SCRUM 4

What I do this SCRUM?

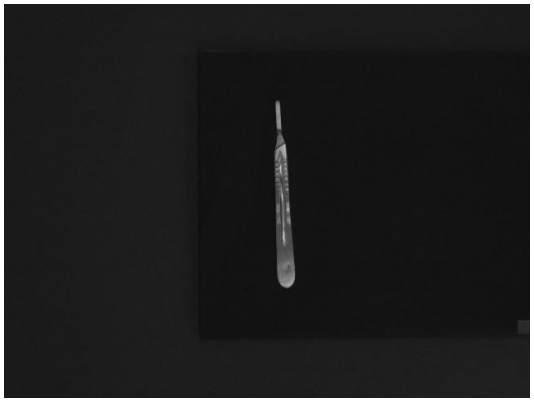
1. Model for Fine-Grained Image Classification
2. Evaluate Performance of models

From last SCRUM

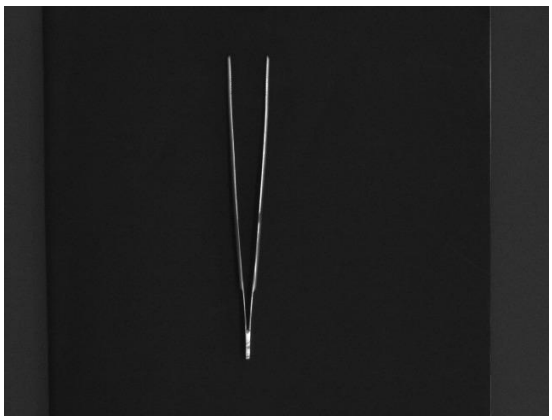
According to dataset I choose is surgical tools dataset which have 4 classes :

1. Curved Mayo Scissor

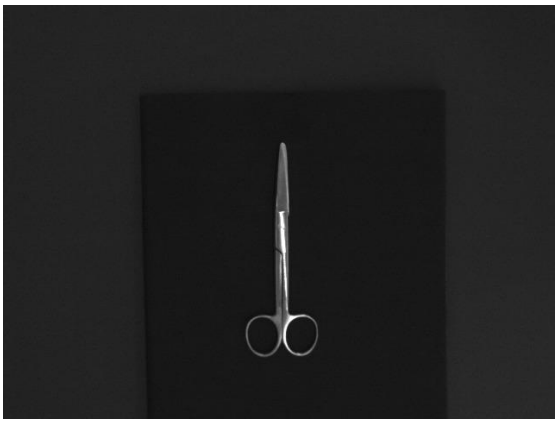
2. Scalpel



3. Straight Dissection Clamp



4. Straight Mayo Scissor



The dataset have 2010 image so I split the dataset with ratio : 70:10:20

It's will have

Train set	:	1407	image
Validation set	:	201	image
Test set	:	402	image

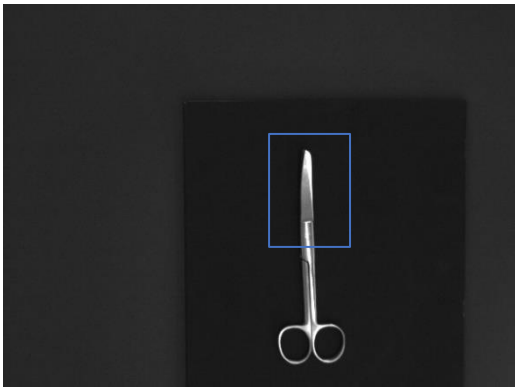
SEResNet50 Classification report

	precision	recall	f1-score	support
Curved Mayo Scissor	0.88	0.93	0.91	104
Scalpel	1.00	1.00	1.00	111
Straight Dissection Clamp	1.00	1.00	1.00	83
Straight Mayo Scissor	0.93	0.88	0.90	105
accuracy			0.95	403
macro avg	0.95	0.95	0.95	403
weighted avg	0.95	0.95	0.95	403

