GLAUCOMA DETECTION USING VISION TRANSFORMERS ON FUNDUS IMAGES
A TAKE ON THE AIROGS LITE CHALLENGE

Robert Ioiart, Valentin Buchner, Rutger van der Linden, Fauve Wevers

Vrije Universiteit Amsterdam, The Netherlands

ABSTRACT

Glaucoma is significant cause of blindness worldwide even though it can be effectively treated if detected early. As such, glaucoma screening is essential. However conventional screening, done by a qualified physician, is not cost-effective or practical at a large scale. Therefore, machine learning can crucially support this process. We present a machine learning model which takes color fundus retinal photographs as input, detects the optic disc and then performs classification on the optic disk centered and cropped images. The YOLOv5 frameworks was used for optic disk localisation. For classification we experimented with different cropping strategies, image augmentation strategies, and with different classifier architectures such as Vision Transformers and ResNets with different pre-trained weights. In our experiment we found that besides centering on the optic disc the biggest impact on performance is the choice of classifier architecture and pre-trained weights and, surprisingly, data augmentation makes little difference. Our best model achieves a partial AUROC score of 0.9448 and a has a sensitivity of 93.7% at a specificity of 95%.

1. INTRODUCTION

Worldwide, glaucoma is the second leading cause of blindness [1]. Glaucoma is a group of disorders which all share a few common features. These features are: progressive degeneration of the optic nerve, death of retinal ganglion cells, thinning of the retinal nerve fiber layer, and increasing excavation of the optic disc [2].

It leads to irreversible visual impairment and can cause blindness if not treated in a timely manner [3]. Therefore, it is of utmost importance to detect it as soon as possible and avoid visual impairment. Currently, the most common way to detect glaucoma is funduscopic examination of the optic disc and the retinal nerve fiber layer. In addition, the intraocular pressure is measured, for a pressure too high can indicate glaucoma [2]. Still glaucoma is often detected too late where the patient already experiences visual impairment [3]. Therefore screening is needed for early detection. Unfortunately, studies have shown that population-based screening is not cost-effective [4]. The application of Artificial Intelligence (AI) techniques could be a potential approach to increasing the cost-effectiveness of screening, because it would eliminate the need for highly skilled labour on-site. Therefore, this paper aims to create a high-performance machine learning model to detect glaucoma in real-world ocular fundus images, while giving insights in the experiments we did when developing the model.

2. METHODS

2.1. Classification Pipeline

The top three competitors in the AIROGS competition have made use of optic disc (OD) detection as a preprocessing step [5]. Claro et al. have also showed that OD detection can lead to a higher accuracy when detecting glaucoma [6]. Therefore, we have also used an OD detector. In addition, we also tried to detect the fovea because Kim et al. argued that the fovea could be useful in detecting glaucoma [7]. The authors, who are not experts in the medical domain, have annotated 1000 images where they have placed two points at the most distant ends of the optic disc, and one point at the fovea. Similar to entries in the AIROGS [5] competition, we have used the YOLOv5 framework for OD and fovea localisation. Eventually, our classification pipeline consisted of 4 steps:

- 1. OD detection using YOLOv5
- 2. OD cropping based on detector predictions
- 3. Glaucoma classification using cropped OD
- 4. Ensemble prediction of classifiers trained with early stopping on different train/validation splits.

2.2. YOLOv5

2.2.1. YOLOv5 metrics

YOLOv5 outputs the metrics precision, recall, mAP@0.5 and mAP@0.5:0.95. Metric Mean Average Precision (mAP) is based on the precision, recall and Intersection over Union (IoU) [8].

2.2.2. Preprocessing images and data augmentation

The standard image size for YOLOv5 is 640 so we first we make the original images square by cropping them to fully contain the circular field of view of the CFP images and fill any remaining empty space. Then the square images are resized to the 640 resolution. The YOLOv5 training procedure applies augmentation each time a training image is loaded [9] therefore we did not need to implement data augmentation ourselves.

2.2.3. Baseline model and early stopping

There are five pretrained YOLOv5 models. We have used these pretrained models because we have a small training set. It is advised to use the two largest models for cloud deployments. We have used the second largest model (YOLOv5l) as a baseline model, we didn't use the largest model to decrease our initial runtime [10, 11]. We didn't change any hyperparameters and we've used the standard number of epochs (300), and auto batch size. A training/test split of 80/20 was applied. We've further split the training data into training and validation data via the same ratio. The best model is determined by weighing the four metrics on the validation set. Based on the baseline model we've also determined the patience for early stopping to decrease runtimes [12].

