CHAPTER 4

ENDOSCOPIC ARTEFACT DETECTOR

4.1 INTRODUCTION

Revolutionary object detection models are used to find artefacts in endoscopic images. It can be grouped into a non-neural network approach and a neural network approach. The former includes Voila-Jones, Histogram of Oriented Gradients (HOG), Deformable Part Models (DPM) and Scale-Invariant Feature Transform (SIFT). The neural network approach comprises YOLO, SSD, R-CNN, Fast R-CNN and Faster R-CNN. Based on the architecture, neural network algorithms that depend on DL are classified into single-stage, two-stage, multi-stage and anchor-free object detectors.

The single-stage object detector predicts the class probabilities and makes bounding boxes in one run, hence the name single-stage object detection. YOLO is one among such detector. As soon as the input is passed into the algorithm, YOLO divides the image into smaller grids of 3x3. It tries to make meaningful predictions in every small grid. If found, it draws a bounding box and predicts the class probabilities. YOLO algorithm is meant for its high speed. It is comparatively accurate as two-stage detectors.

R-CNN, a two – stage object detector work based on the following principle The algorithm receives the input image. First, the model proposes a RoI using selective search. CNN takes all these proposals in a fixed shape. Hence all are reshaped and passed into the ConvNet. Based on the features extracted by the CNN, a bounding box regressor is used to draw a bounding box over the region is identified. A classifier is used to classify the predicted regions into classes. All the steps described consume more time. Hence making predictions for real-time applications is almost impossible. The major improved versions of R-CNN include Fast R-CNN, faster R-CNN and mask R-CNN. The enhanced versions are said to perform faster with the better accuracy.

Multi-stage object detectors are an extended style of faster R-CNN, striving to achieve better accuracy than two-stage detectors and also faster than single-stage detectors. Cascaded R-CNN is an elite example of a multi-stage object detector.

The traditional object detection algorithms work based on anchors to detect the objects. An anchor-free object detector uses key-point detection instead of anchors. It generates a heat map to achieve the same. Such detectors struggle to segment dense and overlapping object.

4.2 METHODOLOGY

This research uses three well-known object detection algorithms to detect endoscopic artefacts. The algorithms are YOLOv3, YOLOv4 and Faster R-CNN. YOLOv3 uses Darknet-53 as backbone. The algorithm is said to have improved mAP and IoU. YOLOv4 is an upgraded version of YOLOv3. It uses CSPDarknet53 as the backbone. Faster R- is known for accuracy. Faster R-CNN uses ResNet 50 as backbone.

In most of the research outcomes in the literature compare and calculate the algorithm's performance using mAP and IoU. Accuracy and inference time are equally important to deploy any application into medical imaging and diagnosis. Hence, accurate and fast object detection models are selected for this research. All three models considered are trained and ensembled for final predictions.

4.3 PROPOSED ENSEMBLE METHOD FOR ENDOSCOPIC ARTEFACT DETECTION

The architecture and the training procedure of all three futuristic object detection models, namely YOLOv3, YOLOV4 and Faster R-CNN, and the proposed ensemble model architecture are discussed in this section. The datasets are initially divided into train, validation and test. The training and validation set of images are data augmented. The augmented dataset is used to train all the three models. The images in the train and validation set are retained uniform across all the three models. Google Co-laboratory's high-performance GPU is used for rigorous training. All three models and the final ensemble model is coded using python programming language.

Other models trained for artefact detection include Cascaded R-CNN, RetinaNet and Key point R-CNN. The combination of YOLO and R-CNN yielded better results compared to other combinations. Thus, the detection models used for proposed ensemble method is

separately discussed in this chapter. The common steps to perform artefact detection are described in the following block diagram shown in Figure 4.1.

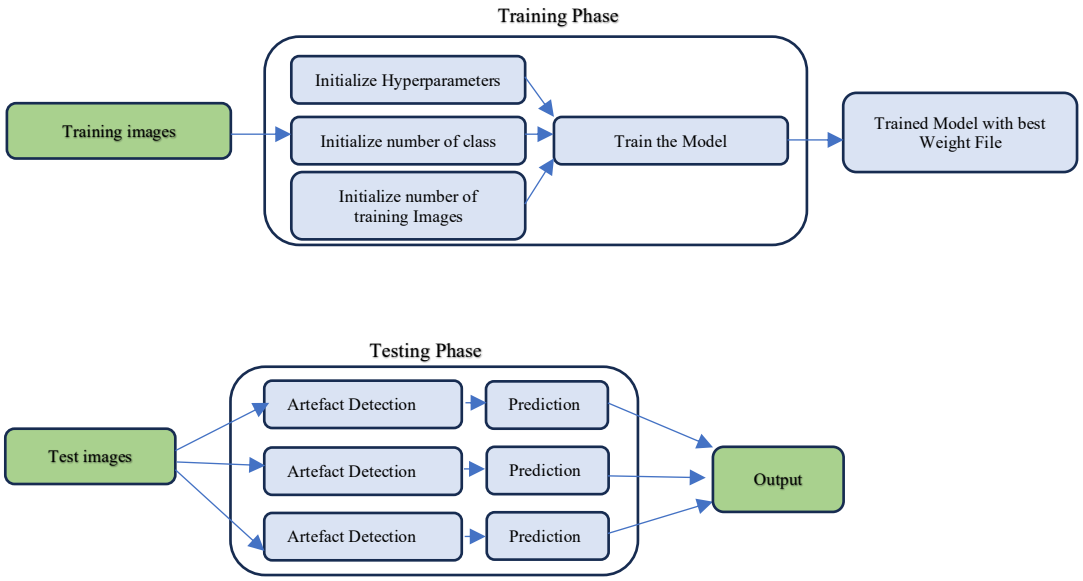


Figure 4.1 Training and Testing Phase of Detection Models

4.3.1 YOLOv3 for Endoscopic Artefact Detection

YOLO first came into sight in the year 2016 followed by many updated versions of YOLO in the later years. YOLOv3 is renowned for having the fastest performance. It is used in a wide variety of real-time applications. YOLOv3 divides the entire image into tiny grids after taking a single, full look at the image. If there is a significant object inside each gird, boundary box will be drawn. Thus, it is claimed that its predictions are global. Calculations are made to determine how closely the predictions resemble the stated classes. A positive detection is thought to have occurred when the score is high.

Darknet53 is used as the backbone for YOLOv3. The backbone is used to extract features as it is comparatively faster and accurate than all the other backbones. Darknet (https://pjreddie.com/darknet/) is an open-source neural network. It is written using languages such as, C and Computer Unified Device Architecture (CUDA). It supports both GPU and Central processing Unit (CPU) computations. Figure 4.2 shows the Darknet 53 backbone. The term Darknet 53 is because it has 53 successive 3 x 3 and 1 x 1 convolutional layers. The convolutional layers are the main building blocks of the network. This makes the backbone

more powerful. The backbone is proved to be 1.5 times faster than ResNet-101 and also the backbone is accurate as ResNet-152 (Redmon & Farhadi, 2018). Hence Darknet 53 backbone is chosen. These 53 layers are pretrained using images from ImageNet dataset. The trained network can perform 1457 billion floating point operations per second making it faster than all existing network.

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Figure 4.2 Darknet 53 Backbone of YOLOv3 (Redmon & Farhadi, 2018)

Another 53 more layers are stacked upon the darknet 53 network. Hence the total number of layers becomes 106. This 106-layer rich network is now termed as YOLOv3. This network incorporates skip connections with concatenations. This characteristic is advantageous in detecting small artefacts like specular reflections and artefacts like contrast and instrument which are prominent in nature. Figure 4.3 illustrates the model architecture of YOLOv3.

