NHL STENDEN
computer vision & data science

Anomaly detection in human X-Ray images with YOLOv5

NHL Stenden Professorship in Computer Vision & Data Science

This concept paper is under review by the supervisors.

M. Alejandro Villalobos C.

Supervisors: Willem Dijkstra, Klaas Dijkstra

Abstract— This research aims to evaluate the effectiveness of YOLOv5, a state-of-the-art machine learning architecture for object detection, in identifying contraband items in X-ray images. The model will be trained on a dataset of human X-ray images and tested on its ability to detect anomalies. The baseline model uses the full image as input and outputs predicted bounding boxes for each class. The study then conducted several experiments to address supporting questions, including the impact of data augmentation, the relationship between tile size/number of tiles and model performance, and the effect of the number of classes and class distribution on model performance. The ultimate goal is to determine the viability of using YOLOv5 to identify anomalies on x-ray images. While the results are promising, it is desirable to improve the model's performance for the application.

Index Terms— YOLOv5. Object detection. Computer vision. Convolutional Neural Networks. Anomaly detection. X-Ray images.

1 Introduction

It is crucial to maintain a safe environment in correctional facilities, as these can easily become dangerous places for both staff and prisoners. For this reason, many items are deemed illegal for prisoners to possess, such as drugs and weapons. These items are known as contraband and, unfortunately, they are still smuggled into the facilities. "Prison staff needs to be able to detect and confiscate contraband quickly to prevent drug abuse, violence, and the commission of further crimes" [1]

For this reason, visitors and staff must be searched before entering the facilities and inmates after having contact with the outside world. However, traditional searching methods can be intrusive and time-consuming. Additionally, it is possible to miss contraband if it is well-hidden or stored in a person's cavities. This highlights the need for an inspection method that is faster and more reliable without being overly intrusive and why X-ray imaging is introduced. X-ray-based systems allow for quick, reliable, and safe inspections of inmates without being invasive. A trained human operator only needs to examine the X-ray image to visualize anomalies and easily detect most types of contraband, such as weapons or cellphones. In this case an anomaly is defined as an object that is not part of the human anatomy, this definition is used throughout the paper.

Even though this inspection method is superior, it is still dependent on an operator, leaving room for human error. In this arrangement, the operator becomes a single point of failure (SPoF); meaning the entire system depends on their prediction. Adding a second layer creates redundancy thus eliminating the SPoF problem. Furthermore, some contraband items are exceptionally hard to detect for the human eye, such as drugs. However, a trained model might be able to detect these elusive items. For these reasons, it is desirable to aid operators in the task of detecting contraband by adding an object detector without fundamentally changing the system [2].

- *M. Alejandro Villalobos C. is a Computing Science student at the NHL Stenden University of Applied Sciences, E-mail: mauricio.villalobos@student.nhlstenden.com.*
- *Willem Dijkstra is a researcher at the NHL Stenden Professorship in Computer Vision & Data Science, E-mail: willem.dijkstra@nhlstenden.com.*
- *Klaas Dijkstra is a lector at the NHL Stenden Professorship in Computer Vision & Data Science, E-mail: klaas.dijkstra@nhlstenden.com.*

1.1 Goal

To perform research on the viability of YOLOv5 as an object detector to identify contraband and other anomalies from X-Ray images. The model will be trained on a dataset of human X-Ray images and will be evaluated on its ability to detect anomalies.

1.2 Research Question

What is the performance of an object detection model for identifying anomalies in x-ray images?

1.2.1 Supporting Questions

- How does the model performance compare when using the full image or random positive tiles?
- Also, what is the relationship between tile size/number of tiles and model performance?
- What data augmentations increase performance on the model effectively?
- How is the model's performance affected by the number of classes?

2 State of the Art

In this section, a thorough overview of the current state of the art in object detection in x-ray images is presented. It begins by discussing the most recent advancements and breakthroughs

in the field, followed by a review of the key techniques and methods currently being used.