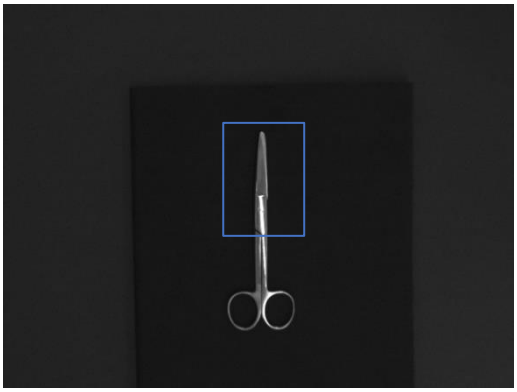
Based on previous SCRUM, the SEResNet50 model was able to effectively classify two out of the four classes in the dataset. Specifically, it was able to accurately classify the Scalpel and Straight Dissection Clamp. However, the model struggled with the other two classes, the Curved mayo scissor and the Straight Mayo Scissor. These two classes have a similar shape, with the only difference being the end of the scissor. The Curved Mayo Scissor has a curved end while the Straight Mayo Scissor has a straight end.

The difference of class Curved mayo scissor and Straight Mayo Scissor

Curved Mayo Scissor



Straight Mayo Scissor



These two classes have a similar shape, with the only difference being the end of the scissor. The Curved Mayo Scissor has a curved end while the Straight Mayo Scissor has a straight end. According to blue box in image above.

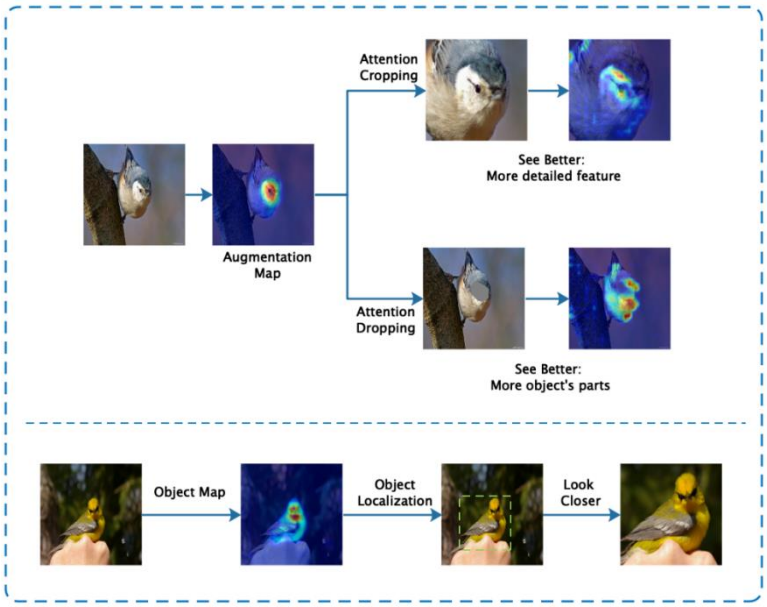
To classify objects that look similar, the task would fall under the domain of fine-grained classification.

WSDAN

(Weakly Supervised Data Augmentation Network for Fine-Grained Visual Classification)

from research paper WSDAN

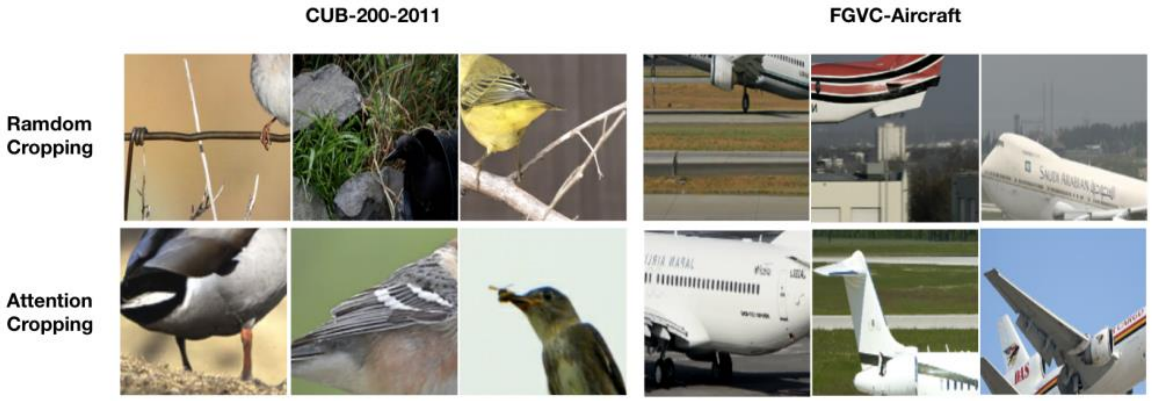
In **Abstract** it's said "WS-DAN improves the classification accuracy in two folds. In the first stage, images can be seen better since more discriminative parts' features will be extracted. In the second stage, attention regions provide accurate location of object, which ensures our model to look at the object closer and further improve the performance."



