Exploring gaze behavior through object detection for walking pedestrians with *Homonymous Hemianopia*

AYUSH KUMAR[1], DHRUV SHETH[1], SHRINIVAS PUNDLIK[1], AND GANG LUO[1]

[1] *Schepens Eye Research Institute, Harvard Medical School MA, USA*
[*] *Corresponding author: ayush$_k$umar@meei.harvard.edu*

Compiled October 31, 2022

To be finalised. (15Nov)

http://dx.doi.org/10.1364/ao.XX.XXXXXX

1. INTRODUCTION

Examining information in temporal gaze patterns can reveal insights into gaze behaviour essential for studying patterns in visual defect conditions. These patterns provide insight into how a gaze changes it's spatial location and orientation under a landscape and the reason why a specific gaze is focused on a particular set of objects for a larger period of time. The notion here is that it changes widely in patients concerning different visual defects and specifically examining the patterns exhibited in different visual defects may potentially be of clinical relevance for diagnosis and prognosis. Specifically examining patterns in patients with Homonymous Hemianopia allows extrapolating temporal trends to other visual defects as well. Quantitative analysis of these patterns are central to drawing reasonable inferences in behavioral patterns associated with each visual defect. Such an autonomous and non-invasive approach for identification of behavioral patterns provides a low-cost alternative in domains where MRI is not readily accessible or required.

Homonymous Hemianopia is a visual defect associated with the damage of the occipital lobe and the visual cortex leading to loss in visual information from the contralateral visual field. Previous studies focusing on oculography focused on manually observing quantitative patterns using approximated methods to classify them - Meienberg et al. [1]. These studies observed patients to consciously or unconsciously exhibit specific common strategies to spatially locate and fixate objects observed through gaze patterns. This helped draw assumptions that locating and fixating strategies are specific to visual defects and can be used to identify the defect. However, we still observe limitations in oculographic methods to manually classify patterns in longer temporal-trends and hence we propose this Research to employ gaze estimation for localizing quantifiable patterns temporally.

Comparing strategies previously used for diagnosis and prognosis of visual defects is significantly distinctive specifying the evaluation strategies for gaze checks which commonly rely on rapid vibration or change in object around a fixation point throughout the focal view to identify signs of visual defects, in this case Homonymous Hemianopia. Evaluation experiments relying on acquisition of visual stimuli specific to tracking an object around the fixation point does not give insight into behavioural analysis of the same patient under contemporary real world conditions. The following research develops a data acquisition system tracking gazes as well as relative head angles and spatial movements for a Homonymous Hemianopic patient walking in locations namely classified as empty street walking which involves gaze directional towards objects of other classes than 'person' evaluating the nature of objects and trends in gaze over specific object characteristics, crosswalks or lane-crossing mainly to evaluate simultaneous attention distribution over vehicles, people and crosswalk over the focal view, under stationary conditions assessing trends in tendency to observe objects under a certain quadrant of the focal view (characteristic distinctive to Homonymous Hemaniopia) and other similar instances.

The task of calculating gaze extends to the standard gaze estimation method under fixation point experiments where the head position is fixed as a function of time. Under simulated real-world conditions with moving head position in the coordinate space, gaze estimation is achieved through eye gaze estimation relative to the head spatial location and angle, and calculation of head coordinates in space relative to the ground frame. This creates an ensemble of both these calculation algorithms working simultaneously to achieve accurate gaze estimation for objects irrespective of the orientation of head in space. The calculation of head location is based on the information retrieved through IMU sensors positioned on the Homonymous Hemaniopic patient.

Temporally, the gaze is tracked and estimated for the person in-motion using a camera positioned to capture the gaze; simultaneously another camera positioned horizontally outwards captures the focal view of the landscape to correspond to the objects being observed by the gaze. Concerning studies involving mobile subjects with head and gaze position varying with time, most research focused solely on tracking gaze in-head orientation without considering the impact of head orientation over eye-gaze results. (Fotios, Uttley et al.[2], Li, Munn et al.[3]). Since head-orientation wasn't taken into account, an accurate picture of what the eye-gaze is directed towards could not be made. Relative analysis to some-extent may reveal important results; however an absolute estimation of spa-

