Cell Detection in Whole Slide Images With Out-of-Focus Corruption

Erikas Sokolovas
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands
e.sokolovas@student.utwente.nl

ABSTRACT

The widespread adoption of digital whole slide scanners in fields like histopathology has introduced computer assisted into whole slide image (WSI) analysis. Machine learning (ML) techniques have successfully automated laborious tasks such as cell detection in WSI analysis. Unfortunately, various types of image corruptions tend to be introduced during the WSI creation process such as out-of-focus (blurry) regions. Out-of-focus corruptions can degrade ML accuracy - decreasing the reliability of automated analysis. In the literature, convolutional neural networks (CNNs) have shown positive results in cell detection and focus quality assessment tasks. So far, no paper has combined both approaches for blur resistant cell detection in WSIs. This paper intends to combine both approaches for an accurate and robust cell detection method by developing a novel pipeline utilizing two differently trained models and a blur classifier. The pipeline was developed using two different CNN types: image segmentation (represented by Unet models) and object detection (represented by Yolov4). The novel pipeline showed no appreciable performance difference in the cell counting task between a control model trained on both in-focus and out-of-focus images. The output of image segmentation models required additional processing to derive cell counts, the method used for this step was naive and provided poor results. This meant that object detection based pipeline substantially outperformed the image segmentation based pipeline. However, the control models trained on both in-focus and out-of-focus images provided overall reasonable performance indicating that some degree of robustness against blur could be achieved through the inclusion of blurry images into model training datasets.

Keywords

WSI, Blur, Convolutional Neural Network, CNN, Yolov4, Unet, Cell Counting, Cell Detection

1. INTRODUCTION

Digital whole slide imaging allows for the creation of a high resolution digital image of a tissue sample (usually called a whole slide image or WSI), which can then be interacted with on a workstation computer as if it was viewed through a microscope [7].

One use case of WSI's are for histopathological analysis, that is tissue sample analysis for medical diagnosis. E.g. a cancer diagnosis or cancer sub-type can be established via WSI analysis of a tissue sample taken from a patient [25].

Digital whole slide imaging has seen substantial growth over the last couple of decades in fields such as histopathology. There are many factors contributing to the adoption of digital whole slide imaging such as: workflow improvements, cost savings, performance, access to services in under-resourced locations [7].

Besides the above benefits, WSI digitization also brings about the possibility semi-automated or fully-automated WSI analysis [7]. Semi-automated analysis might allow pathologists to mark areas of interest in the WSI to perform certain tasks automatically, such as cell counting and or identification [21], the results of the task may then be used by the attending pathologist to make diagnostic decisions. While in fully automated WSI analysis a (simplified) analysis pipeline might, for example, automatically identify regions of interests [15], perform cell identification and counting [21] in these regions and predict a diagnosis based on the results [22].

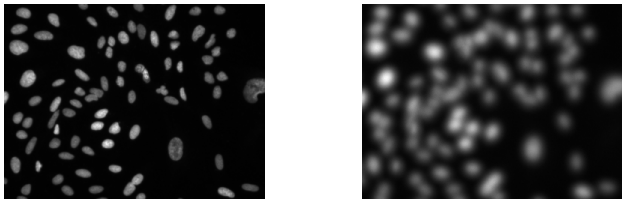
Convolutional Neural Network (CNN) based approaches have proven successful in automated WSI analysis - matching or even exceeding the accuracy of professional pathologists when asked to evaluate the same data set of whole slide images [25, 22].

However, during the slide (thin glass plate and covering to hold the sample) preparation and scanning process various batch effects (non-biological factors that affect the WSI) can be unintentionally introduced into the resulting WSI [5]. Batch effects can present themselves as various image artifacts: pen markings on the slide, cracked slide glass, contrast and hue variations, image blurriness, etc. [10]. This can make automated analysis by machine learning algorithms less accurate, as the algorithms can become biased if they have to model batch effects [5].

Blurring in WSI's can occur because sample tissue may slightly vary in height within a slide - this can lead to cells in the sample being noncoplanar [20]. To account for this a set of focus points at different focal planes (z-depths), which are properly aligned with the tissue height in the given region of the sample, are necessary to produce a sharp WSI [20]. If an incorrect focal plane for the height of the tissue in a given region of the sample is chosen - the entire region can become blurry in the final WSI [20]. An example of this type of blurring can be seen in Figure 1.

The current approach to dealing with blur artifacts in a

35th Twente Student Conference on IT July. 2nd, 2021, Enschede, The Netherlands.



(a) In focus cell nuclei (b) Out of focus cell nuclei

Figure 1: Example images from the Broad Bioimage Benchmark Collection image set BBBC006 showing the difference between (a) in-focus and (b) out-of-focus U2OS Hoechst stain images of cell nuclei [12].

WSI is to detect these blurry regions and to not evaluate the blurry regions [10]. If too many blurry regions are present - it might be necessary to discard the entire WSI [10]. But this approach can be problematic in certain pathological analyses. E.g., in lung adenocarcinoma histopathological analysis, the spacial distribution of cells throughout the tissue can convey important information [23]. In S.Wang et al. this is referred to as the tumor micro-environment [23]. If the blurry regions of the WSI are not numerous enough for it to be discarded but sufficiently pervasive to ignore important regions for analysis, it could lead to incorrect diagnoses. This example shows that the current method of dealing with blurry regions in WSI's of excluding blurry WSI regions from analysis can exclude important information from analysis. As such a different approach to WSI region analysis that is robust against blur is needed.

Several CNN models have been successfully applied in automated WSI analysis [25, 22]. These studies have demonstrated that CNN models can achieve performance on par with human pathologists when evaluating the same dataset.

There are models that have been shown to perform well in the cell detection and counting tasks, such as Unet [16] which produces an image mask that indicates the presence of a cell. The object detection model Yolov4 has also demonstrated good performance in the cell detection and counting task [11].

CNN based approaches to blur detection have been developed. C. Senaras et al. [20] developed a CNN that is capable of classifying WSI's as either in-focus or out-of-focus (binary classification). S.J. Yang et al. [26] developed a CNN capable of classifying a WSI patch into 11 absolute blur levels. Non-machine learning approaches also exist, such as the Laplace matrix approach used in HistQC [10], the laplace matrix approach provides a binary classification of in-focus or out-of-focus for a patch of a WSI.