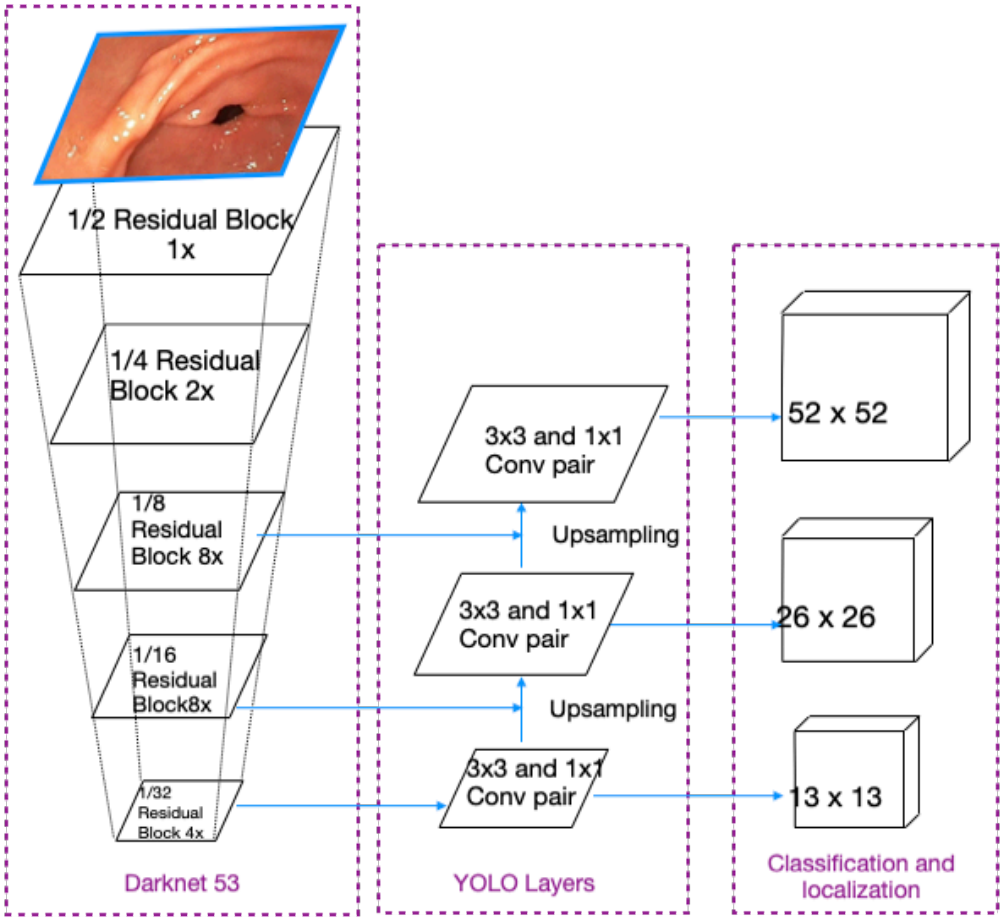


Figure 4.3 Network Architecture of YOLOv3

In the figure, the Darknet 53 network contains residual blocks, mentioned as 1x,2x,8x and 4x with varying residual groups. These blocks are placed after every convolutional layer as evident from Figure 4.2. The spatial dimension of the produced feature maps should be down sampled, stride convolution is used. The stride factor is chosen to be 2. Such convolution is done in prior to every residual groups. This technique avoids loss of low-level features.

Features from three different scales are extracted from darknet 53. The scales include 52 x 52, 26 x 26, 13 x 13. These feature vectors are fed from the classification and localization head to the detector head. These three feature vectors are responsible for detection objects at three different scales that is small, medium and large. Hence object detectors like YOLOv3 can detect tiny artefacts to artefacts that cover larger area at a faster rate. After predicting the artefacts across various scales, class to which the artefact belong to

must be predicted. Finally, the confidence score computed by YOLOv3 for each artefact reflects the presence of an object bounded by bounding box. The score is calculated using the Equation 4.1.

$$\text{Class confidence score} = \text{Box confidence score} * \text{conditional class probability} \quad (4.1)$$

When the result of the class confidence score is high, the bounding box is said to have an object in it. The loss function used in YOLOv3 is a weighted sum of three different loss functions. They are confidence loss, classification loss and localization loss. The former two loss functions use BCE and the later use mean square error. Therefore, the total loss is an addition of localization loss, classification loss and confidence loss.

With the above designed network, few parameters are expected to fed in as input. Along with the initializations for the data augmentation, the training parameters are set up. The classes are set to 8 in each of the three YOLO layers because there are a total of 8 classes that need to be detected Concurrently, the filters parameters must be changed based on the formula given in Equation 4.2.

$$Filters = (Class + 5) * 3 \quad (4.2)$$

Therefore, in each convolutional layer before the YOLO layers, the filter parameter should be set to 39. The input parameter set for training YOLOv3 is displayed in Table 4.1.

Table 4.1 YOLOv3 Input Parameters

Parameter Name	Parameter Value
Image Size (in pixels)	416 x 416 (Height x Width)
Batch Size	64
Subdivisions	16
Initial Learning Rate	0.001
Number of Classes	8
Model Checkpoint	1000th Iteration

Particularly in terms of spatial resolution, the network output is adjusted to be 1/32 times smaller than that of the input images. The training began after all the initializations. It persisted until the network's minimum average loss is reached. After 55,000 iterations the average loss is flattened and it is no longer found reducing. Hence the iterations are stopped. Else, if the loss has reached 0.05 for a small dataset or 3.0 for massive dataset the iterations can be stopped. The average loss did not improve much after 55,000 iterations. The training is stopped at 70,000 iterations. Various weight files extracted during training are tested for their performance in terms of mAP and IoU. Weight file extracted at $55{,}000^{th}$ iteration gave its best results.

4.3.2 YOLOv4 for Endoscopic Artefact Detection

YOLOv4 is the fourth prominent member added to the YOLO family and it is the research outcome of Alexey Bochkovskiy in 2020. The advantage of the network, YOLOv4 can be easily trained using a single GPU and deployed for real time object detection process.

YOLOv4 is composed of three basic parts such as, backbone, head and neck architectures. For this research YOLOv4 is set with the following features. The network considers heavier backbones such as, DenseNet based CSPResNeXt50, CSPDarknet53 and EfficientNet-B3 as backbones for final implementations. After simulations, CSPdarknet53 is selected as the backbone. It is found that the CSPdarknet53 backbone is fast with good receptive field.

Selecting the neck is the subsequent step in the YOLOv4 architecture. Existing networks such as, FPN, Adaptively Spatial Feature Fusion (ASFF), Path Aggregation Network (PANet), Neural Architecture Search (NAS)-FPN, Bi-directional FPN (BiFPN) and Scale-wise Feature Aggregation Module (SFAM) are considered. For a good network it is essential to consider three notable points. It is vital to have higher input network resolution. Such higher resolution helps to detect even tiny artefacts. Second, more network layers to cover larger receptive fields. Last, more the parameters more the capacity of the model to predict artefacts of various sizes.

Having the parameters in mind, finally PANet is chosen as the neck after simulations. The network groups crucial backbone features using SPP functions. Choosing the head is the

last stage. Head assigns class names and forecasts the bounding box's coordinates. YOLOv4 uses YOLOv3 head. The same network is retained as head. Figure. 4.4 illustrates the YOLOv4 model workflow.