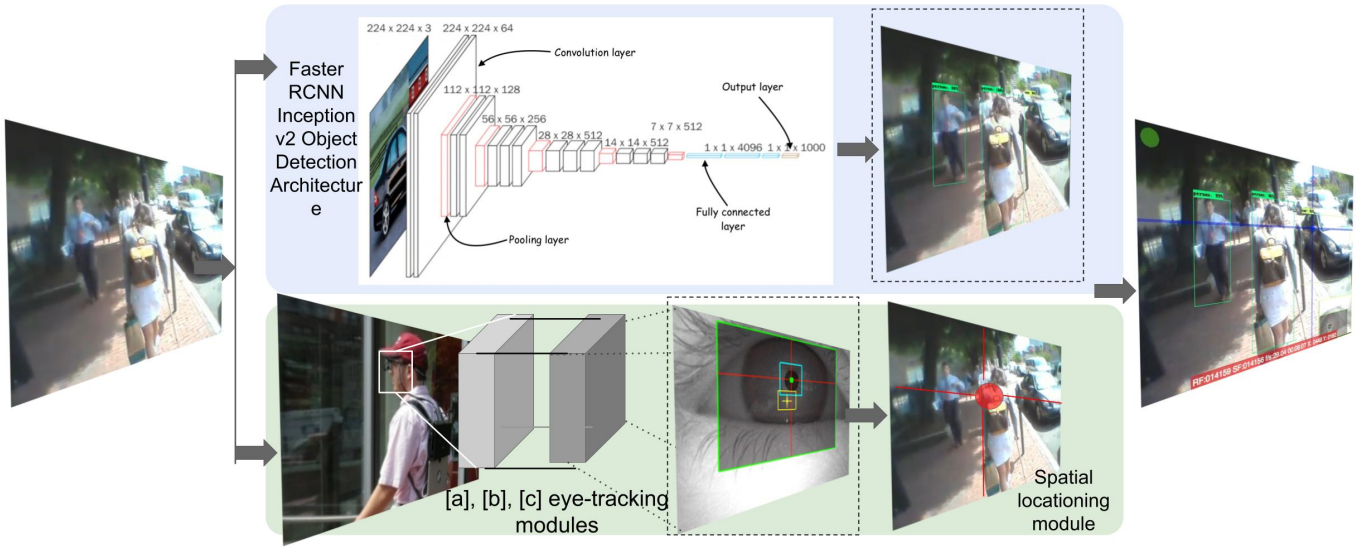

Fig. 1. Designed methodology of the proposed research. Here, modules [a], [b], [c] denotes a feature detection module, thresholding module and realtime pupil and optional corneal reflection tracking module which form an ensemble network for eye-gaze extraction. A Faster RCNN Inception V2 Object Detection framework is used for detecting objects for identifying salient gazed objects.

tial coordinates helps decipher accurate results with respect to the ground frame. Recent improvements are observed in tracking by estimating spatial positions of head movement with respect to confined environments using walking simulators of head-in-space motion capture systems (Barabas, Goldstein et al., [3]; Bowers, Ananyev et al., [4]; Cesqui, de Langenberg et al., [5])

Considering the case of Homonymous Hemianopia can be observed as uni-hemispherical peripheral visual field loss, quantification of temporal patterns in data through gaze tracking provides important behavioural insights which possibly might not be accomplished through eye tracking alone. According to Luo et al. [6], patients with tunnel vision observed saccadic eye-movements large enough to match those of normal vision person under specific conditions while walking outdoors. This makes it hard to distinguish between specific cases in people with visual defects compared to those with normal vision under different environments and using only in-head eye-movements. Extrapolating such data might not yield accurate results and might also contradict findings of previous experiments. Contrary to the frequent methods published to solely identify Homonymous Hemianopia, this paper provides a deeper insight in the behavioural and temporal gaze patterns observed for the patients under various outdoor conditions.

2. BACKGROUND

3. DATASET

Write about dataset in details. I will give you 2-3 papers which has used this dataset

4. METHOD

One of our main aims in this research is to to analyse and recognize the qualitative nature of the objects guided by a gaze. Here, we make use of Object detection to detect and classify objects based on their extrinsic properties and use these qualitative temporal patterns to identify Homonymous Hemianopic bio-markers. Object detection has been at the epicentre for different computer vision applications. Over the years, Object detection and classification frameworks have been widely used over a spectrum of applications from Autonomous Driving Assistance Systems (ADAS), Healthcare, Robotics etc. and contextually, the mode of deployment of the developed framework varies as well which can be observed in the form of two subsequent algorithms performing similar task extracting completely different patterns of data. Essentially, this research focuses on an Object detection model with a notion of saliency to understand gaze-aware relative importance of different classes and its demographics. The post-processing of our results focuses on identifying relative saliency of detected object classes distinguishing between observant behavioural patterns in Homonymous Hemianopic patients.

We start by comparing different Object detection models suited for our research environment and going with the one which yields the most accurate detections specific to the objects in our curated dataset. The detected objects are classified as whether they are gazed or not using three different levels of classification algorithms namely a Euclidean thresholding method through the centroid of the detecting bounding boxes and the other two being geometrical approaches calculating whether a gaze fixation at a spatial frame exists inside the bounding box temporally and the last one being a combination of the previous two; using a Euclidean thresholding method for the foot of perpendicular from the spatial gaze coordinates onto each localised segment of the bounding box using the distance for the perpendicular as the threshold for classification. This will be elaborated upon later.

