Proceedings of Business and Economic Studies, 2021, Volume 4, Issue 3
http://ojs.bbwpublisher.com/index.php/PBES
ISSN Online: 2209-265X
ISSN Print: 2209-2641

A Study on Intelligent Guide Stick Using YOLOv3 Algorithm – Improving Spatial Awareness with Self-made Data Set

Stone-Yan*

Shanghai American School, Hongkong, China

***Corresponding author:** Stone-Yan, stone01px2022@saschina.org

Abstract: The increasing negligence of "blind lanes" on streets in metropolises such as Shanghai is an inconvenience to the blind population, therefore, alternatives should be explored. When walking on these lanes, there should not be any obstacles for the users, yet bikes can be seen parked there in addition to littered objects. This makes it potentially more dangerous to walk on them in comparison to walking on non-tactile paved lanes. An alternative to these lanes has been discovered which is a camera that is attached to a blind staff. This camera provides auditory feedbacks in regard to the user's surroundings. Using YOLOv3 (You Only Look Once, Version 3), the software is trained using 140 images to identify three different classes which are blind lanes, waist-high obstacles, and dog feces, as well as the right direction of these objects. If the camera captures any of these three categories, it will provide and voice feedback, hence, warning the user. With this system, the blind can essentially have functional vision that would better guarantee their safety when walking on streets.

Keywords: Blind lane; Object identification; YOLOv3

Publication date: June 2021; ***Online publication:*** June 30, 2021

1. Introduction

With only 17.5 million blind people out of 1.393 billion people, the society in China has developed without considering their needs.[1] One of the difficulties in being blind is movement, especially when moving around the city. As countermeasures to these hurdles, the local government has introduced "blind lanes" for the blind population in order for them to be aware of whether they are walking on a safe path. While feeling the texture on these bumpy lanes, walking outside becomes much more convenient as it helps the blind to identify directions. To further support the blind population, methods should be proposed to assist the identification of these "blind lanes" from a distance. Currently, the blind population are unable to identify these lanes without stepping on them first. These paving are also constantly being interrupted due to slopes and intersections; hence, it becomes difficult for the blind to accurately identify the start of the next segment. This is extremely dangerous and inconvenient as blind people would either require the service of another person for guidance or they would potentially face fatal threats such as poles or cars while walking along the streets. The modifications of city infrastructures would be able to provide suitable substitutes for these lanes, however, different technologies should also be developed and implemented to maximize users' satisfaction.

Currently, China's government have employed solutions such as these "blind lanes" and also, the use of guide dogs. These lanes function to raise awareness via the sense of touch. While walking on these tactile paving, the blind should be able to determine the direction they are moving in. These "blind lanes" are constructed by placing bumps onto the surface of the road. Hence, the blind would be able to traverse along

the streets without worrying about the direction or obstacles. However, when there is an intersection, these lanes would break off, making it difficult for a blind person to be aware of where the other side of the street is. Additionally, due to the increasing negligence of the citizens, bikes can be seen parked on these lanes. This defeats the purpose of having the lanes as it becomes potentially more dangerous for a blind person to walk on compared to walking on non-tactile paving.

On the other hand, guide dogs offer a different kind of solution. After being trained, guide dogs would be able to effectively lead the blind to walk along the streets while avoiding obstacles such as cars or bikes. Guide dogs are also capable of notifying their owners of steps and curbs by stopping ahead of these obstacles. Guide dogs are usually exempted from the normal restrictions of dogs in public areas. Hence, allowing them to be important companions for the blind to always have access to. However, to train a guide dog, it does not only take two whole years, but it also costs $50,000, in which their daily expenses are not yet covered. In consideration that it would be a costly affair, it may be an intangible option for several blind people. In addition to that, the blind can be overly reliant on their guide dogs as they slowly lose awareness of their surroundings. Even if guide dogs make mistakes, the owners would not be able to recognize them.

In order to solve these issues, software such as YOLOv3 should be utilized.[3] YOLOv3 was created by Joseph Redmon to increase the efficiency of existing object detection software. YOLOv3 was acknowledged as a significant milestone in its field as it revolutionizes the process in which the desired object can be separated from the rest of its surroundings.