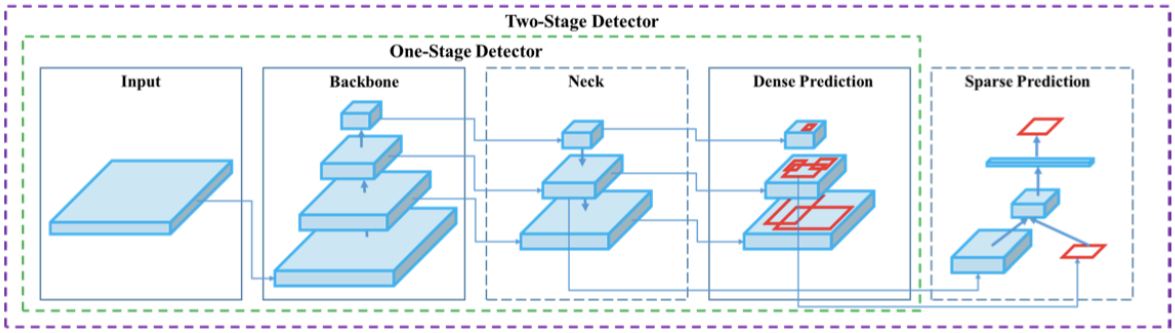


Figure 4.4 YOLOv4 Workflow (Bochkovskiy et al., 2020)

To leverage the performance of the network, few universal features are available. It includes, new run time augmentation technique, Cross Stage Partial connections (CSP), mish activation, Cross mini-Batch Normalization (CmBN), weighted residual connections, Complete IoU (CIoU) loss and drop block regularization. One Applications can combine one or more of the newly added functionalities to provide cutting-edge outcomes. The dataset determines the combination. They are divided into the Bag of Freebies (BoF) and the Bag of Specials categories (BoS).

Special features of YOLOv4 from BoF and BoS are handpicked for the best results. The features of BoF are given in the Figure 4.5. Selective techniques are chosen from BoF. There are few conventional colour-based augmentation techniques such as, adjusting brightness, hue, contrast and saturation. The other geometry-based augmentation techniques such as, noise, flipping, rotation, scaling and cropping. The additional augmentation techniques in YOLOv4 include cut-out, mix-up, and mosaic augmentation. Out of all the available augmentation techniques colour based augmentation techniques such as, modifying the hue, saturation, contrast and geometry-based augmentation techniques such as, scaling and cropping, multiple image augmentation techniques such as, cut-out and mosaic augmentation techniques are chosen based on trial-and-error experimentations. Self-Adversarial Training (SAT), conceals the dependence and forces the network to generalize

new features. Hence SAT is adopted. It helps to make the model robust and reduces overfitting.

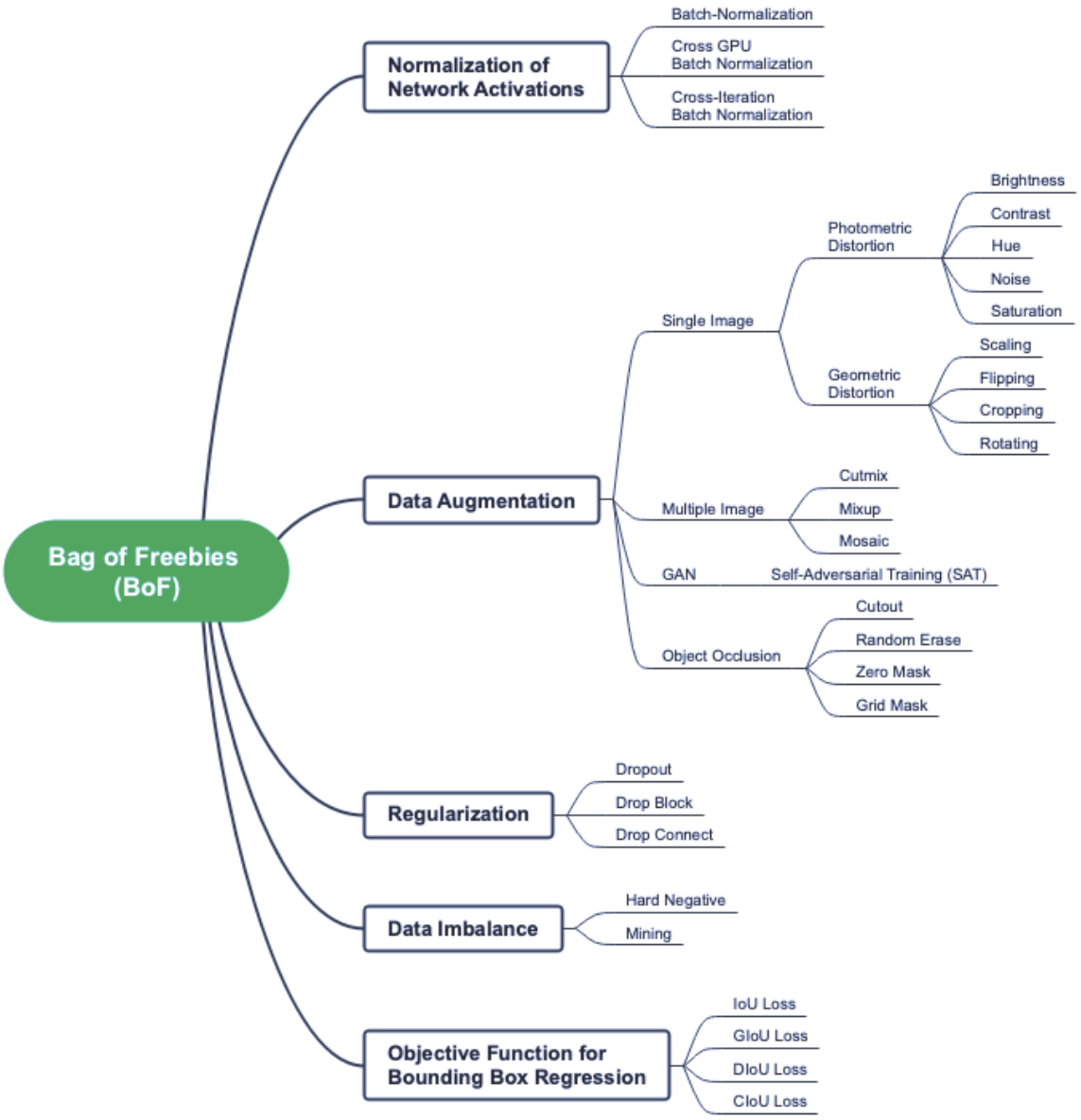


Figure 4.5 Bag of Freebies from YOLOv4

The model includes CIoU, as it improves convergence and accuracy. Drop block regularization technique is adopted. Batch normalization is also adopted to speed up the training process. These methods, which are categorized under BoF, help to increase model accuracy without retarding model inference.

BoS is committed to accelerate the inference time and performance. Figure 4.6 potrays various choices available under BoS. The BoS strategies help to increase accuracy at the expense of a higher inference times. Therefore, it is up to the researchers to choose the best approaches for the best outcomes. The proposed YOLOv4 utilizes features such as, DIoU NMS, SAM block, PAN block, SPP block, and mish activation function. These techniques are selectively chosen after literature study and trial and error experimentations.