2.2.4. Model size and tuning hyperparameters

We have investigated the influence of the YOLOv5 model size on the performance. If a smaller model leads to similar results as a larger model, we could decrease the runtime. YOLOv5 also supports hyperparameter optimization via a genetic algorithm. The hyperparameter optimizations doesn't support tuning of the batch size, so we have done this separately [13, 14].

2.2.5. Influence number of images on training results

We were also interested in the influence of the number of training images on the performance. We've let the trained model predict on 1000 images and we manually corrected errors. Aimyshev et al. also followed this approach in the original AIROGS competition [15]. We've used the same validation and test images as we've used for the baseline model to make a fair comparison.

2.3. Glaucoma classification

To approach the binary classification problem of whether a fundus image contains glaucoma or not, we experimented with several preprocessing and augmentation techniques and classification models. For each parameter we experimented with, we ran the model at least 3 times and calculated the metrics on the ensembled(averaged) predictions.

2.3.1. Optic Disk Cropping

We investigated 3 options to crop the raw, rectangular image: (1) Not using YOLOv5 predictions. The image is cropped by separating the retina from its black background using a threshold of 10. This was done by representing the image as a numpy array and using the first column with a maximum pixel value above 10 as the left end of the retina, and the last column with a pixel value above 10 as the right end of the retina. If the retina was cut off at the top or bottom, black frames were added to square the image. (2) Using YOLOv5 predictions for the OD. Yolo predicted a rectangular box around the OD. The side lengths of this rectangular box were averaged to obtain a square around the OD. For cropping, the side length of the OD was multiplied by a cropping factor, for which we tested multiple values in an experiment. (3) Using YOLOv5 predictions for both OD and fovea. The motivations for using the fovea location stems from the fact that cropping based on OD diameter leads to the all images presenting the same OD area, so information on whether the OD is enlarged or shrunk can not be used by the classifier. In order to recover this information the crop size can be made related to the distance between the OD and fovea centres. More specifically, the side length of the square around the OD was now determined by $factor * c * euclidean_distance(odc, fc)$, where c is a constant such that for a typical eye it is true that $c * euclidean_distance(odc, fc) = OD_diameter$. This was done with the reasoning that one's OD diameter might relate to a glaucoma diagnosis, and that maintaining the true OD diameter may help the model during classification.

2.3.2. Histogram Equalisation

Histogram equalization enhances image contrast and was therefore expected to be of benefit to the classification model [16]. Again, we investigated 3 methods: (1) No histogram equalization. (2) Histogram equalization on the cropped image including its black border. (3) Histogram equalization on the cropped image, by ignoring the black border. As in some cases, the OD was located on the edge of the retinal image, the cropped square around it would then include the black background. This background was cropped out by first generating a mask based on a pixel value threshold of 10, and then applying morphological closing. The appendix (fig 2) contains an example of histogram equalisation.

2.3.3. Data Augmentation

Due to timing considerations, only a few augmentation methods were applied. We rotated the images up to 40 degrees, translated the images up to 40% of their width, scaled the images up to 40% of the OD diameter and applied horizonatal flipping. Translation and scaling were done with the intention to account for YOLOv5 not perfectly predicting the OD center and bounding box.

2.3.4. Classifier backbone

Several pre-trained classification models were used for classification. For brevity we will only list a short identifier for each as a hyperlink to the model and the main article as a reference: vit-base-patch32 [17], swin-base-patch4-window12 [18], swin-large-patch4-window12 [18], ViT_B_16 [17] using 3 different pre-trained weights (ImageNet-1K, Swag, SwagLinear), ViT_B_32 [17] with ImageNet-1K pre-trained weights, tv-224-swin_b [18] with ImageNet-1K weights, resnext50_32x4d [19] with Imagenet-1K-v2 weights. Image normalization was done as in the pre-training process of the respective classification model and the image was then resized to either 384x384 of 224x224 pixels.

2.3.5. Focal Loss

As only 10% of the dataset consists of referrable glaucoma, the dataset has a significant class imbalance. We addressed this by applying the focal loss function [20] with different values for parameter alpha.

2.3.6. label smoothing

To account for the possibility of labelling noise, label smoothing was applied with epsilon values of [0.05, 0.1] [21].

3. RESULTS

3.1. YOLOv5

3.1.1. Baseline model and early stopping

The resulting batch size from auto batch size was 16. In Table 1 are the results for the baseline model. We can observe that the model is better in detecting the OD than the fovea.