By using this software, cameras would essentially become eyes for the blind. YOLOv3 is a software with the ability to identify any object in a video or image, given the time for training. Via the use of this software along with the codes for voice feedback, the blind would be able to hear the direction of where these “blind lanes” are located. By using center point coordinates, the location of a blind lane in relation to the position of the camera can be computed. For example, assuming the camera is held directly in front of the user, the user would be able to determine whether the lane is on the right, left, or directly ahead.

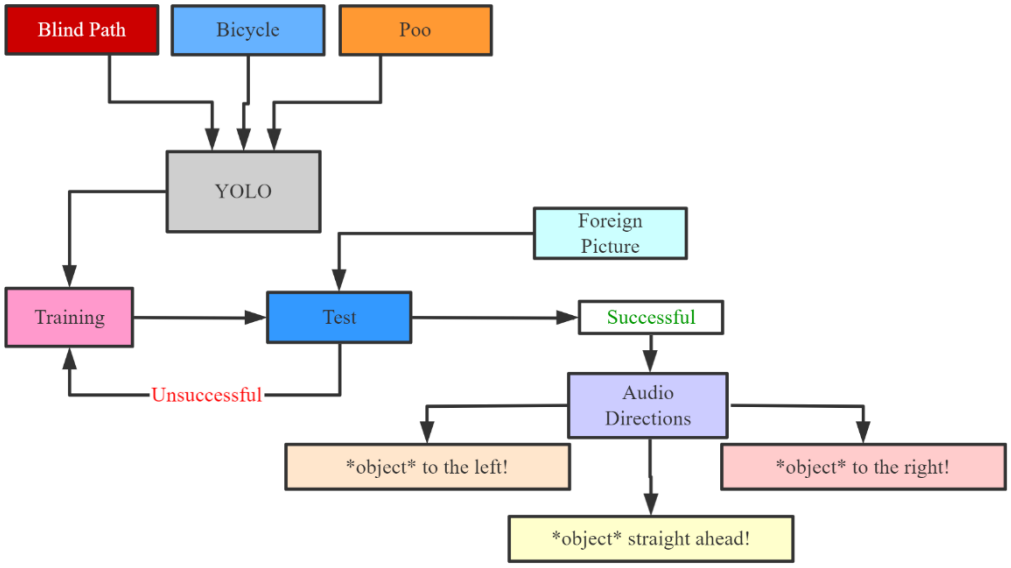


Figure 1. Intelligent blind cane system

Although this provides a solution, without proper maintenance and care of these lanes, there is a possibility that they would erode and potentially be more dangerous or inconvenient for blind people to walk on. Due to the negligence of the blind, people have been stacking items on these lanes for their own convenience and occasionally, trash can be found lying here and there. In addition to that, bikes are also

parked along these lanes. At times, the government may carry out construction works on telephone poles, fire hydrants, and manholes. Hence, there are occasions when these manholes would be without any lids. Under normal circumstances, these obstacles can be easily avoidable but without vision, avoiding them may become exponentially difficult. With the use of YOLOv3, the task to avoid these kind of obstacles become much easier for those visually impaired.

2. Research Plan

Week 1	Planning out the project and related research.
Week 2 - 7	Learning to code, understanding the basics of python in order to be able to interpret codes, examining the codes from other sources, and determining those which fits the project best.
Week 8	Learning about object detection and the method of most software in performing the task.
Week 9 - 10	Gathering dataset, adjusting the codes to fit desired dataset, and training the software.
Week 11 - 12	Learning about voice commands and method to integrate them into the existing codes.
Week 13 - 14	Performing experiments to ensure that the software functions well.
Week 15	Writing the project essay and revise accordingly.

Figure 2. Research plan

3. Methodology

YOLOv3-tiny uses labelled images to train itself to draw bounding boxes around identified objects.

First, pictures would be separated into grids, the identified object is then classified, and the bounding box is drawn. The size of the bounding box is estimated by using the center point as well as the length and width of a grid. The center point is determined through the following function:

Grid width + horizontal distance from grid = X value of the center point

The function is then repeated to find the y-value:

Grid height + vertical distance from grid = Y value of center point

To determine the height and width of the bounding box, the YOLOv3 software estimates the probability of the given object existing within the bounding box, then it provides the height and width to the closest degree. Subsequently, those values are multiplied by the error margins to maximize the precision. After these processes, the given object should be perfectly encapsulated by the bounding box.