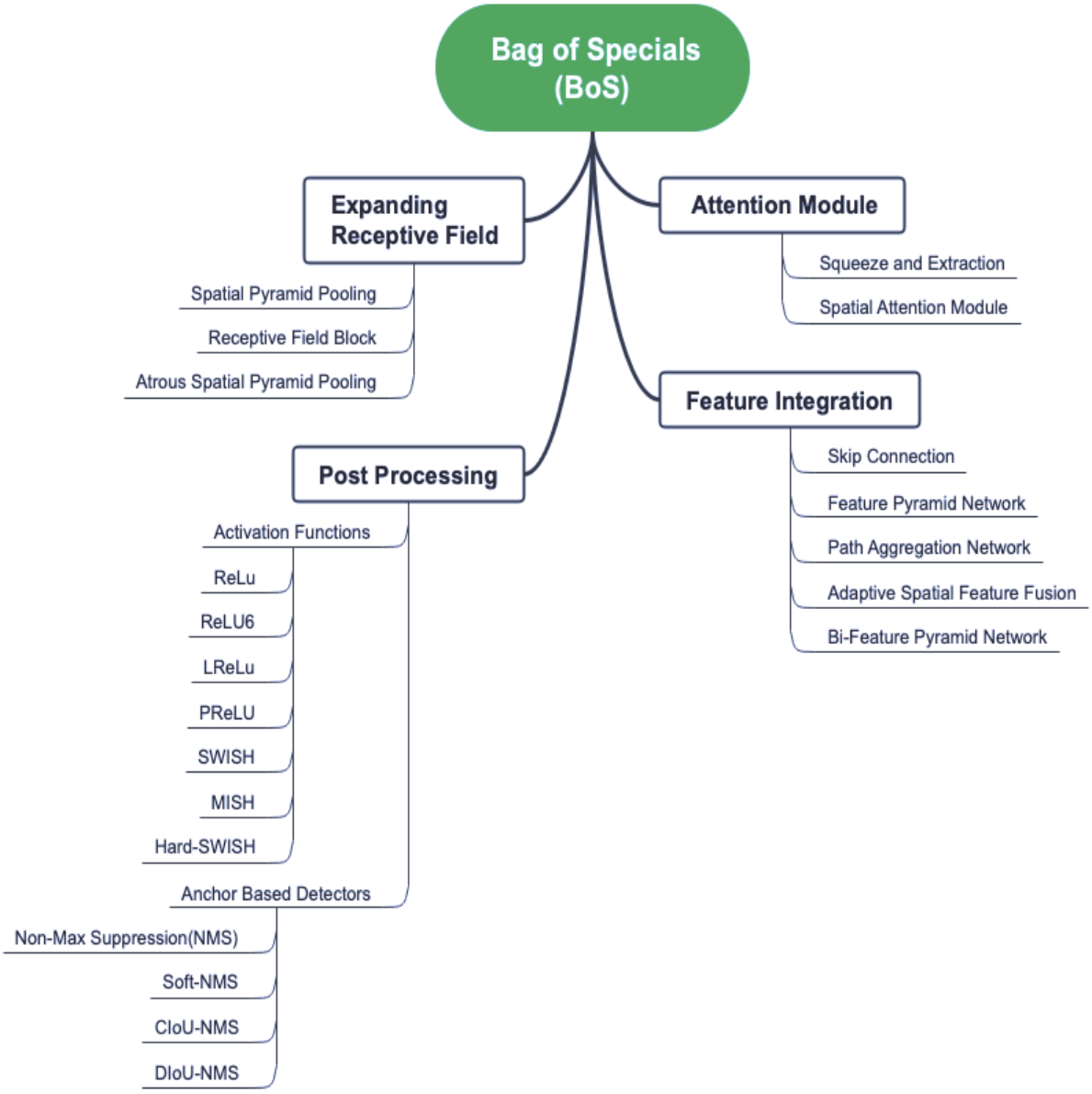


Figure 4.6 Bag of Specials from YOLOv4

Pre trained weights for YOLOv4 are employed to eliminate training from scratch. Google Co-lab (https://colab.research.google.com) with a single GPU is used for training. With all the features been selected from BoF and BoS, other critical initial hyper-parameters required for training is set as tabulated in Table 4.2.

Table 4.2 YOLOv4 Input Parameters

Parameter Name	Parameter Value
Image Size (in pixels)	512 x 512 (Height x Width)
Batch Size	64
Subdivisions	64
Initial Learning Rate	0.013
Number of Classes	8
Model Checkpoint	1000^{th} Iteration

With the parameter values given in Table 4.2 training endured till 85,000 iterations. After which the loss did not converge. Thus, the training is terminated after 95,000 iterations. The best performing weights files are examined and the weight file extracted during $76{,}000^{th}$ iteration is selected. The weight file under examination provided an appropriate balance of mAP, IoU, and inference time.

4.3.3 Faster R-CNN for Endoscopic Artefact Detection

Faster R-CNN, gets the input image and passes through its backbone. The model is said to have deeper networks. For artefact detection Faster R-CNN uses ResNet backbone. This backbone network gives the output feature maps along with the bounding boxes. The output feature map is considered for the next step. Each point in the derived feature map is called as anchors. Every anchor is said to generates multiple bounding boxes of several sizes and shapes. The bounding box is generated to capture the artefacts present in the input endoscopic image. Faster R-CNN uses 1 x 1 convolution layers. It helps to predict the category of the anchor boxes. These anchor boxes may be positive or negative. If it is positive then the box is said to contain an object (artefact) in it. If it is negative then it is considered as background. Binary cross entropy loss is used to classify the boxes. The predicted anchor

boxes may not align with the original ground truth bounding boxes. Offsets are used during training to align the bounding box. Faster R-CNN adopts L2 regularization loss in order to learn the offsets. Hence the region proposal that contain an artefact is determined. Later, the region proposals are shifted to next stage to predict the class of the artefact. The proposals are of various size and shape, using RoI pooling they proposals are resized. Multiple classes can be predicted using cross entropy loss. The weighted combination of both losses is considered as final loss. During prediction, the test image is passes through the backbone as first step. This stage generates anchor boxes. To increase the detection speed top 300 boxes are chosen. They are ranked based on the classification score. The classes are predicted, duplicate boxes are removed using NMS. Finally, the model outputs one box per artefact with the corresponding confidence score. A simple version of faster R-NN is displayed in the Figure 4.7.

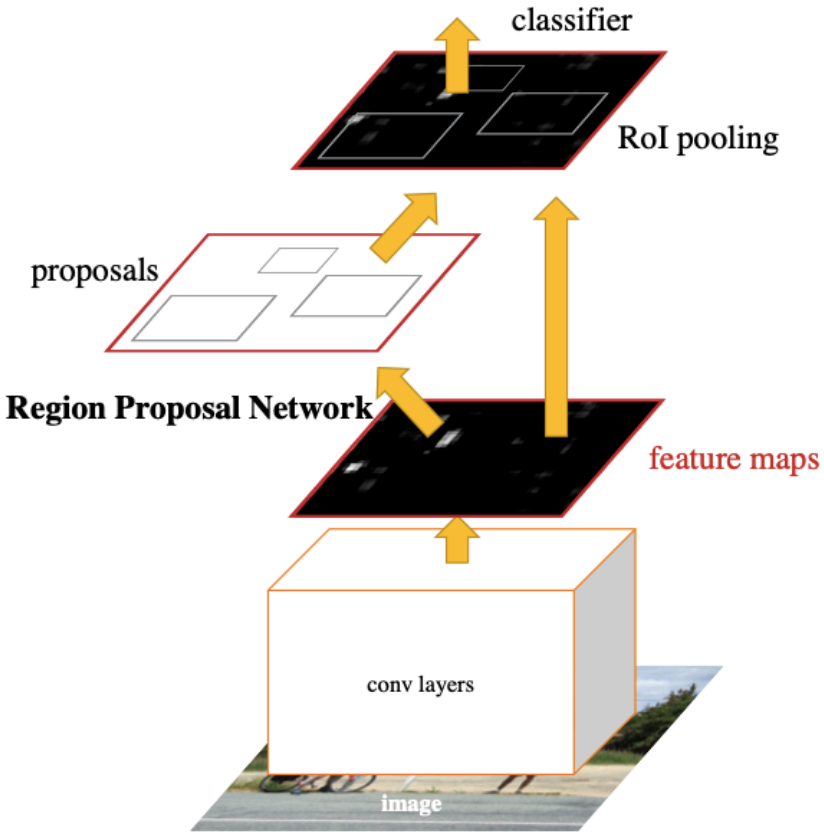


Figure 4.7 Faster R-CNN (Girshick R., 2015)

For this research faster R-CNN is adopted from detectron2 (https://ai.facebook.com/tools/detectron2/). It is created by Facebook AI Research (FAIR). It supports rapid implementation of various object detection algorithms. It also supports instance image segmentation and panoptic segmentation. It has multiple research baselines and a model zoo made up of trained model files for quicker implementations. The Faster R-training CNN's parameters are listed in Table 4.3.

