Detection and Localization of Sperm Cells

Leveraging Computer Vision Techniques for Microscopic Image Analysis

Derrick Mayall, Eric Chaves, Shuai Shao

AGENDA

Problem Statement & Challenges

We propose leveraging Convolutional Neural Networks (CNNs) to automate and expedite the process of sperm counting, offering a faster and more reliable alternative to manual methods.

Challenges:

- Finding data due to sensitivity
- Computing power and resources
- Model Design and Development

Motivation

- Rapekits are on a huge backlog, estimated to be anywhere from 200,000-400,000 still waiting to be processed [1]
- This problem only increases as this crime in the US continues to grow, self reporting alone has doubled in recent years [2]
- One large cause of this backlog is the amount of time it takes to actually process a kit, with steps of verifying enough counts for a good DNA sample, and then needing to actually extract that DNA [3]

1: https://en.wikipedia.org/wiki/Rape_kit#:~:text=A%20rape%20kit%20is%20considered,the%20forensic%20labs%20for%20analysis.
2: https://www.nsvrc.org/statistics/statistics-depth#:~:text=The%20self%2Dreported%20incidence%20of,the%20United%20States%20in%202018.
3:https://www.endthebacklog.org/what-is-the-backlog/#:~:text=When%20tested%2C%20DNA%20evidence%20contained,of%20receipt%20by%20the%20lab.

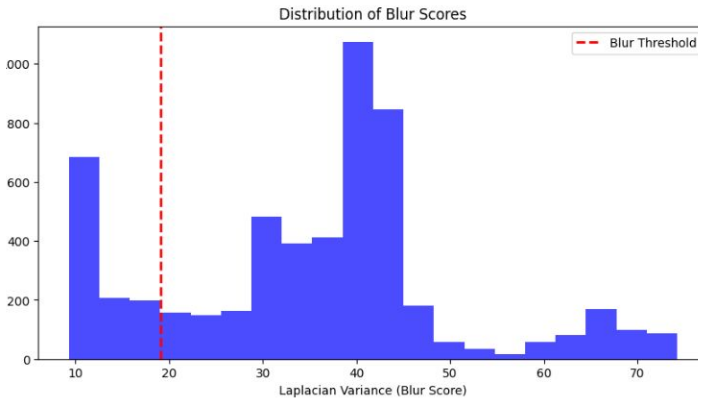
Dataset

- The dataset is the VISEM-Tracking
    - This is a sperm motility dataset used to teach models to count and analyze the state of a sperm cell (damaged versus healthy)
- We modified this dataset to better fit our subject
    - We did not worry about the sperm state since that would not matter in this case
    - Limited the dataset to smaller counts to account for how little the kits will be able to obtain a lot of the time

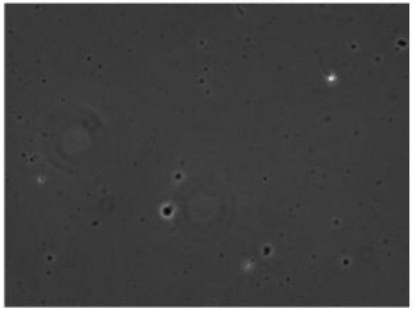
	frame_name	sperm_count	fid	bb0	bb1	bb2	bb3
0	14_frame_0	3	ckyw6zzlj001r3867thf0fuy7	0.208594	0.825000	0.035937	0.037500
1	14_frame_0	3	ckyw704kw001v3867kvyjtx6k	0.796094	0.797917	0.029687	0.037500
2	14_frame_0	3	ckyw708pn001z386779fr849h	0.827344	0.123958	0.035937	0.039583
3	14_frame_1	3	ckyw6zzlj001r3867thf0fuy7	0.208594	0.814583	0.035937	0.037500
4	14_frame_1	3	ckyw704kw001v3867kvyjtx6k	0.764062	0.781250	0.029687	0.037500
...	...	...	...	...	...	...	...
19836	52_frame_427	6	cl1x89ze6000o3f6baic4zklo	0.464062	0.462500	0.037500	0.054167
19837	52_frame_427	6	cl1x8a7j3000s3f6bvn1rm1tw	0.416406	0.400000	0.029687	0.045833
19838	52_frame_427	6	cl1x8ih7n00103f6b67egfbw2	0.057031	0.803125	0.032813	0.043750
19839	52_frame_427	6	cl52g254u000o3b6gcyhoi1fk	0.132031	0.112500	0.026562	0.037500
19840	52_frame_427	6	cl52g4ex1000s3b6g2941ltlj	0.297656	0.729167	0.039062	0.050000