2.1 X-Ray images

The state-of-the-art in object detection on X-ray images has advanced significantly in recent years [3][4], thanks to the development of powerful machine learning algorithms and large datasets for training. Currently, the most widely used methods for object detection on X-ray images are deep learning-based, utilizing convolutional neural networks (CNNs) to learn and identify objects in images.

One of the key challenges in object detection is the large variations in object appearance, pose, and scale. This makes it difficult for traditional, hand-crafted feature extraction methods to accurately detect objects. To address this issue, many recent studies have used CNNs trained on large datasets of X-ray images to learn robust and discriminative features automatically such as [5]. X-ray images have the added challenge of presenting a large amount of noise. This is specially the case for non-medical x-ray images, which often use a lower dose of radiation for health and safety reasons.

2.2 YOLOv5

The state-of-the-art in object detection is constantly evolving, with new methods and techniques being developed and published regularly. YOLOv5 (You Only Look Once version 5) [6][7] is one of the most widely used object detection models, known for its high accuracy and impressive performance in various benchmarks, including the popular MS COCO dataset.

YOLOv5 was developed by the team at Ultralytics and builds upon the success of previous YOLO models. It uses a single convolutional neural network to predict multiple bounding boxes and class probabilities for objects in an image.

One of the key components of YOLO architectures is their use of a novel network architecture called SPP-Net (Spatial Pyramid Pooling Network), which enables the model to handle a wide range of input sizes without the need for fixed-size feature maps. This allows YOLOv5 to make predictions on high-resolution images without sacrificing performance.

Additionally, YOLOv5 recommends the usage of a variety of data augmentation methods - such as Mosaic, Copy Paste, Random Affine, Mixup, etc - which creates a larger, virtual training dataset. This can help improve the generalization of the model and increases its performance on a variety of object classes.

Overall, YOLO architectures are powerful and efficient object detection models that have demonstrated state-of-the-art performance on a variety of benchmarks and real-world applications [8].

3 Materials and Methods

The research will be performed on a single core from an Intel i9-7960X CPU and an NVIDIA GeForce RTX 2070 GPU. The software used will be Python 3.8.13 and the libraries on the annexed conda environment.

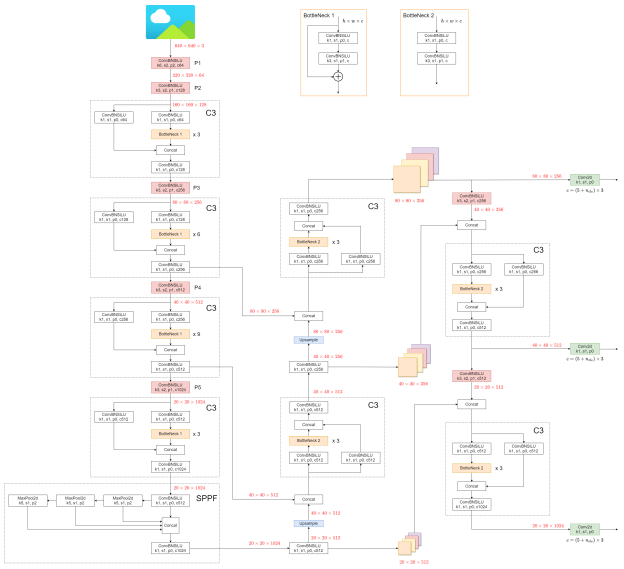

Fig. 1. YOLOv5 architecture [9]

3.1 Model Architecture

The model architecture chosen for this study was that of YOLOv5. As mentioned before, it is a state-of-the-art family of models that are trained on the COCO dataset and are available in different sizes. The models in the latest YOLO generation are YOLOv5s, YOLOv5m and YOLOv5l. The last 2 are 2x and 4x larger than the smallest model respectively; these larger models have the added benefit of being more accurate at the significant cost of speed and computational expense. The model chosen for this study is the smallest one, YOLOv5s, because it is the fastest one and least computationally expensive, making it the most suitable one for this application, as it calls for quick predictions without demanding to many resources. In Figure 1 you can observe the YOLOv5 architecture diagram.