Table 1: Performance YOLOv5l baseline model

Class	*Precision*	*Recall*	*mAP@0.5*	*mAP@0.5:0.95*
OD + fovea	0.968	0.971	0.983	0.778
OD	0.995	1	0.995	0.873
Fovea	0.940	0.942	0.970	0.683

In Figure 1 of the appendix are the results shown over the epochs. The obj_loss (confidence that object is present [22]) of the validation data increases after 50 epochs. Therefore, a patience of 20 should be sufficient for early stopping.

3.1.2. Model size

As we can see in Table 2 the model size doesn't influence the performance. The larger models required less epochs, but these epochs costed more time to run. We further tuned the small model because it had a good performance and a low run-time. We were reluctant in using the nano model because it may not be very sensitive to tuning.

Table 2: Performance different model sizes

Model size	*Precision*	*Recall*	*mAP@0.5*	*mAP@0.5:0.95*	*Epochs*
nano	0.975	0.964	0.983	0.789	87
small	0.981	0.965	0.975	0.787	74
medium	0.973	0.97	0.978	0.782	74
large	0.968	0.971	0.983	0.778	60
Xlarge	0.980	0.970	0.981	0.739	27

3.1.3. Tuning the batch size and other hyper-parameters

We have trained the YOLOv5s model with a batch size of 8, 16, 24 and 32. As we can see in Table 1 in the appendix, the differences in results between the batch sizes are very small. However, it is advised to use the largest batch size as possible in YOLOv5 [11]. Therefore, we have tuned the model with a batch size of 32.

We've also tried out hyper-parameter evolution, but we soon came to the conclusion that it wasn't feasible. It is advised to use at least 300 generations [14], but this would take days to run. For completeness we ran 15 generations, but the tuned hyper-parameters didn't lead to better results.

3.1.4. Influence number of images and the final model

We've trained a YOLOv5s on the earlier used training images, and the generated predictions (1000 images). The results can be seen in Table 3. If we compare this table with the results of batch size 32 in Table 1 we can see that more training images lead to higher metric values.

Table 3: Performance YOLOv5s 1640 training images

Class	*Precision*	*Recall*	*mAP@0.5*	*mAP@0.5:0.95*
OD + fovea	0.981	0.975	0.989	0.805
OD	0.997	1	0.995	0.871
Fovea	0.64	0.949	0.983	0.739

The final model is the YOLOv5s model with a batch size of 32. The training set is created by concatenating the training set of size 1640, and the test set of size 200. The validation set was needed for model selection via early stopping.

3.2. Classification Model

3.2.1. Histogram Equalization

As can be seen in Table 5, Histogram equalization did not improve the model performance. This might be the case because on some images, histogram equalization introduced artefacts that obscured the data.

3.2.2. Data Augmentation

Adding rotations of up to 10 degrees and translation of up to 20% of the image side improved the partial AUC slightly, but

Table 4: Equalization Performance

	No	Yes	Ignoring Background
AUROC 90-100%	**0.941**	0.907	0.914
Sens@95%Sp	**0.918**	0.864	0.887

not the Sens@95%Sp, however the differences seem negligible. However, scaling seems to improve the model.

Table 5: Data Augmentation Performance

	Rotation degree			Translation		
	10	20	40	0.1	0.2	0.4
pAUC	**0.944**	0.941	0.937	0.941	**0.943**	0.938
Sn95Sp	0.915	0.921	0.912	0.915	0.912	0.812

Table 5: Data Augmentation Performance continued

	Scaling			
	0.0	0.1	0.2	0.3
pAUC	0.939	**0.946**	0.944	0.939
Sn95Sp	0.921	**0.931**	0.924	0.905

3.2.3. Optic Disc Cropping

Cropping the optic disc had a relatively large impact on the model performance, and cropping based on the OD size outperformed the other methods. Applying this method, the cropping factor did not have a large impact on performance, but OD cropping factor 1.5 seemed the best.

Table 6: Cropping Performance

	Fovea Dist	OD Width			
Factor	c*1.2	1	1.2	1.5	2
AUROC 90-100%	0.939	0.941	**0.941**	0.941	0.933
Sens@95%Sp	0.921	0.918	0.905	**0.928**	0.905

3.2.4. Label Smoothing

Label smoothing did not improve model performance as measured by the metrics of interest.

Table 7: Label Smoothing Performance

	No	Label Smoothing	
		0.1	0.5
AUROC 90-100%	**0.941**	0.938	0.932
Sens@95%Sp	**0.927**	0.915	0.905

3.2.5. Focal Loss

Despite the imbalanced dataset, using the focal loss function did not improve performance according to the two metrics considered. The F1-score did actually improve slightly as well as sensitivity, but this is not our objective. Essentially the focal loss seems to just have shifted the RO-curve.

