Automatic ovarian follicle detection using object detection models

Maya Haj Hassan, Eric Reiter, Misbah Razzaq

Automatic ovarian follicle detection using object detection models

Maya Haj Hassan
Inrae
Eric Reiter
Inrae
Misbah Razzaq
misbah.razzaq@inrae.fr

Inrae

Research Article

Keywords: Artificial intelligence, object detection, computer vision annotation, deep learning, folliculogenesis, corpus luteum, antral follicle, reproduction

Posted Date: June 27th, 2024

DOI: https://doi.org/10.21203/rs.3.rs-4637709/v1

Additional Declarations: The authors declare no competing interests.

Automatic ovarian follicle detection using object detection models

Maya Haj Hassan[1], Eric Reiter[1,2], and Misbah Razzaq[1,*]

[1]INRAE, CNRS, Université de Tours, PRC, 37380, Nouzilly, France
[2]Université Paris-Saclay, Inria, Inria Saclay-Île-de-France, 91120, Palaiseau, France
[1]maya.haj-hassan@inrae.fr, eric.reiter@inrae.fr
[*]Corresponding author: misbah.razzaq@inrae.fr

June 26, 2024

Abstract

Ovaries are of paramount importance in reproduction as they produce female gametes through a complex developmental process known as folliculogenesis. In the prospect of better understanding the mechanisms of folliculogenesis and of developing novel pharmacological approaches to control it, it is important to accurately and quantitatively assess the later stages of ovarian folliculogenesis (i.e. the formation of antral follicles and corpus lutea). Manual counting from histological sections is commonly employed to determine the number of these follicular structures, however it is a laborious and error prone task. In this work, we show the benefits of deep learning models for counting antral follicles and corpus lutea in ovarian histology sections. Here, we use various backbone architectures to build two one-stage object detection models, i.e. YOLO and RetinaNet. We employ transfer learning, early stopping, and data augmentation approaches to improve the generalizability of the object detectors. Furthermore, we use sampling strategy to mitigate the foreground-foreground class imbalance and focal loss to reduce the imbalance between the foreground-background classes. Our models were trained and validated using a dataset containing only 1000 images. With RetinaNet, we achieved a mean average precision of 83% whereas with YOLO of 75% on the testing dataset. Our results demonstrate that deep learning methods are useful to speed up the follicle counting process and improve accuracy by correcting manual counting errors.

Keywords— Artificial intelligence, object detection, computer vision annotation, deep learning, folliculogenesis, corpus luteum, antral follicle, reproduction.

1 Introduction

Folliculogenesis is a highly complex and dynamic process which culminates with the ovulation of one or more oocyte(s) at each cycle. During each estrous cycle, the follicles develop from a dormant primordial pool. The oocytes start to grow and maturate while surrounded by an increasing number of granulosa cells. Various classifications have been used to describe the different stages of oocyte and follicle development[1, 2]. Briefly, primordial follicles contain a partial or complete single layer of squamous granulosa cells. Primary follicles contain a single layer of cuboidal granulosa cells. Antral follicles are characterized by multiple layers of granulosa cells and a cavity named antrum. The remaining of the antral follicle following ovulation is called corpus luteum. It is composed of granulosa cells, thecal cells and blood vessels. The evaluation of follicle numbers across these different classes at various stages of development and/or upon exposure to hormonal/pharmacological treatments is crucial in many fields of biology. The number of follicles and corpus luteum can vary between estrus cycles in response to physiological and non-physiological factors. These factors include endocrine-disrupting chemicals [3, 4], maternal aging, chemotherapy [5], infection [6], and inflammation [7]. All of them have been shown to affect ovarian reserve. As the antral follicles and corpus lutea represent the hallmark of late follicular development and ovulation, counting their number is necessary when studying infertilities, improving assisted reproduction technologies or evaluating of the effects of drugs.

Research and pre-clinical phase of drug development extensively use rodents as experimental models to evaluate potential efficacy and/or repro-toxicity [8, 9]. Consequently, the refinement of follicle quantification

methods has gained heightened significance. It is imperative for researchers to understand the strengths and weaknesses of the available counting approaches to ensure the accurate interpretation of results. Histological counting have been widely accepted and used in reproductive biology research to estimate the number of follicles. It enables the distinction of various types of follicles including, primordial, primary, secondary, and antral follicles. It offers spatial distribution and organization of follicles within the ovary. However, it remains a tedious and time-consuming technique based on the visual assessment by the expert. This approach rely entirely on the expert and, therefore, is prone to human errors.

Artificial intelligence (AI)-based methods have gained popularity in recent years and have been successfully applied in many domains such as image recognition [10], robotics [11], speech recognition [12], life sciences [13, 14, 15, 16], etc. Advances in biomedical technologies are providing us with large amounts of data such as proteomics, genomics, and medical images [17]. They have shown great performance in image analysis due to the availability of large amount of labelled dataset, sometimes even surpassing the contributions of experts [18]. Follicle detection from histology images using deep learning methods remains largely an uncharted territory. The high resolution of the whole slide digital images (WSI) obtained from digital slide scanners combined with different AI methods, can reduce the workload and inconsistencies of current methods [19, 20]. This paper highlights the benefits of AI methods, particularly deep learning, for counting antral follicles and corpus lutea. Regarding deep learning methods to count follicles in mouse ovaries, in [21] authors proposed a convolutional neural network (CNN) with a sliding window algorithm to count primordial follicles. They used data of 9 million images of mouse ovaries to train the model and 3 million images to test the model. They achieved precision of 65% and recall of 91%. Later in [22] authors performed detection of 5 classes of follicles (primordial, primary, preantral, secondary, and tertiary). They started with generating sub-images from the input image, then these sub-images were classified into edge, follicle, and background classes, and finally a binary image was created representing the background and follicle class, and position of these binary sub-images were drawn on the original input image. In the final classification phase, all follicles localized in the input image were classified into 5 classes. Their dataset consists of 1750 images for the training set and 222 images for the testing set. On the testing dataset, they achieved a mean accuracy of 95%. In contrast to earlier research, where primordial, primary, preantral, secondary, and tertiary were detected, our primary focus of this study are late follicles, i.e., the antral follicles and corpus luteum. Furthermore, we report state-of-the-art mean average precision (MaP) metric for the evaluation of the proposed object detection models which is absent in the aforementioned methods. The proposed model represents a first step towards automating the quantitative assessment of late folliculogenesis.

The remainder of the paper is structured as follows. In Section 2, we introduce the background on late follicles, their annotation, and describe the proposed machine learning framework for follicles counting. In Section 3, we present and compare our results. In Section 4, we identify limitations, give concluding remarks and describe future work.

2 Methods

2.1 Animals

All experimental and care procedures were by the European and French Directives and approved by the local ethical committee CEEA Val de Loire N°19 and the French ministry of teaching, research, and innovation (APAFIS #18035 − 2018120518194796). C57BL/6JOlaHsd mice were purchased from Inotiv.inc. These female mice of 12 − 20 week-old were housed in the rodent animal facility, experimental unit: UEPAO (PAO, INRAE: Animal Physiology Facility, https://doi.org/10.15454/1.5573896321728955E12) in an environmentally controlled room maintained at 21°C, humidity of 55 percent with a 12h light − 12h dark photoperiod, ad libitum access to food and water.

2.2 Tissue collecting and processing

The mice were sacrified by cervical dislocation. Ovaries were collected, trimmed from the fat pad and fixed in Bouin's solution (Sigma Aldrich, HT10132) at 4°C overnight. The samples were dehydrated using ethanol water sequential incubations and embedded in paraffine blocks (see figure 1). They were sequentially sectioned into $7\mu m$ using a microtome (Leica HistoCore AUTOCUT). The whole sections were mounted on microsocope superfrost plus slides. Between 7 to 15 consecutive sections were placed onto a single slide. After 48h at room temperature, each slide was deparaffinized, rehydrated and stained with hematoxylin-eosin (Sigma Aldrich, HHS32, HT110132) for morphological observation. The sections were mounted in Depex (DPX new, Merck GaA, Darmstadt, Germany).