3.2 Datasets

For this study, a privately owned dataset of X-Ray images called ”THEIA dataset” was obtained. The dataset contains 247 images of X-Ray scans of prisoners, visitors and staff; all images are 16-bit images sized 1500 x 1216 pixels. This dataset was then split into a training set, a validation set and a test set. The training set contains 148 (60%), the validation set contains 49 (20%) and the test set contains 49 (20%) images. The images were split into 3 sets to ensure that the model was not overfitting to the training set. This is a comparatively small dataset and a larger dataset would be ideal; this is further discussed in the 5.3 Future Work subsection. The dataset is not publicly available, as it contains sensitive information.

To annotate the data, a variety of filters were used to aid in the visualization of the anomalies on the image. Such filters adjusted to different degrees brightness, contrast, and gamma values or by applying algorithms such as contrast limited adaptive histogram equalization (CLAHE) or global normalization. Figure 2 shows a comparison between the original image (left) and two different filters. The data is

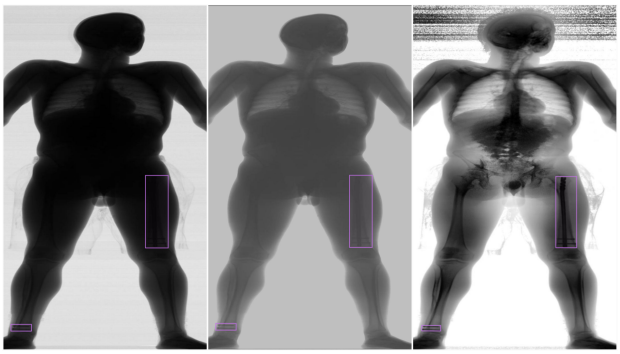

Fig. 2. Image without filter (left), image with contrast and brightness filter (middle), and image with gamma correction and CLAHE filter (right). Subject has two surgical implants, marked with bounding boxes

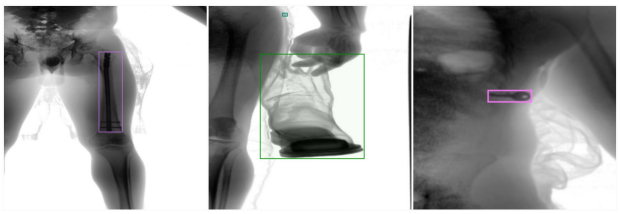

Fig. 3. Surgical implant (left), holding shoes (middle), and key (right)

annotated with simple bounding boxes without any rotation. These are then exported to an XML file in PASCAL VOC format.

There is a total of 18 classes in the dataset, these are considered anomalies and it is desired to detect them. The classes are: zipper, watch, unknown anomaly, surgical implant, shoes, piercing, key, jewelry, holding shoes, glasses, drugs, cuffs, bra, belt buckle, zipper head, ankle monitor, shirt button, and pants button. Some examples can be seen in Figure 3. It is important to mention there is class imbalance in the dataset, as some classes are much more common than others (See class distribution in Figure 4). For example, the classes pants button, unknown anomaly, shirt button and zipper collectively represent 65% of the anomalies. This class imbalance could be addressed by using a weighted loss function [10][11][12], which will give more importance to the classes that are less common. However, this falls outside the scope of this study but is discussed again in the 5.3 Future Work subsection. Another way to address this issue is to remap the classes to a smaller set of classes. This would essentially group the classes that are similar together and would reduce the number of classes. This second approach is what was done in this study and discussed in more detail in the 4 Experiments and Results section.

3.3 Evaluation

During training, the model will use the validation set to determine when to save the model. The model will be saved when the validation loss is at its lowest, thus functioning as a form of early stopping. The model will be evaluated every other epoch, as validating the model every epoch would be computationally expensive. This process continues until the model has been trained for 150 epochs; at which point the

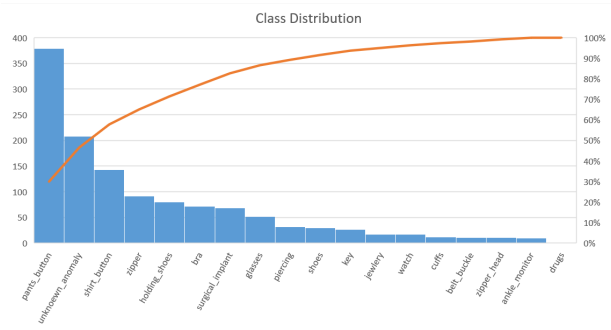

Fig. 4. Histogram of the class distribution showing text-book class imbalance

model will be evaluated one last time on the test set.