Initially in this research, we gauge a comparison between 6 different Object detection frameworks each trained on the COCO (Common objects in context) dataset namely YOLO v4 Darknet (You Only Look Once), Faster R-CNN Resnet 50 COCO, SSD Mobilenet V1 COCO- (quantized), SSDlite Mobilenet V2 COCO, SSD Incpetion V2 COCO, Faster R-CNN Inception V2 COCO. It must be noted here that we gauge the frameworks such that a trade-off for speed or processing time for accuracy is preferred in each case. This is because the research focuses on evaluating behavioural patterns and this does not require a real-time on-device processing out-of-the-box and hence post-processing accurate methods (even though computationally heavy) are preferred. Additionally, even though we try to

Fig. 2. Comparison of 31/6500 frame from all networks to visually understand comments based on subjective comparison from **??**

Framework	Speed(ms)	COCO mAP	Comments
Faster R-CNN Inception v2	58	28	Efficient to Deploy, Accurate and low latency on the given input video sequence.
SSDlite Mobilenet V2 COCO	27	22	Efficient and real-time however not as accurate results .
SSD Inception V2 COCO	42	24	Results comparable to Faster R-CNN Inception, however few missed objects in occluded scenes
SSD Mobilenet V1 COCO - quantized	30	21	Slightly depreciated performance as compared to SSDlite Mobilenet V2, however lower latency.
Faster R-CNN Resnet50 COCO	89	30	Not efficient enough and multiple false-negatives for other classes, prominent flickering between frames.
Yolo V4 Darknet	73	47.5	Efficient, however produces few false negatives and performs worse in occluded scenarios

Table 1. Initial Objective and Subjective comparison of all tested frameworks on the HMS Homonymous Hemanopia Dataset. As observed in the next section, YOLOv4 Darknet and Faster R-CNN Inception v2 performs better than most other compared.

minimise false negatives as well as false positives in our framework, there is a general preference towards having no-detections in any given frame rather than false-positives since those might yield arbitrary conclusions which might drive our findings onto a different tangent which is avoided. The models trained on COCO dataset are preferred here since the classes in the set of COCO list are representative of most common objects influencing major proportion of behavioural patterns reflective of the visual information perceived. Withing this set of COCO classes, we primarily focus on 24 classes overly dominant in each spatial frame combined temporally which primarily constitute the person class, 4 major *vehicle* classes, road signs and symbols classes and a few other influencing gaze patterns. We use a subjective method of ranking different Object detection frameworks relative to our use-case by observing:

- The accuracy of detected objects in major classes.
- Consistency of the detected objects over temporal frames
- Performance of occluded regions of objects

- Percentage of false-negatives observed in random sample frames and overall calculating the information retrieved through the detected objects overall.

Additionally, we consider the quantitative results such as mAP for the dataset (COCO) and speed(ms) in case two frameworks observe identical performance. Even though in the short term we do not intend on using a real-time Object detection framework for analysing behavioral patterns, we still include SSD Mobilenet v1 and SSDlite Mobilenet v2 COCO for understanding their relative performance on this dataset for future studies which are real-time Object detection models which can be deployed on an edge-device.

A. Subjective Evaluation of Frameworks

According to our subjective assertion in **??**, we narrow our focus on comparing the two most optimum frameworks i.e. YOLO v4 Darknet and Faster R-CNN Inception v2 COCO by evaluating the subjective comments in the table. Narrowing our focus down to two frameworks after subjectively and objectively testing the others helps eliminate any other ambiguity in our evaluation. Despite both the frameworks being trained on the COCO dataset, the object detection method both employ widely varies according to the application and in this case, both frameworks tend to vary in their performance depending on the tasks. Assigning each task a weightage according to relative importance and then deciding the most optimum framework based on our discretion 4.

A.1. YOLO V4 Darknet

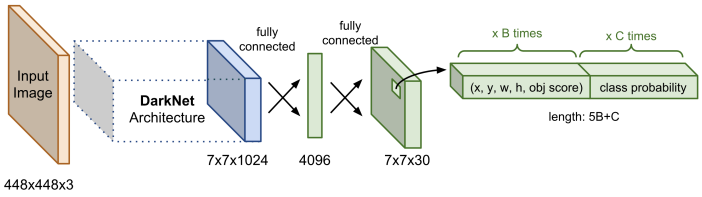

Fig. 3. A generalized network architecture for YOLO

YOLOv4 is a single stage detector framework with it's core of target detection algorithm in it's efficient calculation and small size. The model is capable of effectively localizing the objects in an image which makes it efficient in various applications for object detection. YOLO V4 skips the region proposal network in it's algorithm and tends towards making a trade-off for efficient and faster detection by running over a dense sampling of all possible locations using a one-stage object detection approach. In theory this might impact the performance to some margin, however the magnitude of this difference varies with different applications. YOLO's efforts in minimizing the overlapping of detection boxes lies in using the global image for detection, which encodes the global information reducing the detections of background as an object. In general, YOLO v4 Darknet outperforms most other networks in scenarios where the object is distant in the focal view and has a clear consistent visible region temporally over the frames. Additionally, YOLO is known for its consistency in detecting objects in temporal frames which preserve spatial appearance.

- **Backbone** - CSPDarknet53 is a modified version of the Darknet-53 network. The number of convolution layers in the network are represented by the number '53' and CSP stands for Cross-stage-partial connections.
- **Neck:** Path Aggression Network (PAN) and Spatial Pyramid Pooling (SPP) form the components of the neck of the network. PAN serves as the method of parameter aggregation for different detector levels and SPP is used to increase the receptive field by a significant amount which separates the most significant context features without compromising the network operaton speed.
- **Head:** YOLOv3 is used as the end of the chain object detector for dense predictions and detections.