Table 4.3 Faster R-CNN Input Parameters

Parameter Name	Parameter Value
Image Size (in pixels)	512 x 512 (Height x Width)
Batch Size	4
Subdivisions	4
Initial Learning Rate	0.1
Number of Classes	8
Model Checkpoint	1000^{th} Iteration

Model check point at 1000^{th} Iteration is used to allow the model to save the trained weight file at every 1000^{th} iteration. With all the input allowable parameters set the training of Faster R-CNN is started. Initially maximum iterations are set to 10,000. So that after every 10,000 iteration the performance of the artefact detector can be evaluated and monitored. Batch wise the training lasted till the model reached 80,000 iterations. The weight file of $70,000^{th}$ iteration and $80,000^{th}$ iteration is compared. No notable difference found between the performance of both the weight files in terms of mAP and IoU. Hence the training is stopped. The weight file of every 1000^{th} iteration from 70,000 iteration is tested. The best results are from 74,000th iteration. Hence the same iteration and its corresponding weight is considered to be better compared to others.

4.3.4 Ensemble Model

Ensemble learning model incorporates predictions from multiple object detection model predictions to improve the overall performance. In applications where the dataset is not commanding or hyper parameters are tough to identify and tune, the resulting model is

said to be weak. Ensemble learning may help in a scenario where two or more weaker models can be combined to produce better result.

Every individual object detection model for an input image I, returns a list of artefact detections given by $D=[d_1,d_2,d_3,\ldots..d_N]$ where d_i is a list that must contain a triplet values that includes bounding box information b_i, class to which it belongs as c_i and the confidence score s_i.

Input to the ensemble object detection model is a list $L = [D_1, D_2,D_3,\ldots..D_m]$, Di with $i = \{1,2,3,\ldots.m\}$ is nothing but a list of artefact detections. In this research three different base learners are used. Hence there will be 3 lists D_1, D_2 and D_3. That is D_i is a list of detections of one of the artefact detection models. Each artefact detection model is denoted as M_i. Such D_i are generated for each input image.

In order to obtain the final ensemble results algorithm runs through five steps as discussed:

Step 1: List L must be flattened and renamed as $F = [d_1, d_2, d_3,\ldots.d_n]$ where $i = \{1,2,3\ldots n\}$. Every d_i are the predictions.

Step 2: Each detections d_i of the flattened list F must be grouped considering the overlap of bounding boxes along with their corresponding classes.

Step 3: IoU metric is deployed on the grouped elements as the elements are grouped based on the cohesion of bounding box and classes.

Step 4: This metric measure will result in a new list $G = [D_1G, D_2G,..., D_mG]$ with detection which has IoU Score>0.5. In the expression D_iG, denotes the detections where $i \in \{1...m\}$ and m is the total number of detections.

Step 5: The decision of whether the bounding box contains an object is decided by various voting strategies it includes,

Affirmative: All the base learners are allowed to predict objects in the given input test image. The bounding box predicted by even one of the models will be considered for the final ensemble model. That is, every prediction present in the list G, are considered.

Unanimous: All the models predict the objects present in the input image and draws a bounding box. If all the three models predict the same object, and if the IoU is greater than 0.5, then that object is considered for final predictions.

Consensus: A bounding box around an object will be considered only if most models generate the same box. The majority is measured by assuming the size of the list. When the length is greater than m/2, only that particular D_i^G lists are retained.

Any one voting strategies can be chosen. In this research, consensus strategy is deployed. If more than one model predicts an artifact the possibility of the presence of artefact is higher. Hence consensus voting strategy is chosen. If affirmative is considered it may lead to false detection as the result can be positive if one model predicts which may even be a weaker model. If a unanimous approach is selected every model must detect an artefact which increases training cost of each model. Consensus gives a balance between these two approaches. The output of deploying such a strategy will end up with a new list G' $\subseteq$ G. Now this list may contain several bounding box, for the same artefact. Hence only one bounding box that covers the artefact precisely must be chosen. It can be attained using NMS. After applying NMS the final result will be a list F= [f_1,f_2,f_3…f_n] which is the final ensemble predictions where fi, where i = {1,2,…n}. Each fi is now a triplet containing bounding box information b_i, class to which it belongs as c_i and the confidence score s_i.

The ensemble model proposed in this research combines the benefits of single-stage and two-stage object detectors. YOLO architectures are fast and good enough in predicting objects in the images that are tiny. Faster R-CNN model has good accuracy when compared to YOLO. For predictions, the trained model files of YOLOv3, YOLOv4, and Faster R-CNN are combined. The model receives a test image. Every artefact in the image is predicted by each of the three trained models. Final predictions will be produced using the ensemble method that is selected. The flow chart shown in the Figure 4.8 illustrates the process.

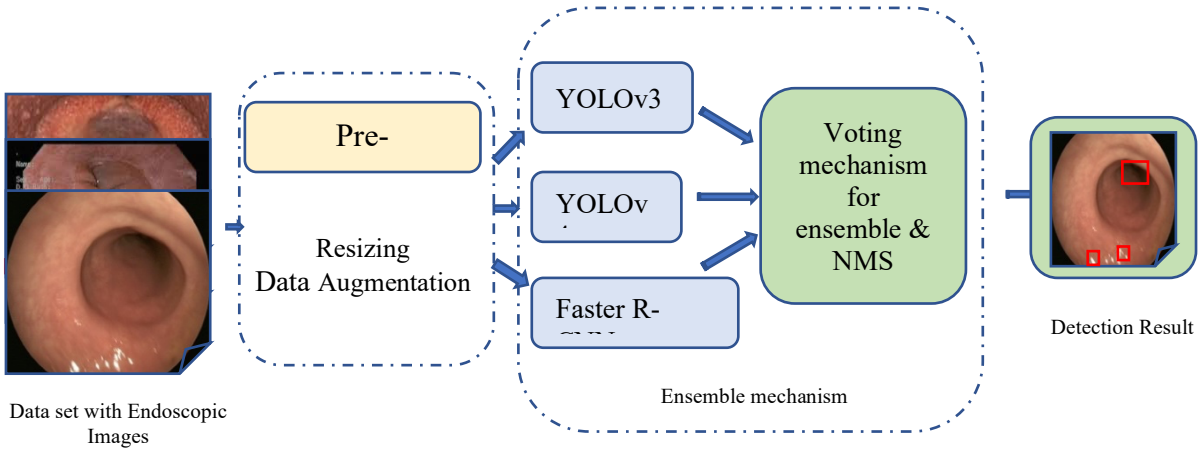


Figure 4.8 Ensemble Artefact Detection Model

In this research, all the three models, namely YOLOv3, YOLOv4 and Faster R-CNN, predicts every artefact in the given test image. Every model generates a list and the generated list is analyzed using all the three voting strategies. The analysis resulted in consensus approach and the results stands out the better for multiple artefact predictions.

4.4 PERFORMANCE METRICS AND SIMULATION RESULTS

Section 4.4.1 throws light on datasets used to evaluate detection algorithms' accuracy and inference time. The following sections describe the performance metrics and results obtained by the artefact detection model.

4.4.1 Test Dataset

Images from EAD and custom datasets are selectively chosen to form the test database for endoscopic artefact detection. The test set contains 20% of the total images. The selection criteria include careful handpicking of images covering all artefacts, organs and modalities.

4.4.2 Performance Metrics

To evaluate and model, few metrics are considered to be standard across the globe. They are IoU, mAP, Precision and Recall.

- ***Intersection over Union***

$$IoU = \frac{A \cap B}{A \cup B} \tag{4.3}$$

In Equation 4.3, the ground truth bounding box is denoted by B, and A is the anticipated bounding box. The intersection of predicted and ground truth bounding boxes to the union of predicted and ground truth bounding boxes is measured by the IoU metric. The outcome falls between 0 and 1. When the result is 0, there is no overlap between A and B; else, when the result is 1, both A and B overlap completely. Throughout the research, the results obtained are measured with a threshold of 0.5. Threshold determines the margin above which the overlap can be considered for positive prediction.