On testing, the model will be evaluated using the following metrics: precision, recall, and F1 score. The precision is the ratio of true positives to the total number of positives (Equation 1). The recall is the ratio of true positives to the total number of true positives and false negatives (Equation 2). The F1 score is the harmonic mean of precision and recall (Equation 3). With these metrics we will be able to get a deep insight as to the model's limitations and performance. This evaluation will be done on the test set; the model has not interacted with these images during training nor validation. The test images will be tiled to match the image size on which the model was trained on. This is a fixed tiling method, in which the entire image is evenly split. The model will be evaluated on the test set without data augmentation, as the real-world images will not be augmented.

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives} \tag{1}$$

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives} \tag{2}$$

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{3}$$

3.4 Data augmentation

Data augmentation is a technique used to increase the size of the training dataset by creating new samples from existing ones. This can done by applying a variety of transformations to the images, such as cropping, flipping, rotating the image (among others). The goal of data augmentation is to increase the diversity of the training dataset, thus improving the generalization abilities of the model. However, some data augmentation techniques can be detrimental to the model. For example, if our dataset contains images of alphanumeric characters, rotating or flipping the image could render the character unreadable, as "p", "q", "b" and "d" would be indistinguishable. Another example is if the dataset contains fine details, adding noise or blurring the image might destroy these details. This is why data augmentations should be carefully chosen for each dataset.

Keeping in mind that the THEIA dataset consists of gray-scale human x-ray images, we know we can apply transformations such as horizontal flipping, noise, blurring,

brightness and contrast. These transformations won't void the images from any meaning and have been used and proved extensively in the literature. For these reasons, the augmentations chosen for this study are: random horizontal flip, Gaussian noise, Gaussian blur and random brightness and contrast. These augmentations will be applied to the training set only, as the validation and test sets should not be augmented. The augmentations will be applied randomly, with a probability of 50% for each augmentation. This is done to ensure that the model is not overfitting to the augmentations. If the image is augmented the bounding boxes will be adjusted accordingly.

3.5 Random positive tiling

In some cases it is desirable to resize images from a dataset, as the images might be in different resolutions or they require too much memory to train efficiently. When this is the case, traditional methods include resizing and/or padding the image into the desired size. However, a different method is tiling; in this case the image is split into smaller portions or tiles. Random positive tiling is a data augmentation technique in which the image is tiled but the resulting segment is guaranteed to contain an annotated object. Positive tiling is useful to avoid overfitting the model to negative samples that could occur by tiling the entire image at fixed intervals or random spots.

Benefits of random positive tiling include an inherent translation invariance, as the tile guarantees to contain the object, however the position of the object in the tile is random. Also, the model is trained on a larger dataset, as each image is cropped into multiple tiles and can be reused several times before overfitting. Finally, the tiles are smaller than the original image, which means that the model can train on multiple images without running out of memory. One disadvantage of random positive tiling is that the context of the object is lost, as the area surrounding the object will be reduced or entirely removed. The impact these benefits and disadvantages have on the model performance will depend on the dataset, as context around the object might be important for some datasets and not for others.

4 Experiments & Results

There are countless variables that have the potential of impacting the performance of the model. For this paper, it was decided to experiment with tiling parameters, data augmentations, and class remapping. These were chosen because they target the three main challenges in the study; memory availability, lack of data and class imbalance.

Some important hyper parameters and their values are as follows: learning rate was set to 0.001, batch size to 8, and optimizer to Adam.