Analysis

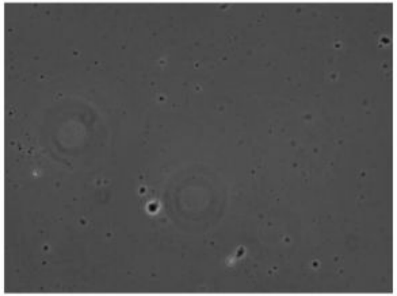
- First we looked the the blur scores of the images using the Laplacian Variance
- We found about ¼ of the dataset to be blurred (defined as ½ of the median)
- Thinking about the approach and problem at hand we decided to keep all images because not every kit will be perfectly readable making the blur even more important



23_frame_1411
Blur Score: 10.48

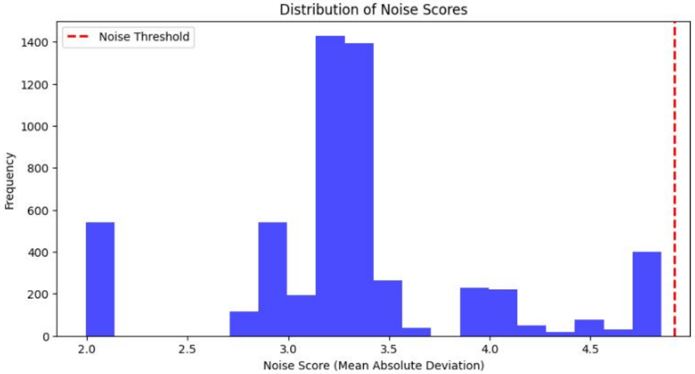


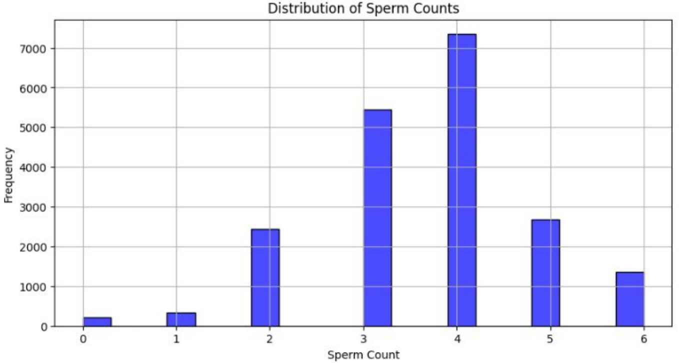
23_frame_1044
Blur Score: 11.24



Analysis

- Next we looked at noise and determined that no images needed to be removed due to noise (threshold defined as 1.5 times greater than median)
- Next we looked at count distributions ranging from 0-6 as we decided would be best range of lower counts, looking at the distribution we were satisfied and thought it seemed like a good balance





Existing Methods

- The largest existing paper we found that compares to our method idea was a research paper done on sperm motility of obese men using the same dataset
- They used a yolo v5 model to determine BB of sperm cells and whether the cell was damaged or not
- Using a fitness value shown below they found the Yolo v5l performs the best with a fitness score of 0.0920 with an estimated IoU of 0.5-0.75
    - This was estimated based on looking at the mAP

