IMAGE TILING METHODOLOGIES AND EFFECTIVENESS:

HIGH-RESOLUTION IMAGE CLASSIFICATION

March 15, 2023

Prepared by: Cameron Bass

Reviewed by: Xinyu Zhang

Department of Electrical and Computer Engineering

Mississippi State University

413 Hardy Road, Box 9571

Mississippi State, Mississippi 39762

Email: wcb262@msstate.edu

Research by Undergraduate Assistant Cameron Bass for Dr. Yu Luo on behalf of the Department of Electrical and Computer Engineering at Bagley College of Engineering, Mississippi State University. Additional assistance from Xinyu Zhang.

1.1 ABSTRACT

High-resolution datasets are notoriously difficult to process without loss of accuracy or performance in AI/ML image classification models. To assess data within these sets, models resize images to fit their network dimensions before processing. This typically results in missed detections and loss of quality that scales with image size. To alleviate the strain put on networks, we have conducted research into image tiling-based approaches for image classification. Collectively, there are three core image tiling methodologies that we wish to expound upon: static tiling, attentive tiling, and predictive tiling. The image tiling methodologies were each analyzed based off its efficiency and performance across high-resolution datasets. Results were organized according to the following parameters: latency, loss rate, intersection over union (IoU), and frames per second (FPS). All tests were performed on a NVIDIA Jetson Nano, one of our target devices for further research. After concluding testing, we found that accurate image tiling is largely dependent on the average target size expected from each dataset. That being said, the attentive tiling approach resulted in the highest flexibility across test cases, being the most likely to pick up both large and small detections. However, predictive tiling also offers a unique range of benefits for datasets expecting sufficiently small detections. All approaches created noticeable increases in the accuracy of detections at performance losses proportional to the amount of crops passed into each network. As we have concluded, image tiling, and particularly attentive image tiling are tremendously advantageous to achieving fast, accurate detections in high-resolution images. As the need for processing speed increases, demand for precise, reduced-impact techniques such as these offer an increasingly desired approach to AI/ML image classification and object detection.

1.2 CONTENTS

2.1 OVERVIEW

Computer vision tasks, such as image classification and object detection, perform well on low-resolution images. Higher-resolution datasets, specifically those of 4k and 8k resolutions, drastically decrease the accuracy of even state of the art detection models. One recently explored solution to improving the performance on high-resolution datasets, is the tiling-based approach. Simplified, this approach involves cropping a high-resolution image into smaller, more easily detectable crops before passing the images into a network. This approach, however, may heavily increase the load upon a device, taking a toll on performance. The following sections will outline and expand upon different techniques used for tiling-based methodologies, as well as their net benefits and losses.

3.1 METHODOLOGIES

3.2 Static Tiling

In traditional AI/ML training models, images are scaled to match a network's dimensions before being processed. This results in inaccuracy caused from down scaling. There are some models that offer other configurations, such as image tiling to remedy this. Static image tiling allows a high-resolution image to be processed as several lower resolution image crops that are overlayed onto the base image. Typically, crops contain a padding value that specifies the overlap between crops to prevent misses between intersections. The resulting tiles contribute to a general increase in detection accuracy, at the expense of a device's performance. This process can cause lengthy delays and will always have a time complexity proportional to the crop size. Below are some examples of static image tiling.