See Better: Attention maps represent discriminative parts of the object. Randomly choose one of the part regions, then drop it to generate more discriminative object's parts or crop it to extract more detailed part feature.

Look Closer: The whole object is localized from attention maps and enlarged to further improve the accuracy.

What different about normal augmented cropping and attention cropping

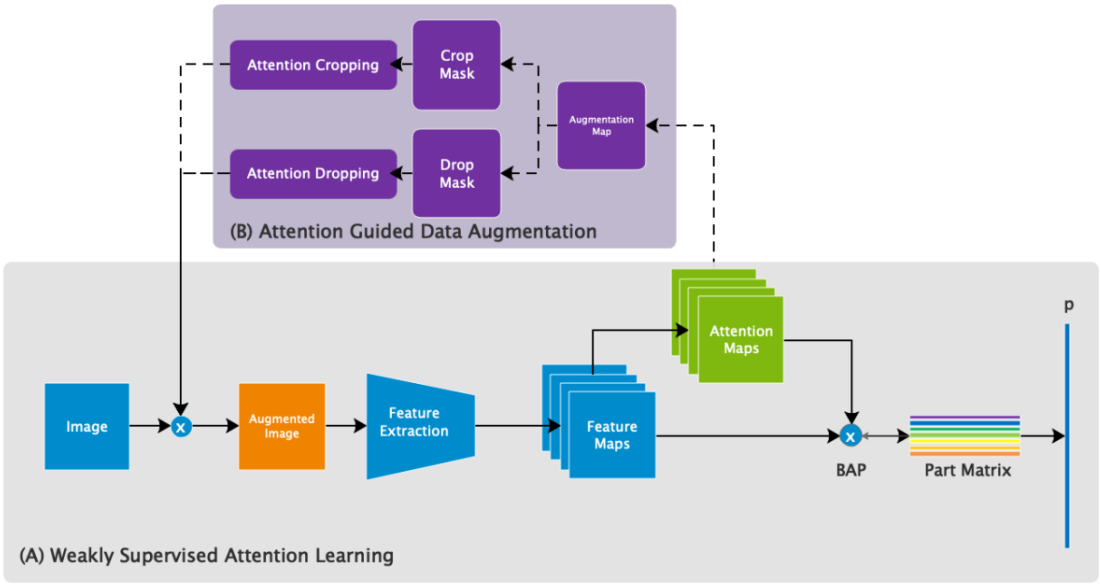


Comparison between Attention Cropping and Random Cropping. Random cropping is very likely to include a high percentage of background as input image, while attention cropping knows exactly where to crop to see better.

What different about normal augmented dropping and attention dropping

Comparison between Attention Dropping and Random Dropping. Random dropping might erase the whole object out of the image or just erase background. Attention dropping is more efficient for erasing the discriminative object parts and promoting multiple attention.