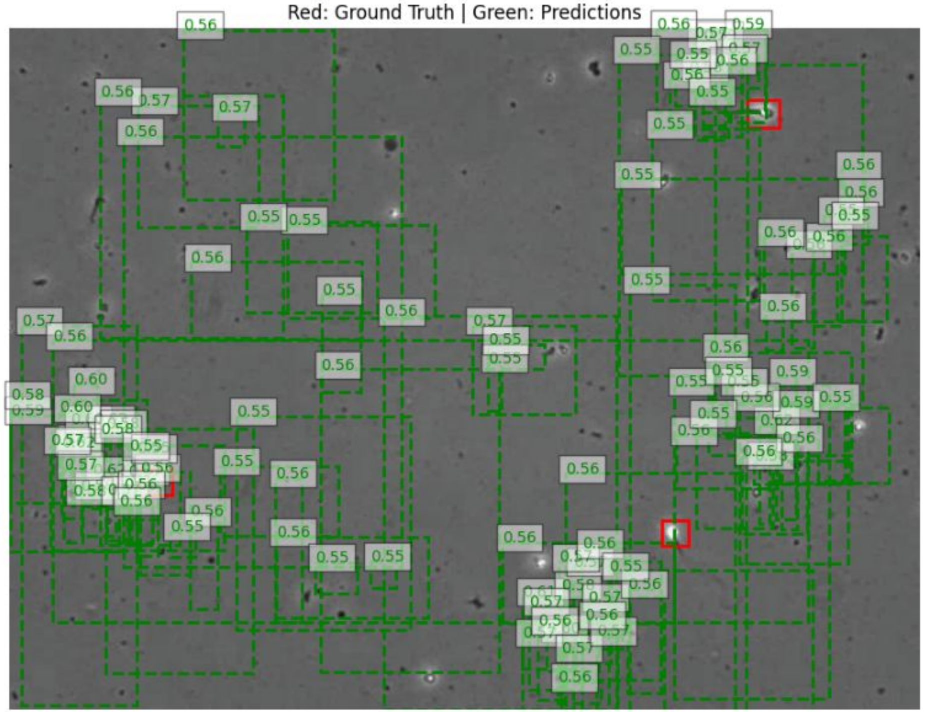
$$Fitness\_value = (0.1 \times mAP\_0.5 + 0.9 \times mAP\_0.95)$$

Derrick’s Method

- Backbone:
  - EfficientNet-B0 is a lightweight and efficient convolutional network known for its excellent trade-off between speed and accuracy
  - Using the feature extraction part to output high-level image features
- Pooling:
  - Used the MultiScaleRoIAlign to convert variable-size regions intop 7x7 feature maps
- CNN:
  - Used the FastRCNN to put everything into a pipeline and do the object detection

Derrick Results

YoloV5: 0.5 - 0.75
Baseline: 0.0061
IoU: 0.767134



Shuai's Method

- YOLO-NAS:
  - Speed and Accuracy: Optimized using NAS for a better tradeoff between real-time inference and high precision.
  - Lightweight Structure: Efficient architecture suited for edge devices.
  - Pretrained Weights: Trained initially on COCO dataset, easily fine-tuned for custom datasets.
- Fuzzy Logic:
  - Introduced to softly evaluate the reliability of detections.
  - Input Variable: Detection confidence score (ranging from 0 to 1).
  - Output Variable: Reliability score (ranging from 0 to 1).

Eric’s Method

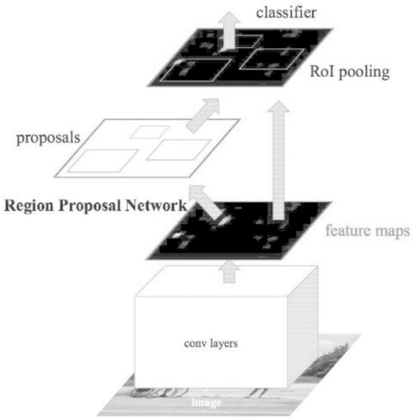


Figure 5: The latest incarnation of the R-CNN family, Faster R-CNN, introduces a **Region Proposal Network (RPN)** that bakes region proposals *directly* in the architecture, alleviating the need for the Selective Search algorithm. (Image credit: Figure 2 of Girshick et al., 2015)

Faster R-CNN: Two-stage object detector.

- Stage 1 (RPN): suggests candidate object regions (proposals).
- Stage 2 (ROI Head): Refines and classifies the proposals.

Backbone:

- ResNet-50 pretrained on ImageNet.
- Used as a feature extractor (final classification layers removed).

Modifications for this project:

- Converted grayscale images to 3-channel RGB to match ResNet50 input requirements.
- Tuned anchor sizes and aspect ratios to detect small sperm cells.
- Customized NMS threshold and Score threshold for better proposal filtering.

Architecture Highlights:

- Anchor Sizes: (8, 16, 32, 64, 128)
- Aspect Ratios: (0.5, 1.0, 1.5, 2.0)
- NMS Threshold: 0.5
- Score Threshold: 0.7

Loss Functions:

- RPN
  - Binary cross entropy (object vs background)
  - Smooth L1 Loss (anchor box regression).
- RoI Head
  - Cross-Entropy(sperm vs background)
  - Smooth L1 Loss (bounding box regression).

https://pyimagesearch.com/2023/11/13/faster-r-cnns/

Preprocessing

- **YOLO to Corner Conversion**
  - Converted bounding boxes from YOLO format (x_center, y_center, width, height) to corner coordinates `(xmin, ymin, xmax, ymax)` in pixel space using original image dimensions.
- **Grayscale to RGB Conversion**
  - Images loaded as single-channel grayscale and expanded into 3-channel RGB by duplication to meet ResNet-50 input requirements.
- **No image resizing**
  - Images retained original resolution (640×480).
- **Transforms**
  - Basic transformations (e.g., tensor conversion).
    No heavy augmentation applied during training.