In general, various different versions of YOLO have shown high computation speed and real-time inference, however since our application focuses on accuracy over real-time detection, the real-time factor isn't taken into consideration while choosing frameworks. The YOLO algorithm is designed to employ an image into a grid of multiple cells and the confidence scores inside the region of the bounding box are predicted for each cell and assigned a class probability. This is done by using IOU metric (Intersection over Union), a common method for evaluating semantic segmentation models, which measure the extent of overlap between detected object and the defined ground truth as a fraction of the total area spanned by the union of the two. Looking at a brief cycle of improvements in the YOLO algorithm, YOLOv2 [7] consisted of anchor boxes- a pre-determined set of boxes to predict the offsets from these pre-defined anchor boxes rather than directly predicting the bounding box. However, YOLOv3 [8] included bounding box prediction over different scales and the inclusion of 53 layers in Darknet. Currently, the most optimum of all proposed, YOLOv4 [9] was superior in terms of both speed and accuracy. As seen in **??**, YOLOv4 provided 47.5% mAP over MS COCO Dataset at a speed of 73ms per frame. In theory, YOLOv4 can be summarised as follows:

It might be further worth noting the workflow of YOLO which is a one-stage detector explained through 5:

- An image is split into a grid of NxN cells, where locally those cells in which the object's centre is located, are responsible for the detection of the object. Associated with each cell is a location of B bounding boxes, confidence score and the probability of an object class dependent on the existence of an object in the bounding box.
- A tuple of 4 values normalized between [0,1] contain the coordinates of the bounding box namely (center x-coordinate, center y-coordinate, width, height) in the format (x, y, w, h) where x and y are conditioned depending on the cell location.
- Each cell is associated with its individual confidence score which indicates the likelihood of the presence of object. Pr(containing an object) x IoU(pred, truth) . Pr: Probability, IoU: Intersection over Union.
- The probability of an object contained by a cell belonging to every class $C_i, i = 1 \ldots K$ is predicted by Pr(the object belongs to the class $C_i \mid$ containing an object). This allows the model to predict only one set of class probabilities per cell regardless of the number of bounding boxes, B.
- Cumulatively, an image contains $NxNxB$ bounding boxes where each box corresponds to 4 location predictions, 1 confidence score, and K conditional probabilities for object classification.
- Finally, the final layer of YOLO's CNN Network outputs a tensor of size $NxNx(5B+K)$

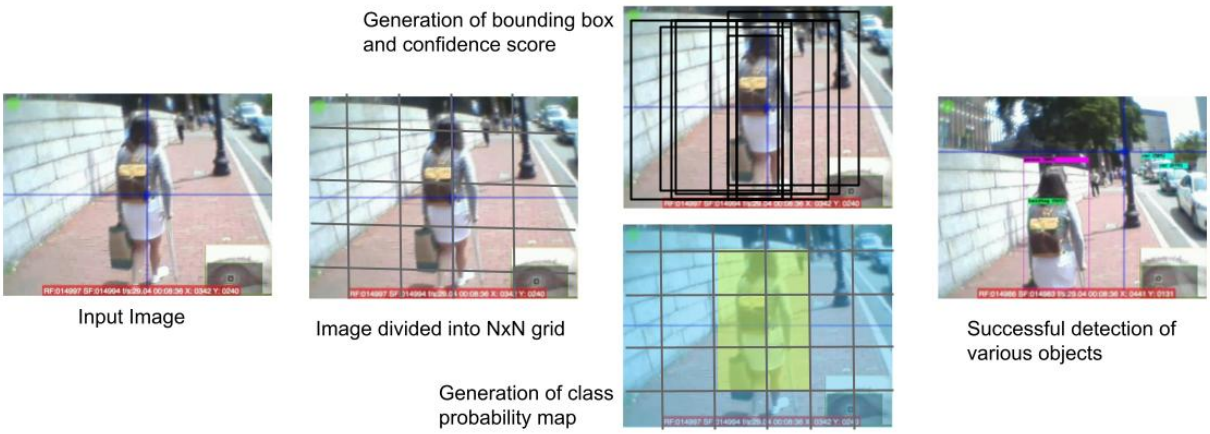

Fig. 4. Different stages of object detection explained through A.1 for a standard one-stage YOLO detection algorithm, further generalized to YOLOv4 Darknet.