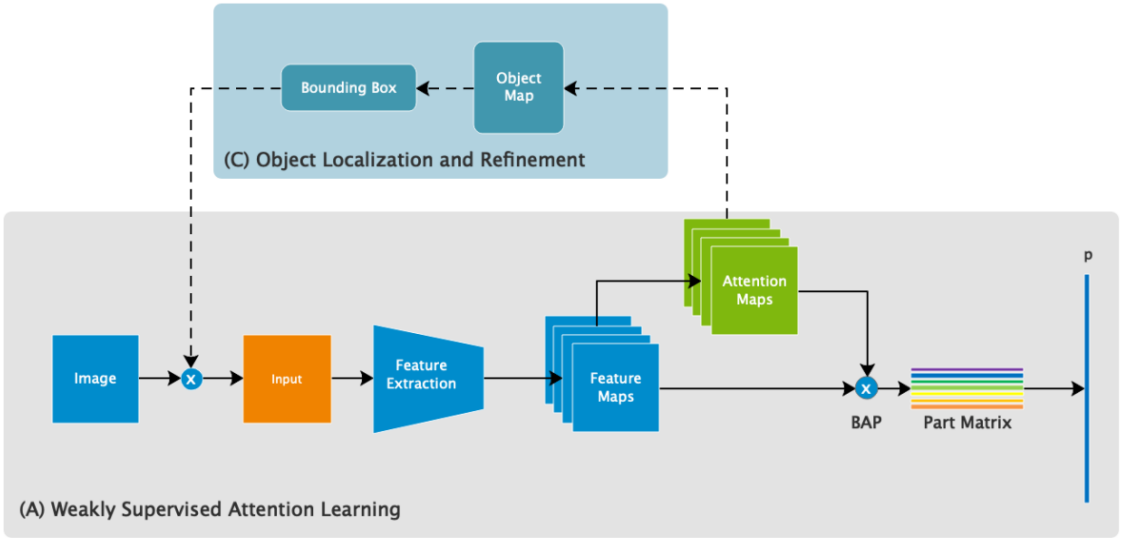
Training process of WSDAN



(A) Weakly Supervised Attention Learning. For each training image, attention maps will be generated to represent the object's discriminative parts by weakly supervised attention learning.

(B) Attention-Guided Data Augmentation. One attention map is randomly selected to augment this image, including attention cropping and attention dropping. Finally, the raw and augmented data will be trained as input data.

Training process of WSDAN



Firstly, object's categories probability and attention maps will be outputted from raw image by

(A). Secondly, object will be located according to

(C) and then be enlarged to refine the categories probability. Finally, the above two probabilities will be combined as the final prediction.

Train model using own Dataset.

Using : inception_mixed_6e as feature extractor

Batch size : 4

Epoch : 100

Evaluation

Sensitivity: measure of how well a model can detect positive instances.

Sensitivity = (True Positive)/(True Positive + False Negative)

Specificity: measures the proportion of true negatives that are correctly identified by the model.

Specificity = (True Negative)/(True Negative + False Positive)

Result

Classification report

	precision	recall	f1-score	support
Curved Mayo Scissor	0.96	0.99	0.97	110
Scalpel	1.00	1.00	1.00	110
Straight Dissection Clamp	1.00	1.00	1.00	92
Straight Mayo Scissor	0.99	0.94	0.97	90
accuracy			0.99	402
macro avg	0.99	0.98	0.98	402
weighted avg	0.99	0.99	0.99	402

1. models and classify Scalpel and Straight Dissection Clamp perfectly
2. if according F1-score models quite classify both of scissor very well with 0.97 f1-score

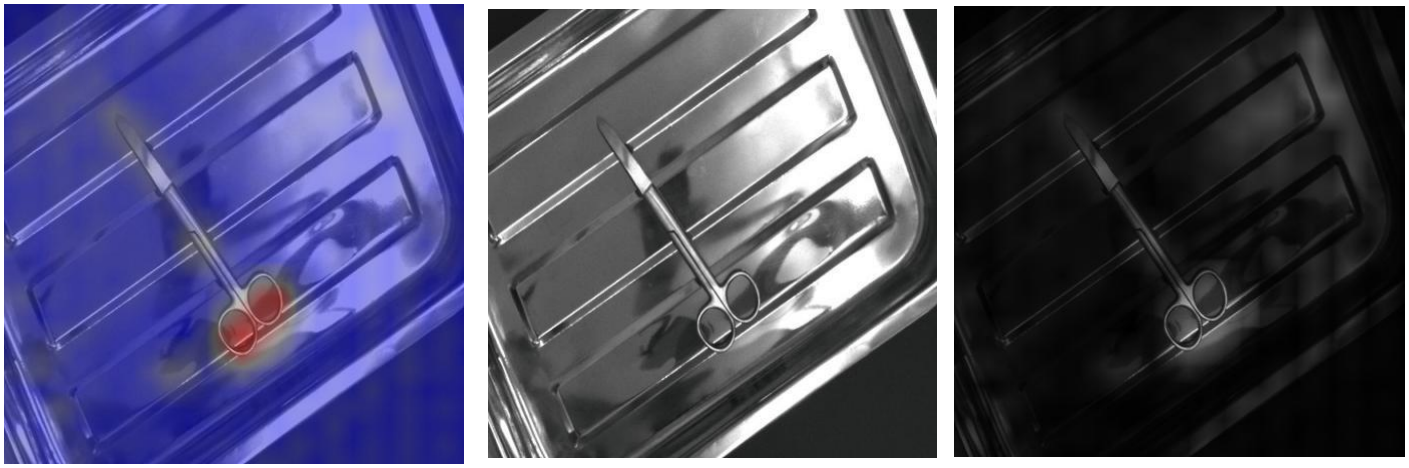
Sensitivity and specificity

Class	sensitivity	specificity
Curved Mayo Scissor	0.9909	0.9829
Scalpel	1.0	1.0
Straight Dissection Clamp	1.0	1.0
Straight Mayo Scissor	0.9444	0.9968