Experiments and Hyperparameter Search

- **Initial Random Search (IoU - 0.6949)**
  - Explored wide ranges of learning rate, weight decay, step size, gamma, anchor sizes, aspect ratios, box NMS threshold, and score threshold.
  - Ran 20 configurations for 2 epochs each
- **Next Random Search (IoU - 0.7494)**
  - Focused search around the best-performing configurations to further improve IoU.
  - Ran 20 configurations for 2 epochs each
- **Final Grid Search (IoU - 0.8088)**
  - Centered around the best configuration from the previous random search.
  - Total of 48 configurations
  - 3 epochs per configuration

- **Final Training (IoU - 0.9143)**
  - Best hyperparameters used for final training:
    - lr: **7.5e-05**
    - weight_decay: **0.0001**
    - step_size: **6**
    - gamma: **0.2**
    - anchor_sizes: **(8, 16, 32, 64, 128)**
    - aspect_ratios: **(0.5, 1.0, 1.5, 2.0)** ×5
    - box_nms_thresh: **0.5**
    - score_thresh: **0.7**
  - Final Mean IoU after full training (30 epochs): 0.9143

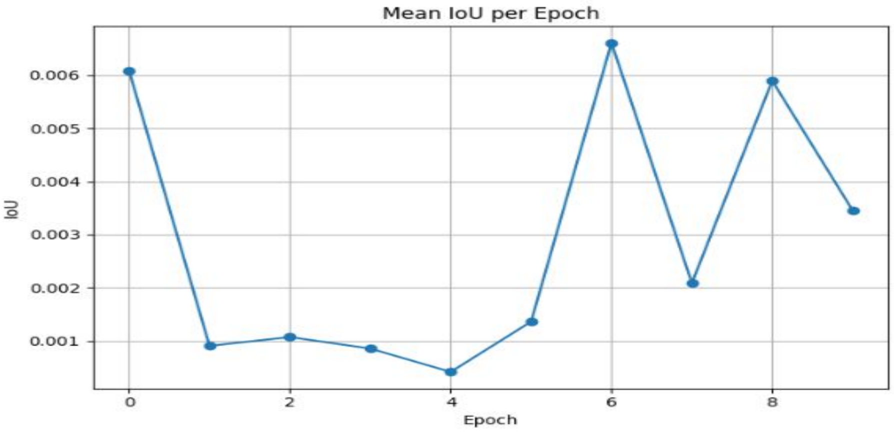
Comparison of Results

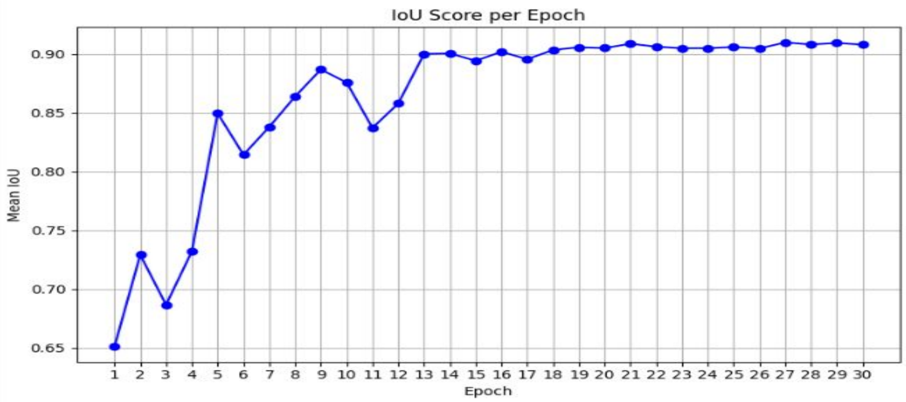
Baseline CNN Model

- Mean IoU across epochs ≈ 0.001 to 0.006.
- Poor localization ability.
- MSE loss decreased during training, but bounding box predictions were inaccurate.
- Model architecture too simple (only one convolutional layer).