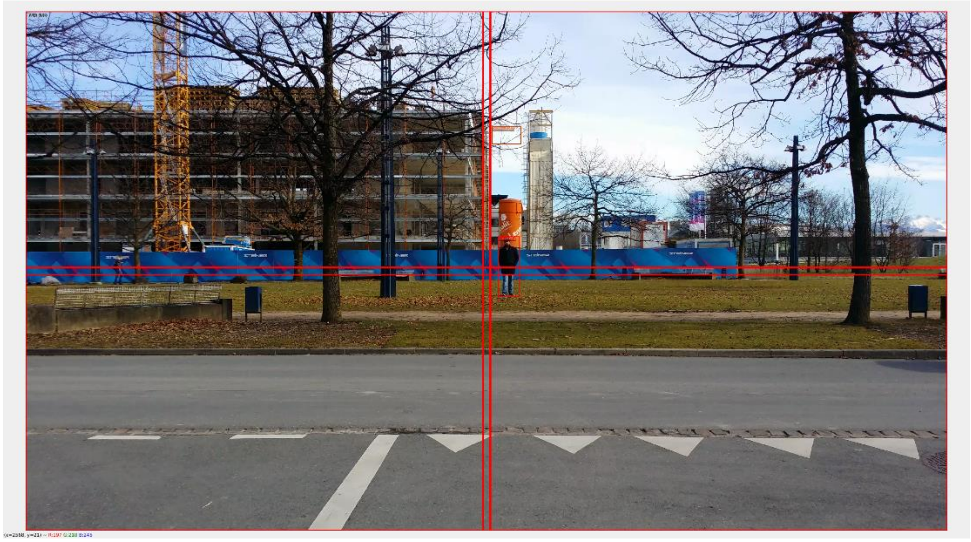

Figure 3.1: Static tiling example of image using 4 crops with 15 pixels of padding.

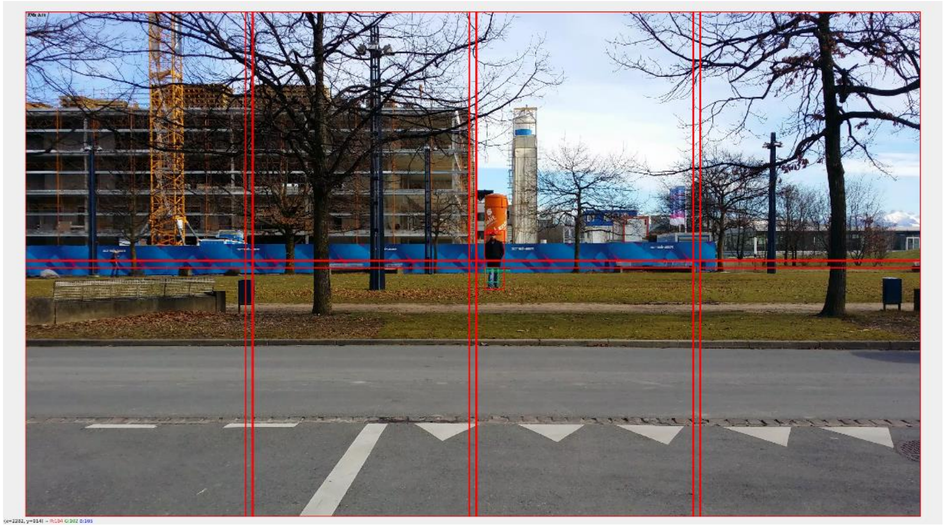

Figure 3.2: Static tiling example of image using 8 crops with 15 pixels of padding.

The output of image tiling will result in **N** times the precision of a model, where **N** is the crop size. Thus, statically tiling an image will reduce the shrinking factor when passed into a network by **N**.

3.3 Attentive Tiling

In 2018, researchers Vıt Ruzicka and Franz Franchetti from the Department of Electrical and Computer Engineering, Carnegie Mellon University developed the **Previtus Attention Pipeline** [1]. The Previtus Attention Pipeline is an implementation of a technique that I will be ascribing as an attentive tiling approach to high-resolution object detection. As outlined by the Previtus team, the process begins by performing a low-resolution crop over the base image. These crops are then scaled to the network dimensions within the model. After processing, the active bounding boxes are collected and crops containing bounding boxes are split into smaller high-resolution crops. These new crops are then passed into the network. The result is more precise image classification and object detection while maintaining greater performance than the static approach on the device. The process is outlined below.