2.3 Manual follicle counting

The slides were digitized after 72h using a histology slide scanner Axio scan Z.1 Zeiss, running under Zen software (ZENblue 3.5 edition) with a magnifcation of 10x (numerical aperture 0.45) (see figure 1). Follicles containing multiple layers of granulosa cells and a follicular antrum were designed as antral follicles. To avoid counting the same follicle on serial sections, only those containing a clear visible oocyte were scored. The total number of antral follicles is the sum of the antral follicles from all sections of a complete ovary. The corpus luteum is more a solid structure, made of granulosa cells (rounded cells), theca cells (elongated cells), and blood vessels.

2.4 Deep learning framework for automatic counting ovarian follicles

2.4.1 CVAT annotation and data extraction

The sections on the slides were extracted in Joint Photographic Experts Group format (JPG) by using ZENblue 3.5 edition to annotate with Computer Vision Annotation Tool (CVAT), a free open source, suitable for image and video labeling [23]. We decided to use bounding boxes for annotation purposes for two reasons: 1) Boxes require relatively less workload to annotate as compared to other formats such as polygon, etc. 2) It has been shown previously that other formats do not necessarily increase performance [24] by large margin while causing more workload for the annotator. Three structures were annotated: 1) antral follicle (AF), 2) antral follicle without ovocyte (AFWO), and 3) corpus luteum (CL). In figure 1, we show the manual annotation workflow.

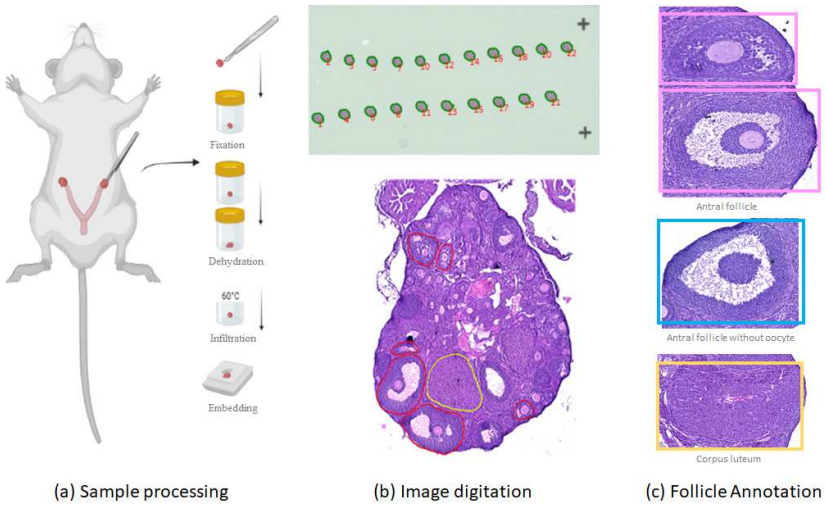


Figure 1: Data extraction process for annotation of whole slide images. (a) Sample processing - Ovaries were dissected out and trimmed from the fat pad, fixed in Bouin's solution, dehydrated and embedded in paraffine blocks. (b) Image digitation - The stained slides were digitized by Axio scan Z.1 Zeiss with a magnification of 10x. (c) Follicle annotation - Follicles containing multiple layers of granulosa cells and a follicular antrum were designed as antral follicles (Pink box), The antral follicles lacking a visible nucleus were labelled as antral follicle without oocytes (Blue box) and the temporary endocrine structures formed from the remanants of the ovarian follicle after ovulation were identified as corpora luteum and annotated (Yellow box).

2.4.2 Data augmentation

In our work, we use on-the-fly data augmentation since it eliminates the need to save additional datasets and enhances computational efficiency by dynamically generating augmented data during the training process, thereby improving model generalization and robustness.

2.4.3 Class imbalance

Classification algorithms are known to be very sensitive to unbalanced data when the aim is to derive classification and prediction tools for categorical classes. In general, the algorithms will correctly classify the most frequent classes and lead to higher misclassification rates for the minority classes, which are often the most interesting ones. In our case, we observe that the AFWO is an over-represented class. We can see from table 1

that we have approximately 1.5x more samples of AFWO as compared to AF and approximately 2x more samples of AFWO as compared to CL. To deal with the data imbalance, we sampled the dataset as shown in the sampled dataset column. In object detection tasks, we have foreground-background class imbalance in addition to foreground-foreground class imbalance. It is unavoidable because majority of the boxes are labeled as the background during the training process. In this paper, we employ sampling methods to deal with the foreground-background imblance. Refer to [26] for a detailed review of various data imbalance strategies.

2.4.4 Object detection models

Object detection includes both locating and classifying objects of interest in images (in our case, full sections of ovaries). In general, two types of detectors can be used: two-stage and one-stage detectors. Two-stage detectors are slower but more accurate due to their complicated approach which involves first generating region proposals before detection and classification tasks. One-stage detectors are more efficient in their approach to object detection and classification, as they do not require filtering of the region proposals, making it faster but slightly less accurate than two-stage detectors.

In our work, we compare the performance of two one-stage detectors: RetinaNet [27] and YOLO [28]. Since our objective is to choose an architecture with real-time detection capabilities, we decided to develop model based on one-stage approach. Furthermore, we adopt the transfer learning approach to improve performance of our models by utilizing pre-trained models on other tasks [29]. Finally, we use the early stopping criteria to halt the training of a model when its performance no longer improves.

In figure 2, we show high level overview of RetinaNet and YOLO (You Only Look Once). RetinaNet is a popular one-stage detector which combines feature pyramid network (FPN) for detecting multi scale objects and focal loss to handle imbalance between background and foreground class. RetinaNet has two separate output detection head, one for the classification and one for the bounidng box regression. These heads are shared among all the features of the FPN. We used different backbone (ResNet50, CSPDarkNet, MobileNetV3, EfficientNetV2) architecture trained previously on ImageNet 2012 to gauge difference in performance and to establish baseline performance. Our goal is to select a backbone model which can be used to build a tool used by biologists with real-time performance.

YOLO is extremely fast, real-time one-stage detector where a single neural network is used to simultaneously predict multiple boxes and their classification probabilities. YOLO is relatively less accurate but incredibly fast object detection architecture, it is commonly utilized in security cameras. YOLO divides an entire image into a S x S grid, and if the center of a certain object falls within the grid cell, then this grid is responsible for detecting that specific object. Each grid cell can predict several bounding boxes along with their confidence scores and classes. The confidence score indicates whether or not the object has been detected. YOLO multiplies the class probabilities for each grid cell with confidence scores of the bounding box to obtain final detections. In our work, we use the most latest variation of YOLO called YOLOv8 with a small backbone YOLOV8 pretrained on COCO dataset. This architecture comes with two major modification, i.e., anchor-free detection and mosaic data augmentation.