4.1 Tiling Parameters

When tiling the images, it is possible to choose the size of the tiles, and the number of positive tiles. The size of the tiles determines the amount of information that the model will have access to for any given tile. On the other hand, the number of positive tiles is important because it determines how many positive samples the model will have access to per epoch. However, the tile size and amount of tiles is limited by the available memory. For this reason, an experiment was designed to find the optimal tile size and number of positive tiles. The experiment consisted of training the model with increasingly more tiles and the largest tile size possible for the amount of tiles. As a baseline or control group, it is also compared without any tiling, meaning that the entire image was used. The results of the experiment are shown in Figure 5.

Tile Size vs Number of Tiles (Batch Size 8)				
num_tiles	tile_size	precision	recall	F1
no tile	full image	0.6020	0.2719	0.3746
1	1024	0.6518	0.3364	0.4438
2	704	0.7388	0.4562	0.5641
3	576	0.5930	0.4700	0.5244
4	512	0.7229	0.5530	0.6266
5	448	0.7721	0.4839	0.5949
6	384	0.6725	0.5300	0.5928
7	384	0.7535	0.4931	0.5961
8	320	0.6994	0.5576	0.6205
9	320	0.7097	0.5069	0.5914
10	320	0.6458	0.5714	0.6064
11	256	0.6800	0.5484	0.6071
12	256	0.6456	0.4700	0.5440
13	256	0.6237	0.5346	0.5757
14	256	0.6011	0.5069	0.5500
15	256	0.6649	0.5760	0.6173
16	256	0.6558	0.4654	0.5445

Fig. 5. Tiling experiment table

The results of the experiment show that the model's performance does increase as we introduce more smaller tiles. However, this growth slows down significantly as we increase the number of tiles. Furthermore, while inspecting the model's loss graphs, we can observe two tendencies. The first is a significant difference when the number of tiles is less than three, as instead of obtaining a smooth loss curve, the loss curve becomes jagged and jumpy. This is likely due to the fact that the model is not being trained on enough positive samples. The second is that with more smaller tiles the model requires more epochs before converging. This is important as the training process becomes lengthy while performance stays nearly the same. The tiling arrangement that performed the best was 4 tiles of size 512 x 512 as it produced the best F1 score. The results of the experiment are shown in Figure 6.

It should be noted that because tiling provided such a significant improvement in performance, the rest of the experiments were performed exclusively with tiling. The experiments were not repeated without tiling to save time and resources and to simplify the visualization and analysis of the results.

4.2 Data Augmentation

Data augmentation is well known to increase the performance in machine learning models, especially on limited datasets. Additionally, the THEIA dataset in comparably small to object detection datasets. For these reasons, augmentations might be beneficial and it is desirable to identify which augmentations have the best impact on the model's performance.

The other data augmentation methods used were random horizontal flips (H. Flip), Gaussian blur (G. Blur), random brightness and contrast (RBC), and Gaussian noise (G. Noise).

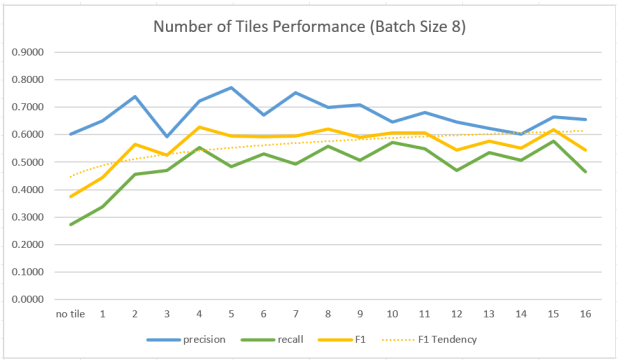

Fig. 6. Tiling experiment graph

Augmentations (Constant P = 0.5) - Overall			
augmentation	Precision	Recall	F1
None	0.7229	0.5530	0.6266
G. Blur	0.6647	0.5207	0.5840
H. Flip	0.6374	0.5346	0.5815
G. Noise	0.6981	0.3410	0.4582
RBC	0.7203	0.4747	0.5722
Combined	0.8070	0.6359	0.7113

Fig. 7. Data augmentation table

Each augmentation was tested individually in conjunction with random positive tiling utilizing the results of the previous experiments: 4 tiles sized 512x512, batch size 8. Finally, there is the control group with no additional augmentations other than random positive tiling and without performing the datatype change. The results of the experiment are shown in Figure 7 and Figure 8.