An approach that is robust to blur was performed by J. Lu et al. [14] and was found to achieve performance on par with a human. However, The approach in J. Lu et al. relies on having multiple different WSIs of the same slide at different z-depths to construct the sharpest possible WSI for the classification task. The sharp WSI constructed by the following method: a cell nucleus is first detected at the central focus level (middle z-depth), the location of the detected nucleus is then used to cut out the same image coordinates in all the WSI's, then the average pixel value in each cutout is computed, this is then used to compute the variance between all neighboring (z-depth wise) cutouts. The pair of cutouts with the highest variance between is then selected, as the idea is that the largest variance between z-depths will be observed when an image

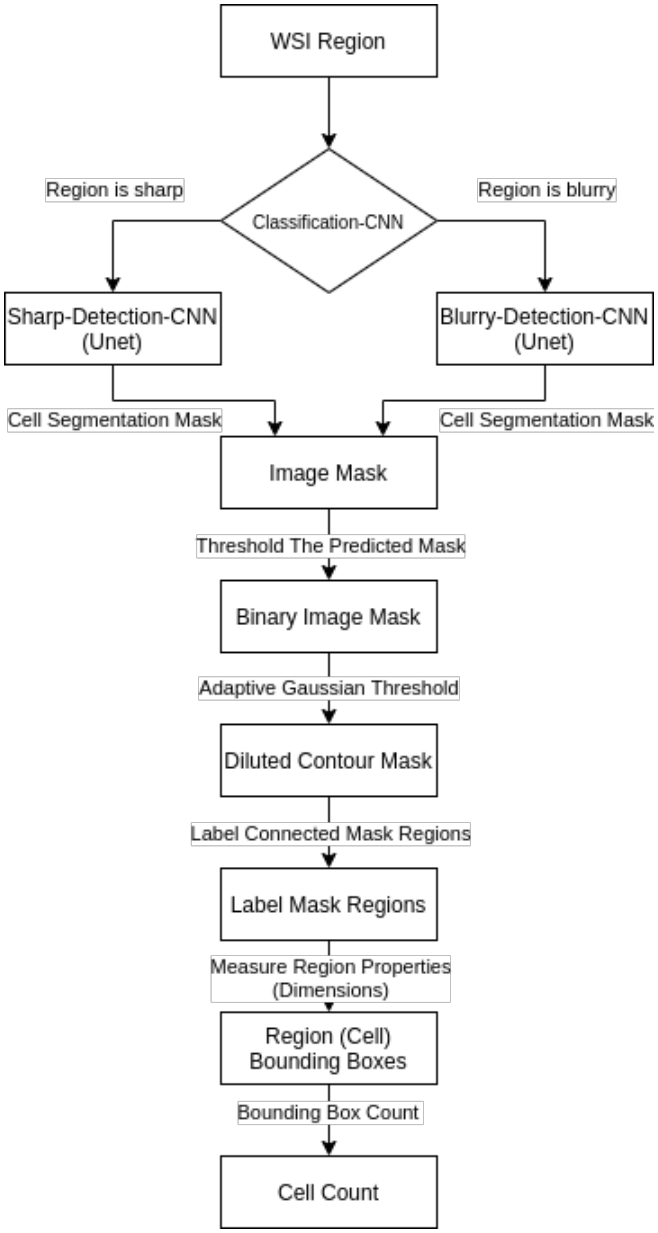


Figure 2: Image Segmentation Pipeline

moves between being in focus or being out-of-focus. Next, the in-focus image in the pair needs to be determined, a method developed in J. Guan et al. [8] called Edge Model based Blur Metric (EMBM) is used to determine which image in the pair is the in-focus one. The in-focus cutout then replaces the cell in the middle z-depth WSI. This process is then repeated for all detected cells. It should be noted that reconstruction is only done after cell nuclei detection to improve cell classification reliability, not detection reliability.

2. METHODOLOGY

Currently in the literature there exists an approach for blur robustness that relies on taking several images of the same slide at different z-depths [14]. While more resistant to blur artifacts, it makes the slide imaging more cumbersome due to the need of several WSI's of the same slide.

We propose an easier approach without the need for multiple WSI's of the same slide at varying z-depths. The proposed approach is a pipeline that can be described as such.