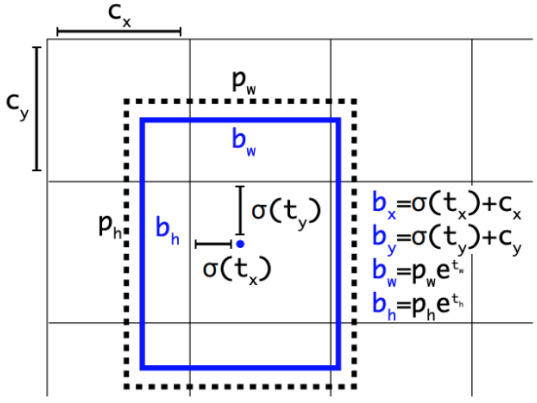


Figure 3. Bounding box of YOLO[2]

To determine the numbers of bounding boxes and the probabilities of the classes existing in the bounding boxes are expressed in this function: $N \times N \times [3 * (4 + 1 + 1)]$. The two Ns represent the numbers of grids created in the picture, while 4 represents the four coordinate points of the bounding box, the first 1 represents the probability of identifying the class, and the final 1 represents the number of classes that exists.

In this context, [3*(4+1+1)] determines the output and results of the identification and the numbers of bounding boxes that could be drawn per grid. Combining them, the output is organized into a column where the probability of each class and the four coordinate points would be identified. By only having a single class, this column would only have (4 + 1 + 1) = 6 rows.

There was a total of 6 clusters in the Blind Lane dataset: (10,14), (23,27), (37,58), (81,82), (135,169), (344,319). These are the possible numbers of grids (N × N) that could be attained by division in the picture.

In order to correctly identify the object, 53 convolutional layers were used to extract the required features needed to identify the object.[3] Each layer would identify a specialized trait of the recognized object; hence, resulting in a higher probability of the final outcome being correct. By closely analyzing the patterns of the RGB (red, green, and blue) values in the pictures, the 53 convolutional layers can discover numerical patterns that isolate the object. These patterns enable the software to identify similar patterns that exist in any other given pictures. Pictures which display the same color pattern would be recognized as the object. Darknet-53 is shown as below.[3]

	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. Intelligent blind cane system diagram

To ensure accuracy, 140 pictures were labelled and processed to train the model. It was trained three times in which the first time, the epoch was at 100, the second, at 300, and the third was at 500. The goal of the training was to drive objectness and GIoU (generalized intersection over union) down while increasing the precision and recall values. The graphs generated by the program suggested that the higher the epochs, the better the precision and recall values were while the objectness and GIoU were driven down. This indicated that the training model functions as intended.

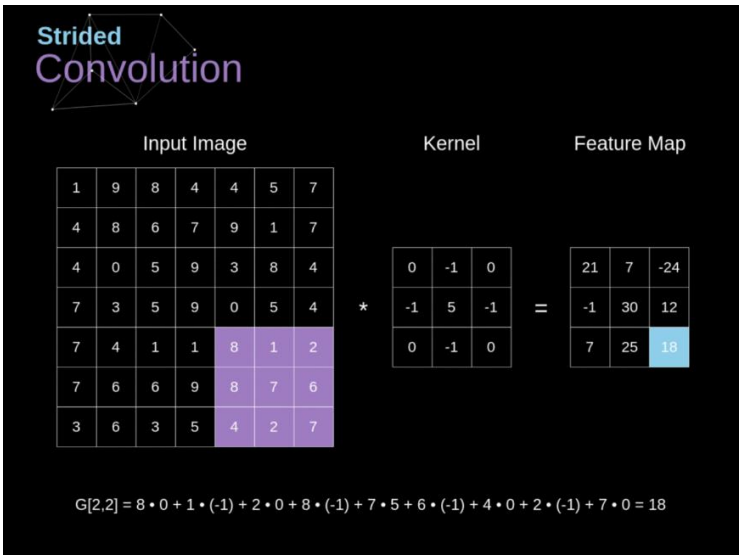
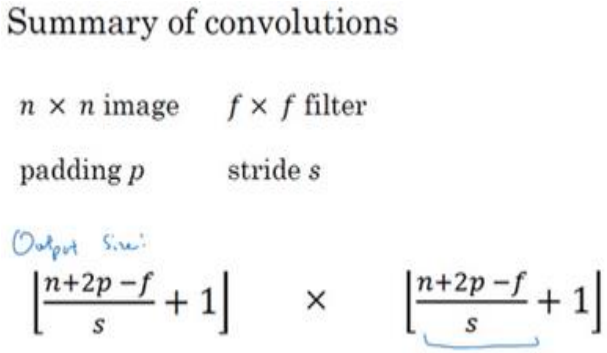