The results of the experiment show that the base model has a higher F1 score compared to individual augmentations. It appears as if individual augmentations have a negative impact, which is unexpected. The experiment was then repeated with a combination of the augmentations (Combined), resulting in an F1 score that is significantly higher than the base model's. It should be noted that the Gaussian Noise augmentation was not used in the combined augmentations as it produced a considerable negative impact on the model's performance. This is likely due to the fact that the images are already noisy and adding more noise makes it difficult for the model to identify smaller anomalies.

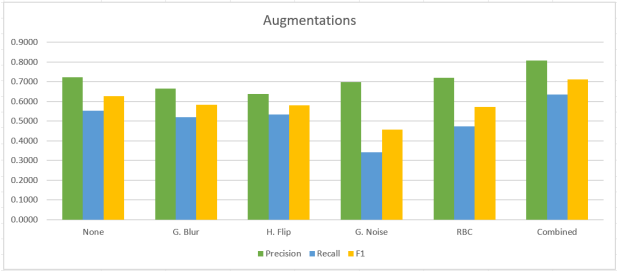

Fig. 8. Data augmentation graph

Remap			
Grouping	precision	recall	F1
Original	0.7229	0.5530	0.6266
Score	0.7125	0.5253	0.6048
Size	0.7665	0.5899	0.6667
Binary	0.7070	0.5115	0.5936
Agnostic	0.7333	0.6590	0.6942

Fig. 9. Class remapping table

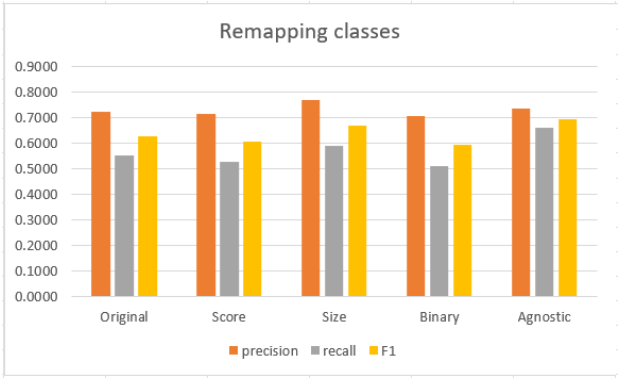

Fig. 10. Class remapping graph

4.3 Class Remapping

The last experiment that was performed was to remap the classes. To do this, the classes were grouped in four different ways. 1) Classes grouped by F1 score: the F1 score was calculated and stored for each class. Then they were classified as "best" ($F1 > 0.75$), "high" ($0.5 < F1 < 0.75$), "medium" ($0.25 < F1 < 0.5$), and "low" ($F1 < 0.25$). 2) Classes grouped by size of the anomaly: the classes were separated by the average size of the anomaly. The classes were classified as "large", "medium", "small", and "tiny". 3) Classes grouped as a binary classification: the classes were classified as either "clothing" or "anomaly". 4) Classes grouped all classes as one: all classes were grouped as one; creating a class-agnostic model. The results of the experiment are shown in Figure 9 and Figure 10.

With class remapping we aim to group similar classes to achieve two things: diminish class imbalance and encourage the model to learn desirable features. For example, the density of the material (as denser materials appear darker) or the size of the object. On the other hand, class agnostic grouping opens the opportunity to identify anomalies that are not present in the dataset. For example, weapons and cellphones are anomalies that are not represented in the current dataset but are present in the real world. A class agnostic model could be used to identify these and more.