Fig. 5. Frame-wise comparison of Faster R-CNN network with YOLOv4 Darknet

A.2. Faster R-CNN

R-CNN, coined by [10], is a Convolutional Neural Network (CNN) with an added region-proposal algorithm which hypothesizes object locations. This network employs a selective search to initially extract a fixed number of regions (2000). Further, using a greedy algorith, this network merges similar regions together to obtain the selected regions where object detection is applied. Since R-CNN came with a speed bottleneck, both in terms of training and testing, the authors designed an enhanced algorithm entitled Fast R-CNN [11] using a shared convolutional feature map that the convolutional neural network would generated from the input image which is used to extract the Regions of Interest (RoI). While Fast R-CNN was arguably better in terms of both training and testing time, the improvement was not dramatic because the region proposals were generated separately by another model which tends to be expensive. However, Ren et al. [12] proposed a Faster R-CNN algorithm which introduced the Region Proposal Network (RPN) by integrating the region proposal algorithm in the CNN model itself. Faster-RCNN introduces the construction of an ensemble of a unified model composed of RPN and Fast R-CNN with shared convolutional feature layers which is trained end-to-end to predict both- the object bounding boxes and objectness scores in a computationally inexpensive manner (10ms per frame).

Workflow explaining the Faster R-CNN architecture in 7:

- The RPN (region proposal network) is fine-tuned end-to-end for the region proposal task, further initialized by the pre-train image classifier. The positive samples are assigned an IoU score > 0.7 and negative samples an IoU score < 0.3 which might be indirectly treated as a threshold.
- A spatial window cover of size $n \times n$ is slid over the convolutional feature map of the frame.
- At the center of each sliding window, multiple regions of varying scales and ratios are predicted simultaneously. An anchor is a combination of (sliding window center, scale, ratio) .

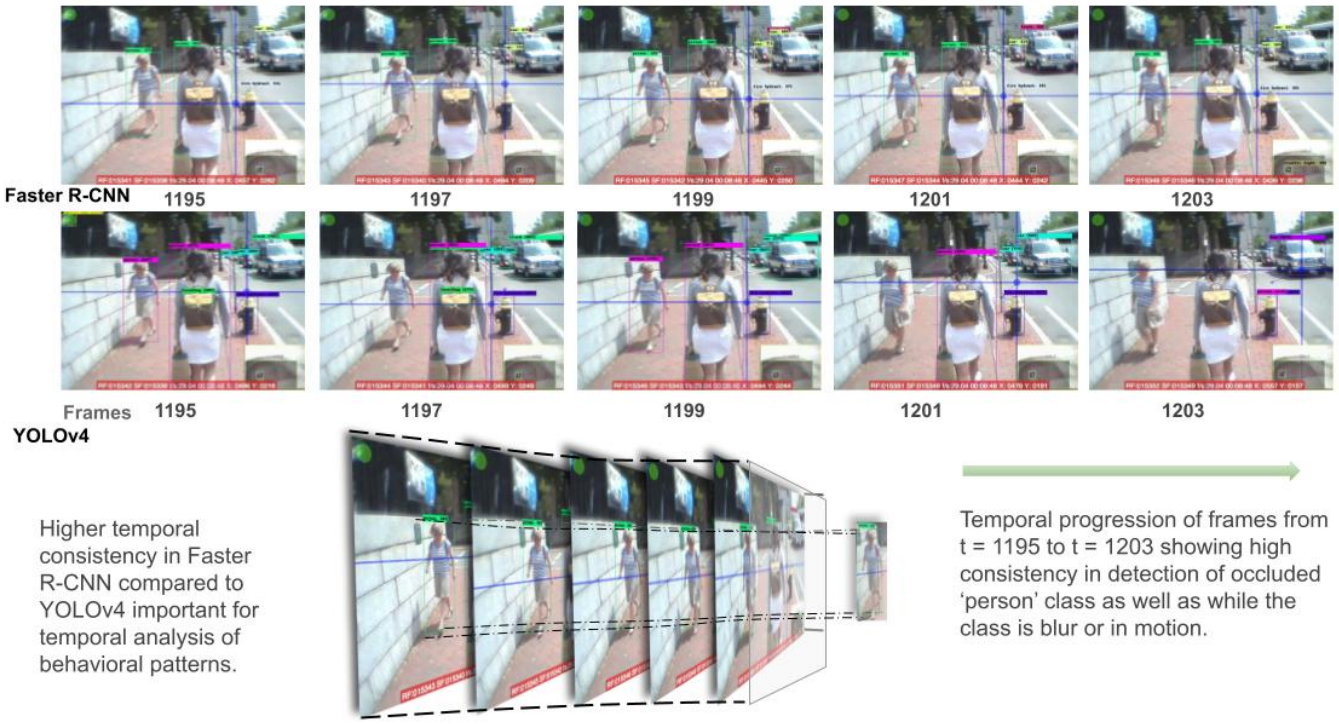

Fig. 6. Consistency analysis of both networks - YOLOv4 Darknet and Faster R-CNN Inception v2

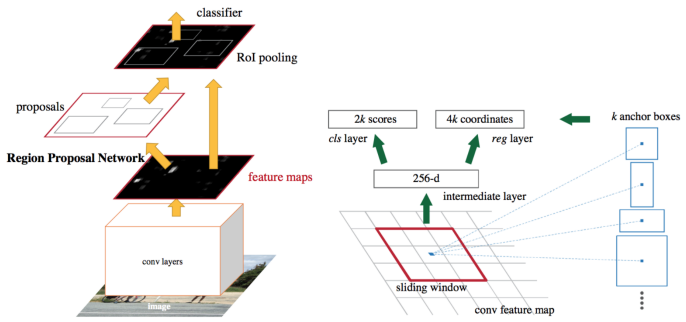

Fig. 7. Network architecture for Faster-RCNN model

For example, 3 scales + 3 ratios => k=9 anchorsat each sliding position.