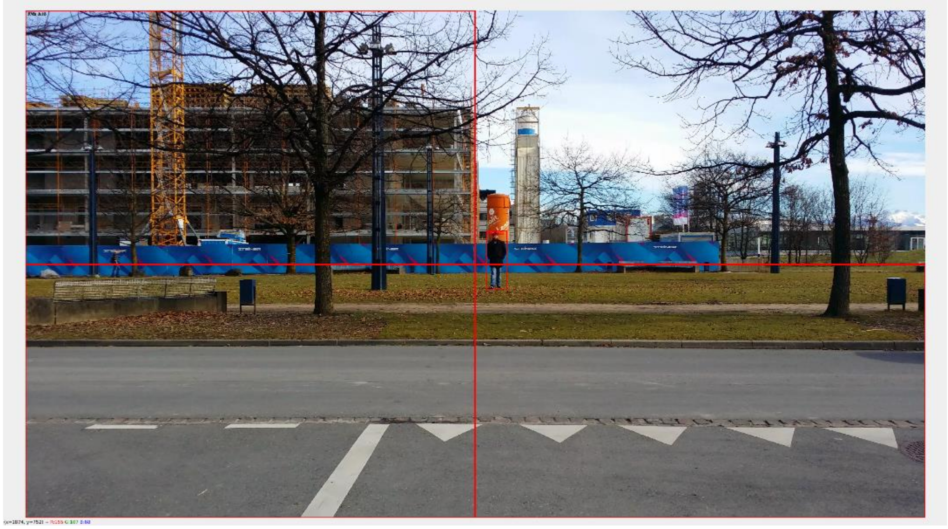

Figure 3.3 Image is split into a set of low-resolution crops to be evaluated.

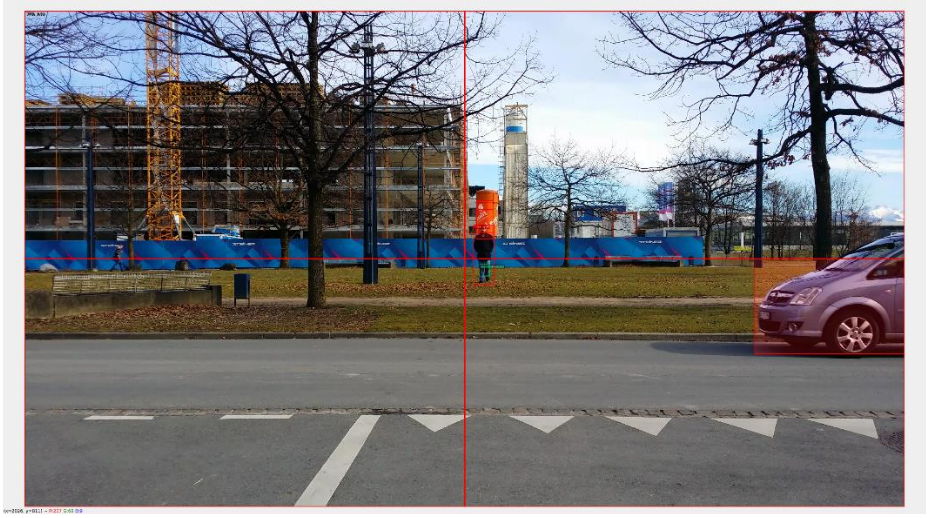

Figure 3.4 After being evaluated, a detection is flagged.

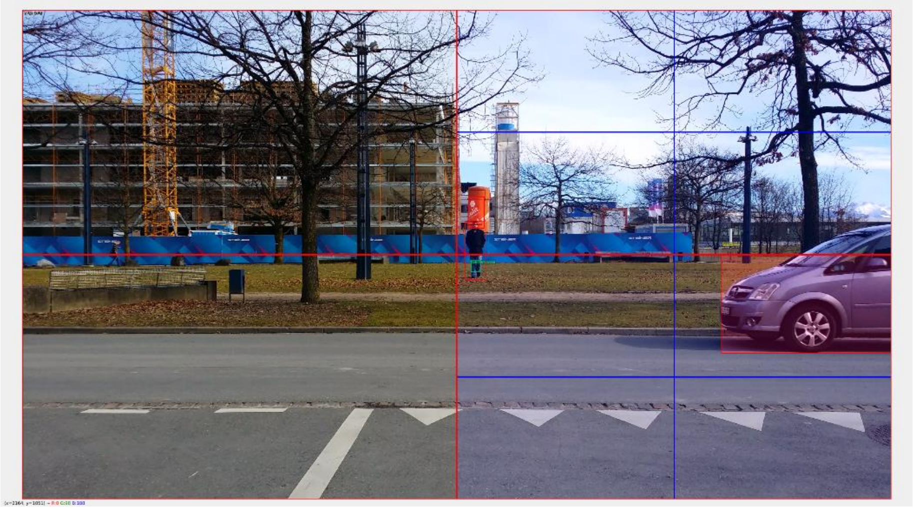

Figure 3.5 The low-resolution crop is split into smaller, high-resolution tiles that undergo final evaluation.

The previous steps are summarized by the researchers behind the Previtus Attention Pipeline in the graphic below.

