Progress Report

Jakob Balkovec
Thu Jul 24th 2025

1. Recap

The pipeline has been finalized for extracting patches. The pipeline does the bare minimum to extract the patches, this means there isn't any augmentation or preprocessing applied to the patches. The patches are extracted from the original images and saved in a directory structure that is suitable for training classification models.

The current focus shifted toward training classification models. This involves multi-label classification of four lesion types:

- microaneurysms (`MA`)
- hemorrhages (`HE`)
- hard Exudates (`EX`)
- soft Exudates (`SE`)

---

2. Dataset Composition

As noted before, the dataset is composed of patches extracted from the original images. The patches are labeled based on the presence of lesions. The labels are "multi-hot" encoded vectors where a `1` in a certain position corresponds to a lesion type.

The labeled dataset contains three categories:

- `lesion` patches (label vector with at least one `1` in any position)
- `healthy` patches (label vector: `[0, 0, 0, 0]`)
- `black` /uninformative patches (excluded from training)

Now due to the imbalance in the dataset, the original model performed poorly. I unfortunately did not keep track of the exact numbers, but the dataset was heavily skewed towards healthy patches, with very few lesion patches (see below).

Original Dataset

```
Healthy patches: 136237 | of those 'black': 23091
Label sums: [11993 28829 17647  3448]
```

Now due to this imbalance, and poor performance, I decided to "rework" the dataset. The goal was to create a more balanced dataset that would allow for better training of the classification models.

Here is what I was mostly focused on:

- Preserve lesion diversity
- Avoid overfitting to common lesions (`HE/EX`)
- Prevent collapse to predicting [0,0,0,0]

Strategy:

- Balanced lesion patches by undersampling to the smallest class count.
- Sampled 2× the lesion count in healthy patches to retain a strong negative signal.
- Resulted in ~balanced dataset with black patches excluded entirely.

Reworked Dataset

```
Healthy patches (after resampling): 47914
Label sums (after resampling): [11991 28805 17640  3432]
```

Healthy
Lesion

---

3. Baseline Model: Patch-Only Classifier

I started this with a shallow custom CNN model, but since that performed poorly, I switched to a more robust architecture. I chose a pre-trained ResNet18 model, which according to some papers is a common choice for image classification tasks.

Architecture

The backbone is a ResNet18 model pretrained on ImageNet, with the final layer modified to output four sigmoid activations for multi-label classification. I was initially computing the loss using a binary cross-entropy loss with logits, which in theory is suitable for multi-label classification tasks, but switched to Focal Loss to better handle class imbalance. I also used per-class inverse-frequency weights to dynamically adjust the loss contribution of each class. I used the Adam optimizer with a weight decay of `1e-5` .

Loss Weighting

I passed in per-class inverse-frequency weights (normalized) into the focal loss to tackle class imbalance more dynamically.

Configuration

I performed some simple data augmentation, since the model wasn't performing well at the start.

- `RandomHorizontalFlip` , `RandomRotation` , and `ColorJitter`

Here is what I think is happening:
I think my issue was that the model wasn't able to learn the difference between healthy patches and lesion patches. This happened because the pathces were extracted via a grid system, and were not centered around the lesion. This means that even if 1 pixel of a lesion is present in the patch, it's tagged as a lesion patch, but the model can't learn to differentiate between a healthy patch and that lesion patch. This is why the polygon approach might've been better, as it would have centered the patches around the lesions.

> **Note:** Again, this is just me speculating, since I don;t really know how to test for this, but I think this is the issue...

- **Threshold**: Fixed sigmoid threshold of 0.5 for multi-label binarization
- **Eval**: Micro/macro F1, per-class F1, Hamming loss, subset accuracy, precision, and recall

---

> **Note:** Note that this model is only being trained for 10 epochs, since I want to see if the model is able to learn anything at all. If it does, I will continue training it for more epochs...No reason to train it for more epochs if it can't learn anything, and waste time and resources...