- Finally, a 'Fast R-CNN' object detection model is trained using the proposals generated by the RPN.
- The Fast R-CNN network is used to then initialize RPN training. The Shared Convolutional layers are kept and simultaneously, the RPN-specific layers are fine-tuned. Now, the detection network and RPN have shared convolutional layers which completes the workflow.

Loss function for Faster R-CNN network: The loss function of the Faster R-CNN network can be described in the following manner-

The multi-task loss function combines the losses of classification and bounding box regression:

$$\mathscr{L} = \mathscr{L}_{\text{cls}} + \mathscr{L}_{\text{box}}$$

$$\mathscr{L}\left(\{p_i\},\{t_i\}\right) = \frac{1}{N_{\text{cls}}}\sum_i \mathscr{L}_{\text{cls}}\left(p_i, p_i^*\right) + \frac{\lambda}{N_{\text{box}}}\sum_i p_i^* \cdot L_1^{\text{smooth}}\left(t_i - t_i^*\right)$$

where $\mathscr{L}_{\text{cls}}$ is the log loss function over two classes, as we can easily translate a multi-class classification into a binary classification by predicting a sample being a target object versus not. L_1^{smooth} is the smooth L1 loss.

$$\mathscr{L}_{\text{cls}}\left(p_i, p_i^*\right) = -p_i^* \log p_i - \left(1 - p_i^*\right)\log\left(1 - p_i\right)$$

The variables used to defined the loss function can be enlisted in the form: $|p_i|$ Predicted probability of anchor i being an object. $||\ p_i^*\ |$ Ground truth label (binary) of whether anchor i is an object. $||t_i|$ Predicted four parameterized coordinates. $||t_i^*|$ Ground truth coordinates. $|\ |\ N_{\text{cls}}\ |$ Normalization term, set to be mini-batch size (256) in the paper. $|\ N_{\text{box}}\ |$ Normalization term, set to the number of anchor locations (2400) in the paper. $||\ \lambda\ |$ A balancing parameter, set to be ~ 10 in the paper (so that both $\mathscr{L}_{\text{cls}}$ and $\mathscr{L}_{\text{box}}$ terms are roughly equally weighted).

B. Comparison of Both Frameworks through Standard Metrics

Since YOLOv4 performance widely varies as per context and the dataset, we subjectively carried out our evaluation by localizing the detected objects in each frame, penalizing the performance for each undesired object detected or object missed out with separate categories for occlusion based missing or inconsistent frame tracking. Table 1 proposes a direct comparison between YOLOv4 Darknet and Faster R-CNN based on theoretical information on the architecture of the given models.

Quantifying rate of False-positives and False-negatives in both networks Amar et al. [13] conducted a comparative study between the frameworks (Faster R-CNN Inception V2, Faster R-CNN Resnet 50, YOLO V3 and YOLO V4) and found that the difference is False positives, negatives and precision is closely intertwined with the dataset chosen for the study. While the above research was conducted on two aerial datasets, the performance of both models widely varied with the nature of the dataset- (Stanford Dataset and Prince Sultan University (PSU) Dataset). The representation of the images and nature of images widely influenced the performance of the models.

	YOLOv4	Faster R-CNN
Phases	Concurrent bounding box regression, and classification.	RPN + Fast R-CNN object detector.
Neural network type	Fully convolutional.	Fully convolutional (RPN and 4 detection network).
Backbone feature extractor	CSPDarknet53 (53 convolutional layers).	VGG-16 or Zeiler Fergus(ZF). Other feature extractors can also be incorporated.
Location detection	Anchor-based	Anchor-based
Number of anchor boxes	Using multiple anchors for a single ground truth	3 scales and 3 aspect ratios, yielding k = 9 anchors at each sliding position.
Default Anchor sizes	(12,16), (19,36), (40,28), (36,75), (76,55), (72,146), (142,110), (192,243), (459,401)	Scales: (128,128), (256,256), (512,512). Aspect ratios: 1:1, 1:2, 2:1.
IoU thresholds	One (at 0.213)	Two (at 0.3 and 0.7).
Loss function	Complete IoU loss: CIoU	Multi-task loss: - Log loss for classification. - Smooth L1 for regression.
Input size	Different possible input sizes (n × n with n multiple of 32).	- Conserves the aspect ratio of the original image. - Either the smallest dimension is 600, or the largest dimension is 1024.
Batch size	Default value: 64.	Default value: 1

The Stanford dataset had nearly 30 times as many object instances to be trained on than PSU dataset per image. Following this, one of the conclusions made by the authors were that the YOLO V4 was specifically tuned for the COCO dataset and did not yield as high Average Precision values for the rest of the datasets. It could also be subjectively observed that YOLO V4 had a higher tendency of detecting fainter, distant objects which Faster R-CNN could not easily catch, and while this hints at YOLO V4 being a better algorithm for this niche of datasets, these results cannot be smoothly extrapolated in our case. The reason is that Faster R-CNN performs better than YOLO V4 for occluded images considering the nature of the algorithm; and with most of the dataset consisting of occlusions, the result changes drastically.