Figure 5. Convolutional layer calculation in a schematic diagram[4]

This image explains the process of downsampling using convolutional layers. The process of computing the output of a processed image is to multiply the values in the purple box to each of their corresponding values in the kernel and then adding all the numbers together to achieve the number in the blue box. The example given is shown above as in the equation of G[2.2] which demonstrated the concept of feature extraction performed by the computer. The kernel was artificially inputted. In this process, developers would first acquire a randomized kernel layer, then, they would continuously tweak until the error between the input and the output is minimized. As mentioned above, the values in the input images are the RGB values and each layer of the input image is either red, green, or blue. Through the identification of the RGB values and their trends, the trends of the object would then be recognized in unfamiliar images.

With closer examination, it is apparent that there are more values in the input than the output, despite the supposedly 9:1 decrease in size. The drastic decrease in size of the box from 7x7 to 3x3 is due to the set stride which skipped values when calculating the output. Although the accuracy would be lower due to the compressed nature of the output, the time spent training the model would be significantly shorter in view of fewer operations needed to compute the output.



Source : Professor Andrew Ng, Stanford University

Figure 6. Convolutional calculation of the size change

With the function listed above, the size of the output box can be calculated. Padding is circuits of zeroes which enable the values at the edge of the box to be computed multiple times. Normally, these values are only computed once which is detrimental to the model because other values would then be computed

multiple times. The computer would ignore the trends shown at the edges of the image because of single consideration. Padding would solve this issue as values at the side would be considered along with the zeroes added to the edges of the box. Padding was added to the edges of the input and integrated into the output.

4. Experiment and Result Analysis

4.1. Bounding box

In order to evaluate the variables that provide optimal conditions for users, each variable was tested to investigate the minimum number of epochs to optimize proficiency. By looking at GIoU and objectness in the training results, the optimal number of epochs can be determined. There is a threshold at which the GIoU and objectness would cease to increase. Any more epochs beyond this threshold would be considered as a waste of time, however, if too there are too few epochs, the model would not provide any benefit in view of its accuracy.

GIoU is the difference in area between the artificially labelled bounding box and the computer-generated predicted box. By minimizing the difference, the prediction will be improved. The value of the GIoU is usually provided by the software, however, the value can also be derived based on the equation below:

$$IoU = \frac{|A \cap B|}{|A \cup B|} = \frac{|I|}{|U|}$$

Source: Hamid Rezatofighi

In this equation, A and B represents the predicted and labelled bounding boxes respectively.[6]

140 images were used to train this model. For each image, the different classes that could be discovered were hand-labelled. This included the three classes which were the blind lanes, dog feces, and waist-high obstacles. The software *LabelImg* was used to complete the task. As seen below, a box was drawn around the desired object. Due to the limitation of shapes, not all blind lanes, waist-high obstacles, or dog feces were fully encapsulated or covered. However, this did not severely impact the effectiveness of the program as the program still separated color patterns.

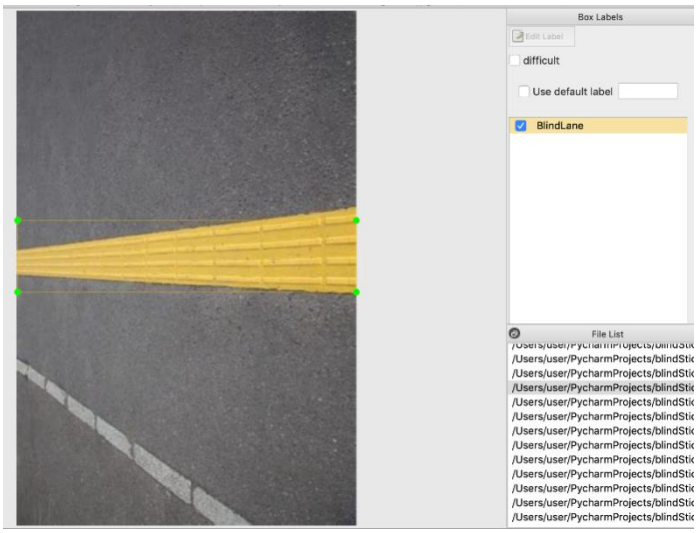


Figure 7. *LabelImg* schematic diagram

4.2. Training process

4.2.1. 100 epochs

The results of running 100 epochs showed that the program was able to identify the images with precision at approximately 0.15. This denotes a 15% chance that the program would identify the object in a given image. This is extremely low and is probably not the maximum precision this program can achieve. On the other hand, via the GIoU, objectness, classification, and precision, all of them showed trends that support the idea that by running of more epochs, the computer would be better at identifying the desired classes.