4. Results & Observations

With `BCEWithLogitsLoss`

- Subpar performance, especially for rare lesions
- F1 scores for minority classes remained near 0
- Learning stagnated around Epoch 3–4

With `FocalLoss`

- Significant improvement in per-class F1, especially for hemorrhages and exudates
- Precision remained lower than recall, indicating over-prediction
- Soft exudates (SE) remain the hardest class to classify accurately for some reason

Example before Re-Sampling (Epoch 10/10)

```
# [MA, HE, EX, SE]
Epoch 10/10 | Loss: 0.1998 | F1: 0.3326 | Hamming: 0.0719 | Subset Acc: 0.7608
F1 per class: [0.0, 0.1683, 0.6883, 0.0]
```

Example (Epoch 5/20)

```
[TRAIN] Epoch 5/20 | Loss: 0.0572 | F1: 0.4534 | Hamming: 0.3285 | Subset Acc: 0.1543
[TRAIN] F1 per class: [0.2655, 0.5085, 0.7188, 0.3459]
[TRAIN] Precision per class: [0.1575, 0.337, 0.6957, 0.4461]
[TRAIN] Recall per class: [0.8448, 0.9722, 0.7435, 0.2825]

[VAL] Epoch 5/20 | F1: 0.4784 | Hamming: 0.2814 | Subset Acc: 0.2298
[VAL] F1 per class: [0.2793, 0.5036, 0.7208, 0.2679]
[VAL] Precision per class: [0.1775, 0.3419, 0.6926, 0.5882]
[VAL] Recall per class: [0.6538, 0.9553, 0.7689, 0.1734]
```

The Good Stuff

- `F1` is improving steadily across epochs, particularly for EX and HE.
- Class 2 (EX) is dominating performance (`F1` reaches `0.729` by epoch 3)
- Precision and recall are both strong (0.75+)
- Focal loss seems to be working, it's helping push harder on difficult examples.
- Capturing some signal in SE (class 3) in just 3 epochs, which was nearly silent before down sampling.

Issues and my Concerns

- Subset Accuracy is low (~0.1–0.17):
  - This is kind of expected with multi-label tasks, but worth tracking and noting
  - It could improve with threshold tuning per class later...
- Class 0 (MA):
  - Stagnates at ~0.25 F1. Precision is low (~0.15), recall is high (~0.85), which suggests:
    - Too many false positives, means the mModel is over-predicting MA.
- Class 3 (SE):
  - Extremely poor on epoch 1, some lift by epoch 3...
  - Still suffers from low recall, which could be due to:
    - Scarce examples
    - Weak visual patterns (probably the case)
    - Mislabeling or poor annotation in source patches (not likely, but possible)

> **Note:** I am running the model for 20 epochs, since `ResNet18` needs more epochs to converge, compared to the custom CNN model I was using before.

> **Note:** The F1 scores are still low, but they are significantly better than before. The model is able to learn something, and the F1 scores are improving over time. The precision is still low, which indicates that the model is over-predicting, but the recall is high, which means that the model is able to find most of the lesions.

---

5. Challenges

- There is a severe class imbalance, despite downsampling, some classes like soft exudates have far fewer samples. I can downsample further (1:1 ratio), but this would lead to a very small dataset.
- The model is overfitting, subset accuracy remains low despite rising F1, possibly due to noisy or borderline samples.
- Precision–Recall tradeoff. The model is prone to over-prediction; may benefit from threshold tuning or a recall-precision balancing scheme.

---

6. My Next Steps

> Planned

1. Global Context Modeling:
   - Encode entire fundus images using ResNet or Swin Transformer
   - Combine global and patch features (early or late fusion)
2. Fusion Strategy:
   - Concatenate or apply cross-attention on patch + image features
   - Experiment with attention pooling on patches
3. Threshold Tuning:
   - Optimize per-class sigmoid thresholds using validation F1/precision–recall curves
4. Training Extensions:
   - Incorporate semi-hard triplet mining or contrastive loss if patch similarity info becomes available

My other plan was to move straight to the image based model. I'm hoping that the global context will help the new model learn better, and therefore get better performance. On one hand I would like to build the framework for the hybrid model, and tune later, but on the other hand I want to have a solid baseline for the patch-only model before moving on...I'm still contemplating this, but considering the fact that I spent 12 hours trying to get better performance out of my patch-only model (solely by tunning it), I think it's time to move on to the next step.

I'll updated you with any progress I make on the next steps, and will keep you posted on the results of the image-based model.