1. Class Curved Mayo Scissor have sensitivity 0.9909 that mean models can classify Curved Mayo Scissor class well
2. Class Straight Mayo Scissor have sensitivity 0.944 that mean models predict this class wrong more than curved Mayo Scissor and due to Scalpel and Clamp have 1.0 sensitivity. Seem models predict Straight Mayo Scissor wrong as Curved Mayo Scissor
3. That amount of data of Straight Mayo Scissor is less than Curved Mayo Scissor and that will effect by imbalanced data (not sure)

Some Attention Maps Visualization

Heat Attention Map **Raw Image** **Image x Attention Map**

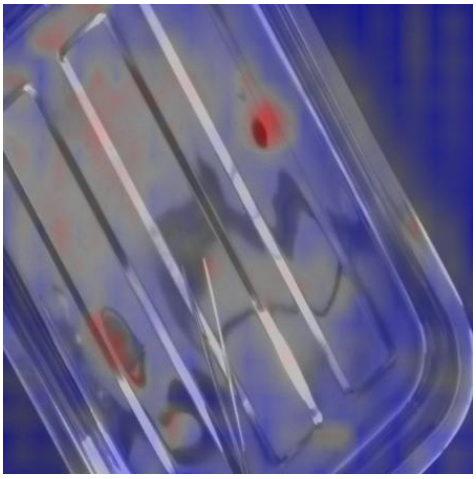
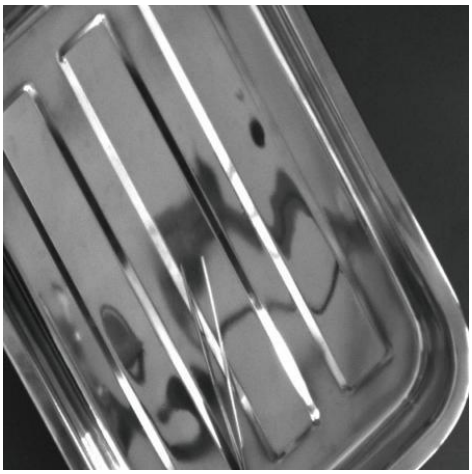
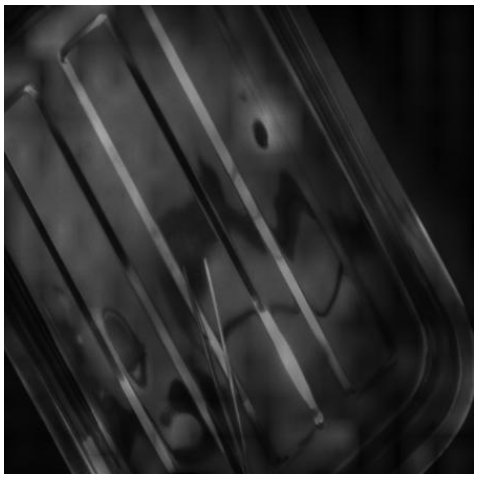


Heat Attention Map **Raw Image** **Image x Attention Map**

The models can localize the object. And can generate attention maps to represent the object's discriminative parts.