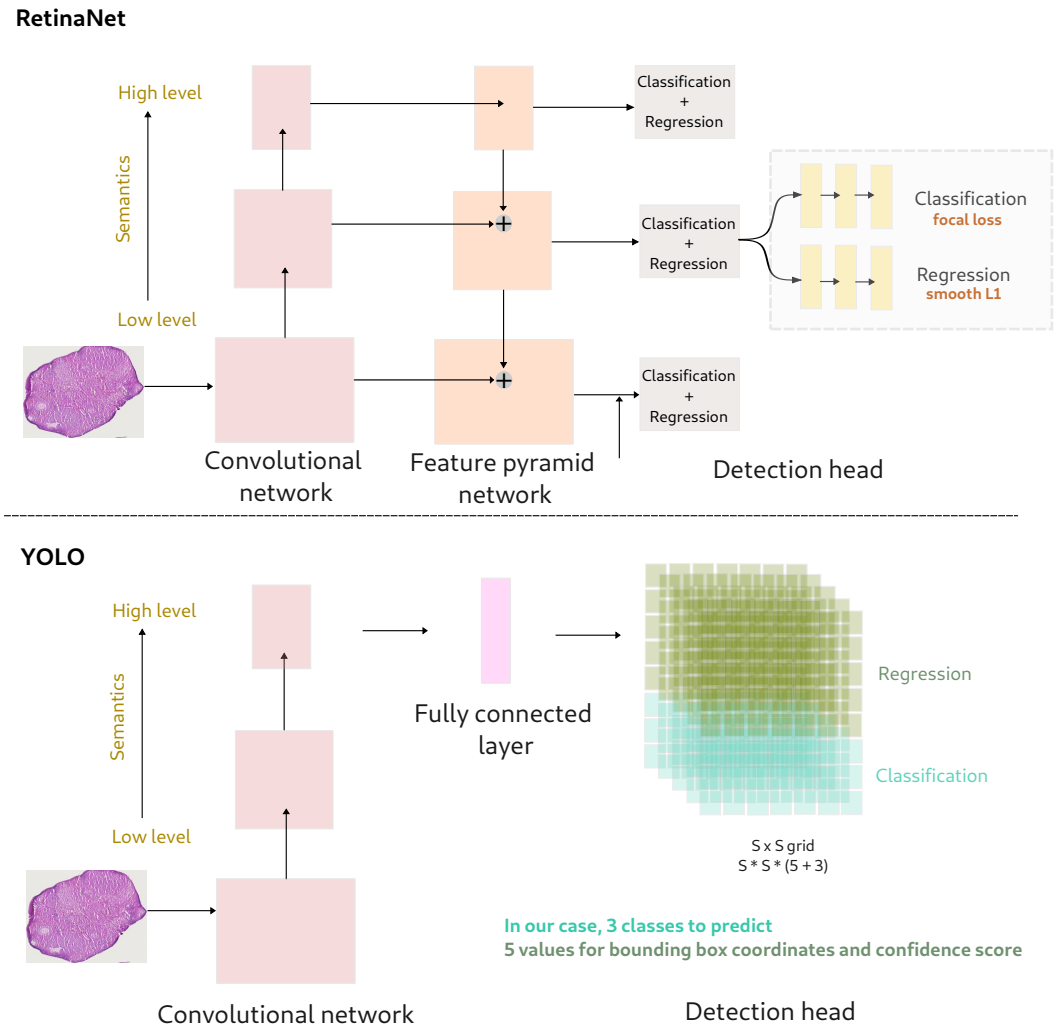


Figure 2: High level overview of one-stage detectors. RetinaNet employ feature pyramid network to extract multiple scales information. Detection heads takes information from multiple levels to perform prediction. YOLO takes the whole image, resize it, apply single convolutional network over it, and finally employs non-max suppression based on model's confidence scores.

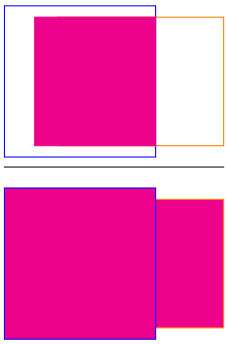
2.5 Evaluation metrics

In this paper, we use state-of-the-art metrics to evaluate the performance of our object detection models, i.e., average precision (AP). AP relies on precision, recall, and intersection over union (IoU) metrics.

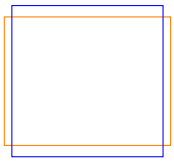
2.5.1 Intersection over Union

In object detection, we localize objects and predict their classes using boundary boxes. Object detection models take into account the quality of the predictions by calculating the intersection between the ground truth object box G and the predicted bounding box $\overline{G}$. IoU represents the ratio of the intersection over union between the ground truth and predicted boxes (see eq 1). We generally measure the performance of the object detection model using various IoU thresholds. In figure 3, we highlight different IoU thresholds, orange represents the predicted box while blue represents the ground truth box.

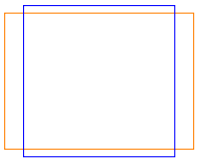
$$IoU(G, \overline{G}) = \frac{A(G \cap \overline{G})}{A(G \cup \overline{G})} \quad (1)$$



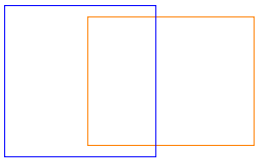
where A represents the area of the bounding box.



(a) Excellent with IoU = 0.95



(b) Good with IoU = 0.75



(c) Poor with IoU = 0.30

Figure 3: Intersection over Union (IoU) with different threshold showing the quality of prediction.

2.5.2 Precision and Recall

The term "precision" is used to describe how precise our model is, i.e., how many of the total detections for a given class actually belonged to that class. Recall refers to the number of cases of particular class instances that our model is able to predict out of the total number of ground truths for that particular class. There is usually a trade-off between precision and recall; increasing one value can result in a drop of the other. We aim to increase the precision and recall values as much as possible. The precision and recall are calculated using equations 2 and 3, respectively.

$$P = \frac{TP}{all; detections} = \frac{TP}{TP + FP} \quad (2)$$

$$R = \frac{TP}{all; ground; truths} = \frac{TP}{TP + FN} \quad (3)$$

A true positive (TP) represents the number of true predicted boxes where IoU is equal to or greater than a certain threshold. A false positive (FP) is a predicted bounding box that either does not match any ground truth box (IoU is below the threshold), incorrectly matches the class label, or is an extra detection when multiple boxes are predicted for the same object (only one is kept as TP). A false negative (FN) indicates the model's inability to identify an object present within the image, i.e., no predicted box overlaps with the ground truth box above the IoU threshold or the predicted box overlaps but the class is not correctly identified.

2.5.3 Average precision

Average precision (AP) can be used to summarize the precision and recall values into a scaler. It represent the area under the precision-recall curve and is calculated using equation 4 for each class:

$$AP = \int_{r=0}^{1} p(r)dr \quad (4)$$

where p is the precision and r is the recall.

We calculate mean average precision (MaP) using using eq 5:

$$MaP = \frac{1}{N} \sum_{i=1}^{N} AP_i \quad (5)$$

where N is the total number of classes, in our case 3, AP_i is the average precision for class i, and MaP is the average precision of all class's average precision.

3 Results

3.1 Image annotations

Since we are performing object detection task in a supervised manner, it is necessary to obtain labeled data of objects of interest in different histology images. We use CVAT to manually annotate images to identify categories of follicles and their boundary box coordinates. In figure 4, we show the annotation of one whole slide image in the CVAT software. Approximately 60 hours were required to annotate the whole data set. A total of 1373 antral follicles with nucleus, 1941 antral follicle without nucleus and 869 corpus luteum.

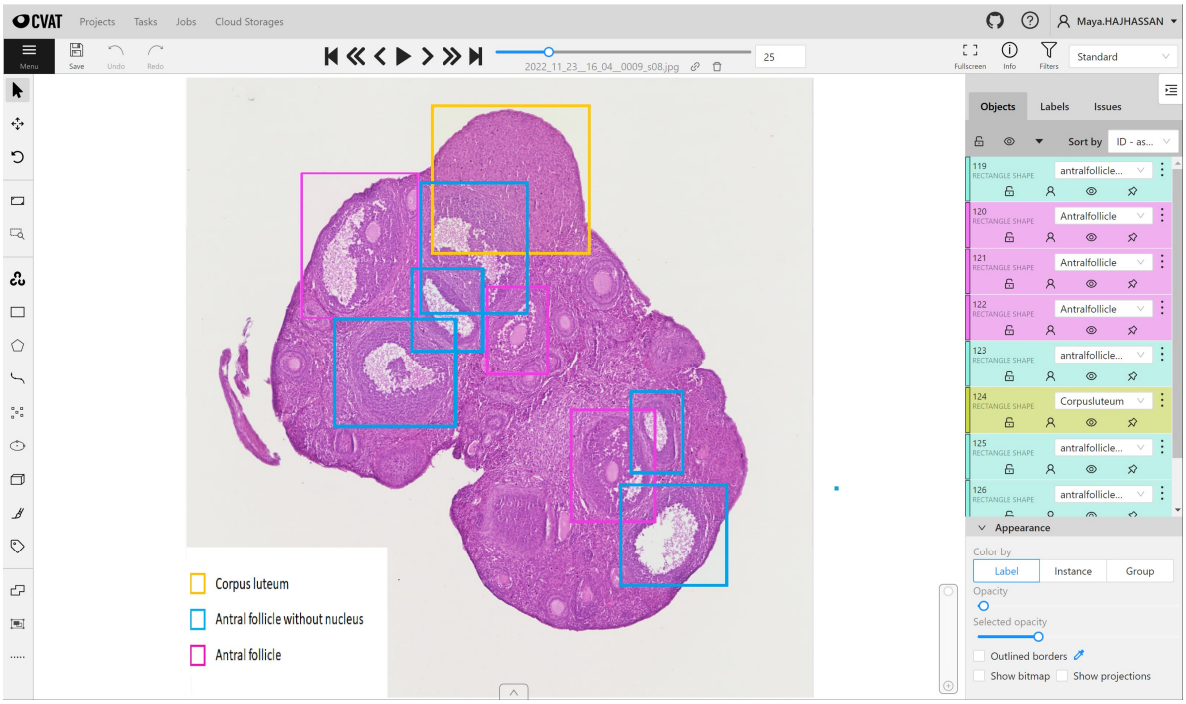


Figure 4: Annotation of whole slide images (WSI) using the Computer Vision Annotation Tool (CVAT). This figure illustrates the CVAT interface used for the annotation of ovarian structures for image analysis. The interface displays an image of an ovary section with annotated regions highlited by colored boxes. The yellow box indicated the area corresponding to the corpus luteum. The blue box delineates an antral follicle lacking a visible nucleus. The pink box represented an antral follicle with a discernible nucleus.