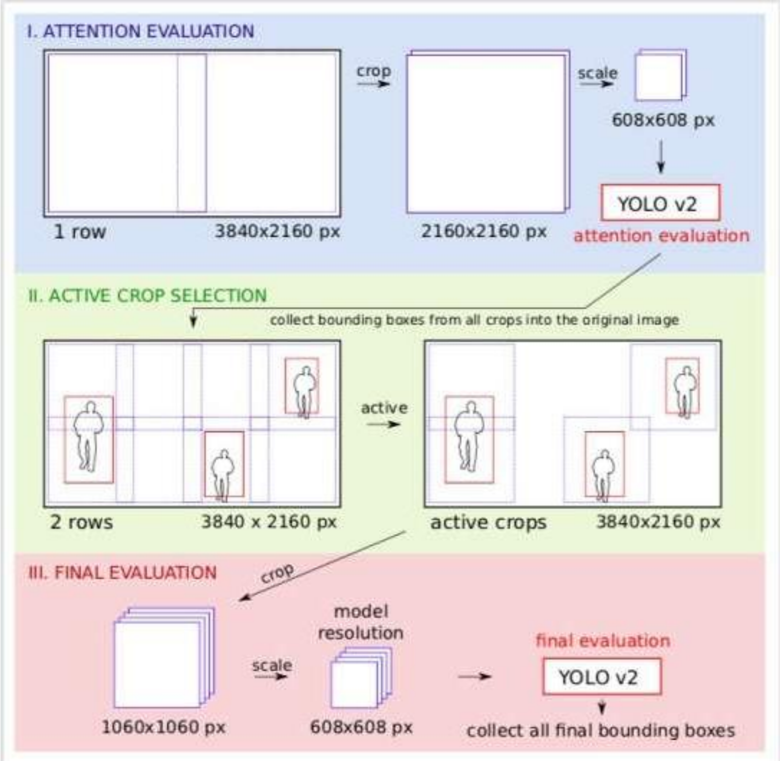

Figure 3.6: The attention pipeline as provided by the Previtus Attention Pipeline. [1]

Unfortunately, this solution does have the potential to miss smaller objects during the first evaluation phase. However, as an alternative to standard tiling, it has a noticeable increase on performance when detections are sparse.

3.4 Predictive Tiling

Predictive tiling utilizes a simple predictive algorithm to create crops based off previous detections. First, this model creates a border of crops equal to the desired dimension size. These crops are passed into the network for detection. If a target is detected, its bounding box is recorded for the next pass. During subsequent passes, new crops are predicted based on their proximity to the bounding box. Parameters may be added to the algorithm to predict the pixels per frame (PPS) for more precise crop activation. This effectively creates a perimeter around the image, only processing the information that is needed at any given instance. The predictive process is illustrated below:

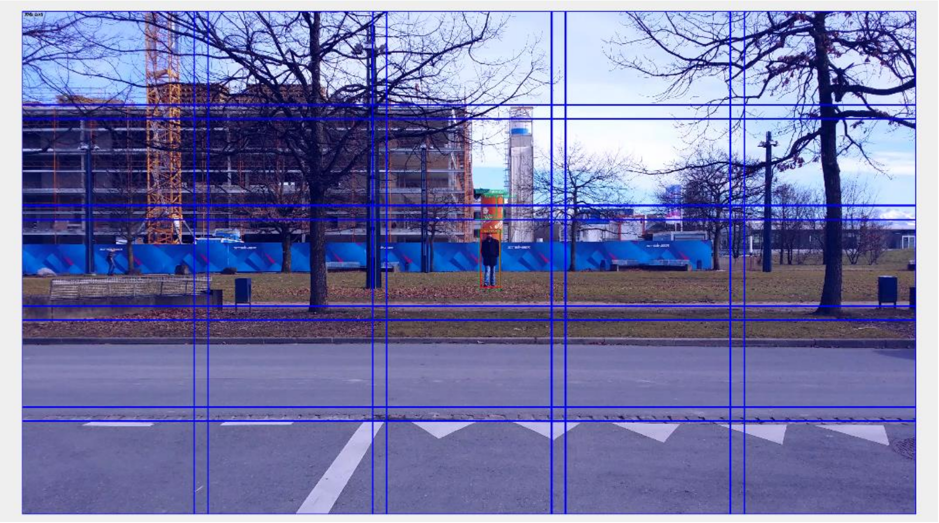

Figure 3.7: Initial worst-case scenario is performed.