Table 8: Focal Loss Performance

	No	Focal loss (gamma=2)	
		alpha=0.5	alpha=0.66
AUROC 90-100%	**0.941**	0.940	0.939
Sens@95%Sp	**0.927**	0.908	0.918

4. DISCUSSION

Training the small YOLOv5 model with a batch size of 32, and 1640 training images led in our case to the best OD and fovea detection results. Other researchers have also investigated the detection of the OD in fundus images, for example Park et al. [23] who have compared different architectures. They got a mAP@0.5 score of 1 with architectures YOLOv3 and DenseNet, this is higher than our best mAP@0.5 score of 0.995. It could be that YOLOv5 is less suitable for detecting the OD than YOLOv3 and DenseNet. However, Park et al. also used a different image resolution, more training images and exclusion criteria for the training images. In addition, we could also annotate more images to get a larger training set. The approach where we have created a larger training set by letting a trained model predict on unseen images may also be sub-optimal. Of course we corrected wrong predictions, but this approach may lead to over-fitting.

When considering the classifier, several interesting observations can be made. The choice of architecture made a significant difference in performance, even when choosing between models with similar number of parameters: swin transformers outperform vanilla transformers. We do not have an experiment to establish if transformers would outperform a ResNet with the same number of parameters though. Some noteworthy observation can be made about the augmentations tried out: surprisingly for the image classification domain rotations and translations did not improve performance. We speculate that this is because rotation does not naturally occur in the dataset since patients look straight ahead when CFP are taken. It is also possible that specifically oriented features around the OD are informative for glaucoma and this information is lost through rotation. Translation also hurts the performance, possibly because it unnecessarily makes the classification task more difficult for transformers as they are not translation equivariant like some CNNs. Interestingly scaling does improve performance slightly but only with a low value, likely because the OD apparent size in the crop data does vary due to imperfect bounding box prediction of the OD detector.

5. REFERENCES

[1] H. Quigley and A. Broman, "The number of people with glaucoma worldwide in 2010 and 2020." *Br J Ophthalmol*, vol. 90, pp. 262–267, 2006.

[2] A. Schuster, C. Erb, E. Hoffmann, T. Dietlein, and N. Pfeiffer, "The diagnosis and treatment of glaucoma." *Dtsch Arztebl Int.*, vol. 117(13), pp. 225–234, 03 2020.

[3] A. Katie, L. Wollstein, and G. Wollstein, "Structural and functional evaluations for the early detection of glaucoma." *Expert Review of Ophthalmology*, vol. 11(5), pp. 367–376, 2016.

[4] S. Mohammadi, G. Saeedi-Anari, C. Alinia, E. Ashrafi, R. Daneshvar, and A. Sommer, "Is screening for glaucoma necessary? a policy guide and analysis." *J Ophthalmic Vis Res.*, vol. 9(1), pp. 3–6, 01 2014.

[5] "Airogs - grand challenge leaderboard." [Online]. Available: https://airogs.grand-challenge.org/evaluation/final-test-phase/leaderboard/

[6] S. Shahinfar, P. Meek, and G. Falzon, ""how many images do i need?" understanding how sample size per class affects deep learning model performance metrics for balanced designs in autonomous wildlife monitoring," *Ecological Informatics*, vol. 57, p. 101085, 03 2020.

[7] H. Kim, J. S. Lee, H. M. Park, H. Cho, H. W. Lim, M. Seong, J. Park, and W. J. Lee, "A wide-field optical coherence tomography normative database considering the fovea-disc relationship for glaucoma detection," *Translational Vision Science & Technology*, vol. 10, 2021.

[8] D. Cochard, "Mapnbsp;: Evaluation metric for object detection models," Oct 2021. [Online]. Available: https://medium.com/axinc-ai/map-evaluation-metric-of-object-detection-model-dd20e2de2472

[9] "Augmentation." [Online]. Available: https://docs.ultralytics.com/FAQ/augmentation/

[10] D. Dlužnevskij, P. Stefanovič, and S. Ramanauskaitė, "Investigation of yolov5 efficiency in iphone supported systems," *Baltic Journal of Modern Computing*, vol. 9, 01 2021.

[11] Ultralytics, "Tips for best training results · ultralytics/yolov5 wiki." [Online]. Available: https://github.com/ultralytics/yolov5/wiki/Tips-for-Best-Training-Results

[12] L. Prechelt, *Early Stopping — But When?*, 01 2012, pp. 53–67.