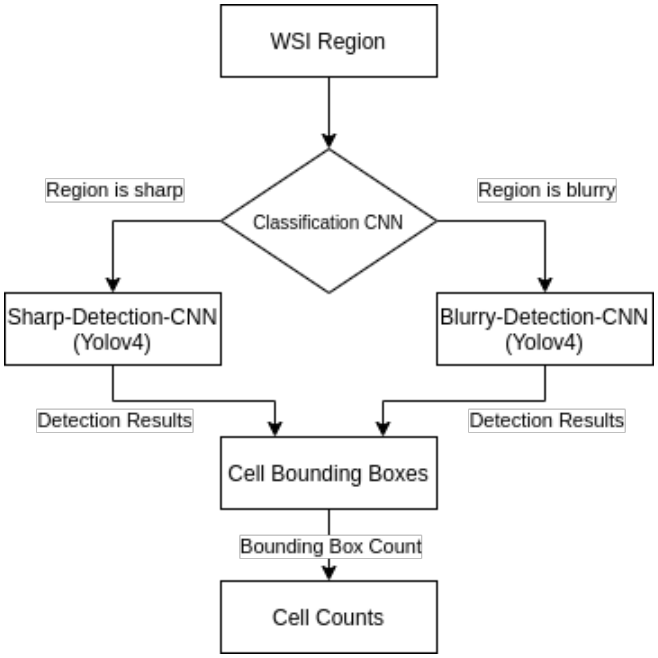


Figure 3: Object Detection Pipeline

First, a CNN that is capable of detecting blur in images will be used to classify the input-image as either blurry or sharp, this CNN will be referred to as the classification-CNN. Second, depending on the result of the classification-CNN one of two other CNN's will be chosen to (indirectly) predict the cell count of the image, these CNN's will be referred to as the detection CNN's. Each detection CNN model will be trained on a different dataset. The first detection CNN (referred to as the sharp-detection-CNN) will be trained with a dataset consisting only of images determined to be in-focus. The Second detection CNN (referred to as the blurry-detection-CNN) will be trained with a dataset consisting only of images determined to be out-of-focus. Which of the detection CNN's is used to predict the cell count in the image is decided by the classification-CNN, that is: if the classification-CNN classifies an image as sharp (in-focus) the sharp-detection-CNN will be used to predict the cell count in the input-image, or if the classification-CNN classifies an image as blurry (out-of-focus) the blurry-detection-CNN will be used to predict the cell count in the input-image. Consider the following example of the pipelines operation: consider an input-image A (an image of a region of a WSI), the input-image is first provided to the classification-CNN that processes the input-image and classifies it as either sharp or blurry, if the input-image was classified as sharp it is then given to the sharp-detection-CNN for cell detection, or if it was classified as blurry it is then given to the blurry-detection-CNN for cell detection, then the output of the detection-CNN is used to determine the number of cells in the image. It should be noted that the detection-CNN output may require additional processing to extract the cell count from the output of the detection-CNN depending on the type of model used for the detection-CNN's, e.g. a object detection CNN's outputs are bounding boxes- so the number of cells in the image can be derived by simply taking the number of bounding boxes that represent a cell class object, while with an image segmentation CNN, which outputs an image mask, may require additional processing to derive the number of cells in the image- extending the pipeline.

To avoid confusion further in the paper, the above proposed cell detection pipeline will be referred to as the **dual-model pipeline** from this point-onward.

Additionally, two different CNN types will be used for the detection-CNN's in the dual-model pipeline: object detection CNN (represented by Yolov4 model [1]) and an image segmentation CNN (represented by a Unet model [17]). To clarify, the CNN types (object detection, image segmentation) will not be mixed in the dual-model pipeline, but two different dual-model pipelines will be built using different model types, e.g. one using only the Yolov4 models for the detection-CNN's and a second one only using Unet models for the detection-CNN's. The diagram of the dual-model proposed pipelines can be seen in Figures 2 and 3.

In total six models will be trained: Yolov4-Sharp, Yolov4-Blurry, Yolov4-Combined, Unet-Sharp, Unet-Blurry and Unet-Combined. The -Sharp models will be used as the sharp-detection-CNN's in the dual-model pipelines, these models will be trained only with in-focus images. The -Blurry models will be used as the blurry-detection-CNN's in the dual-model pipelines, they will be trained only with out-of-focus images. The -Combined models will be trained with both in-focus and out-of-focus images and will act as a control group to compare the dual-model pipelines against to see if the dual-model pipeline is more robust against blur than a single model trained with both in-focus and out-of-focus images.

3. RESEARCH QUESTIONS

In this paper we seek to answer the following research questions:

1. What are the cross-evaluation (all models and all dual-model pipelines compared against all other models and all other dual-model pipelines) results?

2. Is the performance of the dual-model pipeline comparable to the performance of existing cell detection techniques?

3. How does the performance of the segmentation based model (Unet) compare against the object detection based model (Yolov4) performance?

4. EXPERIMENTS

4.1 Dataset

The BBBC006 dataset from the Broad Bioimage Benchmark Collection [13] was selected as the dataset to be used to train and validate the models and dual-model-pipelines as the BBBC006 dataset contains a set of WSI region images at 34 different z-depths (focus planes) with some z-depths being in-focus and others being out-of-focus. This allows for the sharp-detection-CNN's and blurry-detection-CNN's to be trained on essentially the same images except for the blur levels. Z-depths between 11-23 are considered to be in-focus, while z-depths of 0-10 and 24-33 are considered to be out-of-focus as a ground truth. Z-depth of 16 is considered to be the optimal focus plane. Each z-depth indicates a difference of 2m of the focal plane from neighboring z-depths, preceding z-depths are below the focal plane by 2m, while succeeding z-depths are 2m above the focal plane for a given z-depth. The dataset also provides ground truth cell counts for the WSI images and ground truth cell segmentation masks are also provided, an example of WSI image and its corresponding mask can be seen in Figure 5.

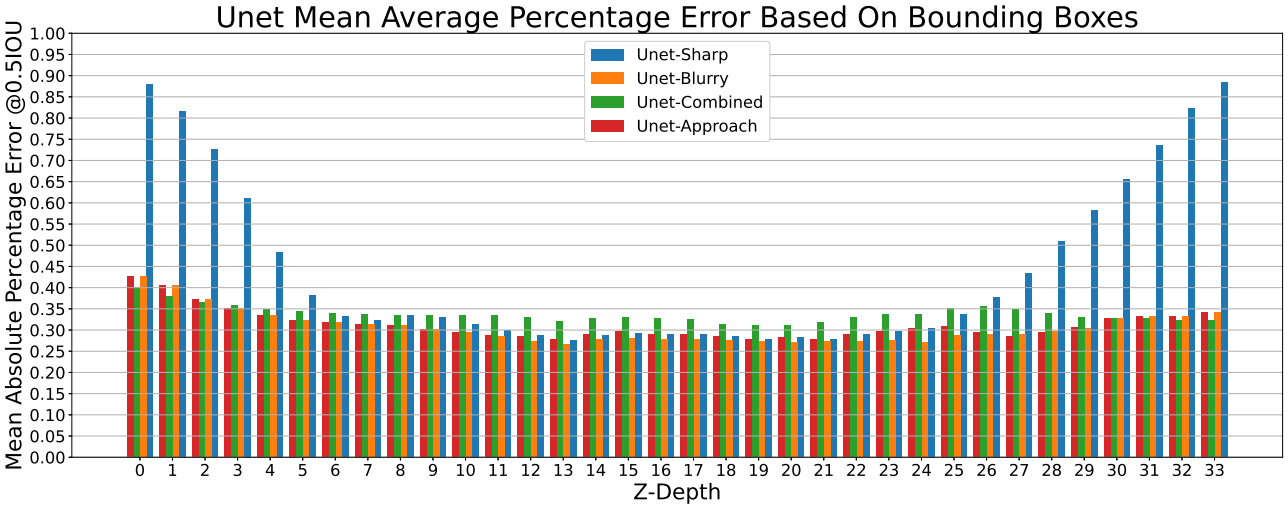
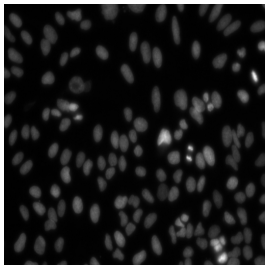
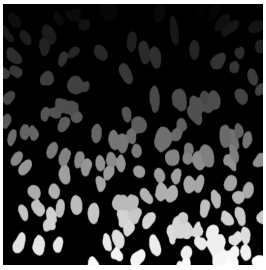


Figure 4: MAPE values for each z-depth for Unet models and image segmentation based dual-model pipeline. The calculations are based on bounding box derived TP, FP, FN values.