The input data of our model consists of four numeric values which represent where the object is located on the image and one more value that shows what kind of an object it is. Different formats can be used to represent boundary boxes. For example, one can define the boundary box by specifying a set of corner coordinates, width and height. Another way is to describe it using the coordinates of its top-left and bottom-right corners, i.e., $[x_1, y_1, x_2, y_2]$. To specify coordinates in our study, we use the $[x, y, w, h]$ format. The top-left corner is represented by two values x and y, width and height are represented by w and h values.

3.2 Data description

Our full dataset consists of 1209 ovarian sections. Within these sections, we counted 1373 antral follicles, 1941 antral follicles without oocytes, and 869 corpus lutea. The sampled dataset consists of 999 images with 1373 antral follicles, 1549 antral follicles without oocytes, and 869 corpus lutea objects. Both datasets were randomly divided into training (70%), validation (15%) and testing (15%) sets. In table 1, we show characteristics of our dataset.

	Full data			Sampled data		
	AF	**AFWO**	**CL**	**AF**	**AFWO**	**CL**
Training	981	1352	620	948	1100	650
Validation	199	297	102	232	233	92
Testing	193	292	147	193	216	127
Total	1373	**1941**	869	1373	**1549**	869

Table 1: Training, testing, and validation sets. AF : Antral Follicle, WO : Antral Follicle without oocytes, CL : Corpus luteum

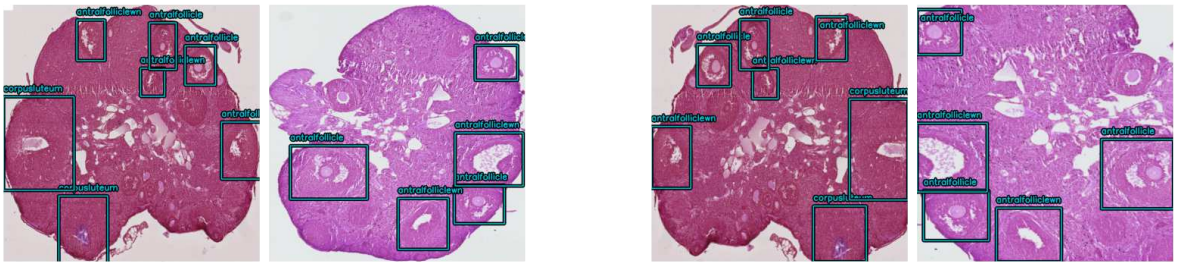
3.3 Preprocessing

We converted XML files obtained through the CVAT software to CSV files to perform object detection tasks. We use the OpenCV python library [30] to remove the excessive white patches from histology images.

It is well established that image resolution has a direct impact on the classification and localization of objects. In particular, it is difficult to detect small objects in low-resolution images. In general, for CNN-based detectors, images are down-sampled to a resolution of 256 x 256. Our initial findings confirmed that resolution does impact the performance of object detectors. We observed that high resolution, although computationally more expensive, does not translate systematically into improved performance, which has also been shown previously in other studies [31, 32]. In our case, we set the image resolution to 640 x 640 pixels.

To deal with the background-foreground class imbalance, we used focal loss [27]. This loss function modifies classic cross entropy function in a way that it down weights the loss contribution of well classified examples (background class) and quickly leading the model to focus on difficult examples. In order to deal with the object class imbalance, we used the data sampling technique. We kept only those images that had at least one of the antral follicle or corpus luteum objects, leading to the elimination of 210 images. Note that these images did not have any antral folllicle and corpus luteum objects. In this paper, we refer to this dataset as a sampled dataset.

Data augmentation in object detection models is more challenging and complex as compared to simple classification models as we must take into account the underlying bounding boxes during various transformations. We performed on-the-fly data augmentation using Keras-CV library [33], which propose native support for data augmentation with bounding boxes. We used random flip and jittered resize techniques for increasing the diversity of our dataset. In figure 5, we show two images from our training set before (see figure 5a) and after augmentation (see figure 5b).



(a) Before data augmentation

(b) After data augmentation

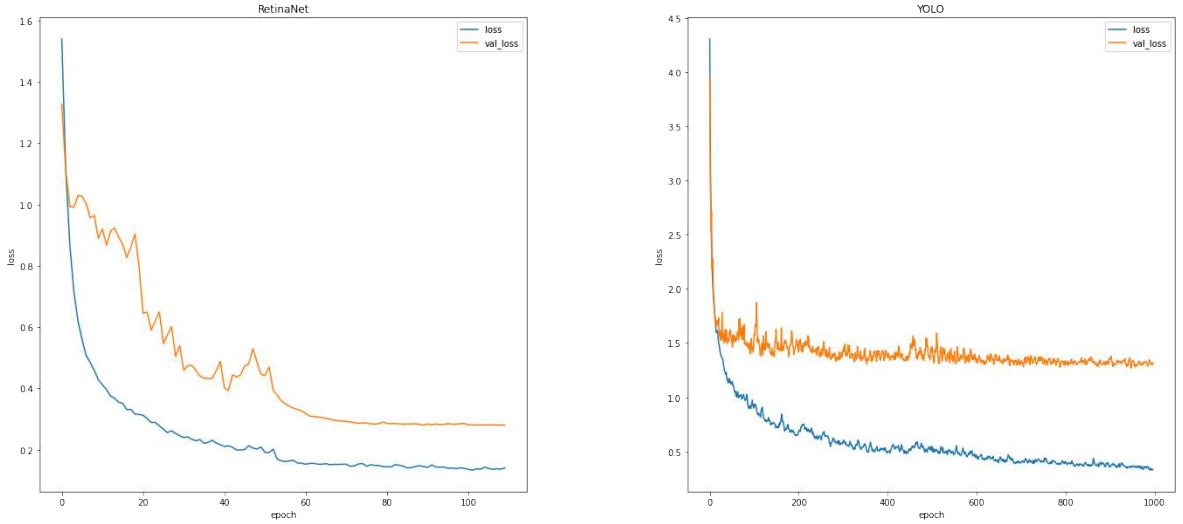
Figure 5: Images before and after data augmentation. We observe that the images were flipped and/or resized while preserving the bounding box coordinates during transformation.

3.4 Comparative models

The experiments were performed on a server with a NVIDIA A30 24GB PCIe NonCEC Accelerator GPU card. We build our dataset using tensor flow data API and models using keras-CV library [33].

The dataset is divided into training, validation and testing sets. During the training phase, the model parameters (weights and biases) are updated. During model training, the validation of the model is carried out by calculating the MaP using the validation dataset. In general, neural network training imply two main phases: (i) forward propagation and (ii) backward propagation. In the forward propagation, outputs of all nodes while moving from the input layer to the output layer are generated. At the output layer, error between the predicted output and the expected output is computed. In the second phase, the error is backpropagated to update the network parameters. These phases are iterated so as to minimize the final error by adjusting the

values of parameters. Once training is finished, the neural network can be used to generate predictions for the testing dataset, and several criteria are used to assess the accuracy of these predictions. In our case, we save the model with the highest MaP, and then later use it to make predictions on the testing dataset. We employed a stochastic gradient descent (SDG) optimizer with a batch size of 16. Exploding gradient is a common problem that arises when developing object detection models. The huge update to the model parameters caused by the high gradient values results in an unstable network. Gradient clipping, which relies on a threshold value, is used to cope with the exploding gradient problem. When a gradient value exceeds a predetermined threshold, the value is set to the threshold's value. After completion of the training phase, we use testing dataset to measure the model's generalizability. In figure 6, we show loss function of the object detectors, orange represents the validation loss and blue represents the training loss. We observe that there is a sudden change in gradient in the beginning of the training which stabilizes with time.