The results of the experiment show that remapping the classes by score or binary has a small yet measurable negative impact on the model's performance. Whereas size and class agnostic remapping improves the model's significantly. The class agnostic remapping provides de largest benefit and has the additional benefit previously mentioned. Even though information is being lost with class agnostic remapping, it is considered the best remapping method because in this

model	precision	recall	F1
Base	0.60	0.27	0.37
Ideal	0.85	0.69	0.76

Fig. 11. Ideal parameters table

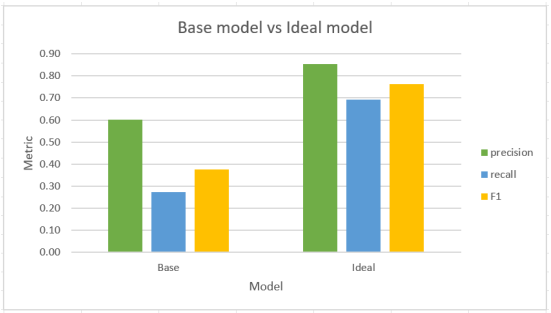

Fig. 12. Ideal parameters graph

application being able to detect an anomaly takes priority over identifying its class.

4.4 Ideal Parameters

Utilizing the results of the previous experiments, the ideal parameters for the model were identified. The ideal parameters are: random positive tiling with 4 tiles sized 512x512, batch size 8, combined data augmentations (Gaussian blur, horizontal flip and random brightness and contrast), and class agnostic remapping. A model was trained with these combined parameters and allowed to be trained up to 500 epochs; once again only saving the model if the validation loss is lower than the best loss. The results of the experiment are shown in Figure 11 and Figure 12.

The ideal parameters produce a model with an F1 score of 0.76. This is a significant improvement over the base model (without tilling, augmentations nor remapping) which had a F1 score of 0.37. However, it is still not enough to be considered reliable enough for the application. This is because the model is still not able to identify all objects consistently.

5 Discussion, Conclusion & Future Work

After analyzing the results of the experiments, the following conclusions can be made.

5.1 Discussion

Something that is evident throughout the experiments is that precision is consistently higher than recall. Upon further inspection of the metrics per class indicates that the model learns to identify and classify reliably one class but fails to consistently identify all other classes. This is likely due to the fact that the model is not being trained on enough positive samples and the huge class imbalance. This is a problem that was addressed with the class remapping but can be further improved upon by increasing the dataset, which will give more examples for the model to learn on and with weighted loss; this is discussed in the future work section.

However, it is still possible to learn from the results obtained during the experiments. It is still possible to identify the ideal

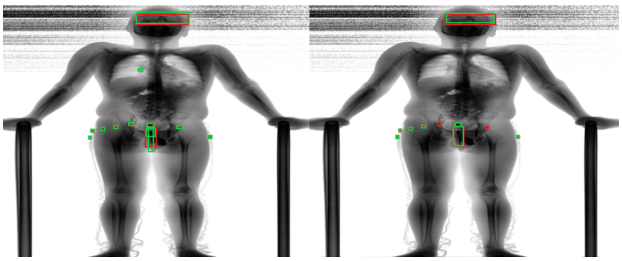

Fig. 13. Base model (left) performs similarly to the best model (right). Ground truth in red and predictions in green.

tiling, augmentation, and remapping parameters for the model.

5.2 Conclusion

Going back to the research questions, the following conclusions can be made:

- **SQ1: How does the model performance compare when using the full image or random positive tiles?** The model performs 67% better when using random positive tiles.
- **SQ2: What is the relationship between tile size/number of tiles and model performance?** The model tends to perform better when using more tiles. However, the returns diminish as the tile size decreases. The ideal tile size is 512x512 with 4 tiles.
- **SQ3: What data augmentations increase performance on the model effectively?** The model performs 13% better when using Gaussian blur, horizontal flip and random brightness and contrast in combination.
- **SQ4: How is the model's performance affected by the number of classes?** While the number of classes does not correlate with the model's performance, the way the classes are grouped does. The model performs 11% better when using class agnostic grouping.