(a) Sample Hoechst stain image



(b) Corresponding ground truth mask

Figure 5: Example image (a) from the BBBC006 dataset and the corresponding cell segmentation mask (b). The Ground truth mask (b) is not binary as each individual cell nuclei in the image has a unique shade of gray associated with it to allow easier segmentation of individual cell nuclei.

4.2 Data Pre-processing

The dataset consists of two types of images Hoechst and phalloidin stains, the phalloidin stains were excluded from the dataset as phalloidin stains primarily provide information about the structure of a cell rather than nuclei counts.

Additionally, during the examination of the dataset a discrepancy was discovered between the cell count and mask ground truths. Eight images had non-zero cell counts listed in the cell count ground truth, but had completely blank mask ground truths. These images and their ground truths were excluded (across all z-depths) from the dataset.

After the dataset cleaning there were 25840 images remaining in the dataset across 34 z-depths (760 images per z-depth).

Two separate versions of the dataset were created - one for training Yolov4 models and one for training Unet models as they require different ground truths for training. The Yolov4 models required cell bounding boxes as a ground truth, while the Unet models required cell segmentation masks as ground truths.

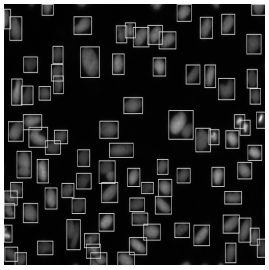
To create the Unet training dataset the ground cell segmentation masks in the dataset were thresholded so that all non-zero pixel values in the mask became one- this was done to make the problem that the Unet model was trying to solve a binary classification problem (is the cell present at a given location in the image or not).

To train the Yolov4 model bounding boxes for the cells needed to be created as they were not provided as a ground truth by the dataset. The bounding boxes were created using an automated method by labeling connected regions of the ground truth mask and then deriving the bounding box by measuring the properties of the labeled regions (dimensions) [19], this method of deriving bounding boxes from a cell segmentation mask will be referred to as the region properties method from here on. The region properties method was deemed sufficient for the creation of bounding boxes. The placement accuracy of the bounding boxes could not be quantified, but visual inspection of the created bounding boxes indicates that the placement is correct, but there are some issues with separating clustered cells into individual cells, see Figure 6b. The mean absolute percentage error (MAPE) was calculated using the bounding boxes produced by the region properties method to evaluate the accurate of the region properties method. MAPE value was calculated to be 1.51% using the following formula:

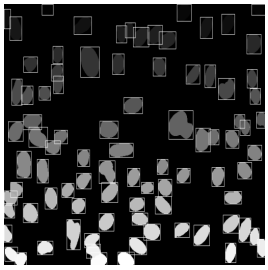
$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - G_t}{A_t}\right|$$

- A_t - ground truth cell nuclei count for ground truth cell segmentation mask with index t
- G_t - number of bounding boxes generated by the region properties method for ground truth cell segmentation mask with index t
- n - the number of ground truth cell segmentation masks

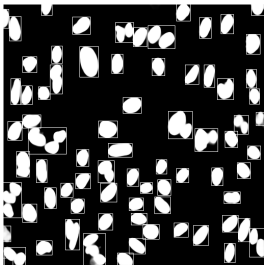
In both versions of the dataset the training-validation split was 90%:10%. The validation training-validation splits were the also the same across all the It was ensured that the validation sets across the two different versions were of the same images.



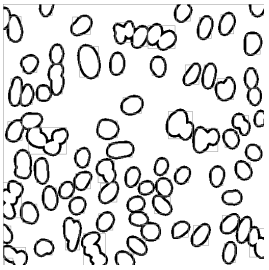
(a) Source WSI patch image at z-depth 16 with ground truth bounding boxes



(b) Ground truth cell segmentation mask for (a) with cell bounding boxes drawn over the mask.



(c) Unet-Combined model output segmentation mask for (a) with predicted cell bounding boxes.



(d) Predicted mask of (a) at the end of the pipeline in figure 2 with predicted bounding boxes.

Figure 6: Example image (a) from the BBBC006 dataset with bounding boxes derived from the ground truth mask (b) using the region properties method. The ground truth mask (b) is not binary as each individual cell in the mask has a unique shade of gray assigned to it. The predicted cell segmentation mask (c) is not binary as the output layer on the Unet is a sigmoid, for further processing the mask is thresholded at a pixel value of 0.5, this mask would be produced at the Image Mask state in the dual-model image segmentation pipeline, which can be seen in Figure 2. The post-processed mask (d), is used to determine the cell bounding boxes using the region properties method, however as can be seen by the bounding boxes the post-processing steps are not able to properly separate clusters of cells and multiple cells end up with only a single bounding box.

For both datasets the images and masks needed to be resized from 696x520 to 512x512 as both the Yolov4 model and Unet model had an input layer resolution of 512x512.

Additionally, it was important to also ensure that the same validation images were excluded across all z-depths to ensure that the models did not see ground truth values of the validation dataset. This is important to produce a good model as the same image across two neighbouring images can be extremely similar if not identical, letting the model memorize the ground truth.