(a) RetinaNet with MobileNet as backbone

(b) YOLOv8 with coco weights

Figure 6: Graphs of loss function. The x-axis represents the number of epochs during the training process. The y-axis represents the value of the loss function. The goal during training is to minimize this value, as a lower loss indicates better alignment between the model's predictions and the actual data.

In table 2, we show comparison of full dataset and sampled dataset using RetinaNet detector. It takes $\approx$ $22h$ to train on a full dataset without any performance gain as can be observed from table 2, while it takes approximately $\approx$ $12h$ hours of training on sampled dataset. In addition to the computational overhead, model trained on the full dataset does not performed well on CL class as compared to the WO and AF class using validation and testing sets (IoU 0.50). Furthermore, our model built using sampled dataset is able to classify the remaining 200 images with a MaP of 0.83 (IoU 0.50) which were initially deleted from the dataset to create a more balanced dataset. We include the results with two different IoU threshold, i.e., 0.50 and 0.75. It is worth noting that different thresholds can lead to some side effects. For example in case of high threshold, we may filter or eliminate predicted boxes for overlapping objects, which is not a desirable behavior. There is no ideal IoU threshold value, it varies between 0 and 1. In practice, the common IoU threshold is 0.5 and may need adaptation in a particular context to obtain meaningful outcomes.

	IoU	Full dataset				Sampled dataset			
		AF	AFWO	CL	Avg	AF	AFWO	CL	Avg
Training	0.50	0.94	0.94	0.95	0.94	0.94	0.95	0.95	0.95
	0.75	0.82	0.83	0.81	0.82	0.81	0.83	0.84	0.83
Validation	0.50	0.84	0.81	0.77	0.80	0.87	0.85	0.85	0.86
	0.75	0.63	0.60	0.54	0.59	0.65	0.61	0.57	0.61
Testing	0.50	0.80	0.79	0.75	0.78	0.86	0.84	0.79	0.83
	0.75	0.60	0.61	0.53	0.58	0.60	0.54	0.50	0.55

Table 2: The MaP score with different IoUs on training, validation, and testing set.

As a sanity check, we also trained the model without any data augmentation to ensure the robustness of our model. The initial results of this experiment indicate a noticeable overfitting problem; as our model is able to achieve MaP of 0.99 (IoU 0.50) on the training dataset However, these impressive results raise a concerns about generalizability of the model, i.e., predictions quality on the unseen dataset. Taking into account this analysis, we applied on-the-fly data augmentation techniques, which effectively improved the model's generalizability on the unseen dataset. Furthermore, to optimize the computational resources, we converted images into grey-scale. Our results did not show any significant performance gains in the context of our dataset and model.

In table 3, we show comparison of YOLO and RetinaNet detectors with different backbone architectures. When it comes to computing speed, YOLO outperforms RetinaNet by a wide margin. It requires roughly only one hour to complete the training. RetinaNet with MobileNet as backbone surpassed YOLO on the testing dataset with a MaP of 0.86. As compared to RetinaNet, the confidence threshold for predicted bounding boxes was quite low. Additionally, YOLO achieved a MaP of 0.71 (IoU 0.50) on the remaining dataset (absent from the training, testing and validation sets) while RetinaNet scored a MaP of 0.83 (IoU 0.50). We believe that the RetinaNet is a suitable choice where computational time is not an issue and the top priority is MaP. However, where we can tolerate slightly lower MaP for the sake of faster performing predictive algorithm, a YOLO model is more suitable option.

Detector	**Backbone**	**AF**	**AFWO**	**CL**	**MaP**
RetinaNet	MobileNetV3	0.86	0.84	0.79	0.83
RetinaNet	ResNetV1	0.81	0.77	0.76	0.78
RetinaNet	EfficientNetV2	0.80	0.78	0.79	0.79
RetinaNet	CSPDarkNet	0.79	0.77	0.76	0.77
YOLOv8	YOLOV8	0.79	0.73	0.73	0.75

Table 3: MaP score on testing dataset of RetinaNet with different backbone networks and YOLO detector with YOLOV8 backbone architecture.

In figure 7, we show the results on the testing dataset. In blue are the original annotations and in yellow are the predictions obtained through RetinaNet with MobileNetV3 as backbone. We are using the batch size of 8, IoU threshold and confidence score of 0.75. We observe that our model failed to predict two objects in figure 7b and one in figure 7c. Figure 7d displays an antral follicle lacking a nucleus, intentionally left unannotated by the expert but correctly labeled by the detector. Despite being located within a distorted region, the follicle was accurately detected. This distortion is attributed to artifacts arising from slide mounting. Although the artifact affects the region, it does not significantly alter the structure's shape, allowing for the follicle's identification through the predictions. Figure 7f shows an antral follicle that was forgotten by the annotator but detected correctly by the trained model. These were considered as FPs in our evaluation criteria, which means we penalized our model for detecting these follicles. These detections highlight the labeling errors. In future research, we plan to enrich the dataset by correcting these errors which will help to further optimize the training of our models and to create a more general model.

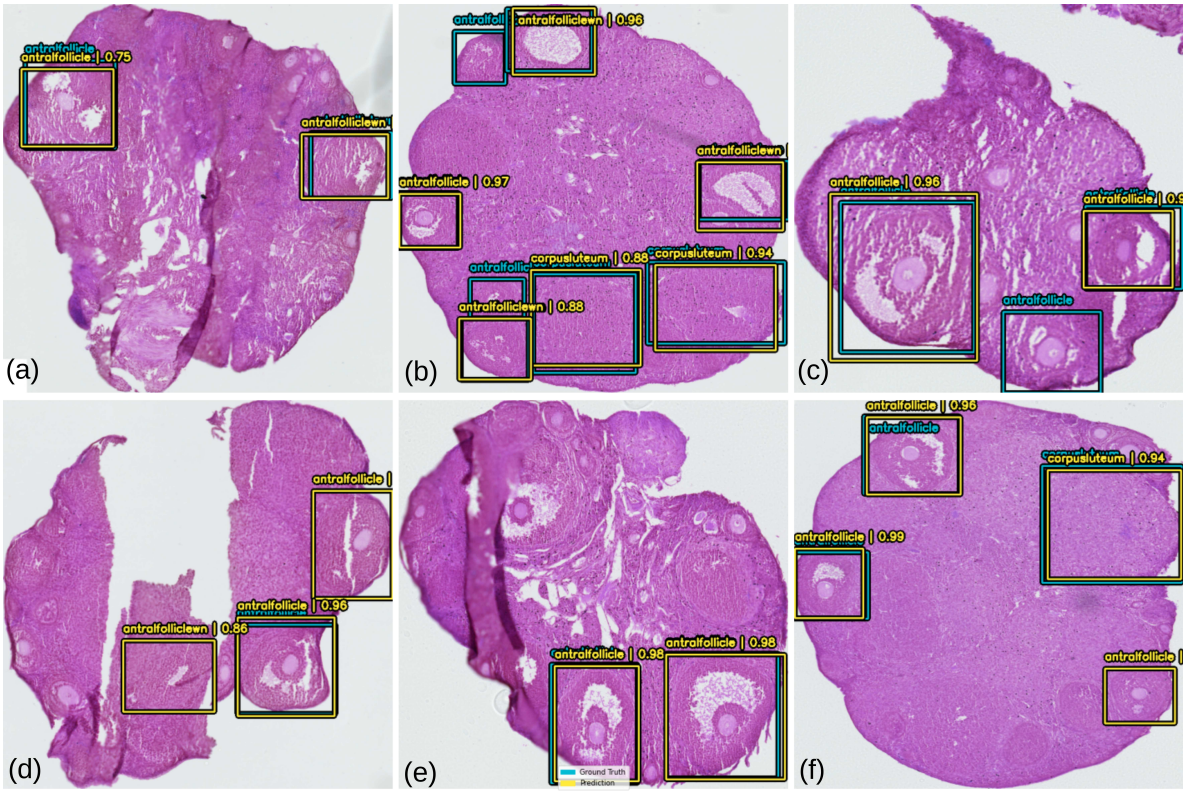


Figure 7: The predictions on validation set of batch size of 8 with IoU of 0.75 and confidence threshold of 0.75. Here blue boxes represent original annotations and yellow are the predictions with a confidence score.