Heat Attention Map **Raw Image** **Image x Attention Map**

Heat Attention Map **Raw Image** **Image x Attention Map**

Some image -models cant localize object well due to it's have reflection of light and something but model still can classify this class correctly maybe it's because of dataset have small number of data classes

What I do next :

1. compose information that I have to do on all SCRUM and write a report.
2. Write final project document.
3. Analysis using performance metrics to compare all models

REF :

Multi-class Classification: Extracting Performance Metrics From The Confusion Matrix :

https://towardsdatascience.com/multi-class-classification-extracting-performance-metrics-from-the-confusion-matrix-b379b427a872

WSDAN :

https://arxiv.org/pdf/1901.09891.pdf

https://github.com/GuYuc/WS-DAN.PyTorch

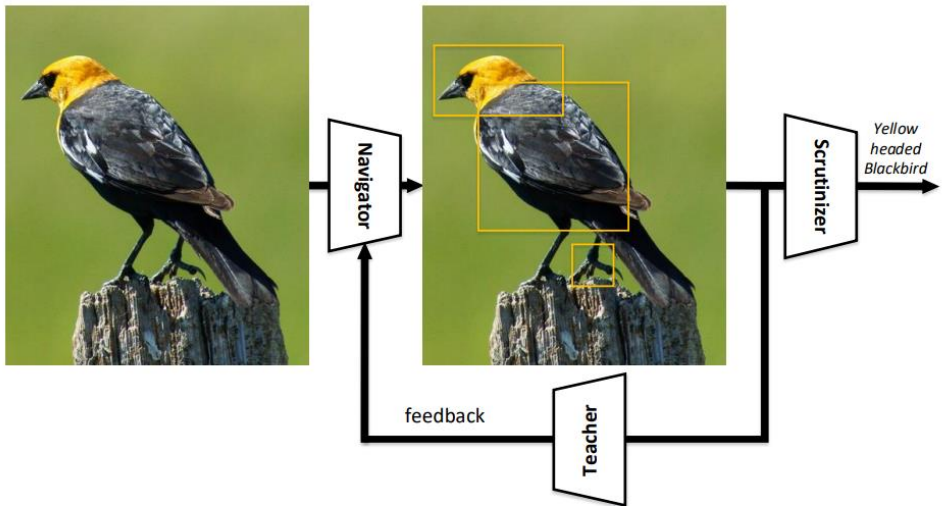
NTS :

https://arxiv.org/pdf/1809.00287v1.pdf

https://github.com/yangze0930/NTS-Net

NTS (Learning to Navigate for Fine-grained Classification)

NTS-Net for Navigator-Teacher-Scrutinizer Network, consists of a Navigator agent, a teacher agent and a Scrutinizer agent. In consideration of intrinsic consistency between informativeness of the regions and their probability being ground-truth class, we design a novel training paradigm, which enables Navigator to detect most informative regions under the guidance from Teacher. After that, the Scrutinizer scrutinizes the proposed regions from Navigator and makes predictions

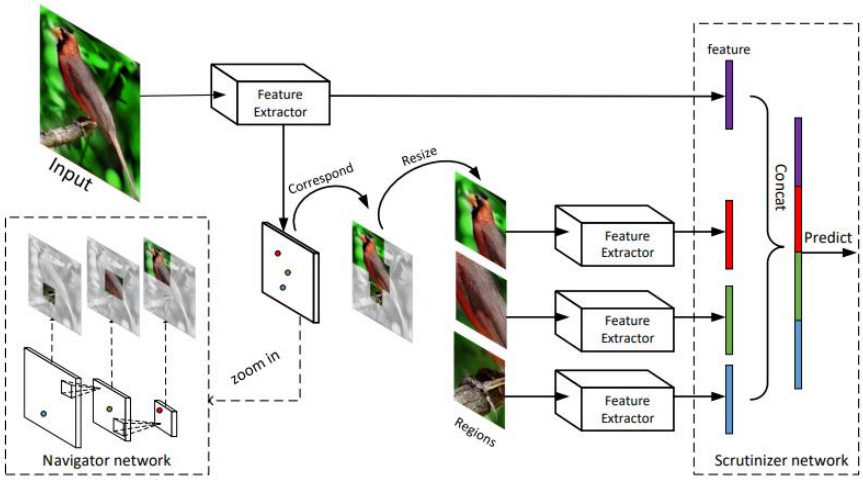


The overview of our model. The Navigator navigates the model to focus on the most informative regions (denoted by yellow rectangles), while Teacher evaluates the regions proposed by Navigator and provides feedback. After that, the Scrutinizer scrutinizes those regions to make predictions.