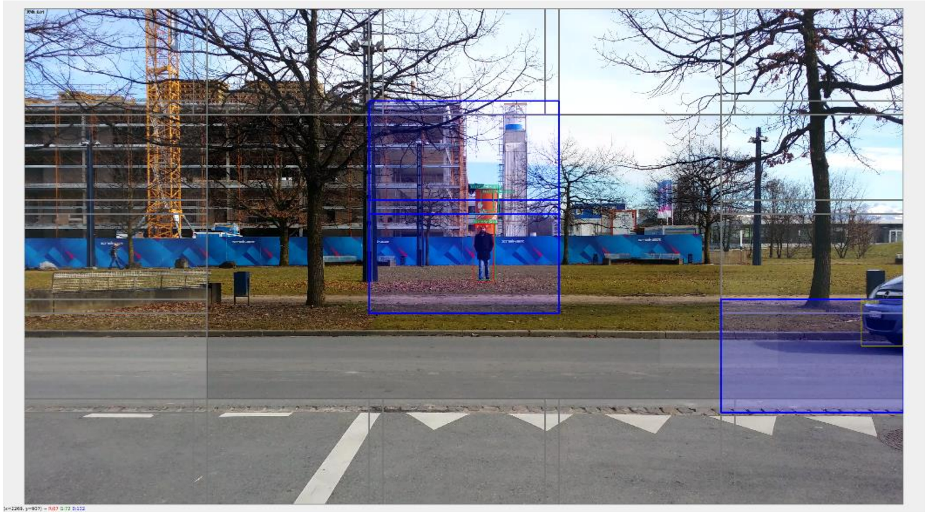

Figure 3.8: Active detections and a border of crops are evaluated along the perimeter of an image.

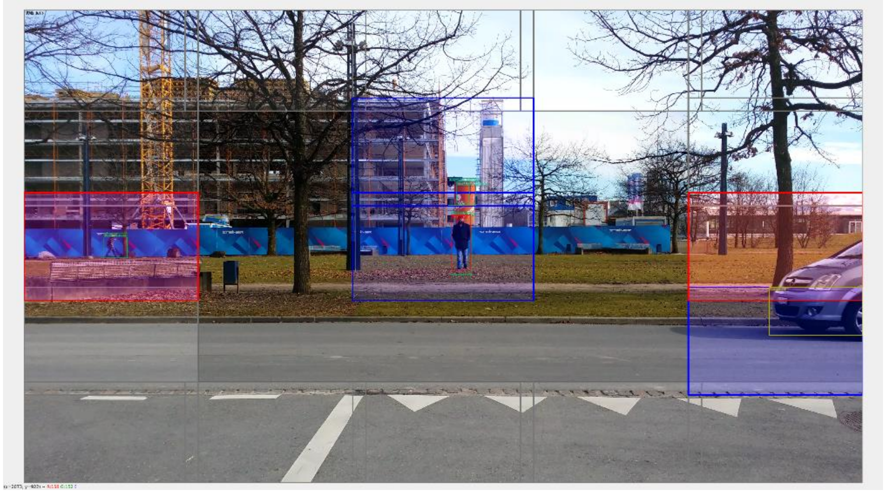

Figure 3.9: The previously expected detection is no longer present.

