow features significantly boosted local performance, but this data wasn't available in the test set.

   Pretraining:
   Next, I experimented with pre-training the CNN model with breslow/ulceration as the target variables and then fine-tuning with relapse as the target. The idea was to help the model learn relevant features from the initial task and transfer that knowledge to relapse prediction. While the local performance gain from this approach was modest, the model generalized better on the test set, indicating effectiveness of this two-step strategy.

   Multiple Instance Learning Approach:
   Building upon recent digital pathology research in Whole Slide Images (WSI) classification, I framed the problem as a Multiple Instance Learning (MIL) task. This involves dividing each WSI into smaller patches, processing them individually with a CNN feature extractor, and aggregating the extracted features to classify the entire WSI.

   To extract tiles from the WSIs, I used the same code that I had previously used in the TissueNet competition (https://github.com/CODAIT/deep-histopath). This algorithm processes WSIs at a lower resolution to create a mask of regions with tissue. It then

divides the image into equal sized tiles and selects tiles from the masked region, assigning a score to each tile based on its color density and tissue quantity. This effectively filters out low quality tiles, ensuring that only the most useful tiles in the WSI are selected for further processing. Once the tile locations are known, regions of interest can be extracted from higher resolution pages of the WSI pyramid without the need to process the entire slide at high resolution, thus minimizing computational resources.

During analysis of the provided expert annotations, I realized that the selected tiles might not fully capture factors that determine melanoma severity, i.e. breslow and ulceration. This was primarily due to the difficulty of selecting representative tiles. For instance, regions marked as ulcerated often occur on the edge of the WSI, which the tile selection algorithm might overlook due to their low tissue content. Additionally, calculating breslow depth requires observing the complete dermal component of the melanoma, which may not be adequately represented by a few choice tiles.

The solution was to select more tiles. Most slides had thousands of potentially useful tiles, but training on approximately 1 million images was not computationally feasible. To address this, I opted to use a pretrained model to extract features from the tiles. The extracted embeddings were then used to train a MIL model on breslow/ ulceration prediction tasks. This method retained an adequate level of representational power while overcoming the computational limitation of training on a large number of images.

I used the SparseConvMIL architecture (https://github.com/MarvinLer/SparseConvMIL), which employs sparse convolutions to preserve the spatial relationship of the tiles. The idea was to preserve structures across multiple tiles that might be useful in identifying ulcerations or calculating breslow.

Finally, out-of-fold predictions for breslow and ulceration were generated and, together with the patient's metadata, used to train an MLP model.

Ensemble Model:
My best solution was a mean ensemble of two MIL/ MLP models trained with tiles extracted from pages 2 and 3 of the WSI, and two CNN models trained on the full slides at different sizes.