For the Unet model training the data was augmented using 90°, 180°, 270°rotations of all the training images, the validation dataset was not augmented. Additionally, the input images and ground truth masks had their values normalized to the [0, 1] range. The Yolov4 model dataset was not augmented using the same image rotations as for the Unet model due to training time constraints. But the Yolov4 dataset was augmented using the default darknet mosaic augmentation (parts from one image being cut into a different one). The reason for the mosaic augmentation was that the mosaic augmentation is a default training option and recommended to be left enabled by the Yolov4's model creators guide on how to train a custom Yolov4 model [2]. Additionally, for Yolov4 the original images were transformed from the TIFF format to JPEG and normalized between [0, 255] due to the darknet library preferring JPEG format images [2].

4.3 Unet Model Training

Three Unet models were constructed and trained using the Tensorflow2 framework following the architecture laid out in the original Unet paper [17]. The channel number in each of the convolutional blocks was decreased by four times. This brought the model parameter size from about 30 million parameters to about 2 million parameters. This was done to decrease the model training times. Even the substantially smaller network should have sufficient learning feature capacity as it is only being trained on a single type of cell and stain so feature variance should not be substantial enough to exceed the learning capacity of the network, that is the dataset is relatively simple.

The network input and output size was adjusted to 512x512x1, that is an image size of 512x512 with a single color channel as the images are gray-scale.

The Unet models output layers activation function is the logistic sigmoid function [18]. The three Unet models that were trained were: Unet-sharp (Unet model trained only on in-focus images), Unet-blurry (Unet model trained only with out-of-focus images) and Unet-combined (Unet model trained with both in-focus and out-of-focus images). The Unet-sharp and Unet-blurry models were trained for use in the proposed approach while the Unet-combined was trained as a control for evaluating the dual-model pipeline performance.

The models were trained with binary cross entropy as the loss function from the Tensorflow2 library. The optimizer was Adam from the Tensorflow2 library with the default parameters in Tensorflow2 version 2.5.0. The evaluation metric was binary accuracy in the Tensorflow2 library with the default threshold of 0.5. All the Unet models were trained for only a single epoch as all of them converged relatively quickly during training and the loss value remained stable after training for only an hour for all models. The batch sizes for each of the models was different as in each batch a training image from each z-depth was included, so the batch sizes were: 13 for Unet-Sharp, 21 for Unet-Blurry and 34 for Unet-Combined.

4.4 Yolov4 Model Training

Three Yolov4 models were trained using the darknet library [2]. Three Yolov4 models were trained: Yolov4-sharp (Yolov4 model trained with only in-focus images), Yolov4-blurry (Yolov4 model trained with only out-of-focus images) and Yolov4-combined (Yolov4 model trained with both sharp and blurry images). Similar to the Unet models, the Yolov4-sharp and Yolov4-blurry models are to be used in the object detection dual-model pipeline, while the Yolov4-combined model acts as a control model.

The basis for the Yolov4 models was the yolov4-custom.cfg template that had certain parameters edited to be appropriate for the custom dataset. The parameters were adjusted according to the instructions in darknet library [2]. The maximum number of bounding boxes that the network can output was changed to 256 boxes to ensure that the network can output detections for all cells in the image (the maximum cell count in the listed cell count

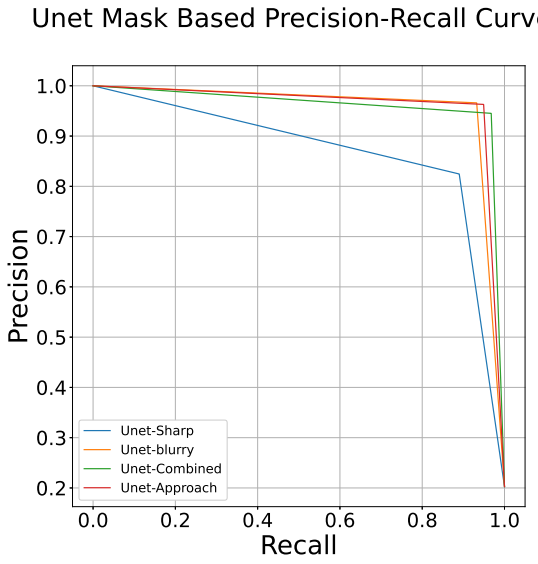


Figure 7: Precision-Recall curve for all Unet model and dual-model pipeline without the post-processing steps (called Unet-Approach in the graph) based on the predicted masks.

ground truth is 176 cells in a single image). The model input and output size was adjusted to 512x512x3 (that is a three color channel 512x512 image). The Hue parameter changed from 0.1 to 1 to force the the darknet library to interpret the images in gray-scale [3] as the framework reads images with three color-channels even if the images are gray-scale. The number of classes in each of the yolo layers was changed to one, since we are training for only a single class. The number of filters in each of the convolutional layers before each yolo layer in the model was changed to 18.

The darknet library uses stochastic gradient descent with momentum and weight decay. The optimizer parameters were kept as default, that is learning rate was kept at 0.001, momentum was kept at 0.949 and decay was kept at 0.0005. Burn in was also kept at the default 1000 batches.

The only training parameters that were adjusted were the max batches and steps values as they depend on the dataset size. For Yolov4-Sharp max batches was set to 8892 and step values were set to 7114 and 8003. For Yolov4-Blurry max batches were set to 14364, while steps were set to 11491 and 12928. For Yolov4-Combined max bathes was set to 23256, while steps were set to 18605 and 20930. The reason for these specific values is the dataset set size, max batches depends on the amount classes being trained according to the formula:

$$MaxBatches = max(classes * 2000, TrainingImages)$$

The models were trained to only detect a single class so the number of training images always wins in the above equation. The step values are just simply 80% and 90% of the max batches value respectively [2].

All the models were trained for 3000 batches with a checkpoint being made every 1000 iterations. The checkpoint with the best evaluation metrics was selected as the final version of a given model. For Yolov4-Sharp the 2000 batch checkpoint was selected. For Yolov4-Blurry the 1000 batch checkpoint was selected. For Yolov4-Combined the 2000 batch checkpoint was selected.