In the end, the main research question (What is the performance of an object detection model for identifying anomalies in x-ray images.) was answered. Through the experiments performed we learned the parameters to train the best model possible within the scope of the study. This model produced by the ideal parameters has an F1 score of 0.76, which is a significant improvement over the base model which had an F1 score of 0.37 (103% increase in F1 score). To illustrate, in Figure 13 it can be observed that, in some cases, both models will performs similarly; while in Figure 14 it is clear that the best model outperforms the base model. Although, an ideal model would detect all contraband ($recall = 1$), which is significantly more than what the current model can achieve. Still, the results are promising and future studies might find further improvements.

5.3 Future Work

As previously stated, the model is not able to identify all classes consistently. Future research can focus on addressing the following issues:

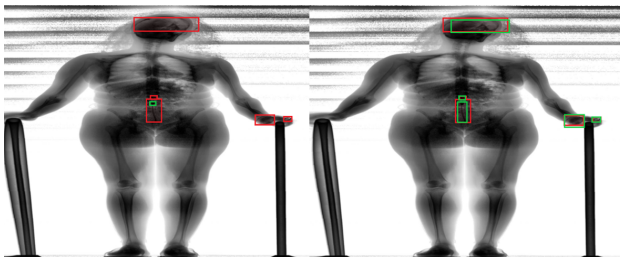

Fig. 14. The best model (right) performs significantly better than the base model (left). Ground truth in red and predictions in green.

- **Dataset size** The model is not being trained on enough positive samples and some classes are not sufficiently represented. It is important to increase the dataset; and, as mentioned before, this is already in progress.
- **Class imbalance** A few classes represent the majority of the dataset. This is a problem that can be solved in a number of ways. One way is to use weighted loss as it has been shown to significantly improve models trained in imbalanced datasets.
- **New classes** There are objects that are not currently represented in the dataset but are present in the real world; such as weapons or cellphones. It could be beneficial to test the ability of the model to identify objects belonging to classes that it has never seen before.
- **Further augmentations** The model could be further improved by using more augmentations. For example, the dataset could be expanded by generating anomalies.

ACKNOWLEDGEMENTS

- This project is made possible by OD Security.
- This project was mentored and supervised by Willem Dijkstra and Klaas Dijkstra; from the Computer Vision and Data Science minor at NHL Stenden.

REFERENCES

[1] National Institute of Justice. Contraband detection and control.

[2] National Institute of Justice. Contraband detection technology: A market survey, 2018.

[3] Amit Kumar Jaiswal, Prayag Tiwari, Sachin Kumar, Deepak Gupta, Ashish Khanna, and Joel J.P.C. Rodrigues. Identifying pneumonia in chest x-rays: A deep learning approach. *Measurement*, 145:511–518, 10 2019.

[4] Jianpeng Zhang, Yutong Xie, Guansong Pang, Zhibin Liao, Johan Verjans, Wenxing Li, Zongji Sun, Jian He, Yi Li, Chunhua Shen, and Yong Xia. Viral pneumonia screening on chest x-rays using confidence-aware anomaly detection. *IEEE Transactions on Medical Imaging*, 40:879–890, 03 2021.

[5] Paolo Napoletano, Flavio Piccoli, and Raimondo Schettini. Anomaly detection in nanofibrous materials by cnn-based self-similarity. *Sensors*, 18:209, 01 2018.

[6] Glenn Jocher. Yolov5 by ultralytics. 05 2020.

[7] Joseph Redmond and Ali Farhadi. Yolov3: An incremental improvement. 04 2018.

[8] Chethan Kumar B., R. Punitha, and Mohana. Yolov3 and yolov4: Multiple object detection for surveillance applications. *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*, 08 2020.

[9] Glenn Jocher. Yolov5 (6.0/6.1) brief summary · issue 6998 · ultralytics/yolov5, 03 2022.

[10] K. Ruwani M. Fernando and Chris P. Tsokos. Dynamically weighted balanced loss: Class imbalanced learning and confidence calibration of deep neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–12, 2021.

[11] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 06 2019.

[12] Ishan Shrivastava. Handling class imbalance by introducing sample weighting in the loss function, 12 2020.