In figure 8, we show the counting by expert and predictions by our model on the testing dataset with IoU threshold of 0.75 and 0.50. The blue color represents the follicle counting performed by the expert, orange represents the predictions by the model, pink signifies the TPs, and green shows the FPs. We see that the TPs rate increases and FPs rate decreases with the most common IoU threshold of 0.50. This figure highlights the effects of IoU threshold which requires careful consideration for different applications.

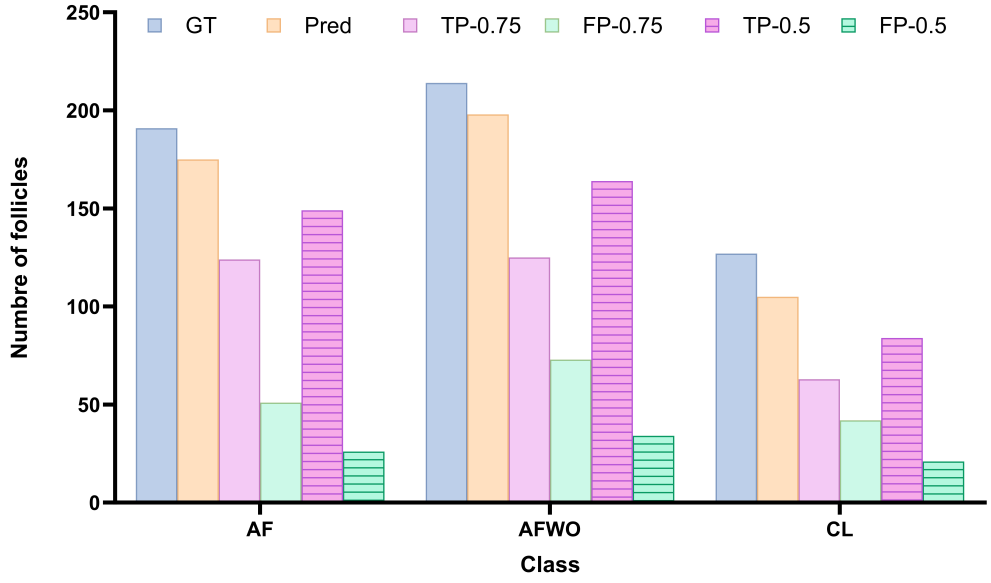


Figure 8: Follicles counting by the expert and the model on the testing dataset.

Furthermore, we analyzed images manually to understand logic behind the predictions of our models, which is crucial for further studies to understand why errors occur and how to address them. This comparative analysis of expert and model identification helps to elucidate the strengths and limitations of AI-driven image classification in the context of ovarian tissue analysis. We examined 151 images manually from the testing set to identify where the model misclassified, confused, or failed to label follicles correctly. What we found particularly intriguing and promising was the agreement between the number of false positives predicted by the model and those by the operator, especially for the antral follicle and the corpus luteum, the two main important structures for us. Upon closer examination, we observed that not all discrepancies were genuine errors made by the model (see figure 9).

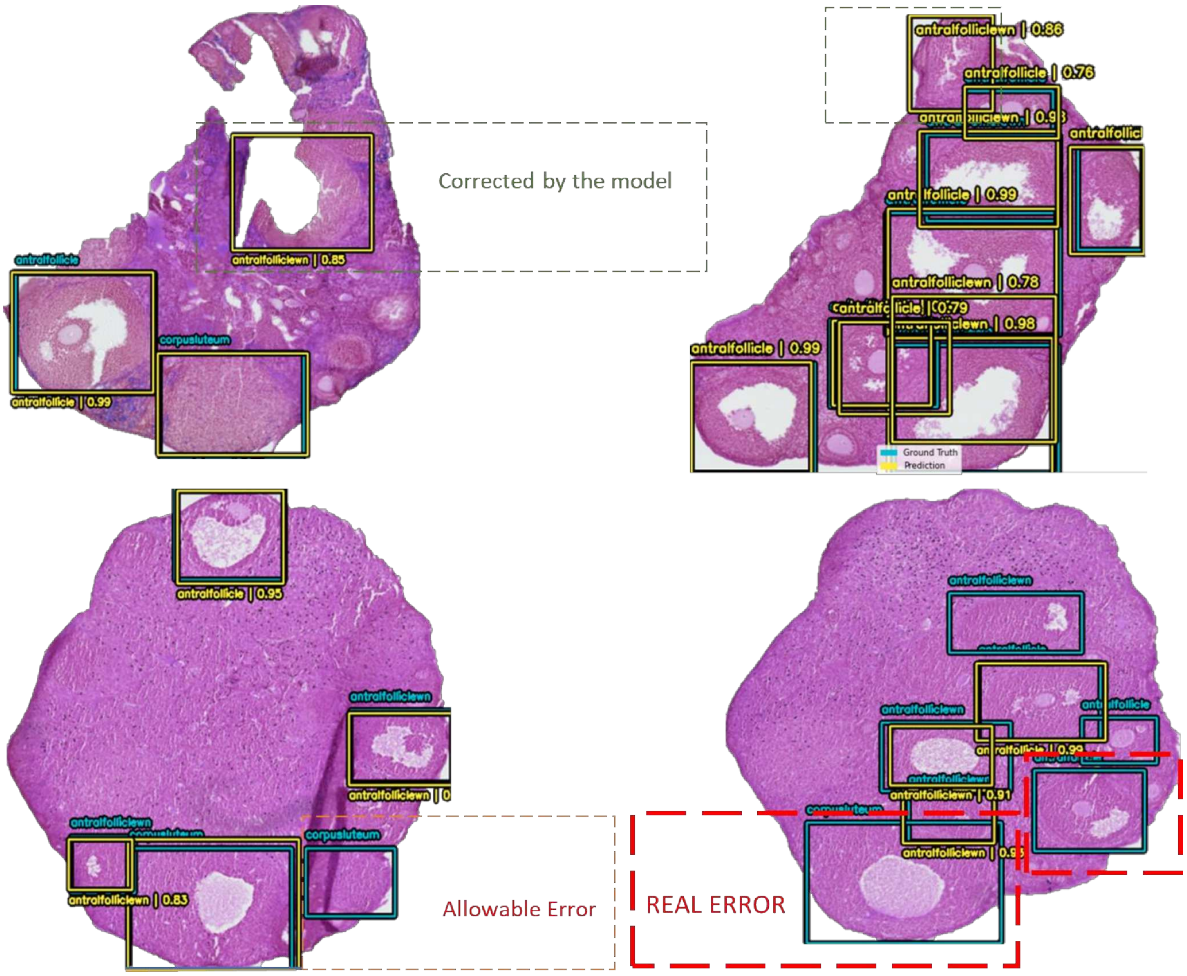


Figure 9: Errors analysis of expert and model identification of ovarian structures. This figure presents section of ovarian tissue with structures identified by both human expert (Blue bounding boxes) and AI model (Yellow bounding boxes). The expert categorized discrepancies into three types of errors: 1) The geniune or real errors (Red boxes) : structures identified by experts but missed by the model. These are clear structures that the model should have recognized but failed to do so, 2) Allowable erros (Light red boxes) : structures recognized by the expert but not labeled by the model due to staining inconsistencies, mounting issues, scanning artefacts, or other technical factors, and 3) Errors corrected by the model (Green Boxes) : structures overlooked by the expert due to factors such as eye fatigue or intentionally unlabelled (tears in the sections) but correctly identified by the AI model. These instances higlight the model's ability to detect structures that may have been missed or intentially omitted by the human expert.

Some regions posed challenges for labeling due to staining inconsistencies, mounting issues, and scanning artifacts, making them difficult even for the operator to label accurately. We categorized these instances as allowable errors given the challenges in image acquisition and processing. Despite these challenges, the system demonstrated an ability to detect and correctly classify some follicles that were missed by the experimenter.

Thereby, after reclassification of these errors into those corrected by the model, those allowable by the system, and genuine/real errors, we noted a reduction in the number of false positives (see figure 10). This is expected to significantly enhance precision and consequently improve the accuracy of the model.