4. Do you have any useful charts, graphs, or visualizations from the process?

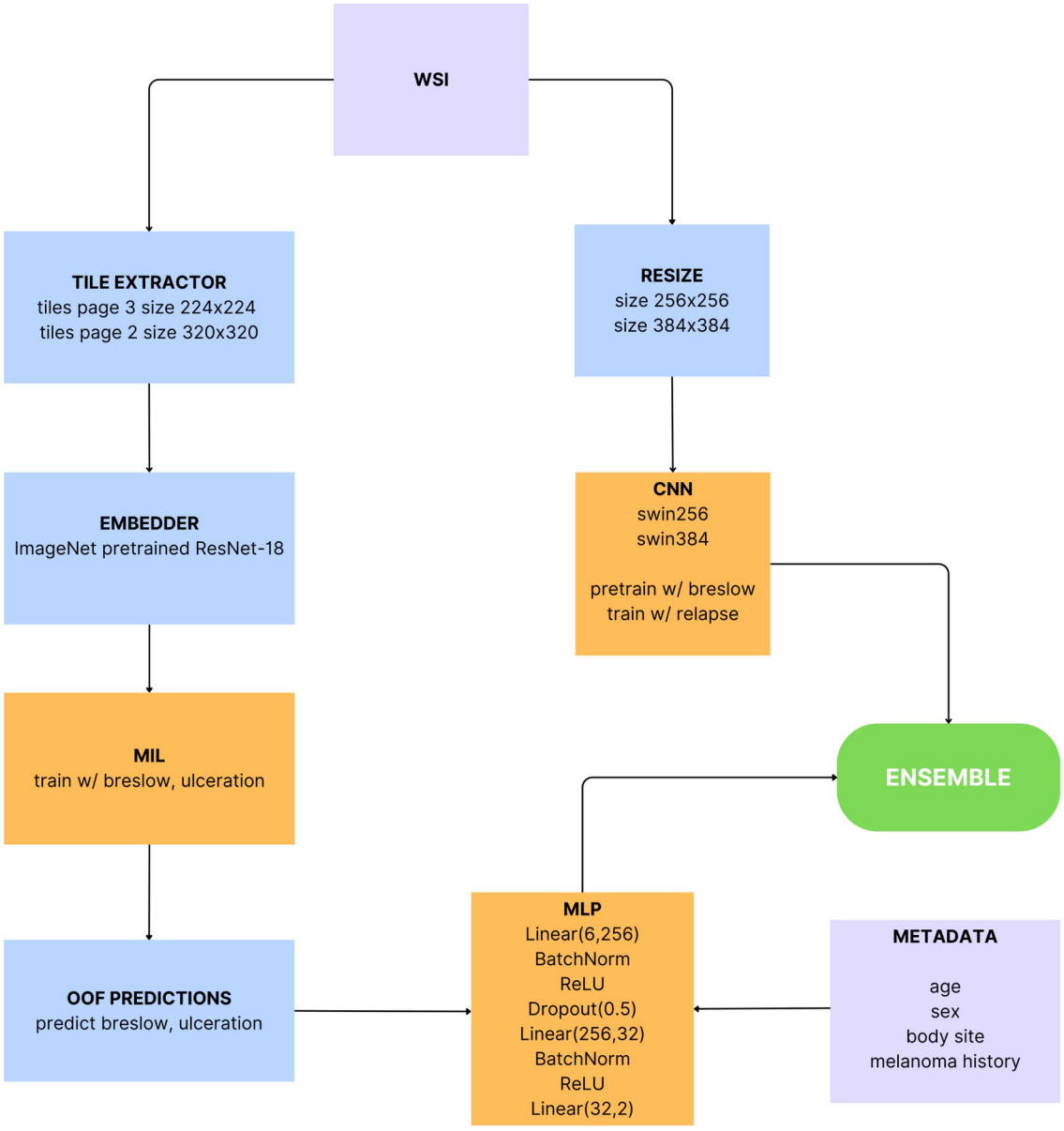

Figure 1: Training pipeline for melanoma relapse prediction model: preprocessing and tile extraction, feature extraction with a pre-trained model, MIL breslow and ulceration training, MLP and CNN training, ensembling for improved performance

	experiment	log loss	accuracy	AUC
1	swin image size 256 no pretraining	0.3669	0.841	0.787
2	swin image size 256 pretrained on breslow	0.3482	0.847	0.819
3	MLP (patient metadata)	0.4121	0.838	0.671
4	*MLP (patient metadata including breslow, ulceration) - local only*	*0.3302*	*0.857*	*0.837*
5	MLP (patient metadata, MIL generated OOF breslow & ulceration)	0.3395	0.855	0.830
6	final ensemble	0.3351	0.856	0.839

Table 1: Performance metrics for various experiments. Test data is not available for experiment 4 and it's only included to show the importance of breslow/ ulceration features

5. Copy and paste the 3 most impactful parts of your code and explain what each does and how it helped your model.

6. Please provide the machine specs and time you used to run your model.
   - CPU (model): Intel(R) Xeon(R) CPU @ 2.20GHz
   - GPU (model or N/A): Tesla P100
   - Memory (GB): 16
   - OS: Ubuntu 20.04.5 LTS
   - Train duration: ~10 hours
   - Inference duration: ~3 hours

7. Anything we should watch out for or be aware of in using your model (e.g. code quirks, memory requirements, numerical stability issues, etc.)?

8. Did you use any tools for data preparation or exploratory data analysis that aren't listed in your code submission?

9. How did you evaluate performance of the model other than the provided metric, if at all?
   All models were trained with early stopping after 5 epochs to prevent overfitting. MSE Loss was used to track breslow score in the pre-training phase.

10. What are some other things you tried that didn't necessarily make it into the final workflow (quick overview)?
    - k-means tile selection - clustering tiles of a WSI and selecting the most representative tiles per cluster
    - laplacian blur detection to eliminate the most blurry tiles in each WSI - worked with mixed results

- stain normalization, adversarial augmentation in an attempt to bridge the performance gap between local cv and the leaderboard

- self-supervised histopathology SimCLR trained model (https://github.com/ozanciga/self-supervised-histopathology) - This model is trained on various histopathology datasets that include different types of cancer. I expected embeddings extracted from the model to work better than the ImageNet pretrained model but the performance was comparable or slightly worse.

11. If you were to continue working on this problem for the next year, what methods or techniques might you try in order to build on your work so far? Are there other fields or features you felt would have been very helpful to have?

    - advanced blur detection technique (e.g. CNN trained on blurry vs non-blurry tiles)
    - advanced stain augmentation
    - image segmentation approach
    - resolution - select pages/ rescale tiles based on resolution
    - My focus was on preprocessing and I spent little time tuning the model. Finding the best learning rate/ scheduler/ optimizer, using a better feature extractor, tuning the MLP architecture - all could further enhance results

12. What simplifications could be made to run your solution faster without sacrificing significant accuracy?
    I trained the MIL/ MLP models on CPU due to the limitations of my environment at the time. Switching to GPU could speed up training/ inference

13. (optional) Whole Slide Images can be challenging to work with due to their size and the significant variation in their contents. What techniques and/or tools did you find helpful for working with WSIs and why?