Navigator : Navigating to possible informative regions can be viewed as a region proposal problem

Teacher : The Teacher evaluates the regions proposed by Navigator and provides feedbacks

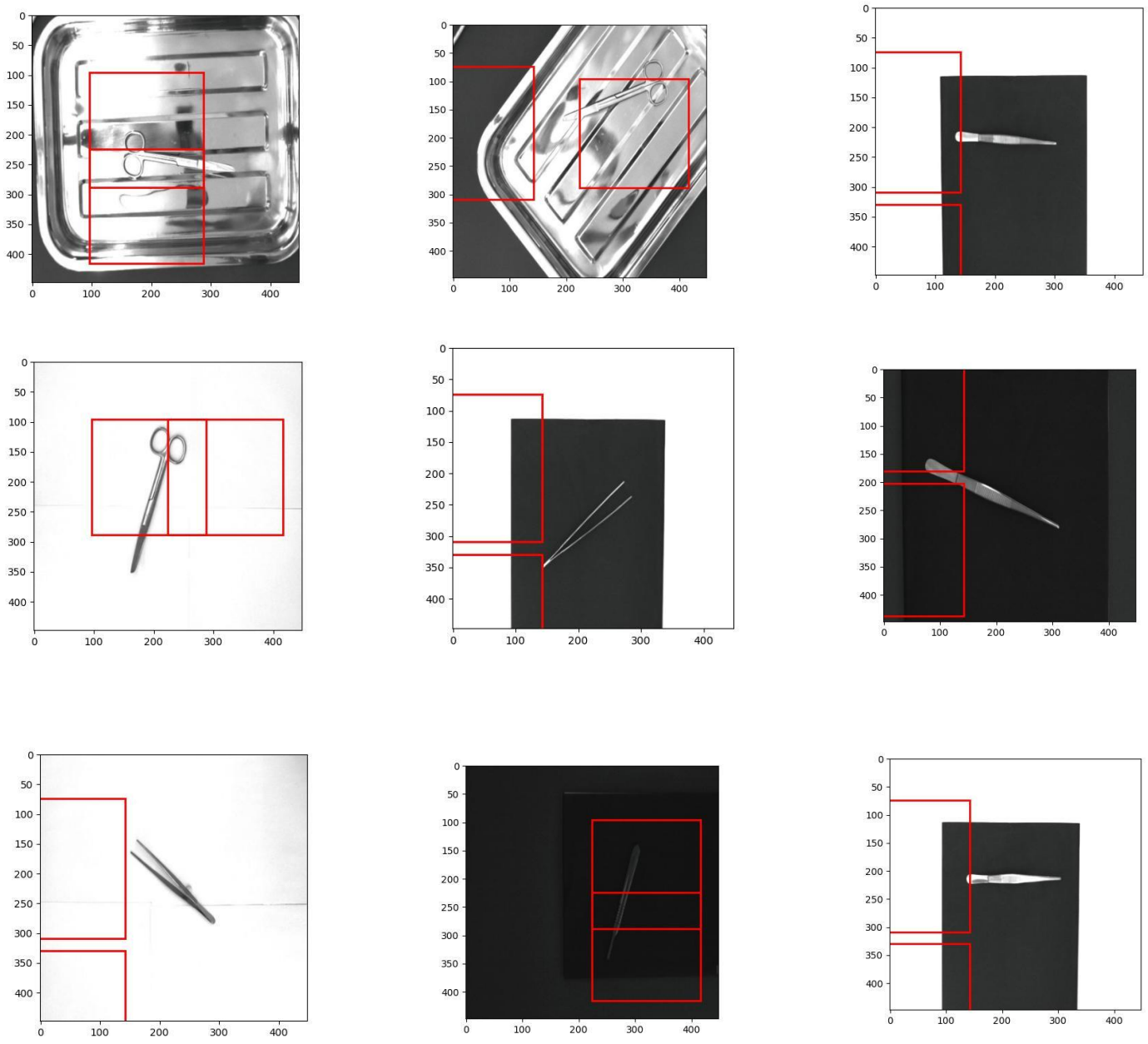
Scrutinizer : Scrutinizer network make decisions



The input image is first fed into feature extractor, then the Navigator network proposes the most informative regions of the input. We crop these regions from the input image and resize them to the pre-defined size, then we use feature extractor to compute the features of these regions and fuse them with the feature of the input image. Finally, the Scrutinizer network processes the fused feature to predict labels.

Visualization

Visualize the anchor box will be part of object



The part of object that obtained by model to use to feature extractor perform bad , the region that models chose isn't part of object. Maybe because of reflection of stainless tray make models learning not well.