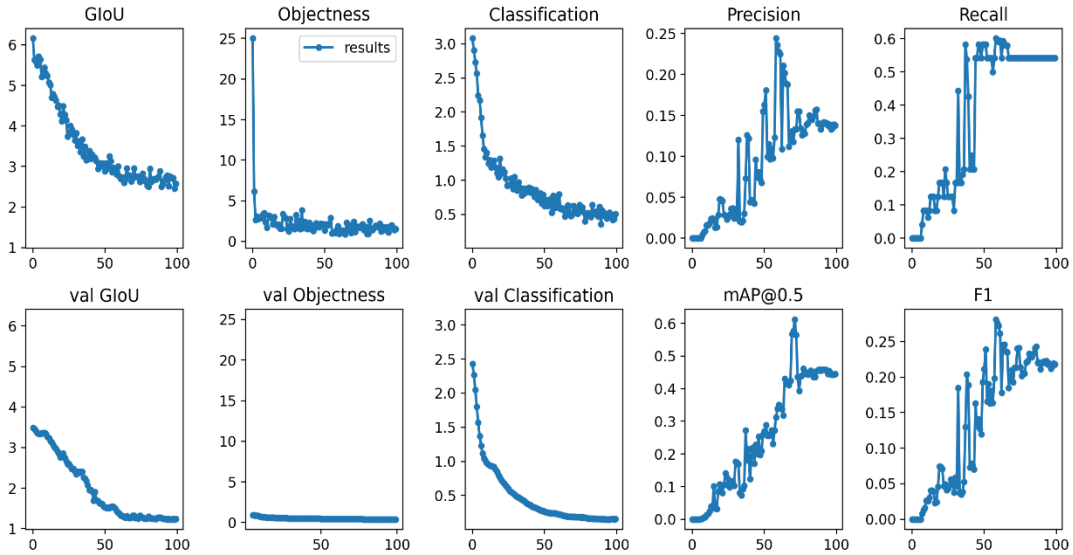


Figure 8. Results of 100 epochs within 17 hours

Figure 9. Detection results of 100 epochs

4.2.2. 300 epochs

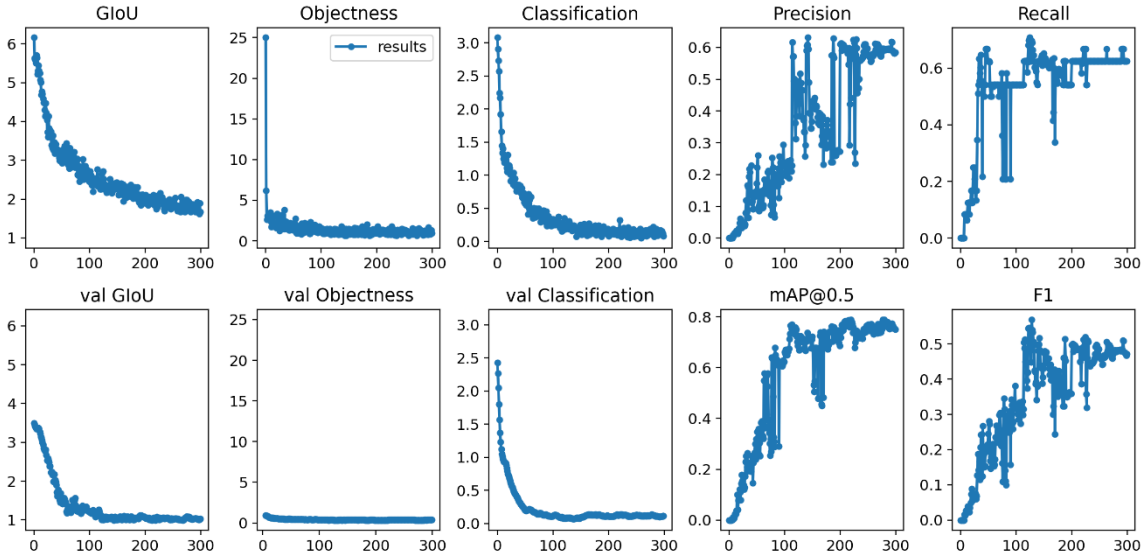


Figure 10. Results of 300 epochs within 17 hours

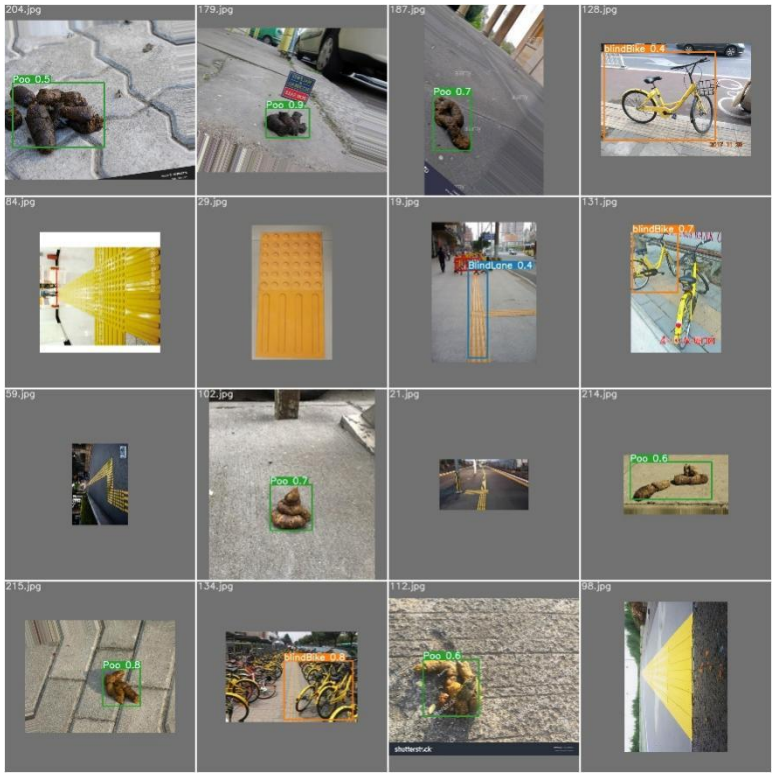


Figure 11. Detection results of 300 epochs

By comparing the results of 100 and 300 epochs, the major difference was the precision of the final epoch ran. When running only 100 epochs, the precision value was 15% while the precision value for 300 epochs was at 60%. This crucial distinction allowed the understanding that machines do share some characteristics with humans in which the more exposures received, the better it was at identifying the given object. Therefore, giving the impression that if the software ran more epochs, the resulting program would be more effective. The existing trends in the 300 epochs program were the same as those in the 100 epochs. GIoU, objectness, and precision all improved as the computer was better at recognizing the objects.