- ***Mean Average Precision***

The trained object detection algorithm's capacity to precisely locate every instance of the ground truths is measured by mAP. The precision-recall curve's area under the curve at recall levels (r1, r2,... rn) can be used to determine AP. The AP score is calculated using Equation 4.4. The letters (p) and (r) in the given equation stand for precision and recall, respectively. The expression to calculate precision and recall is given in Equation 4.6 and 4.7.

$$AP = \sum_{i} (r_{n+1} - r_n) p_{interp}(r_{n+1}) \tag{4.4}$$

where p_{interp}= max p(r). Finally, mAP is assessed by taking the mean of all AP. It can be calculated using the formula given in 4.5.

$$mAP = \frac{1}{N} \sum_{i} AP_i \tag{4.5}$$

If the calculated mAP values are higher, then the performance of the model is said to be better.

- ***Precision and Recall***

The formula to calculate Precision and recall can are given in the Equation 4.6 and 4.7.

$$Precision = \frac{TP}{TP + FP} \tag{4.6}$$

$$Recall = \frac{TP}{TP + FN} \tag{4.7}$$

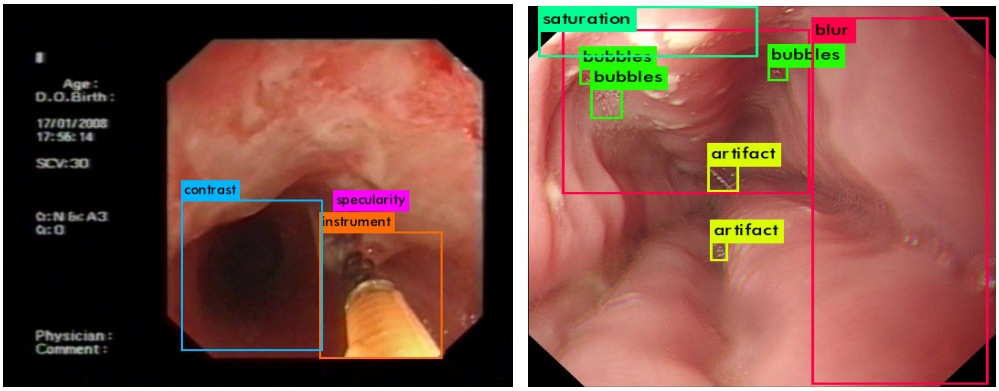
True Positive (TP) is the designation for positive samples that are envisioned as positive. The term "False Positive" (FP) refers to the number of negative samples that are projected to be positive samples, and the term "False Negative" (FN) refers to positive samples that are projected to be negative samples.

4.4.3 Simulation Results

Images from EAD and custom dataset is used to train the base learners of the proposed model. The custom dataset is curated to add more images with blur, saturation and instruments to balance the class distribution when images from EAD and custom dataset are combined.

- **Predictions and performance of YOLOv3**

  This section presents the predictions of YOLOv3 after being trained on EAD and custom dataset. The performance of YOLOv3 as an individual base learner is presented here. The configuration file is written to save the training weights for every 1000^{th} iteration. The performance parameters are measured for every 1000^{th} iteration. The training lasted till the loss function stabilizes. Figure 4.9 (a) and (b) illustrates the detections of multiple artefacts by the trained YOLOv3 model. In the Figure 4.9 (a) it is evident that the model predicted three different artefacts present in the image namely, instrument, contrast and specular reflections (specularity). In Figure 4.9 (b) the model successfully predicted all the artefacts namely saturation, bubbles and miscellaneous artefacts. Hence the trained model is now able to predict artefacts at various scales from small to large.



(a) (b)

Figure 4.9 (a) and (b) Detection of Artefacts by YOLOv3

Figure 4.10 elucidates the mAP vs iteration plot. The plot shows that the network started to stabilize after 45,000 iterations.

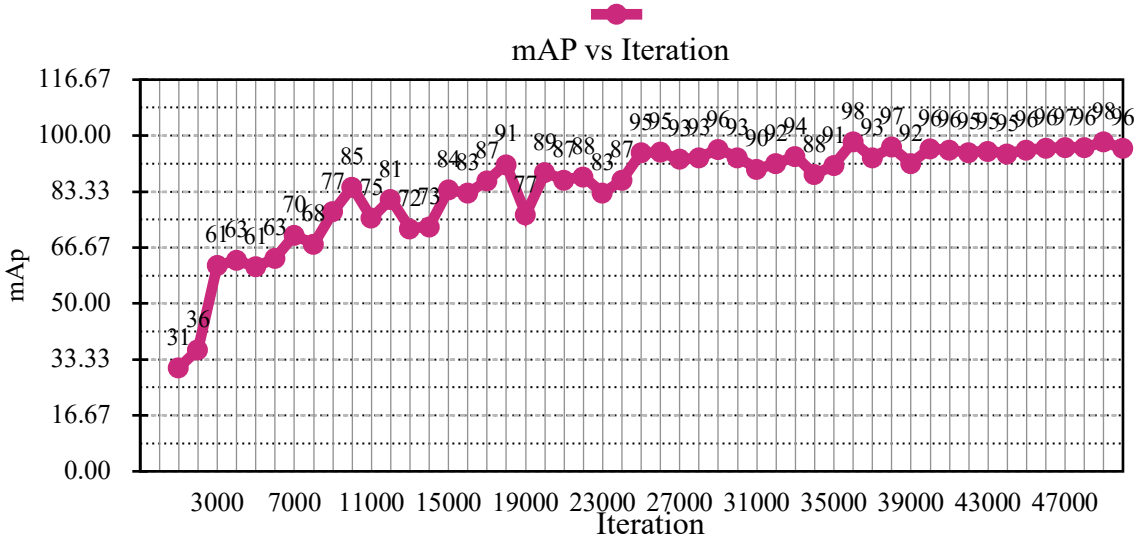


Figure 4.10 mAP vs Iteration Plot of YOLOv3

The best weight file (55000th iteration) extracted from the training is used to validate and test the model. The numeric values mentioned in Table 4.4 are the AP scores obtained across each artefact. From the results it is evident that the AP score for all artefacts except bubbles are better. Bubbles are tedious to detect as the features such as, colour and texture are very similar to the background. Similarly, Table 4.5 shows the overall performance metrics such as, mAP, Precision, Recall and F1 score measured from the model.

Table 4.4 Class-wise AP Score of YOLOv3

Class No.	**Artefact**	**AP Score** (Proposed)	**AP Score** (Ali S, 2019)	**AP Score** (Ren S, 2015)
0	Specular reflection	44.43	34.7	20.7
1	Saturation	46.41	55.7	71.0
2	Artefact	40.91	48.0	35.1
3	Blur	88.89	7.5	14.5
4	Contrast	51.96	72.1	58.7
5	Bubbles	13.63	55.9	42.4
6	Instrument	100.00	-	-
7	Blood	76.23	-	-

Table 4.5 Performance Metrics of YOLOv3

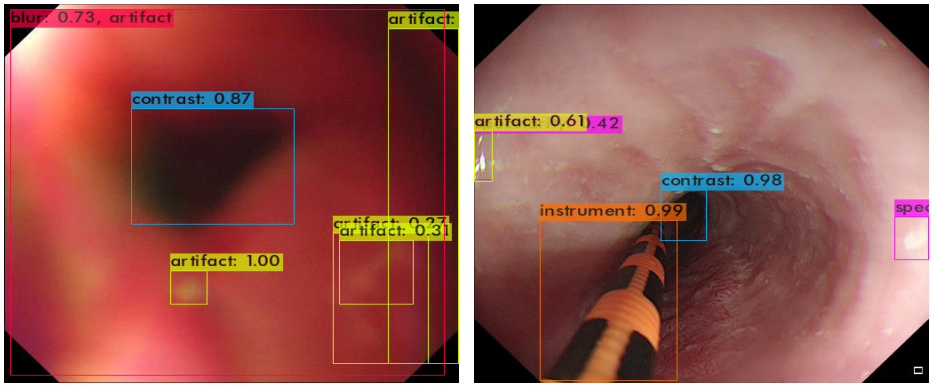
Parameter	Value
mAP	0.308
IoU	0.769

From the simulations, the YOLOv3 artefact detection model is able to score mAP of 0.308 and IoU of 0.769.