[13] V. Chahar, S. Katoch, and S. Chauhan, "A review on genetic algorithm: Past, present, and future," *Multimedia Tools and Applications*, vol. 80, 02 2021.

[14] "Hyperparameter evolution." [Online]. Available: https://docs.ultralytics.com/tutorials/hyperparameter-evolution/

[15] T. Aimyshev1, "Glaucoma detection algorithm for the artificial intelligence for robust glaucoma screening challenge." [Online]. Available: https://rumc-gcorg-p-public.s3.amazonaws.com/evaluation-supplementary/644/34b337f9-551a-4c82-834a-32a9f99ba690/AIROGS.pdf

[16] M. Abdullah-Al-Wadud, M. H. Kabir, M. A. A. Dewan, and O. Chae, "A dynamic histogram equalization for image contrast enhancement," *IEEE Transactions on Consumer Electronics*, vol. 53, no. 2, pp. 593–600, 2007.

[17] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2020. [Online]. Available: https://arxiv.org/abs/2010.11929

[18] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," 2021. [Online]. Available: https://arxiv.org/abs/2103.14030

[19]

[20] K. Pasupa, S. Vatathanavaro, and S. Tungjitnob, "Convolutional neural networks based focal loss for class imbalance problem: a case study of canine red blood cells morphology classification," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–17, 2020.

[21] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision. 2015," *arXiv preprint arXiv:1512.00567*, 2015.

[22] M. Kasper-Eulaers, N. Hahn, S. Berger, T. Sebulonsen, Myrland, and P. Kummervold, "Short communication: Detecting heavy goods vehicles in rest areas in winter conditions using yolov5," *Algorithms*, vol. 14, p. 114, 03 2021.

[23] K. Park, J. Kim, and J.-W. Lee, "Automatic optic nerve head localization and cup-to-disc ratio detection using state-of-the-art deep-learning architectures," *Scientific Reports*, vol. 10, 03 2020.

6. APPENDIX

Code for this work: `https://github.com/trajo/AIROGSLite-AI4MI-VU-2022`

Table 1: Performance different batch sizes on OD + fovea

Batch size	*Precision*	*Recall*	*mAP@0.5*	*mAP@0.5:0.95*	*Epochs*
8	0.982	0.963	0.981	0.766	48
16	0.981	0.965	0.975	0.787	74
24	0.974	0.97	0.985	0.760	51
32	0.985	0.958	0.983	0.733	30

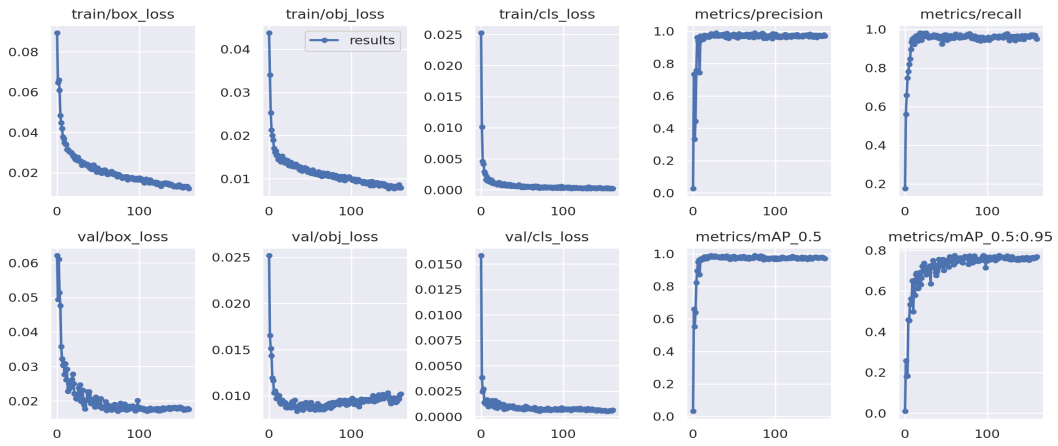
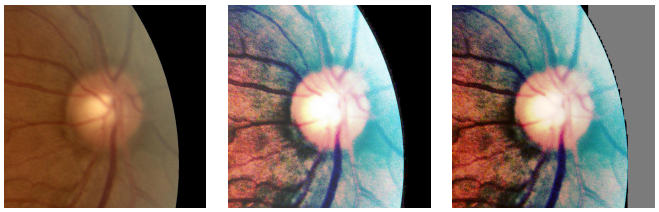


Fig. 1: Loss/performance over epochs YOLOv5l baseline model



(a) (b) (c)

Fig. 2: Equalization methods