5. RESULTS

We used the Faster R-CNN Inception V2 Framework to carry out the object detection for the Focal view of the gaze combined with gaze estimation done discretely and later combined into a single analysis algorithm to calculate the description of different objects being gazed which breaks down into qualitative and quantitative analysis. The results section is also further broken down into two parts:

- Raw data analysis
- Visual description of quantitative results

Work in Progress. Generating visualisations from csv data through google colab.

TBD 6Sept

Algorithm 1. Euclid's algorithm

procedure EUCLID(ab) ▷ The g.c.d. of a and b
  $r \leftarrow a \bmod b$
  while $r \neq 0$ **do** ▷ We have the answer if r is 0
    $a \leftarrow b$
    $b \leftarrow r$
    $r \leftarrow a \bmod b$
  return b ▷ The gcd is b

6. CORRESPONDING AUTHOR

We require manuscripts to identify a single corresponding author. The corresponding author typically is the person who submits the manuscript and handles correspondence throughout the peer review and publication process. If other statements about author contribution and contact are needed, they can be added in addition to the corresponding author designation.

7. EXAMPLES OF ARTICLE COMPONENTS

The sections below show examples of different article components.

8. FIGURES AND TABLES

It is not necessary to place figures and tables at the back of the manuscript. Figures and tables should be sized as they are to appear in the final article. Do not include a separate list of figure captions and table titles.

Figures and Tables should be labelled and referenced in the standard way using the `\label{}` and `\ref{}` commands.

Algorithm	Feature Extractor	Input Size	AP
TP	FN	FP	Precision
Recall	F1 Score	FPS	Inference Time (ms)
Faster R-CNN	Inception v2	992 × 550 (variable)	0.739
548	190	11	0.980
0.743	0.845	9.5	105
Faster R-CNN	Inception v2	608 × 608 (fixed)	0.731
541	197	14	0.975
0.733	0.837	9.5	105
YOLOv4	CSPDarknet-53	320 × 320 (fixed)	0.961
715	23	59	0.924
0.969	0.946	22.4	45
YOLOv4	CSPDarknet-53	416 × 416 (fixed)	0.965
720	18	66	0.916
0.976	0.945	19.4	52
YOLOv4	CSPDarknet-53	608 × 608 (fixed)	0.950
715	23	66	0.915
0.969	0.941	13	77

Table 2. Numerical score for different metrics associated with PSU Dataset.

Algorithm	Feature Extractor	Input Size	AP
TP	FN	FP	Precision
Recall	F1 Score	FPS	Inference Time (ms)
Faster R-CNN	Inception v2	600 × 816 (variable)	0.202
1780	6351	1813	0.495
0.219	0.304	19.2	52
Faster R-CNN	Inception v2	608 × 608 (fixed)	0.317
2916	5215	2654	0.524
0.359	0.426	21.1	47
YOLOv4	CSPDarknet-53	320 × 320 (fixed)	0.157
1278	6853	5	0.996
0.157	0.272	21.1	47
YOLOv4	CSPDarknet-53	416 × 416 (fixed)	0.202
1646	6485	1	0.999
0.202	0.337	18.5	54
YOLOv4	CSPDarknet-53	608 × 608 (fixed)	0.209
1701	6430	64	0.964
0.209	0.344	12.5	80

Table 3. Numerical score for different metrics associated with Stanford Dataset

A. Sample Figure

Figure 8 shows an example figure.

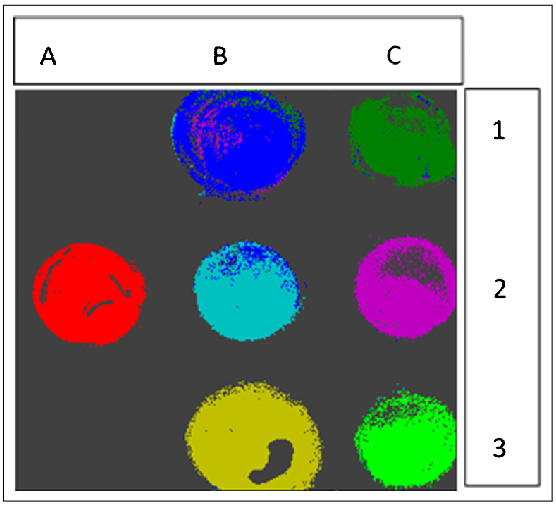

Fig. 8. False-color image, where each pixel is assigned to one of seven reference spectra.

B. Author Photographs

Author photographs. The final printed size of an author photograph is exactly 1 inch wide by 1 1/4 inches long (6 picas × 7 1/2 picas). Please ensure that the author photographs you submit are proportioned similarly.

C. Sample Table

Table 4 shows an example table.