- **Predictions and performance of YOLOv4**

  Similar to YOLOv3, the configuration file is programmed to save weights for every 1000th iteration. The best weight file is considered for final predictions. It is decided to use weight file extracted during 76000th iteration as the best model as the model meets out the expected performance. This section briefly explains the result obtained using the trained YOLOv4. The trained model is allowed to predict artefacts. The model randomly chooses images from test set for predictions.

Figure 4.11 (a) and (b) illustrates the predictions of the YOLOv4. In Figure 4.11 (a) the trained model identified all the artefacts present in the test image. The artefacts identified includes blur, contrast and miscellaneous artefacts. The confidence score in identifying the artefact contrast is 0.87, for blur it is 0.73 and for miscellaneous artefacts the score varies from 0.31 to 1.00 for multiple instances. In Figure 4.11 (b), the predictions include instrument with confidence score of 0.99, contrast with confidence score of 0.98, specular reflections with confidence score of 0.42 and for miscellaneous artefact, the confidence score is 0.61. Hence the trained YOLOv4 is now able to predict and draw a bounding box over different artefacts present in a single image.



(a) (b)

Figure 4.11 (a) and (b) Predictions of YOLOv4

Class wise AP scores measured are tabulated in Table 4.6. The AP scores of YOLOv4 is better when compared with YOLOv3. Particularly the model performance on detecting specular reflections, saturation and bubbles improved. Table 4.7 portrays the mAP and IoU scores of trained YOLOv4 model. The results of YOLOv4, shows an improved performance when compared to YOLOv3 in terms of mAP.

Table 4.6 Class-wise AP Score of YOLOv4

Class No.	**Artefact**	**AP Score**	**AP Score** (Ali S, 2019)	**AP Score** (Ren S, 2015)
0	Specular reflection	46.84	34.7	20.7
1	Saturation	59.46	55.7	71.0
2	Artefact	31.8	48.0	35.1
3	Blur	51.39	7.5	14.5
4	Contrast	43.03	72.1	58.7
5	Bubbles	16.14	55.9	42.4
6	Instrument	100.00	-	-
7	Blood	64.76	-	-

Table 4.7 Performance Metrics of YOLOv4

Parameter	Value
mAP	0.498
IoU	0.439

- **Predictions and performance of faster R-CNN**

Faster R-CNN's performance as an individual base learner is presented in this section. Figure 4.12 (a) – (c) shows the predictions of Faster R-CNN. It is evident from the Figure 4.12 (a) that the model predicts artefacts of various sizes from tiny specular reflection and miscellaneous artefact to large instruments. The confidence score in predicting artefacts include 1.00 for miscellaneous artefacts, 0.57 for instrument, 0.69 to 0.97 for saturation at multiple instances, 0.61 for bubbles and 0.94 for specular reflections. Similarly Figure 4.12 (b) and (c) also presents the predictions of multiple artefacts. It is noticed that in the Figure 4.12 (a) the model misclassified miscellaneous artefact as blood. It is because the chromatic aberrations are found red in colour. In Figure 4.12 (c) the model missed to predict the artefact saturation.

It is also seen form the Table 4.8 that the AP scores are much improved for most of the classes when compared with the other base learners. Table 4.9 portrays the mAP and IoU scores of trained Faster R-CNN model. The model's performance when compared with the other base learners YOLOv3 and YOLOv4, is better in terms of both mAP and IoU. The Faster R-CNN model has scored mAP of 0.513 and IoU score of 0.626.

(a) (b) (c)

Figure 4.12 (a) – (c) Predictions of Faster R-CNN

Table 4.8 Class-wise AP Scores of Faster R-CNN

Class No.	**Artefact**	**AP Score**	**AP Score** (Ali S, 2019)	**AP Score** (Ren S, 2015)
0	Specular reflection	54.81	34.7	20.7
1	Saturation	69.01	55.7	71.0
2	Artefact	43.36	48.0	35.1
3	Blur	44.16	7.5	14.5
4	Contrast	41.43	72.1	58.7
5	Bubbles	36.14	55.9	42.4
6	Instrument	100.00	-	-
7	Blood	84.76	-	-

Table 4.9 Performance Metrics of Faster R-CNN

Parameter	**Value**
mAP	0.513
IoU	0.626

- **Predictions and performance of ensemble model compared to individual models**

The final ensemble model is created by combining all of the trained base learners. The performance of the ensemble model is compared with individual performance of the base learners. It is proved that the ensemble model's performance is comparatively balanced in terms of mAP and IoU. Figure 4.13 illustrates the same.

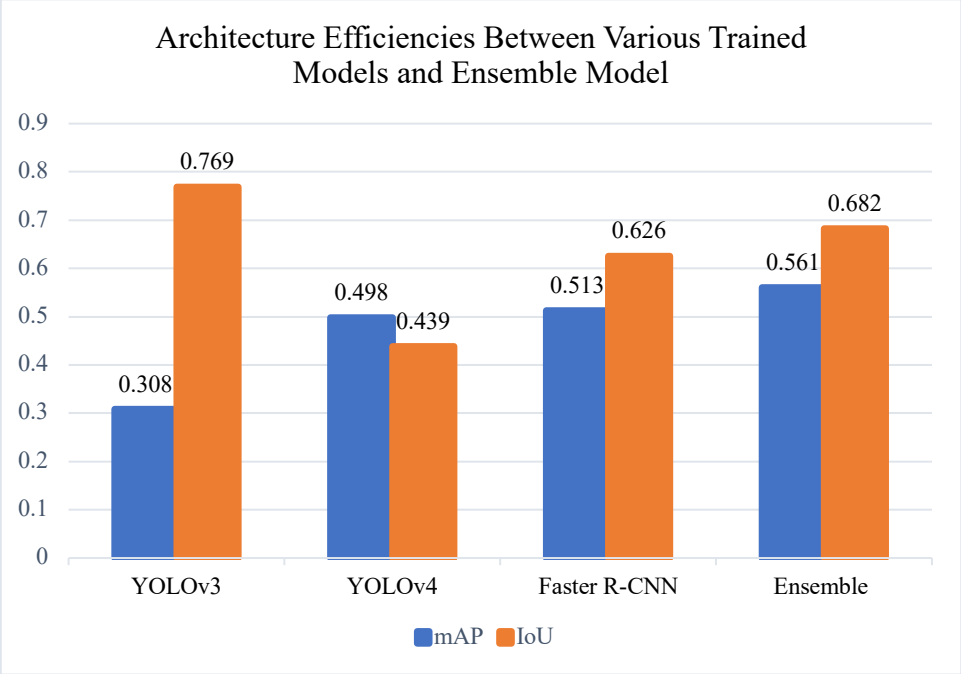
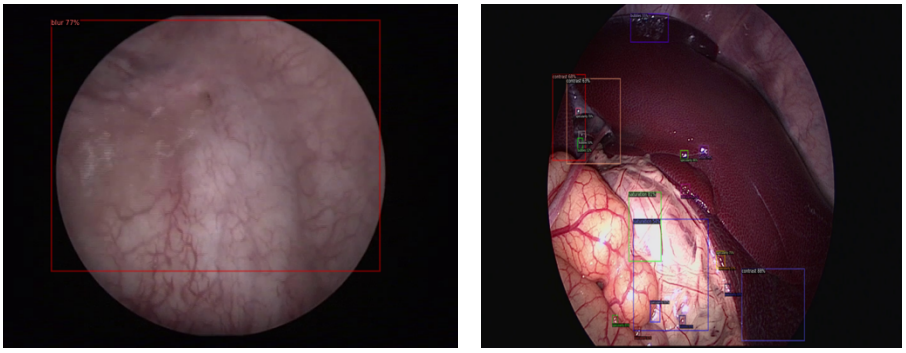


