**Method Overview**

This competition consisted of a dataset with weakly labeled images wherein every image scene contained multiple cells. Using image level labels as cell labels was reasonable for some classes where most of the cells expressed that pattern (e.g. nucleoplasm), while for classes that had a lower percentage of cell pattern expression per image (e.g. mitotic spindle), such labeling was less useful. Additionally, the dataset was extremely imbalanced. These are the main reasons why we opted for a data centric approach to this challenge (in contrast to the classic model centric approach). Our main goal was to create a better dataset with more accurate cell-level labels that would enable even simpler models to achieve highly accurate performance.

The training dataset was created using the 16-bit competition and public HPA images. Cells were segmented using the HPA-Cell-Segmentation segmentator, which generated nuclei and cell masks. All cell images were zero-padded to retain original height to width ratio and then resized to 512x512. All 4 channels were used.

Two heuristics were used for dealing with border images and outliers. Based on the nuclei segmentation masks, calculations were made for each cell in order to approximate how much of each cell is outside of an image scene. Using the cell segmentation masks, sums of cell input values were calculated for red, blue and yellow input channels. Outliers were detected by comparing the red channel sums and the product of the blue and yellow channel sums of each cell with a thresholder average of all cells from its image scene. Cells that were discarded by the first heuristic were ignored when calculating the average values.

To create a clean dataset, rigorous thresholds of the mentioned heuristics were used. The heuristics had a precision of roughly 50%, but a very high recall. In the end, around 20% of the cell images (generated with the mentioned segmentator) were removed from the final training dataset. Since there was an abundance of cell images, data quality was prioritized over data quantity.

A GUI was created for fast manual relabeling and consequently around 150 000 cell-level labels were manually relabeled. While relabeling, positive labels were graded with scores on a scale from 1 to 5 (or 1 to 3), which reflected the confidence that the positive label is correct. This was done for classes that were harder to predict. These scores were mapped to soft labels (e.g. 1 to 0.0, 2 to 0.2, 3 to 0.7, 4 to 0.9, and 5 to 1.0) which were used instead of the given image level labels. The soft label mappings were class specific. This process was done in a fast manner with the intention to focus on ruling out obvious incorrect cell-level labels.

The mitotic spindle class is specific. When present, the pattern is usually expressed only in a few percent of the cells in an image scene. All cell-level images that contained a positive mitotic spindle image-level label were manually relabeled. Around 250 cell-level examples of that pattern were found this way.

Since local validation was hard, a few thousand cell-level images were relabeled for most classes. This was done in a slower manner, but with higher precision. While this was a time-consuming process, it eventually led to better local validation. Around 30 000 cell-level labels were graded and again mapped to class specific soft labels.

With better local validation, it was possible to train better models and use them to further clean the current dataset. One ResNet18 with a single output and a sigmoid function was trained for each class separately. These models were then used to relabel almost all classes, but only positive labels were changed. Relabeling was done when the output of a ResNet18 model would be less than 0.3 for a label that was expected to be positive. Approximately 15% of the cells were relabeled this way. This has led to even better local validation.

By using the inverse approach, false negative examples of the mitotic spindle were detected. Around 100 examples of the mitotic spindle pattern cell-level images were relabeled this way.

All final models are from the EfficientNet family. Three EfficientNetB0 models as well as one EfficientNetB4 model were trained using Adam as an optimizer with either focal loss or binary cross entropy loss. Best models were selected based on local mAP score. Different augmentation techniques were used for training: random resizing, random padding, flipping (horizontal and vertical) and rotation. After augmenting, images were resized to 512x512 when needed. Resizing images to a smaller size and random padding on each side was used to train models on different cell sizes and resolutions. This was inspired by the fact that training images differed in image resolution and cell size.

After ensembling trained models, negatives were calculated by subtracting the maximum output value from the value one. Next, border and outlier cells output prediction scores were decreased based on two previously mentioned heuristics. In the end, the final predictions were weighted with the average cell-level predictions of an image scene, separately for each output label (e.g. 0.7 \* cell output + 0.3 \* average image scene cell output). More details about the final submitted models can be seen on **Supplementary Figure XX** and **Supplementary Table XX**.

**Ablation study**

The ablation study (**Supplementary Table XX**) shows that checkpoint ensembling had a negative effect on the final score. Although the models were trained on 16-bit images, converting all test images to 8-bit showed little impact on the final score. Cell weighting, border and outlier detection showed a positive effect on the final score. The biggest positive benefit was from ensembling.

**Conclusions**

The data centric approach played is the key component in the presented solution. Creating a better training dataset by using all available data, removing outliers and border images along with automatic and manual relabeling showed a bigger impact than training more complex models. The final dataset was still not perfectly labeled, which could explain the better generalization of simpler models in the presented approach. A single EfficientNetB0 model, trained on this dataset, resulted with a mAP score 0.53291 which alone would be enough for 11th place, while ensembling multiple models resulted with scores 0.54389 and 0.54361, both securing 4th place in this competition.