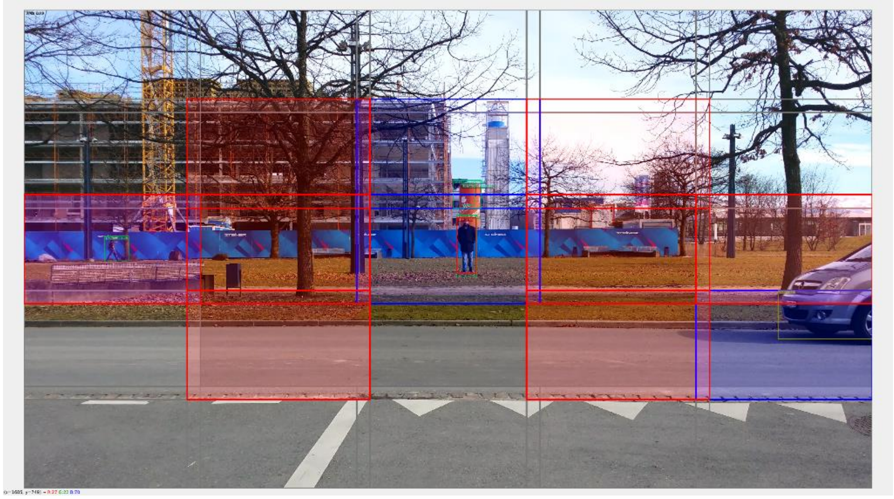

Figure 3.10: Adjacent tiles are evaluated in an attempt to locate the missed detection.

While this method still has a maximum time complexity to equal to that of standard tiling, it has a drastically decreased lower bounds, having to only evaluate the perimeter and crops that have previously included a detection or are adjacent to a missed detection. It should be noted that to differentiate itself from static tiling, a predictive tiling algorithm must have at least 9 tiling crops, this ensures that a border is created around at least one other tile. See note[1].

[1] The first pass of the predictive method must always be equal to the worst-case scenario in order to properly initiate the algorithm.

4.1 EVALUATION

4.2 Test Procedure

Tests were conducted by evaluating the performance across high-resolution (UHD) annotated videos. Test results were then compared against a non-tiled approach utilizing the same configuration and datasets. All tests were carried out on a NVIDIA Jetson Nano using modified versions of the TensorRT Yolov3 Tiny model [2] provided by GitHub user @jkjung-avt with weights by @pjreddie [3]. The images were classified via the standard COCO name set. A confidence threshold of 0.3 and tile padding of 15 pixels were used for all cases. The set of 4k videos used for testing were sampled from the PEViD-UHD dataset [1]. Each tiling method was examined in tile size increments of 2 (up to a maximum of 16), where tile size is equal to the amount of crops overlayed onto an image. See note[2] for utilized formulas.

Table 4.1: Device Specifications [4]

NVIDIA Jetson Nano	
Processor	Quad-core ARM Cortex-A57 MPCore processor
Graphics	NVIDIA Maxwell architecture with 128 NVIDIA CUDA cores
Memory	4 GB 64-bit LPDDR4
Storage	16 GB eMMC 5.1

[2] Loss Rate = $\frac{missed\ dets}{expected\ dets}$ Intersection over Union (IoU) = $\frac{|A \cap B|}{|A \cup B|}$

4.3 Test Results

The results of testing are broken down in the following section. To accurately weigh the benefits gained from different tiling techniques, test results were organized according to efficiency and performance. The parameters included within efficiency are intended to reflect the performance of the algorithm, whereas performance parameters are used to benchmark the impact on the device itself. **Table 4.2** and **Table 4.3** contain the relevant data collected during testing. A sample of the annotated outputs that were generated for each method can be accessed here.

Table 4.2: Efficiency Analysis

	FPS	**Latency (S)**	**Loss Rate**	**IoU**
No Tiling	4.08	0.04	0.89	0.24
Static Tiling				
Tiles: 2	3.60	0.08	0.31	0.21
Tiles: 4	2.77	0.15	0.44	0.20
Tiles: 8	1.93	0.31	0.11	0.19
Tiles: 12	1.45	0.46	0.03	0.25
Attentive Tiling				
Tiles: 2	1.96	0.29	0.18	0.19
Tiles: 4	1.56	0.45	0.25	0.24
Tiles: 8	1.14	0.88	0.09	0.31
Tiles: 12	0.96	0.81	0.09	0.32
Predictive Tiling				
Tiles: 12	1.25	0.56	0.18	0.33
Tiles: 16	1.02	0.74	0.35	0.39

Table 4.3: Performance Analysis

	CPU Usage (%)	GPU Usage (%)	CPU Temperature (C)	GPU Temperature (C)
No Tiling	89.6	84.1	46.9	44.0
Static Tiling				
Tiles: 2	91.5	78.8	46.4	43.7
Tiles: 4	88.6	80.6	48.8	46.1
Tiles: 8	89.8	77.8	55.5	52.8
Tiles: 12	95.2	82.8	47.9	48.3
Attentive Tiling				
Tiles: 2	93.3	78.7	52.0	49.3
Tiles: 4	91.5	79.6	51.0	48.4
Tiles: 8	94.8	83.8	48.0	48.5
Tiles: 12	93.5	85.3	48.1	48.0
Predictive Tiling				
Tiles: 12	89.8	77.9	51.9	49.3
Tiles: 16	90.9	79.5	55.0	52.3