4.5 Blur Detection

For the classification-CNN it was decided to use the blur classifying CNN developed by S.J. Yang et al. [26] as it was simple to use and achieved good performance in the paper. The CNN classifies image into eleven classes of blur from 0 to 10, with 0 being the sharpest and 10 being very blurry. Since the dual-model pipeline needs a binary classification for the blur, classes in the range of [0, 5] were classified as sharp and classes in the range of [6, 10] were classified as blurry. The accuracy of this approach was evaluated per z-depth using the full evaluation dataset by calculating the percentage of evaluation images that were incorrectly classified, the results can be seen in Figure 9

4.6 Evaluation

The dual-model pipelines, all Yolov4 models and all Unet models will be evaluated by bounding boxes that they produce. For the Unet models this will mean that extra post-processing steps to derive cell bounding boxes from the cell segmentation masks will be needed. This process will be same as the one used in the image segmentation dual-model pipeline seen in Figure 2 (the post-processing starts from the Image Mask state). By calculating the percentage area overlap of a predicted bounding boxes area and ground truth bounding boxes area and thresholding this value at 50% it can be determined whether a predicted bounding box is a true positive (50% or more of at least one ground truth bounding boxes area overlaps with the predicted bounding boxes area) or a false positive (a predicted bounding boxes area does not overlap with any ground truth bounding boxes area by at least 50%). The false negative count can be derived by subtracting the number of true positives from the total number of ground truth bounding boxes [9]. These true positive (TP), false positive (FP), false negative (FN) counts will be used to derive the following metrics:

1. Mean Average Precision at 50% threshold of Intersection over Union for the bounding boxes (mAP@0.5). Since only a single class of objects is being detected this is equivalent to Average Precision at 50% threshold of Intersection over Union (AP@0.5) and this value is further equivalent to the Precision-Recall curve Area under the Curve (PR-AUC) value [27]. This value was computed via numerical integration using the midpoint rule at each detection [9].

2. Precision-Recall (PR) curve plots [27]. This was plotted using the precision-recall value pairs at each detection [9]

3. Precision [27], calculated by the following formula:

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives}$$

4. Recall [27], calculated by the following formula:

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives}$$

5. F1-Score - the harmonic mean of precision and recall [24], calculated by the following formula:

$$F1 = 2 * \frac{(Precision * Recall)}{(Precision + Recall)}$$

The mean absolute percentage error (MAPE) [6] was calculated for each model and dual-model pipeline at each z-depth for better insight into performance at each z-depth.

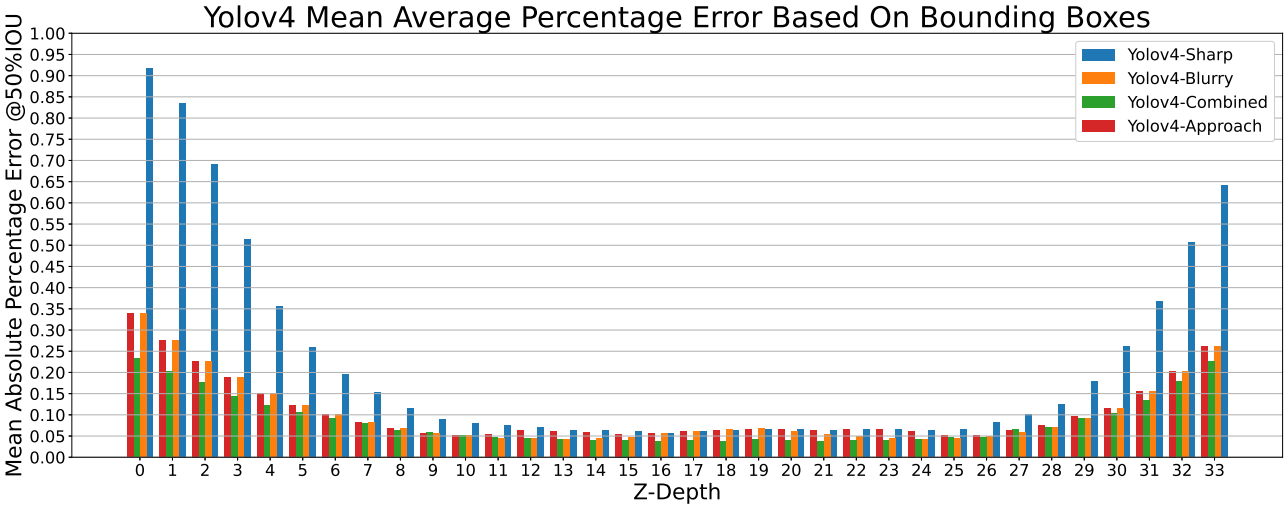


Figure 8: MAPE values for each z-depth for Yolov4 models and object detection based dual-model pipeline. The calculations are based on bounding box derived TP, FP, FN values.

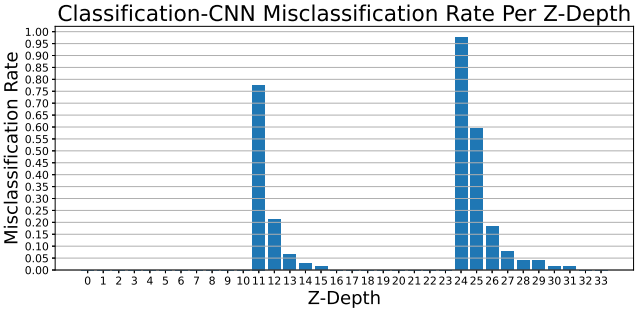


Figure 9: Classification-CNN misclassification rate per z-depth. The classification-CNN has difficulty correctly classifying images at the z-depth border of in-focus and out-of-focus images. The mean misclassification rate across all z-depths is 8.9%.

The formula used for MAPE is:

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right|$$

- A_t - ground truth number of cell in image with index t.
- F_t - the number of predicted bounding boxes that were classified as true positives according to IOU@50% rule for image with index t.
- n - the number of images in the validation dataset.

Additionally, the Unet model cell segmentation mask outputs will also be evaluated using the same metrics as above. However, since bounding boxes require post-processing to retrieve from masks, a different definition for TP, FP and FN will be needed. TP will be pixels who have a value above or equal to 0.5 in the predicted mask and the corresponding pixel in the ground truth mask has a value of 1. FP will be pixels in the predicted mask with a value above or equal to 0.5 and the corresponding pixel in the ground truth mask has a value of 0. FN will be pixels in the predicted mask with a value below 0.5 and the corresponding pixel in the ground truth mask has a value of 1. True negatives (TN) can also be derived but they are not used in the metric mentioned above. It should also be noted that with these TP, FP, FN definitions the mAP@0.5 metric will simply become the PR-AUC metric and the PR curve will have to be derived via logistic regression.