4.2.3. 500 epochs

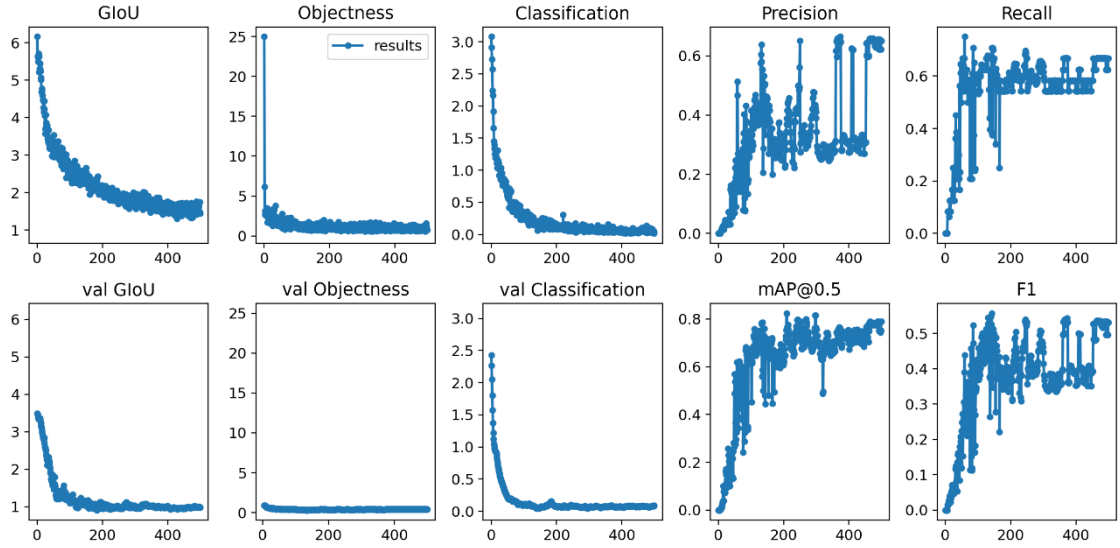


Figure 12. Results of 500 epochs within 40 hours

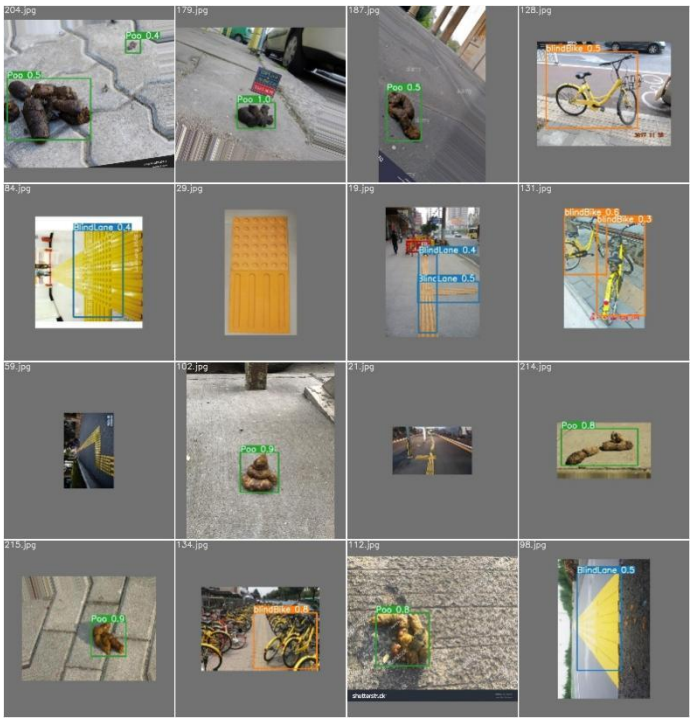


Figure 13. Detection results of 500 epochs

Although running 500 epochs offered approximately the same precision percentage as 300 epochs, it is imperative to compare other values in the graphs. Classification and GIoU were both lower than they were in other runs, therefore suggesting that although the 500 epochs resulted in the same percentage of accuracy, the distinction of the classes was much clearer.

By analyzing the three training sessions, a clear trend could be seen. The more epochs that were ran, the better the performance, and the lower the GIOU and objectness. The epoch threshold remained undetermined, however, even without a threshold, it is certain that the software can successfully perform its job.

The program worked as intended although the recognition probability was low. This meant that the program would only be 50% sure that the lane is truly a "blind lane." This was discovered in the input of a completely foreign image consisting of a blind lane. As shown in the figure below, the largest box comprising most of the surface area of the blind lane had the best percentage, which was 50%. Due to the low percentage, this seemed to be an issue, but it is still valuable to note that the software was able to detect the blind lane. The percentage can be improved by adding reference images during training.

Figure 14. Detection results of test program

Once the object is identified, the program would vocally inform the user of its presence. By using the coordinate points of the bounding box, additional information such as whether the object is on the user's left, right, or front can also be determined. In the image above, the user would hear "there is a blind lane on your right" due to the location of the bounding box. This way, navigation would be more convenient for the blind as they would be able to ascertain the direction they are headed to.

By comparing the results of YOLOv3 to other programs such as PASCAL VOC 2012, the YOLOv3 is still lacking. While PASCAL VOC 2012 has achieved an accuracy of up to 75%[5], this model of YOLOv3 could only achieve precision up to 60%. This implies that the PASCAL VOC 2012 would have better accuracy as a substitute program.

It is possible to increase the precision with the sample size. Although increasing the epochs would not improve precision, the program would gain more insights of an object's appearance by increasing the sample size. High precision is an extremely important component in view of the target audience. Since the blind are unable to use their sense of sight, they cannot verify whether the program has mistakes or not.