4.4 Test Conclusions

As shown from the above test results, it is difficult to come to a firm conclusion on which pattern balances both speed and accuracy desirably. Due to the wide range of shot composition that is included within the PEViD UHD dataset, I found that consistent improvement is generally unique to the expected shot distance of each video. While high-resolution tiling crops may greatly increase the accuracy of faraway detections, the model may miss closer detections that would otherwise be flagged from a lower resolution crop. Intriguingly, I also found that IoU and loss rate are not necessarily linked. Many decreases in loss rate correspond with very little gain in IoU. This has led me to the conclusion that the silhouette of an object to the neural net only changes during a drastic difference of scale when input into the network. That being said, there is an obvious spike in IoU for ultra high-resolution crops, even as more detections are missed from over scaling the input video. Unfortunately, the predictive tiling pattern suffers most from the over scaling brought on by higher resolution tiling crops. In order for the predictive pattern to distinguish itself from a static pattern, it must have 9 crops at a minimum, meaning that this pattern can only be effectively utilized on small, far away targets. Alternatively, the attentive model offers relatively consistent performance across tile sizes.

5.1 Conclusion

During testing, it became apparent that overall performance across methodologies was primarily determined by the composition of the input. As such, great consideration should be given to expected size and distance of detections before deciding on an approach. Due to the possible size variation within input videos, I believe that the attentive tiling pattern best encapsulates flexibility and performance. The attentive approach is most likely to flag targets both close and far, evident from testing. This and other methods also benefit from increased padding between tile crops, allowing the network to detect both larger targets and targets that overlap between crops. Based on my research, I believe that it is safe to conclude that the attentive tiling approach should be implemented in some form to ensure both reliability and speed across a network.

5.2 Optimization

The predictive methodology, though not performing exceedingly well on the mid distance PEViD dataset, has increasing potential for smaller targets/targets at farther distances. Unfortunately, this algorithm can also be easily unnecessarily become bogged down in implementation. The current version of the predictive method used during testing relies on a simple quadrant-based calculation that will check adjacent tiles if an expected detection is no longer present. A much more efficient way of conducting checks for missed detections would be to reformat the quadrant-based approach into a velocity-based approach. Using target identifiers, one could calculate the velocity (pixels per frame) of a detection and based on prediction, use it to determine which crop(s) to access. This has the potential to increase both

FPS and accuracy. As Yolo V3 does not yield unique target identifiers, however, this is not feasible in the current implementation. It could be potentially optimized by running detections through an external neural net to determine a target ID. Though, consideration would need to be given to the potential performance loss of implementing such a network. Other approaches, such as the attentive tiling method, may benefit from extending the functionality of the predictive method when dealing with datasets suited for small target detection. It should also be noted that further experimentation with these methodologies may give more definite results. If implemented with target identifiers, calculations such as accuracy[3] and precision[4] may help expand upon this research. Additionally, though the PEViD datasets are an invaluable resource, much of their ground truth files are left unannotated. This means that, although some detections align with the ground truth, many otherwise accurate detections are left with nothing to compare against. This could easily be addressed by fully annotating the PEViD dataset.

[3] Accuracy = $\frac{TN+TP}{TN+FP+TP+FN}$

[4] Precision = $\frac{TP}{TP+FP}$

where TN, TP, FN, FP == True Negative, True Positive, False Negative, False Positive

6.1 References

[1] P. Korshunov, and T. Ebrahimi. UHD Video Dataset for Evaluation of Privacy. Sixth International Workshop on Quality of Multimedia Experience (QoMEX 2014), Singapore, 2014.

[2] JK Jung, “tensorrt_demos.” github.com https://github.com/jkjung-avt/tensorrt_demos (accessed March 15, 2023).

[3] J. Redmon “Darknet.” pjreddic.com https://pjreddie.com/darknet/ (accessed March 15, 2023).

[4] Nvidia “Jetson Nano Developer Kit.” developer.nvidia.com https://developer.nvidia.com/embedded/jetson-nano-developer-kit (accessed March 15, 2023).