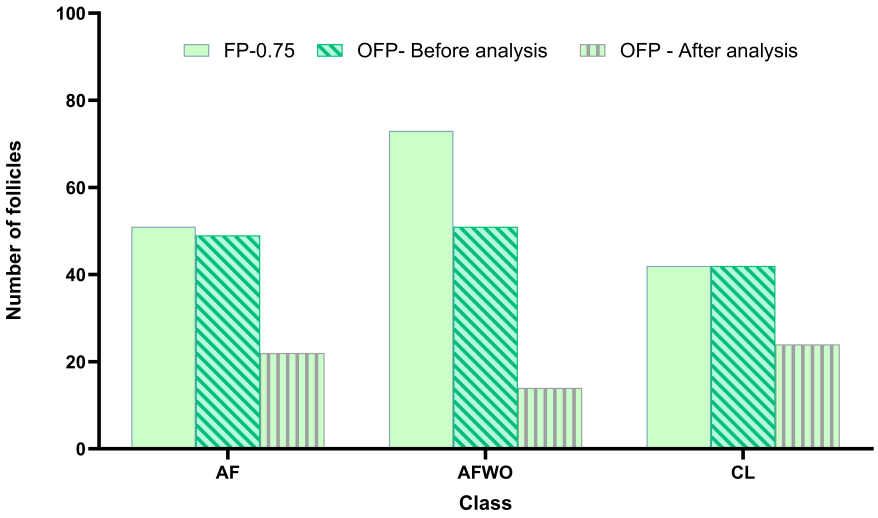


Figure 10: Comparative analysis of False positives (FP) detected by the model and manually by the operator before and after analysis of 151 images from the testing set. The light green bars reprensent the number of FP identified by the model with a precision of 0.75. The green bars with diagonal lines (OFP- Before analysis) depict the number of FP manually counted by the operator upon initial comparaison of the images. The green bars with vertical lines show the number of FP (OFP- After analysis) counted by the operator after reclassifcation of erros between Geniune, allowable and corrected errors. The decrease in the number of FP observed after manual examination and eror correction by the operator demonstrates the potential of enhanced precision and accuracy of the model.

4 Discussion

In this paper, we present a valuable technique to count ovarian follicle from whole slide images (WSI) using transfer learning, early stopping and data augmentation approaches. We annotated the WSI of 20 mice ovaries in CVAT. From these annotations, a model trained; validated and tested itself to distinguish between three different classes of follicles: the antral follicle, the antral follicle without nucleus, and the corpus luteum. We used state-of-the-art one-stage object detection methods, i.e, YOLO and RetinaNet for follicles detection in histology images. We achieved a MaP of 0.83 on the testing dataset with RetinaNet. Furthermore, we identified cases where model was able to correct erros of the annotators.

Histological counting may be one of the most standardized approaches for assessing ovarian reserve and follicle status, it does come with certain limitations: 1) Subjectivity of the operator : even with the presence of skilled personnel, there is a potential inter-observer variability due to the manual nature of histological counting. This variability may lead to inconsistencies in follicle counts, 2) Time-consuming and tedious work: the histological process requires careful preparation, sectioning, staining and examination. Each step in the preparation may affect the accuracy and quality of follicle counting. Studies show that we should take in consideration the impacts of different fixatives (Bouin's fixative versus formalin), embedding material and section thickness [34]. They do not have the same effects on tissue structure. Despite the common use of formalin, Bouin's solution may preserve the cellular morphology better than formalin that may cause tissue shrinkage and consequently changes in follicle dimensions. Results may not be fully representative of the entire ovary if sections at a regular interval were analyzed. Correction factors were used in such cases [19]. In our case, we used Bouin's solution to fix the collected ovaries, a standardized hematoxylin and eosin staining protocol that selectively highlight the different aspects of tissue. The entire ovary was cut and almost all the slides were digitized. There was not any specific selection of the slides which reduce the introduction of sample bias. The use of a scanner helped to digitize faster and with high resolution the Whole Slide Images. The contrast quality of the images improved the ability to analyze. Despite the efforts, staining inconsistency, mounting or scanning artefacts may continue to exist. Hence, some structures were intentionally unlabelled by the experiment especially on slides with mounting or scanning artefact. This was not a problem for the

model, we noticed that the model was able to predict correctly the follicles in some poorly prepared sections with distorted aspect. Some experimenter labeling errors were found demonstrating that fatigue can contribute to errors and were discovered when comparing results obtained through the automated method with manual histological labeling. The scientists need to be always on a focused state for accurate results. One more reason to have at least two experimenters to assess follicle counting. Using AI and mainly this model can indeed reduce the need for multiple experimenters. The model labeled correctly almost all the structures but it only missed some blur structures (mounting or scanning artefacts) that the experimenter was able to annotate when checking the successive slide.

Taking in consideration these limitations is crucial for scientists when designing and interpreting results. Integrating histological counting with additional methodologies like hormone dosage and embracing new technologies, such as artificial intelligence and deep learning can help as we can see to overcome some of the challenges of ovarian follicle counting. Histological counting is often used as a benchmark for validating automated techniques, including those involving artificial intelligence. Comparing results obtained through automated methods with manual histological counting helps ensure the accuracy and reliability of the automated approach. Understanding the nature of errors and discrepancies is crucial for refinning AI algortithms and optimizing their performance in biomedical imaging applications. The model shown in this paper, offers solutions to enhance accuracy, efficiency, and reproducibility of follicle counting, when comparing the predicted labelling and the annotated one.

There are some limitations of this study: 1) the experimental dataset is collected from the same laboratory which means we may not have enough diverse images in our training dataset, 2) optimal threshold to count the true or false positives remains dependent on the application, and 3) black-box nature of the deep learning models which means the thought process behind a particular decision or prediction of DL models is humanly non-interpretable due to complex non-linear internal structure and over parametrization.

In the near future, we want to create a larger and more diverse dataset by collaborating with other labs to create a more general detection models and to futher improve the MaP score. By inspecting the results manually, as discussed above, our model was able to produce predictions where annotator failed to label. In the current model, we penalize it for detecting these structure. In future, we will correct these errors and based on these intuitions, we will perform experiments to define the optimal threshold for each class separately to count the true number of true or false positives.

In a biological context, it is useful to understand the internal workings of DL models and to identify the most essential features or reasoning for the classification or regression tasks. Fortunately, many methods have been developed in the last decade to tackle the problem of the explainability of DL models, such as feature relevance, local or global explanations, and visualizations [for a review, see ([35, 36])]. These approaches, however, are not immediately translatable to the object identification tasks to elucidate decision-making processes. In the case of classification tasks, our output is normally scalar; however, in the case of object detection, the output is multiple bounding boxes per image, i.e., the bounding box coordinates of the detected objects and their category. The task is further complicated because of non-max suppression which is applied to remove overlapping boxes and to keep one box per object. It implies that we cannot translate the input to the output via the usual gradient, which hinders our ability to technically apply methods such as DeepLift, ShAP, etc. Furthermore, we also need explanation not just for the category but also for the location of the objects. Generally, there has been a limited effort to tailor explanation methods for object detection models [37, 38, 39, 40]. These approaches cannot systematically be applied to our models because these methods often depend on the specific characteristics of the object detectors, such as one-stage or two-stage detectors, presence of anchor boxes [37] (our YOLO-based model is anchor-free), absence of individual explanations for the object category and bounding box [40]. The method proposed in [38] is the model agnostic to generate heatmaps for highlighting the important parts of the image, however it is computationally slow and lacks evaluation of the explanation of bounding boxes. In future, we will work on explainability of our object detectors to identify the causes of its predictions for model validation and knowledge discovery. We believe that these insights will help to further improve the performance and usability of our models. It will make them transparent by highlighting the important segments of the images and help to compare between the expert's and model's intuitions.

Acknowledgments

This research was funded with the support of the Institut National de la Recherche pour I'Agriculture, I'Alimmentation et I'Environnement (INRAE), Phase department innovative project, and Bill & Melinda Gates Foundation CONTRABODY grant. We also acknwowledge the support recieved from the dedicated team in the rodent animal facility, experimental unit: UEPAO (PAO, INRAE: Animal Physiology Facility https://doi.org/10.15454/1.5573896321728955E12) for their attention to details and commitment to animal welfare. This work also benefited from the equipment and expertise of the imaging facility Plateau d'Imagerie Cellulaire (PIC) of UMR-PRC.