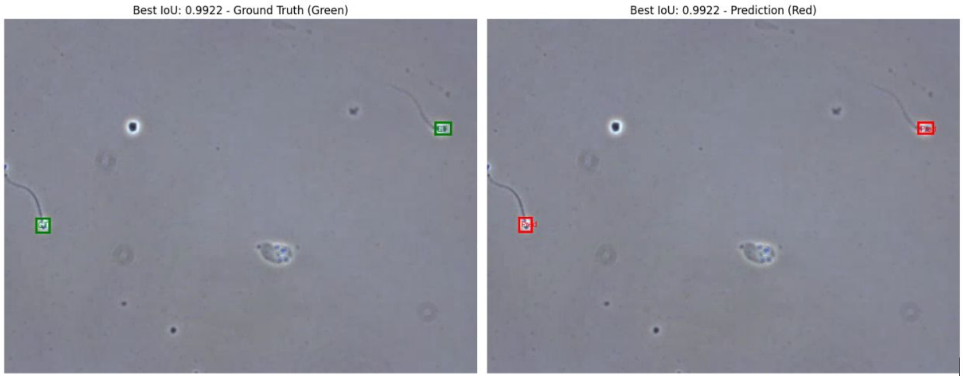
Faster R-CNN Model

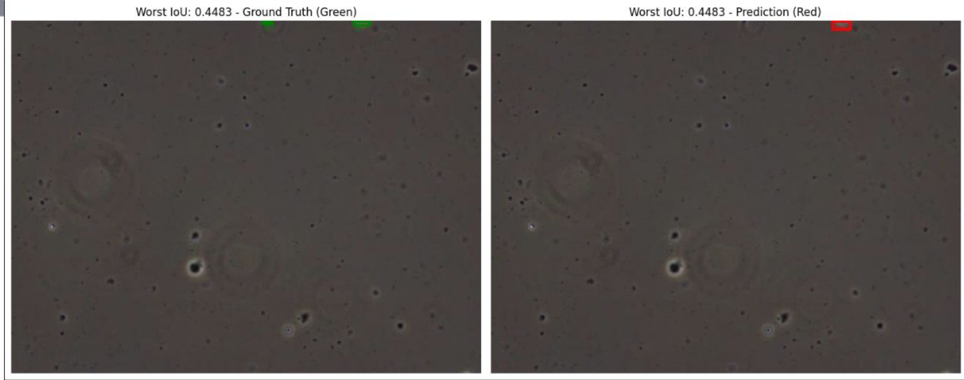
- Final Mean IoU after full training: 0.9143
- Able to accurately localize sperm cells even in cluttered images.
- Very few false positives and false negatives in visual inspections.
- Highly stable validation IoU across epochs after convergence.





Samples


Best IoU: 0.9922 - Ground Truth (Green)
Best IoU: 0.9922 - Prediction (Red)


Worst IoU: 0.4483 - Ground Truth (Green)
Worst IoU: 0.4483 - Prediction (Red)

Observations

- Advanced two-stage detectors (like Faster R-CNN) are crucial for fine-grained object detection.
- Naive CNN models cannot effectively handle bounding box regression and localization without specialized mechanisms.
- Careful hyperparameter tuning (anchors, thresholds, learning rates) was **critical** for achieving high performance.
- Preserving full image resolution helped maintain positional accuracy for small sperm cells.

Conclusion

- Faster R-CNN with ResNet-50 backbone achieved high localization accuracy for small sperm cells.
- Mean IoU of 0.9143 demonstrates strong model reliability.
- YOLO to Corner conversion and grayscale-to-RGB adaptation were crucial preprocessing steps.
- Careful hyperparameter tuning significantly boosted performance.
- Preserving full image resolution helped detect small objects accurately.
- Baseline CNN comparison showed that simple architectures are insufficient for fine-grained detection tasks.

Future Work

- **Upgrade Backbone**
    - Experiment with ResNet-152 backbone for richer feature extraction and possibly higher accuracy.

- **Data Augmentation**
    - Introduce augmentations (e.g., random flips, rotations, brightness/contrast adjustments) to improve model robustness to image variability.
- **Fuzzy Logic**
    - Using adaptive fuzzy systems that learn rules over time.
    - Exporting the model to ONNX format for deployment on mobile or edge devices.