5. Experimental Optimization Research

To assess the functionality of this model, pictures of a bike were taken from various angles and the said object was identified using the model. There were no high expectations in regard to the accuracy of the results as the database used were insufficient to accurately determine the existence of the object in the

image. Nevertheless, this model should be able to predict the object somewhere in the image even if the object was labelled with an extremely low probability.

Three photos were taken to ensure that the program could identify the bike from all angles. The photos were taken at approximately 4pm when the sun was brightly shining; hence, the bike was clearly visible in all the photos. The yellow color bike clearly contrasted the cluttering colors in the picture to allow an easier identification process.

The program successfully identified the object and accurately created a bounding box tightly encapsulating the bike. However, it is worthwhile to note that although the bike was labelled accurately, the probability of the program's estimation was only 22%. This meant that the program was unsure of whether the object was actually a bike or not.

As long as the program is able to identify the object, it can be said that the program has successfully served its function. Therefore, even if the probability is low but the objects are accurately labelled, the program is presumed to be a success. This does not mean that the development of the program should stop at this point. The accuracy of the program is positively correlated with the sample size, whereby with an increase in sample size, the program will be able to improve from that measly 22% to a much higher percentage.

6. Conclusion

With the training of YOLOv3, it can be concluded that with a sample size of 140 images and three different classes, the success rate is at a threshold of approximately 60%. However, it is prevented from increasing because the model has been already optimized. This is significant because in training the model, it takes a long time especially when epochs are overestimated. With a properly trained model, the software will be able to identify and isolate the three classes. With these variables being identified correctly, the convenience of walking along the streets will be drastically improved since potential dangers would be vocally reported to the user. In the trained model, the three objects listed out were blind lanes, dog feces, and waist-high obstacles. In order to identify objects, the program needs to be trained with sample images of artificially labelled bounding boxes. The different classes can then be listed with their corresponding bounding boxes. By identifying the difference in RGB values between the objects in the labelled boxes and objects on the outside, the program can look for patterns that would contribute in the identifying of the objects in foreign images. By labelling the three objects in a set of 140 images, the program would be able to identify all three classes in foreign images. Currently, although only three classes could be identified, other classes should also be able to be identified by increasing the training size and labelling the fourth, fifth, or fiftieth class for the program. As long as the sample size is sufficient, the program would be able to identify the object.

This potential of unlimited classes suggests much greater function for the YOLOv3 framework. In the context of assisting the blind, YOLOv3 can become their eyes. The blind will be able to "see" through the audio cues given by the program. The program will be able to provide directions as well as naming the objects to users. Even with only three classes, this program already offers immense benefits to the lives of the blind. They would be able to identify dangers and inconveniences without having an individual next to them. In the future, there should be a program with all classes labelled and trained to ensure the safety of the blind on the streets.

Disclosure statement

The author declares no conflict of interest.

References

[1] 2018. China country report to world blind union Asia Pacific, general assembly, Ulaanbaatar, Mongolia. World Blind Union Asia Pacific. From: https://wbuap.org/archives/1416

[2] Brownlee J, 2019, A gentle introduction to object recognition with deep learning. Machine Learning Mastery. From: https://machinelearningmastery.com/object-recognition-with-deep-learning/

[3] Rezatofighi H, et al., 2019, Generalized intersection over union: a metric and a loss for bounding box regression. Computer Vision Foundation. From: https://openaccess.thecvf.com/content_CVPR_2019/papers/Rezatofighi_Generalized_Intersection_Over_Union_A_Metric_and_a_Loss_for_CVPR_2019_paper.pdf

[4] Redmon J, Farhadi A, 2020, YOLOv3: an incremental improvement. University of Washington. From: https://pjreddie.com/media/files/papers/YOLOv3.pdf

[5] Skalski P, 2019, Gentle dive into math behind convolutional neural networks. Towards Data Science. From: https://towardsdatascience.com/gentle-dive-into-math-behind-convolutional-neural-networks-79a07dd44cf9

[6] Zhang Y, et al., 2015, Improving object detection with deep convolutional networks via Bayesian optimization and structured prediction. Computer Vision Foundation. From: https://openaccess.thecvf.com/content_cvpr_2015/papers/Zhang_Improving_Object_Detection_2015_CVPR_paper.pdf