Table 4. Shape Functions for Quadratic Line Elements

local node	$\{N\}_m$	$\{\Phi_i\}_m$ $(i = x, y, z)$
$m = 1$	$L_1(2L_1 - 1)$	Φ_{i1}
$m = 2$	$L_2(2L_2 - 1)$	Φ_{i2}
$m = 3$	$L_3 = 4L_1L_2$	Φ_{i3}

9. SAMPLE EQUATION

Let $X_1, X_2, \ldots, X_n$ be a sequence of independent and identically distributed random variables with $\mathrm{E}[X_i] = \mu$ and $\mathrm{Var}[X_i] = \sigma^2 < \infty$, and let

$$S_n = \frac{X_1 + X_2 + \cdots + X_n}{n} = \frac{1}{n}\sum_{i}^{n} X_i \qquad (1)$$

denote their mean. Then as n approaches infinity, the random variables $\sqrt{n}(S_n - \mu)$ converge in distribution to a normal $\mathcal{N}(0, \sigma^2)$.

10. SUPPLEMENTAL MATERIAL

Consult the Author Guidelines for Supplementary Materials in Optica's Journals for details on accepted types of materials and instructions on how to cite them. All materials must be associated with a figure, table, or equation or be referenced in the results section of the manuscript. (1) 2D and 3D image files and video must be labeled "Visualization," not "Movie," "Video," "Figure," etc. (2) Machine-readable data (for example, csv files) must be labeled "Data File." Number data files and visualizations consecutively, e.g., "Visualization 1, Visualization 2. . . ." (3) Large datasets or code files must be placed in an open, archival database. Such items should be mentioned in the text as either "Dataset" or "Code," as appropriate, and also be cited in the references list. For example, "see Dataset 1 (Ref. [1]) and Code 1 (Ref [2])." Here are examples of the references:

A. Sample Dataset Citation

1. M. Partridge, "Spectra evolution during coating," figshare (2014) [retrieved 13 May 2015], http://dx.doi.org/10.6084/m9.figshare.1004612.

B. Sample Code Citation

2. C. Rivers, "Epipy: Python tools for epidemiology," (figshare, 2014) [retrieved 13 May 2015], http://dx.doi.org/10.6084/m9.figshare.1005064.

11. FUNDING AND ACKNOWLEDGMENTS

Formal funding sources should be listed in a separate paragraph block before any other acknowledgment information. Funding sources and any associated grant numbers should match the information entered into the Prism manuscript system. Funders should be listed without any introductory language or use of labels (do not use labels such as "grant no."). The acknowledgments may contain any information that is not related to funding. Here is an example:

FUNDING

National Science Foundation (NSF) (1263236, 0968895, 1102301); The 863 Program (2013AA014402).

ACKNOWLEDGMENTS

The authors thank H. Haase, C. Wiede, and J. Gabler for technical support.

12. REFERENCES

Full references (to aid the editor and reviewers) must be included. This will be produced automatically if you are using a .bib file.

Add citations manually or use BibTeX. See [**? ?**].

REFERENCES

1. O. Meienberg, W. H. Zangemeister, M. Rosenberg, W. F. Hoyt, and L. Stark, "Saccadic eye movement strategies in patients with homonymous hemianopia," Annals Neurol. **9**, 537–544 (1981).
2. F. Li, S. Munn, and J. Pelz, "A model-based approach to video-based eye tracking," J. Mod. Opt. **55**, 503–531 (2008).
3. S. Fotios, J. Uttley, C. Cheal, and N. Hara, "Using eye-tracking to identify pedestrians' critical visual tasks, part 1. dual task approach," Light. Res. & Technol. **47**, 133–148 (2015).
4. A. Bowers, E. Ananev, A. Mandel, R. Goldstein, and E. Peli, "Driving with hemianopia: Iv. head scanning and detection at intersections in a simulator," Investig. ophthalmology visual science **55** (2014).
5. B. Cesqui, R. Langenberg, van de, F. Lacquaniti, and A. D'Avella, "A novel method for measuring gaze orientation in space in unrestrained head conditions," J. vision **13** (2013).
6. G. Luo, F. Vargas-Martin, and E. Peli, "The role of peripheral vision in saccade planning: Learning from people with tunnel vision," J. vision **8**, 25.1–8 (2008).
7. J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2017), pp. 6517–6525.

8. J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” (2018).
9. A. Bochkovskiy, C.-Y. Wang, and H.-y. Liao, “Yolov4: Optimal speed and accuracy of object detection,” (2020).
10. R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *2014 IEEE Conference on Computer Vision and Pattern Recognition,* (2014), pp. 580–587.
11. R. Girshick, “Fast r-cnn,” in *2015 IEEE International Conference on Computer Vision (ICCV),* (2015), pp. 1440–1448.
12. S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” IEEE Transactions on Pattern Analysis Mach. Intell. **39**, 1137–1149 (2017).
13. A. Ammar, A. Koubaa, M. Ahmed, A. Saad, and B. Benjdira, “Vehicle detection from aerial images using deep learning: A comparative study,” Electronics. **10**, 820 (2021).