All models and dual-model pipelines were evaluated on the full evaluation dataset, that is the evaluation dataset included images from all z-depths.

5. RESULTS

The metrics derived from the bounding boxes produced by the models and approaches can be found in Table 1. The main takeaway from these results is that the proposed approach appears to function no better or slightly worse than simply training a single model that is trained on both blurry or sharp images. There are likely three reasons for this. First, from Figures 8 and 4 we can see that the combined models perform almost identically to the sharp model close to the optimal z-depth, but as we move farther away from it the sharp models start to perform worse than the combined models. Second, we can also see that the combined model appears to actually outperform even the blurry model in the blurry z-depths. Third, in Figure 9 we can see classification-CNN does have an 8.9% error rate. These three factors likely contribute most of the difference that we observe.

Even with heavy blurring the Yolov4-Combined model manages to achieve reasonable performance. Additionally, the sharp models performed the worst- which is not surprising as blur changes the features that the CNN is trying to learn and these features change under blur.

The performance of blurry models was relatively high, slightly outperforming the dual-model pipeline, this is likely down to fact that some of the images in the z-depths on the sharp-blurry edge are really a mixed bag about how really blurry-sharp they are. This fact can be in the misclassification rate of the classification-CNN seen in Figure 9.

Another takeaway from Table 1 is that the Unet bounding box predictions perform substantially worse than Yolov4 models, while the precision of the Unet models hovers around 80% (except for the sharp model) the recall is sub-

Table 1: Bounding box evaluation results for all trained models and proposed approach using Unet and Yolov4 models

Metric	Yolov4 Sharp	Yolov4 Blurry	Yolov4 Combined	Yolov4 Approach	Unet Sharp	Unet Blurry	Unet Combined	Unet Approach
mAP@0.5IOU	0.751	0.923	0.936	0.9210	0.549	0.683	0.667	0.666
Precision	0.844	0.902	0.935	0.922	0.716	0.818	0.815	0.813
Recall	0.772	0.899	0.916	0.892	0.530	0.670	0.636	0.662
F1-Score	0.806	0.901	0.925	0.907	0.610	0.737	0.714	0.730

Table 2: Predicted mask evaluation results for Unet models and Unet approach

Metric	Unet Sharp	Unet Blurry	Unet Combined	Unet Approach
PR-AUC	0.930	0.992	0.994	0.993
Precision	0.825	0.966	0.945	0.963
Recall	0.888	0.932	0.968	0.949
F1-Score	0.855	0.949	0.956	0.956

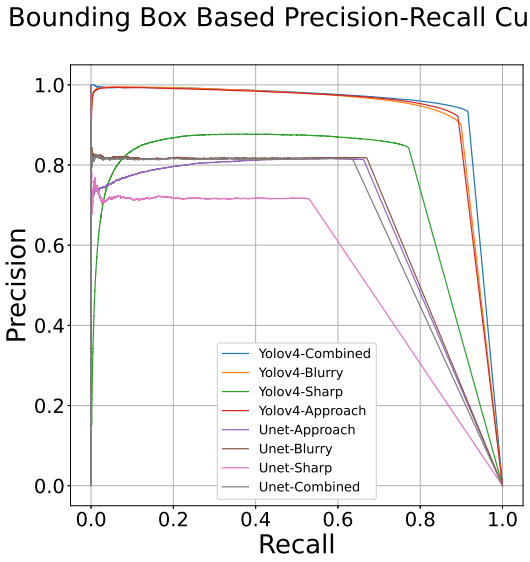


Figure 10: Precision-Recall curve for all the trained models and dual-model pipelines using the generated bounding boxes as evaluation metrics.

stantially lower at around 65% (Unet-Sharp models recall is even worse at 53%). But an examination of the Unet mask predictions metrics in Table 2 indicates that the masks should be accurate- indicating that the issue is with the method of deriving bounding boxes from the masks. An examination of the bounding boxes being produced by the method indicates issues segmenting clustered cells into individual cells, see Figure 6d for an example.

The Precision-Recall curve of the bounding box evaluation of all the models and dual-model pipelines can be seen in Figure 10. The Unet model and modified dual-model mask based Precision-Recall curve can be seen in Figure 7.

Examining the mean absolute percentage error plots of each z-depth for each of the models and dual-model pipelines, seen in Figure 8 for the Yolov4 models and in Figure 4 for the unet models. The figures show a pattern that the error rate starts climbing quickly below the z-depth of 5 and above the z-depth of 30. This would appear to indicate that the models are more easily able to cope with focusing errors before the optimal focal plane than with focusing errors after the optimal focal plane.

6. FUTURE WORK

The conclusion that the Yolov4 models are capable of detecting objects even in blurry images relatively well if trained with both in-focus and out-of-focus images. This indicates that models robust against blur could be achieved by simply including both blurry and sharp images in the training dataset. However, taking many WSI's of the same slide at different z-depths could make the data acquisition process tedious. If artificial blurring could be used it would make the process much easier. As such, an examination whether these properties carry over to artificial blur might be a future work direction.

7. CONCLUSION

The proposed approach turned out to perform no better than simply using a single model trained on both in-focus and out-of-focus images. However the Yolov4 model trained on both in-focus and out-of-focus images demonstrated reasonably good detection metrics indicating that simply including blurry images in training datasets can provide a good improvement in model robustness against blur. While the performance at the heaviest blur levels is still not satisfactory, it is dramatically improved. With further improvements to blur robustness satisfactory levels of performance could be achieved even under heavy blur.

Additionally, there exists a cut off point where the results start becoming substantially worse, specifically under heavy blur conditions at z-depths of around 5 - about 20 micrometers below the optimal z-depth and at z-depths of 30 - about 30 micrometers above the optimal z-depth. This also leads to the conclusion that the Yolov4 model was able to more easily deal with blur produced by focusing incorrectly too close, than focusing incorrectly further away from the optimal focus plane.