References

[1] Torben Pedersen and Hannah Peters. Proposal for a classification of oocytes and follicles in the mouse ovary. Reproduction, 17(3):555–557, 1968.

[2] Robert Hadek. The structure of the mammalian egg. International review of cytology, 18:29–71, 1965.

[3] Wei Ge, Lan Li, Paul W Dyce, Massimo De Felici, and Wei Shen. Establishment and depletion of the ovarian reserve: physiology and impact of environmental chemicals. Cellular and Molecular Life Sciences, 76:1729–1746, 2019.

[4] Eleftheria M Panagiotou, Venla Ojasalo, and Pauliina Damdimopoulou. Phthalates, ovarian function and fertility in adulthood. Best Practice & Research Clinical Endocrinology & Metabolism, 35(5):101552, 2021.

[5] Stephanie Morgan, RA Anderson, C Gourley, WH Wallace, and N Spears. How do chemotherapeutic agents damage the ovary? Human reproduction update, 18(5):525–535, 2012.

[6] Pietro Santulli, Diane de Villardi, Vanessa Gayet, Marie-Christine Lafay Pillet, Louis Marcellin, Valerie Blanchet, Julia Gonnot, Emmanuel Duliousť, Odile Launay, and Charles Chapron. Decreased ovarian reserve in hiv-infected women. Aids, 30(7):1083–1088, 2016.

[7] Linlin Cui, Yan Sheng, Mei Sun, Jingmei Hu, Yingying Qin, and Zi-Jiang Chen. Chronic pelvic inflammation diminished ovarian reserve as indicated by serum anti mülerrian hormone. PloS one, 11(6):e0156130, 2016.

[8] Amy L Winship, Monika Bakai, Urooza Sarma, Seng H Liew, and Karla J Hutt. Dacarbazine depletes the ovarian reserve in mice and depletion is enhanced with age. Scientific Reports, 8(1):6516, 2018.

[9] Amy L Winship, Meaghan Griffiths, Carolina Lliberos Requesens, Urooza Sarma, Kelly-Anne Phillips, and Karla J Hutt. The parp inhibitor, olaparib, depletes the ovarian reserve in mice: implications for fertility preservation. Human Reproduction, 35(8):1864–1874, 2020.

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. Communications of the ACM, 60(6):84–90, 2017.

[11] Artúr István Károly, Péter Galambos, József Kuti, and Imre J Rudas. Deep learning in robotics: Survey on model structures and training strategies. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2020.

[12] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In 2013 IEEE international conference on acoustics, speech and signal processing, pages 6645–6649. IEEE, 2013.

[13] Travers Ching, Daniel S Himmelstein, Brett K Beaulieu-Jones, Alexandr A Kalinin, Brian T Do, Gregory P Way, Enrico Ferrero, Paul-Michael Agapow, Michael Zietz, Michael M Hoffman, et al. Opportunities and obstacles for deep learning in biology and medicine. Journal of The Royal Society Interface, 15(141):20170387, 2018.

[14] Seonwoo Min, Byunghan Lee, and Sungroh Yoon. Deep learning in bioinformatics. Briefings in bioinformatics, 18(5):851–869, 2017.

[15] Muhammad Imran Razzak, Saeeda Naz, and Ahmad Zaib. Deep learning for medical image processing: Overview, challenges and the future. In Classification in BioApps, pages 323–350. Springer, 2018.

[16] Misbah Razzaq, Maria Jesus Iglesias, Manal Ibrahim-Kosta, Louisa Goumidi, Omar Soukarieh, Carole Proust, Maguelonne Roux, Pierre Suchon, Anne Boland, Delphine Daiain, et al. An artificial neural network approach integrating plasma proteomics and genetic data identifies plxna4 as a new susceptibility locus for pulmonary embolism. Scientific Reports, 11(1):14015, 2021.

[17] Christof Angermueller, Tanel Pärnamaa, Leopold Parts, and Oliver Stegle. Deep learning for computational biology. Molecular systems biology, 12(7):878, 2016.

[18] Eric J Topol. High-performance medicine: the convergence of human and artificial intelligence. Nature medicine, 25(1):44–56, 2019.

[19] Jonathan L Tilly. Ovarian follicle counts–not as simple as 1, 2, 3. Reproductive Biology and Endocrinology, 1:1–4, 2003.

[20] Michelle Myers, Kara Louise Britt, Nigel Glen M Wreford, Francis JP Ebling, and Jeffrey Bryce Kerr. Methods for quantifying follicular numbers within the mouse ovary. Reproduction, 127(5):569–580, 2004.

[21] Charlotte Sonigo, Stéphane Jankowski, Olivier Yoo, Olivier Trassard, Nicolas Bousquet, Michael Grynberg, Isabelle Beau, and Nadine Binart. High-throughput ovarian follicle counting by an innovative deep learning approach. Scientific reports, 8(1):13499, 2018.

[22] Özkan İnik, Ayşe Ceyhan, Esra Balcıoğlu, and Erkan Ülker. A new method for automatic counting of ovarian follicles on whole slide histological images based on convolutional neural network. Computers in biology and medicine, 112:103350, 2019.

[23] CVAT.ai Corporation. Computer Vision Annotation Tool (CVAT), September 2022.

[24] James F Mullen Jr, Franklin R Tanner, and Phil A Sallee. Comparing the effects of annotation type on machine learning detection performance. In Proceedings of the ieee/cvf conference on computer vision and pattern recognition workshops, pages 0–0, 2019.

[25] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep learning. MIT press, 2016.

[26] Kemal Oksuz, Baris Can Cam, Sinan Kalkan, and Emre Akbas. Imbalance problems in object detection: A review. IEEE transactions on pattern analysis and machine intelligence, 43(10):3388–3415, 2020.

[27] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In Proceedings of the IEEE international conference on computer vision, pages 2980–2988, 2017.

[28] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 779–788, 2016.

[29] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. IEEE Transactions on knowledge and data engineering, 22(10):1345–1359, 2009.

[30] Itseez. Open source computer vision library. https://github.com/itseez/opencv, 2015.

[31] Carl F Sabottke and Bradley M Spieler. The effect of image resolution on deep learning in radiography. Radiology: Artificial Intelligence, 2(1):e190015, 2020.

[32] Vajira Thambawita, Inga Strümke, Steven A Hicks, Pål Halvorsen, Sravanthi Parasa, and Michael A Riegler. Impact of image resolution on deep learning performance in endoscopy image classification: an experimental study using a large dataset of endoscopic images. Diagnostics, 11(12):2183, 2021.

[33] Luke Wood, Zhenyu Tan, Ian Stenbit, Jonathan Bischof, Scott Zhu, François Chollet, et al. Kerascv. https://github.com/keras-team/keras-cv, 2022.

[34] Urooza C Sarma, Amy L Winship, and Karla J Hutt. Comparison of methods for quantifying primordial follicles in the mouse ovary. Journal of Ovarian Research, 13:1–11, 2020.

[35] Arun Rai. Explainable ai: From black box to glass box. Journal of the Academy of Marketing Science, 48:137–141, 2020.

[36] Guang Yang, Qinghao Ye, and Jun Xia. Unbox the black-box for the medical explainable ai via multi-modal and multi-centre data fusion: A mini-review, two showcases and beyond. Information Fusion, 77:29–52, 2022.

[37] Deepan Chakravarthi Padmanabhan, Paul G Plöger, Octavio Arriaga, and Matias Valdenegro-Toro. Dext: Detector explanation toolkit. In World Conference on Explainable Artificial Intelligence, pages 433–456. Springer, 2023.

[38] Vitali Petsiuk, Rajiv Jain, Varun Manjunatha, Vlad I Morariu, Ashutosh Mehra, Vicente Ordonez, and Kate Saenko. Black-box explanation of object detectors via saliency maps. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 11443–11452, 2021.

[39] Apostolos Karasmanoglou, Marios Antonakakis, and Michalis Zervakis. Heatmap-based explanation of yolov5 object detection with layer-wise relevance propagation. In 2022 IEEE International Conference on Imaging Systems and Techniques (IST), pages 1–6. IEEE, 2022.

[40] Denis Gudovskiy, Alec Hodgkinson, Takuya Yamaguchi, Yasunori Ishii, and Sotaro Tsukizawa. Explain to fix: A framework to interpret and correct dnn object detector predictions. arXiv preprint arXiv:1811.08011, 2018.