Figure 4.13 Architecture Efficiencies Between Various Trained Models and Ensemble Model

Every model based on its learning individually detects artefacts. Each artefact is bounded using a bounding box. The bounding box indicates the predictions. The bounding box is compared with the ground truth. When the overlap of the predicted and ground truth bounding box is more than 50 percent then the bounding box is considered as a positive prediction. In the similar way every bounding box over every artefact is evaluated with 50 percent threshold. Overall performance metrics are calculated and plotted. The predictions of all trained models are evaluated and plotted. The performance of ensemble model is also calculated and compared with the individual base learners.

Figure 4.14 (a) and (b) self-explains the predictions made by the ensemble model. It is found that the confidence in identifying every artefact is more than 75%. Consider Figure 4.14 (a) the artefact blur is detected with 77 % confidence, similarly in Figure 4.14 (b) the artefact contrast is predicted with 82% confidence at one instance and 62% and 68% confidence at two other instances. The artefact saturation and specular reflections are predicted with more than 80% confidence. Finally, the artefact bubble predicted with

the confidence score of 23%. The features of bubbles are harder to learn as both the background and the artefact both are hard to distinct.



(a) (b)

Figure 4.14 (a) and (b) Predictions of Ensemble Model

Many computer-aided detections and diagnosis tools are emerging day by day. Amidst the sudden shoot after AI's advent, various branches of endoscopic imaging have proliferated. Yet few issues are to be addressed. One such is improving the accuracy and inference time of artefact detection and segmentation models. The experiments performed in each phase are carried out with a solid objective to improve methods compared to literature results. Every result obtained during the simulation of various stages of research is recorded in this chapter.

The research results of existing models are compared with the proposed work through graphical representation in the Figure 4.15. The result is compared using standard metrics such as, mAP and IoU. The proposed method outperforms the existing results. The results prove a 61.67% increase in mAP and 63.47% increase in IoU when compared to results from the literature.

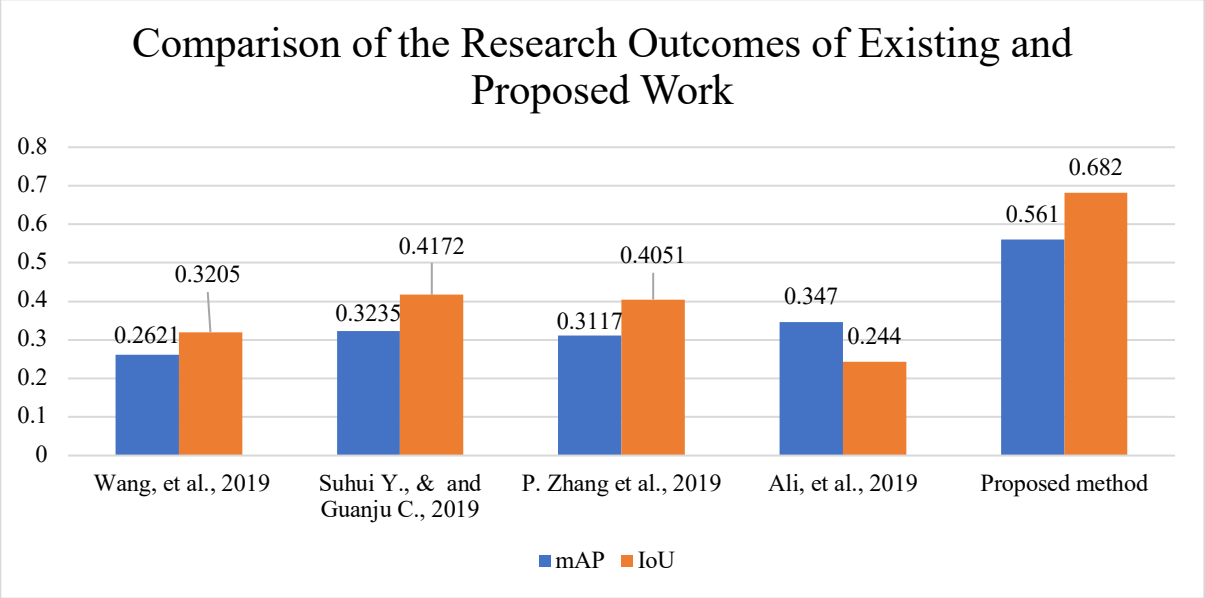


Figure 4.15 Comparison of the Research Outcomes of Existing and Proposed Work

The other primary objective of the research is inference time which is an important parameter in real-time applications. The proposed model reached 80.4ms compared to 88ms of the existing research results, which is portrayed graphically in the Figure 4.16. Consequently, the outcomes demonstrate that the proposed model performs well in terms of mAP, IoU, and inference time.

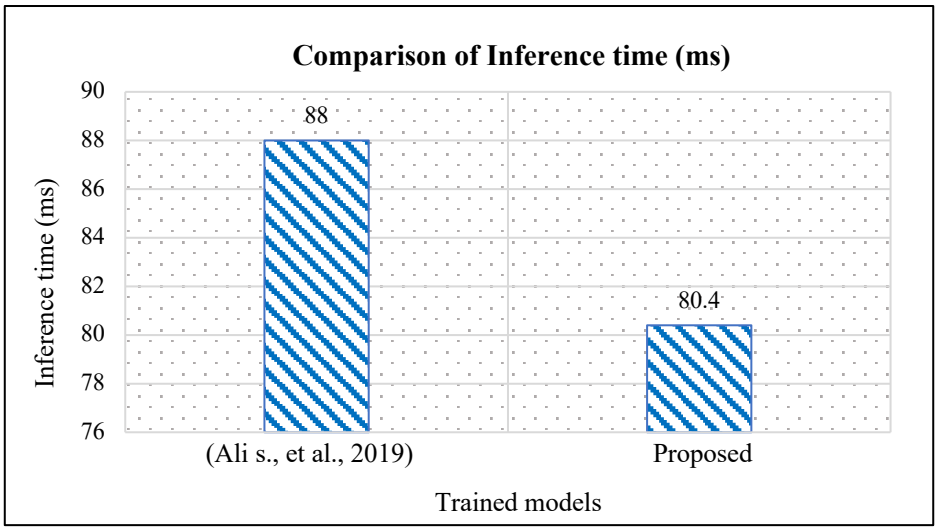


Figure 4.16 Comparison of Inference time

4.5 CHAPTER SUMMARY

The proposed research aimed to develop an artefact detection model, which could be deployed as part of the endoscopic imaging pipeline. An ensemble model with the benefits of single- and two-stage detectors is designed by combining three separate artefact detectors. The combination of three base learners gives a balanced performance when compared using the performance metric mAP, IoU and inference time as well. The inference time is reduced by 8.63%, whereas the mAP and IoU are increased by 61.67% and 63.47% respectively. Images from EAD and custom datasets are used to train all of the ensemble model's base learners This proposed model is effective in detecting all eight major artefacts in an endoscopic image. By analysing the performance of the all the individual models, for majority of the artefacts the individual AP scores are better when compared with existing results. The customized dataset, training strategy, choice of backbone and other features collectively produced a better result than result from literature. The ensemble model results are comparatively better than all the individual models and also the results from the literature.