The closest appropriate comparison for existing cell counting techniques that could be found was S. Chen et al. [4], this study does perform its evaluation on the BBBC006 dataset and derives the same AP@0.5 metric (equivalent to Table 1 mAP@0.5IOU), but only a single z-depth subset (it is unclear, but most likely z-depth of 16) was evaluated making the comparison not entirely appropriate.

The masks produced by the Unet models were quite accurate, but the further processing required to extract bounding boxes from these predicted cell masks was inaccurate as it could not properly segment clusters of cells into individual cells. If better methods of separating the cells in the predicted mask could be applies the performance of Unet models for the cell counting and detection task could be dramatically improved. But due to the poor bounding box extraction method the results, based on bounding boxes, from Unet models are unreliable.

8. REFERENCES

[1] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao. Yolov4: Optimal speed and accuracy of object detection, 2020.

[2] A. Buchkovskiy. Darknet framework.

[3] A. Buchkovskiy. Unable to detect object in grayscale image - darknet.

[4] S. Chen, C. Ding, M. Liu, and D. Tao. Cpp-net: Context-aware polygon proposal network for nucleus segmentation, 2021.

[5] Y. Chen, J. Zee, A. Smith, C. Jayapandian, J. Hodgin, D. Howell, M. Palmer, D. Thomas, C. Cassol, A. B. Farris, and et al. Assessment of a computerized quantitative quality control tool for whole slide images of kidney biopsies. *The Journal of Pathology*, 253(3):268278, 2021.

[6] A. de Myttenaere, B. Golden, B. Le Grand, and F. Rossi. Mean absolute percentage error for regression models. *Neurocomputing*, 192:38–48, 2016. Advances in artificial neural networks, machine learning and computational intelligence.

[7] F. Ghaznavi, A. Evans, A. Madabhushi, and M. Feldman. Digital imaging in pathology: Whole-slide imaging and beyond. *Annual Review of Pathology: Mechanisms of Disease*, 8(1):331359, 2013.

[8] J. Guan, W. Zhang, J. Gu, and H. Ren. No-reference blur assessment based on edge modeling. *Journal of Visual Communication and Image Representation*, 29:1–7, 2015.

[9] J. Hui. map (mean average precision) for object detection, Apr 2019.

[10] A. Janowczyk, R. Zuo, H. Gilmore, M. Feldman, and A. Madabhushi. Histoqc: An open-source quality control tool for digital pathology slides. *JCO Clinical Cancer Informatics*, (3):17, 2019.

[11] R. Khandekar, P. Shastry, S. Jaishankar, O. Faust, and N. Sampathila. Automated blast cell detection for acute lymphoblastic leukemia diagnosis. *Biomedical Signal Processing and Control*, 68:102690, 2021.

[12] V. Ljosa, K. L. Sokolnicki, and A. E. Carpenter. Annotated high-throughput microscopy image sets for validation. *Nature Methods*, 9(7):637637, 2012.

[13] V. Ljosa, K. L. Sokolnicki, and A. E. Carpenter. Annotated high-throughput microscopy image sets for validation. *Nature Methods*, 9(7):637637, 2012.

[14] J. Lu, N. Sladoje, C. Runow Stark, E. Darai Ramqvist, J.-M. Hirsch, and J. Lindblad. A deep learning based pipeline for efficient oral cancer screening on whole slide images. In A. Campilho, F. Karray, and Z. Wang, editors, *Image Analysis and Recognition*, pages 249–261, Cham, 2020. Springer International Publishing.

[15] H. H. N. Pham, M. Futakuchi, A. Bychkov, T. Furukawa, K. Kuroda, and J. Fukuoka. Detection of lung cancer lymph node metastases from whole-slide histopathologic images using a two-step deep learning approach. *The American Journal of Pathology*, 189(12):2428–2439, 2019.

[16] R. M. Rad, P. Saeedi, J. Au, and J. Havelock. Blastomere cell counting and centroid localization in microscopic images of human embryo. In *2018 IEEE 20th International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–6, 2018.

[17] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[18] S. Russell and P. Norvig. *Artificial intelligence: A Modern Approach*. Pearson Education Inc., 3 edition, 2010.

[19] scikit image. Label image regions.

[20] C. Senaras, M. K. K. Niazi, G. Lozanski, and M. N. Gurcan. Deepfocus: Detection of out-of-focus regions in whole slide digital images using deep learning. *Plos One*, 13(10), 2018.

[21] K. Sirinukunwattana, S. E. A. Raza, Y.-W. Tsang, D. R. J. Snead, I. A. Cree, and N. M. Rajpoot. Locality sensitive deep learning for detection and classification of nuclei in routine colon cancer histology images. *IEEE Transactions on Medical Imaging*, 35(5):1196–1206, 2016.

[22] K. S. Wang, G. Yu, C. Xu, X. H. Meng, J. Zhou, C. Zheng, Z. Deng, L. Shang, R. Liu, S. Su, and et al. Accurate diagnosis of colorectal cancer based on histopathology images using artificial intelligence. *BMC Medicine*, 19(1), 2021.

[23] S. Wang, T. Wang, L. Yang, D. M. Yang, J. Fujimoto, F. Yi, X. Luo, Y. Yang, B. Yao, S. Lin, and et al. Convpath: A software tool for lung adenocarcinoma digital pathological image analysis aided by a convolutional neural network. *EBioMedicine*, 50:103110, 2019.

[24] T. Wood. F-score, May 2019.

[25] H. Yang, L. Chen, Z. Cheng, M. Yang, J. Wang, C. Lin, Y. Wang, L. Huang, Y. Chen, S. Peng, and et al. Deep learning-based six-type classifier for lung cancer and mimics from histopathological whole slide images: a retrospective study. *BMC Medicine*, 19(1), 2021.

[26] S. J. Yang, M. Berndl, D. M. Ando, M. Barch, A. Narayanaswamy, E. Christiansen, S. Hoyer, C. Roat, J. Hung, C. T. Rueden, and et al. Assessing microscope image focus quality with deep learning. *BMC Bioinformatics*, 19(1), 2018.

[27] S. Yohanandan. map (mean average precision) might confuse you!, Jun 2020.